Extracção de Recursos de Tradução com base em Dicionários Probabilísticos de Tradução

Alberto Manuel Brandão Simões
(ambs@di.uminho.pt)

Dissertação submetida à Universidade do Minho para obtenção do grau de Doutor em Informática, elaborada sob a orientação de José João Almeida.

Departamento de Informática
Escola de Engenharia
Universidade do Minho

Braga, 2008

Dissertação submetida à **Escola de Engenharia da Universidade do Minho** para a obtenção do grau de **Doutor em Informática** na área de **Inteligência Artificial**.

Financiada por uma bolsa da **Fundação para a Computação Científica Nacional (FCCN)** de Setembro de 2004 a Setembro de 2007 através do projecto **Linguateca**, por sua vez financiado pela **Fundação para a Ciência e Tecnologia (FCT)** através do projecto **POSI/PLP/43931/2001**, co-financiado pelo **POSI** através do projecto **4/1.3/C/NRE** (de 15 de Maio de 2000 a 15 de Dezembro de 2006) e pelo **POSC** através do projecto **339/1.3/C/NAC** (desde 15 de Dezembro de 2006).

Resumo

Os recursos bilingues mais abundantes são os corpora paralelos. Resultam de toda uma história de tradução de instituições e organizações internacionais. Estes corpora constituem um recurso de tradução muito rico, mas que precisa de ser tratado para ser útil: é necessária a sua preparação, realçando conhecimento que se encontra camuflado.

Neste trabalho pretende-se obter conhecimento de diferentes tipos: dicionários de tradução, terminologia bilingue, exemplos de tradução (segmentos equivalentes) ou mesmo n-gramas. Para além de realizar a extracção destes recursos, pretende-se definir uma álgebra que os permita manusear e tratar.

O ponto inicial na extracção de recursos bilingues corresponde à definição de pontes básicas entre as duas línguas: relacionamentos entre palavras, que são representados como dicionários probabilísticos de tradução.

Com base nos corpora paralelos e nos dicionários probabilísticos de tradução são extraídos diferentes tipos de recursos, como sejam exemplos de tradução ou terminologia bilingue.

A necessidade de adaptar os vários recursos bilingues extraídos às situações concretas em que vão ser usados leva a que seja útil um ambiente para o desenvolvimento e prototipagem de processadores de recursos. Este ambiente é constituído por um servidor de recursos e uma API (application programming interface) de ordem superior que os permite manipular.

Os recursos bilingues, para além de poderem ser utilizados de forma programática, são úteis por si só. Neste sentido, é importante a sua disponibilização para consulta interactiva através da Internet, e para uso local através de dicionários off-line.

Dado que todos os métodos usados se baseiam em estatística, e que se pretende uma grande cobertura lexical dos recursos obtidos, é necessário processar corpora de grandes dimensões, pelo que se usam mecanismos de decomposição e ferramentas de suporte ao processamento paralelo que permitem a escalabilidade dos métodos desenvolvidos.

Abstract

The most abundant bilingual resource available are parallel corpora. They are the result of years of human translations performed on international institutions and organizations. These corpora are rich sources of translation knowledge but, to be useful, need to be prepared, enhancing their hidden knowledge.

This main goal of this work is the extraction of different kinds of bilingual knowledge (translations dictionaries, bilingual terminologies, translation examples and n-grams) and the definition of a resources algebra.

The first task of bilingual resources extraction is the identification of basic bridges between two languages: the extraction of relationships between words, that are stored as probabilistic translation dictionaries.

These probabilistic translation dictionaries are used to extract different kinds of bilingual resources from parallel corpora such as translation examples or bilingual terminology.

The extracted resources can be used for different intentions. This makes it important to have a workbench for the development and prototyping of resources processors. This workbench comprises a bilingual resources server and a high order API (application programming interface) over it.

The bilingual resources are useful both for the development of natural language processing applications or by final-users like translators. For these users, it is important to make these resources available. This can be done over the Internet, using an integrated web application, or by releasing off-line dictionaries.

Given that most of the presented methods are based on statistics and that we want a wide lexical coverage, we need to process big corpora. The use of decomposition methods and tools to support parallel processing makes it possible to give scalability to the developed methods.

Agradecimentos

Esta é a parte lamechas, mas também a primeira a ser lida pela maioria dos leitores. A realização de uma dissertação, dada a sua extensão e trabalho, por vezes solitário, leva a que sem apoio não chegue a bom porto. Felizmente tive esse apoio e, portanto, me parece importante fazer alguns agradecimentos.

Um obrigado aos meus dois orientadores, **Diana Santos** e **José João Almeida**, por terem aceite essa árdua tarefa, e me terem aturado na minha desorganização e intermitente motivação.

Aos professores **Pedro Rangel Henriques**, **José Bernardo Barros** e **Luís Soares Barbosa** pela oportunidade que me deram de leccionar no departamento o que me permitiu ganhar experiência e contactos de alunos interessados em realizar projectos conjuntamente.

Nesse correr, um obrigado ao **José Alves de Castro**, **Rúben Fonseca** e **Luís Gomes**, alunos e amigos que contribuíram activamente na construção de ferramentas e recursos.

A outros alunos que, embora não tenham contribuído directamente para a realização desta dissertação me aturaram a experimentar novas abordagens para problemas de PLN, se tornaram bons amigos: **José Marques** e **Luís Miguel Braga**.

Um obrigado pela ajuda e colaboração de um conjunto de investigadores: **Andy Way**, **Xavier Gomez Guinovart**, **Mikel Forcada** e **Djoerd Hiemstra**.

E em último, por ser o maior agradecimento de todos, à minha **Família** que me aturaram, e em especial à minha **Mãe** por todo o carinho e motivação.

A todos, o meu **muito obrigado**!

Alberto

Conteúdo

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

Capítulo 1

Introdução

> [6] *et dixit Dominus: "Ecce unus est populus et unum labium omnibus; et hoc est initium operationis eorum, nec eis erit deinceps difficile, quidquid cogitaverint facere.* [7] *Venite igitur, descendamus et confundamus ibi linguam eorum, ut non intellegat unusquisque vocem proximi sui".*
>
> *Genesis 11, 6-7*

O nosso planeta está dividido em vários continentes e países, que se podem agrupar, de certa forma, de acordo com a sua cultura e língua. Desde sempre[1] que o ser humano precisa de comunicar com culturas diferentes daquelas em que está inserido o que leva à necessidade de estabelecer relacionamentos entre a sua e outras línguas.

Para aprender uma nova língua é habitual[2] preocupar-mo-nos por estabelecer pontes entre palavras em duas línguas. Começamos por aprender associações entre palavras simples, como "*Olá*" e "*Hello*," ou "*Adeus*" e "*Bye*."

[1] Ou, de acordo com a Bíblia (citação do Génesis, 11, 6–7), desde a tentativa Humana da construção da Torre de Babel.

[2] A aprendizagem de uma nova língua pode ser feita usando métodos muito diferentes. A abordagem aqui descrita é uma das possíveis.

Só depois de estarmos confortáveis no relacionamento entre palavras simples é que as tentamos juntar, e criar relacionamentos entre segmentos de palavras. Surge então os habituais "*bom dia*" e "*good morning*," ou o "*boa noite*" e "*good night*[3]."

A tradução nestes casos é composicional[4], ou seja, a tradução do todo pode ser obtida pela tradução das partes[5]:

$$\mathcal{T}(s_1 \cdot s_2) = f(\mathcal{T}(s_1), \mathcal{T}(s_2))$$

Esta função f é na sua forma mais simples a concatenação das traduções, mas pode ser mais complicada, como veremos mais à frente.

Tudo se complica quando as palavras não têm uma correspondência directa, palavra-a-palavra. Basta começarmos a aprender os parentescos para nos confundirem ao associar "*sogra*" à expressão "*mother in law*." Muito perto deste exemplo, temos muitos outros exemplos terminológicos que não são traduções composicionais [6]. As traduções de "*Computer Graphics*" por "*Computação Gráfica*" e de "*Eigen Values*" por "*Valores Próprios*" são emblemáticas. Não faltam exemplos da falta de composicionalidade na tradução.

A falta de composicionalidade torna-se ainda mais notória quando cresce o contexto cultural da expressão em causa, como é o caso das expressões idiomáticas. Embora o exemplo da tradução de "*colocar a carroça à frente dos bois*" por "*putting the cart before the horse*" não seja completamente composicional, é quase uma tradução palavra-a-palavra. Existe apenas alguma diferença cultural que leva a que o animal usado

[3] O uso de "boa noite" em português, ou de "good night" em inglês não é bem o mesmo, já que este último é habitualmente usado apenas como despedida, enquanto que a sua versão portuguesa é também bastante usada como cumprimento.

[4] No âmbito desta dissertação não detalharemos este conceito. A problemática da composicionalidade é bastante rica já que lida com diferentes tipos de composicionalidade: a composicionalidade léxica, a composicionalidade sintáctica/estrutural, a composicionalidade semântica e ainda a cristalização de termos (terminologia).

[5] $\mathcal{T}()$ representa a função de tradução, s_1 e s_2 dois segmentos de palavras, e $\cdot$ a concatenação destes segmentos. O anexo B explica detalhadamente a notação matemática usada.

[6] Definimos *tradução composicional* neste contexto como o facto de a tradução de determinado segmento de palavras poder ser obtido pela tradução das partes, aplicando apenas correcções de concordâncias de género e número.

na expressão seja outro. Outros exemplos, como a expressão "*preso por ter cão e preso por não ter*" não tem uma tradução directa, palavra-a-palavra, em inglês[7]. Possivelmente, a expressão que deveria ser usada como tradução seria, por exemplo, "*robbing Peter to pay Paul*."

Felizmente a tradução por composicionalidade é a regra em grande parte dos casos e, portanto a nossa aprendizagem de uma língua estrangeira não se confina a decorar frases. Por outro lado, a composicionalidade nem sempre é apenas a concatenação das traduções. A aprendizagem de uma nova língua obriga à assimilação de um conjunto de relacionamentos em termos léxicos mas também em termos sintácticos. Um exemplo de um relacionamento do tipo sintáctico é a troca entre os substantivos e adjectivos na tradução de português para inglês[8].

$$\mathcal{T}(w_S \cdot w_A) = \mathcal{T}(w_A) \cdot \mathcal{T}(w_S)$$

Estas regras[9] que temos de conhecer para aprender uma nova língua também são imprescindíveis.

Todo este conhecimento que vamos adquirindo corresponde à construção de pontes, à definição de relacionamentos bilingues quer entre palavras, segmentos de palavras, expressões ou mesmo entre estruturas sintácticas.

Para que estes recursos possam ser usados de forma automática por aplicações informáticas é necessário que contenham, para além da informação linguística, uma classificação qualitativa ou probabilística que permita aos programas optar por uma tradução em relação a outra, ou para permitir algum tipo de desambiguação. Podem ainda incluir um conjunto de predicados ou restrições que tenham de ser validados para que determinado recurso possa ser usado (como verificar a categoria morfológica de palavras antes de aplicar determinada regra).

A todos estes tipos de conhecimento multilingue que foram discutidos chamaremos de recursos bilingues.

[7]Na verdade é possível traduzir literalmente a expressão para inglês, mas a sua tradução não é uma expressão idiomática, pelo que a semântica associada não é a mesma.

[8]Sendo w_A um adjectivo, e w_S substantivo.

[9]Esta regra está descrita de uma forma simplicista, já que há excepções.

Definição 1 *Designaremos por* ***recurso bilingue*** *um qualquer objecto que contenha informação bilingue e que possa ser usado informaticamente.*

Exemplos de ***recursos bilingues*** *são os dicionários de tradução, terminologia bilingue, expressões bilingues, regras de tradução e mesmo corpora paralelos ou comparáveis.*

Embora sejam predominantemente usados para a tradução, estes recursos são úteis em muitas outras situações. O objectivo inicial desta dissertação era a investigação na área da tradução automática[10] e em particular a abordagem da tradução automática denominada por Baseada em Exemplos[11]. Esta abordagem à tradução automática é essencialmente baseada em recursos (corpora paralelos, terminologia bilingue, dicionários de tradução) e não em regras de tradução, o que levou ao estudo e desenvolvimento das ferramentas necessárias para a criação e extracção deste tipo de recursos. Constatou-se que os recursos bilingues extraídos não são úteis apenas para a tradução automática, mas também para a tradução assistida por computador, bem como para outras áreas como a aprendizagem de línguas ou a recolha de informação. Por exemplo, nesta última área existe um fórum de avaliação, denominado CLEF[12] — Cross Language Evaluation Forum — que se dedica à avaliação e comparação de sistemas de recolha de informação em diferentes línguas. Estes sistemas não precisam de incluir um tradutor completo, já que em muitos casos um conjunto de recursos bilingues é suficiente para obter bons resultados neste tipo de tarefas.

[10]Esta é uma das razões do capítulo 2 ser dedicado à Tradução. Na verdade, a Tradução é a área que mais lucra com o trabalho aqui apresentado.

[11]A secção 2.3.2 inclui uma descrição detalhada desta abordagem à tradução automática.

[12]Mais informação sobre o CLEF (*Cross Language Evaluation Forum*) pode ser encontrada em `http://www.clef-campaign.org/`. Durante a realização desta dissertação alguns dos recursos bilingues produzidos foram também usados numa participação neste fórum em 2005 (Cardoso et al., 2005).

> Os recursos bilingues são úteis para a Tradução (seja ela automática ou assistida por computador), mas também para a aprendizagem de línguas, recolha de informação, classificação automática, e outras áreas.

Para que seja possível a criação ou extracção de recursos bilingues é necessária a existência de algum outro recurso que contenha a informação que pretendemos extrair. O ponto de partida por excelência para a extracção de pontes entre duas línguas é o conjunto de todas as traduções que já foram realizadas. Qualquer corpus paralelo[13] corresponde a um ponto de partida para a extracção de recursos bilingues.

> Os corpora paralelos são a fonte por excelência de recursos bilingues.

A extracção de recursos bilingues a partir de corpora paralelos é realizada essencialmente por algoritmos de cariz estatístico. Os corpora são analisados, e são contados factos sobre cada palavra ou segmento de palavras. Estes valores são posteriormente analisados de forma estatística.

Na impossibilidade de usar a população total, a estatística recorre às técnicas de amostragem, sendo sabido que a confiança dos valores obtidos cresce de acordo com o crescimento do tamanho da amostra.

> A qualidade dos recursos extraídos é dependente da quantidade e qualidade dos corpora usados.

Por outro lado, e pela lei de Zipf (Zipf, 1949), à medida que um corpus cresce, aumenta a quantidade de novas palavras. Logo, a cobertura dos recursos obtidos irá também aumentar.

[13] Consideramos um corpus como uma colecção de textos de uma mesma língua e (habitualmente) género linguísticos. Por sua vez, um corpus paralelo pode ser visto como uma colecção de pares de textos. Cada um destes pares corresponde ao texto original e à sua tradução. Na página 69 estes dois conceitos serão definidos formalmente.

A cobertura dos recursos extraídos aumenta de acordo com o crescimento do corpus usado.

O alargamento de um corpus pode ser realizado em duas direcções: a adição de novas áreas temáticas, ou o alargamento com texto homogéneo. Enquanto que a primeira abordagem leva ao aumento da diversidade lexical, também incorpora novos relacionamentos entre palavras, aumentando a ambiguidade. Por exemplo, a junção de artigos técnicos de engenharia civil a um corpus geral da língua inglesa irá resultar em ambiguidade semântica em relação à palavra "*concrete*," já que pode ser um adjectivo (um objecto concreto) ou um substantivo (relativo a cimento).

Defendemos que se pode caminhar nas duas direcções, criando corpora de grandes dimensões para diferentes áreas temáticas. De cada um destes corpora podem ser extraídos recursos que sejam etiquetados com a área do corpus de que foram extraídos. Em caso de necessidade de maior cobertura lexical estes recursos podem ser usados numa mesma ferramenta.

Ou seja: consideremos o processamento pela função f de um conjunto c de diferentes corpora c_i, correspondentes a diferentes temas $t_i = tema(c_i)$. Podemos calcular $f\left(\bigcup_{i=1}^{n} c_i\right)$ obtendo recursos extraídos de um grande corpus multi-temático. Outra alternativa é a extracção dos recursos de cada corpora c_i, aplicando-lhes posteriormente uma função de aglutinação $g(c) = g\left(\{f(c_i)|c_i \in c\}\right)$. A solução que nos parece mais correcta e versátil corresponde ao armazenamento de um mapeamento entre temas e resultados de processamento[14] (que corresponde à etiquetação sugerida): $g(c) = \begin{pmatrix} tema(c_i) \\ f(c_i) \end{pmatrix}_{c_i \in c}$. Este recurso pode, a qualquer instante, ser adaptado dinamicamente às necessidades da ferramenta em causa[15].

[14] Considerando temas diferentes para cada corpus.

[15] A notação matemática usada está descrita no apêndice B.

> A extracção de recursos de diferentes áreas do conhecimento, de uma forma independente, deve ser preferida em relação à extracção de recursos sobre um corpus multi-temático.

Dada a preferência por corpora grandes, é necessário que ao construir protótipos para ensaiar e validar algoritmos se tenha em atenção a sua robustez e escalabilidade. É certo que estes protótipos correspondem a ferramentas em que o algoritmo está a ser afinado e melhorado, ou que não estão prontas para o uso por um utilizador final. Mas, se as ferramentas não forem robustas para processar grandes quantidades de corpora os resultados terão menos qualidade, ou poderemos mesmo estar a falsear experiências.

> Para que sirvam os nossos requisitos, as ferramentas de processamento de corpora têm de escalar de acordo com o tamanho dos corpora envolvidos.

A secção 1.1 descreve o NATools, um conjunto de protótipos desenvolvidos durante a dissertação. Estas ferramentas foram desenvolvidas de acordo com um conjunto de requisitos genéricos, como a sua disponibilização em código aberto, a criação de ferramentas pequenas, composicionais e escaláveis ao processamento de corpora reais.

O desenvolvimento de ferramentas escaláveis tem de ter em consideração a exaustão dos recursos disponíveis durante o processamento. Por exemplo, a extracção de dicionários probabilísticos de tradução obriga à criação de uma matriz esparsa de co-ocorrências, que num corpus real pode ultrapassar as 200000×200000 células. Nos computadores actualmente disponíveis uma matriz com estas dimensões não pode ser armazenado em memória RAM. O uso de uma matriz em disco poderia solucionar o problema mas iria aumentar muito o tempo de execução.

A abordagem de desenvolvimento adoptada baseia-se na divisão de uma tarefa grande num conjunto de tarefas pequenas. Enquanto que o processamento da matriz de co-ocorrências para um corpus real não

pode ser realizado de uma só vez, o seu processamento por fatias já é exequível. Depois da extracção dos dicionários de cada uma destas fatias, os dicionários são somados, obtendo um resultado semelhante ao obtido pelo processamento do corpus como um todo. Esta abordagem, que foi generalizada para várias das ferramentas desenvolvidas, é discutida no capítulo 7.

As várias abordagens de extracção de recursos de tradução a partir de corpora paralelos apresentadas nesta dissertação baseiam-se em dicionários probabilísticos de tradução (Simões, 2004). O capítulo 4 é dedicado à análise do algoritmo de extracção destes dicionários bem como à sua avaliação, que já por si constituem um recurso bilingue útil a vários níveis.

Embora estes dicionários não sejam dicionários de tradução habituais, uma vez que se baseiam na tradução observada nos corpora processados, são uma fonte de pontes ou âncoras entre duas línguas.

> Os dicionários probabilísticos de tradução constituem um relacionamento versátil entre palavras de duas línguas, que permitem a extracção de novos relacionamentos.

Para além das ferramentas de extracção de dicionários probabilísticos de tradução, foram desenvolvidas ferramentas para a extracção e generalização de exemplos de tradução e terminologia bilingue.

Um corpus paralelo e alinhado C é constituído por várias unidades de tradução ($C = TU^{\star}$). Cada uma destas unidades de tradução corresponde a uma ou mais frases e respectivas traduções. Estas unidades são frequentemente grandes, pelo que não são de fácil reutilização. O conceito de exemplo de tradução surgiu com a abordagem de Tradução Automática Baseada em Exemplos. Um exemplo de tradução corresponde habitualmente a uma sub-sequência de uma unidade de tradução (de tamanho reduzido, e com maior reutilização)[16].

[16] A problemática dos exemplos é discutida com mais detalhe na página 7

> As unidades de tradução, dado o seu tamanho habitual, são pouco reutilizáveis. Os exemplos de tradução são, por definição, mais pequenos, o que lhes permite maior reutilização.

Implementaram-se dois algoritmos de extracção de exemplos de tradução:

- Um dos algoritmos é baseado no conceito de palavra-marca: palavras que funcionam como delimitadores de sintagmas. Nesta abordagem cada unidade de tradução é dividida em segmentos tendo em conta as ocorrências das palavras-marca. Estes segmentos são posteriormente associados entre línguas utilizando os dicionários probabilísticos de tradução.
- O segundo algoritmo baseia-se apenas nos dicionários probabilísticos de tradução. Para cada unidade de tradução é construída uma matriz de alinhamento, em que cada célula é preenchida com as probabilidades de tradução para cada par de palavras. Destas células são escolhidas as com maior probabilidade de tradução para servirem de âncoras e delimitadores de segmentos, que são posteriormente extraídos.

Como já foi referido, existe um conjunto de regras intrínsecas ao conhecimento bilingue, como sejam a já referida troca entre substantivo e adjectivo na tradução da língua portuguesa para a língua inglesa. Estas regras podem ser sistematizadas formalmente, pelo que foi definida uma linguagem (*Pattern Description Language*) de definição de padrões de tradução.

> As regras básicas de tradução podem ser formalizadas com uma linguagem simples de padrões.

Estes padrões correspondem essencialmente a segmentos nominais o que levou a que a linguagem de definição de padrões fosse expandida com predicados de restrição (nomeadamente, restrições morfológicas) o

que permitem que se possa escrever padrões certeiros para a extracção de terminologia bilingue.

> A extracção de segmentos nominais é possível mediante um conjunto de padrões bilingues com restrições morfológicas.

Os exemplos de tradução (extraídos com qualquer um dos métodos apresentados) e a terminologia bilingue são mais flexíveis para a tradução automática do que as unidades de tradução. A flexibilidade dos exemplos de tradução e da terminologia pode ser aumentada aplicando uma técnica conhecida por generalização. Esta técnica corresponde à criação de conjuntos de palavras que podem ser substituídas nos exemplos de tradução. Por exemplo, consideremos o conjunto dos dias da semana e respectiva tradução. Estas palavras podem ser substituídas num exemplo que contenha um destes dias da semana, criando assim novos exemplos [17].

> A generalização de exemplos e terminologia permite aumentar a sua aplicabilidade a novas situações.

Nesta dissertação usou-se essencialmente os padrões de tradução para a criação de classes de palavras para a posterior generalização em massa de exemplos e terminologia.

Finalmente, os dicionários, exemplos e terminologia de tradução foram aplicados numa ferramenta de prototipagem de sistemas de tradução como prova de utilidade na área da tradução, e foram disponibilizados como dicionários *off-line* ou através de interfaces Web, para outros usos.

Segue-se uma secção com a descrição das ferramentas desenvolvidas. de forma a permitir uma maior compreensão dos próximos capítulos.

[17] No caso dos dias da semana seria necessário ter algum cuidado com o género das palavras substituídas, para que a concordância fosse realizada correctamente.

Segue-se a secção 1.2 onde são resumidas as contribuições da dissertação, e a secção 1.3 onde é apresentada a estrutura deste documento.

1.1 NATools: Aplicações para Extracção de Recursos de Tradução

Durante a realização desta dissertação foram desenvolvidos vários protótipos. Esta secção visa facilitar a compreensão dos capítulos seguintes, nomeadamente quando referem ferramentas.

O NATools (*Natura*[18] *Alignment Tools*) é um pacote que surgiu como uma ferramenta de extracção de dicionários probabilísticos de tradução, mas que tem vindo a incluir outras ferramentas.

O desenvolvimento destas ferramentas foi guiado por um conjunto de requisitos:

- **código aberto:** o desenvolvimento de ferramentas para uso pessoal leva a um maior desleixo no que se refere à organização do código, facilidade de compilação e instalação e mesmo na documentação. O facto de se colocar o NATools disponível levou a que vários grupos de investigação (Caseli, Nunes, and Forcada, 2005; Specia, Nunes, and Stevenson, 2005; Guinovart and Fontenla, 2004), que lidam com diferentes pares de línguas, tenham instalado as ferramentas, as tenham usado e dado *feedback* sobre as suas funcionalidades. O NATools é código aberto e livre sob licença GPL, e está disponível em `http://natools.sf.net/`.

  > A disponibilização de software de código aberto é imprescindível para obrigar a uma maior disciplina no desenvolvimento e documentação das ferramentas.

- **composicionalidade:** é importante o desenvolvimento de ferramentas pequenas, com fins específicos, que possam ser mais tarde

[18] Natura é o nome do grupo de Processamento de Linguagem Natural do Departamento de Informática, Universidade do Minho, onde esta dissertação foi realizada.

compostas em ferramentas maiores. A abordagem inversa leva ao desenvolvimento de ferramentas monolíticas que embora sejam úteis por si só, não permitem que apenas alguns dos seus constituintes seja usado num novo contexto. Por outro lado, a composicionalidade leva a que em caso de falha existam pontos de teste que permitam detectar rapidamente os componentes em falha.

> A composicionalidade de *software* permite maior reutilização das suas partes e uma maior facilidade no seu *debug*.

- **documentação:** a documentação para o utilizador final de um conjunto de ferramentas deve ser realizado a três níveis: documentar as ferramentas como um todo; documentar cada uma das ferramentas de forma detalhada; e permitir a qualquer instante obter um resumo das opções aceites pela ferramenta. Por outro lado, é importante não esquecer a documentação das API (application programming interface) disponibilizadas para permitir a expansão das ferramentas por terceiros.

> A documentação de uma ferramenta deve ter em conta os utilizadores finais mas também programadores que queiram utilizar e expandir a ferramenta.

- **escalabilidade:** como foi já discutido, é importante que as ferramentas sejam escaláveis e robustas para o processamento de corpora reais;
- **programabilidade:** as ferramentas não devem ser desenvolvidas tendo como objectivo a resolução dos problemas em mãos. Devem ser genéricas e fáceis de estender a novos objectivos e desafios.

O apêndice A descreve os passos básicos de instalação do NATools e de codificação de corpora. Este apêndice não é a documentação de todas as ferramentas disponíveis. Para isso sugere-se a consulta das páginas de manual incluídas (*man pages*) na distribuição. É sim, uma pequena introdução à preparação de corpora paralelos e extracção de dicionários probabilísticos de tradução.

O pacote NATools inclui várias ferramentas, das quais destacamos:

- um **alinhador à frase** baseado no algoritmo (Gale and Church, 1991) e na implementação de (Danielsson and Ridings, 1997). A secção 3.1.2 discute sucintamente a problemática do alinhamento de corpora paralelos ao nível da frase;
- um **extractor de dicionários probabilísticos** de tradução baseado no algoritmo descrito em (Hiemstra, August 1996; Hiemstra, 1998), que foi re-implementado com várias correcções e melhoramentos na sua eficiência (Simões and Almeida, 2003; Simões, 2004). Este extractor, bem como os dicionários obtidos, são discutidos e avaliados no capítulo 4.
- um conjunto de ferramentas integradas para a **consulta de recursos bilingues na Web**. Estas ferramentas são apresentadas com algum detalhe na secção 6.1.
- um **servidor/biblioteca** (Simões and Almeida, 2006b) de disponibilização eficiente **de recursos de tradução** (concordâncias sobre corpora, pesquisas em dicionários probabilísticos de tradução e pesquisa sobre n-gramas), multi-corpora e multi-língua. Este servidor está descrito na secção 7.3.
- uma **linguagem** para a especificação **de padrões de alinhamento** para ajuda na extracção de exemplos (ver secção 5.2) e imprescindível para a extracção de terminologia (ver secção 5.3).
- dois **extractores de exemplos**, de acordo com os algoritmos apresentados nas secções 5.1 e 5.2.
- um conjunto de ferramentas para a **generalização de exemplos**, de acordo com o discutido na secção 5.4.
- uma **API C e Perl** para o manuseamento dos vários objectos criados pelas ferramentas incluídas no pacote NATools.

1.2 Contribuições

Embora a verdadeira secção de contribuições apareça no final do documento optou-se por incluir um resumo para ajudar a leitura.

As contribuições deste trabalho podem ser divididas em três categorias: contribuições científicas, contribuições tecnológicas e recursos linguísticos:

- as contribuições científicas mais relevantes podem ser sumariadas em: análise de diferentes abordagens para a extracção de dicionários probabilísticos de tradução bem como a sua comparação, extracção de exemplos usando a Hipótese das Palavras-Marca na língua Portuguesa, extracção de exemplos por detecção da diagonal na matriz de tradução, e o uso de padrões de alinhamento para a extracção de terminologia bilingue e generalização de exemplos.
- as contribuições tecnológicas podem ser resumidas pelos dois pacotes de software abertos e livres que foram desenvolvidos durante a dissertação: o NATools e o Makefile::Parallel.
- os recursos disponibilizados são vários, desde os corpora que foram criados e filtrados, dicionários probabilísticos de tradução, terminologia bilingue, exemplos de tradução até às simples classes de palavras bilingues.

1.3 Estrutura do Documento

Este documento está estruturado da seguinte forma:

- **Capítulo 1 — Introdução**
  descreve a motivação e o trabalho realizados nesta dissertação;
- **Capítulo 2 — Tradução**
  apresenta a área da tradução, sendo ela manual, assistida por computador ou completamente automática, bem como as várias abordagens actualmente usadas para a tradução automática. São também descritas algumas ferramentas de tradução assistida por computador e de tradução automática actualmente existentes.
- **Capítulo 3 — Corpora Paralelos**
  Este capítulo caracteriza os vários corpora paralelos usados durante a dissertação, comparando-os a nível de conteúdo e de tamanho. São também apresentados métodos para o alinhamento de

corpora paralelos ao nível da frase e para a sua posterior filtragem e melhoria de qualidade.

- **Capítulo 4 — Dicionários Probabilísticos de Tradução**
  Todo o trabalho realizado na extracção de recursos é baseado em dicionários probabilísticos de tradução, trabalho iniciado durante a dissertação de mestrado (Simões, 2004). Este capítulo foi dedicado à avaliação destes dicionários e de diferentes abordagens para a sua extracção. Inclui ainda alguns exemplos de aplicação dos Dicionários Probabilísticos de Tradução para outros fins que não a extracção de recursos bilingues.

- **Capítulo 5 — Extracção de Exemplos de Tradução**
  Os vários algoritmos de extracção de exemplos de tradução e de terminologia bilingue são apresentados neste capítulo, juntamente com uma secção sobre a generalização de exemplos. Cada uma destas abordagens é acompanhada de uma pequena avaliação dos recursos obtidos.

- **Capítulo 6 — Aplicação de Recursos de Tradução**
  A avaliação de recursos não pode ser feita apenas de forma estatística: a correcção de uma amostra de exemplos de tradução não implica que esses exemplos sejam, na verdade, úteis para a tradução automática. Esta é a motivação para que neste capítulo se discuta a aplicação dos recursos extraídos para diferentes finalidades, desde a análise e consulta manual, até à sua incorporação numa ferramenta para a criação de protótipos de sistemas de tradução automática.

- **Capítulo 7 — Estratégia de Desenvolvimento**
  O desenvolvimento de ferramentas escaláveis e robustas tem de ter em consideração o tamanho dos corpora reais, pelo que a estratégia de desenvolvimento tem de ser adequada. Este capítulo discute as abordagens adoptadas para o desenvolvimento das ferramentas implementadas durante esta dissertação.

- **Capítulo 8 — Conclusões e Trabalho Futuro**
  Este capítulo descreve as contribuições desta dissertação, e sumariza os objectivos que se pretendem alcançar em trabalho futuro.

A Título de Conclusão

Os recursos de tradução são imprescindíveis para a tradução automática, tradução assistida por computador, aprendizagem de uma nova língua, recolha de informação e para muitas outras áreas do processamento da linguagem natural. Esta dissertação irá focar a extracção e avaliação destes tipos de recursos.

Dada a necessidade de corpora paralelos para servirem como matéria-prima da extracção de recursos, foram criados e analisados corpora paralelos. O tamanho destes corpora é importante dado o cariz estatístico dos algoritmos implementados. Por outro lado, a qualidade destes corpora também influencia a qualidade dos recursos extraídos, pelo que foram necessários métodos para a filtragem de corpora, de forma a aumentar a sua qualidade.

A dissertação também inclui uma abordagem técnica, que levou ao desenvolvimento de protótipos, escaláveis e robustos, para o processamento de corpora paralelos e extracção de recursos bilingues.

Os algoritmos de extracção de recursos usam como fonte de informação Dicionários Probabilísticos de Tradução, associações probabilísticas entre palavras de duas línguas diferentes. Estas relações mono-palavra permitem estabelecer pontes, e desta forma permitir a extracção de diferentes recursos, desde simples dicionários de tradução, terminologia bilingue e exemplos ou regras de tradução.

Capítulo 2

Tradução

Translation is the art of failure.

Umberto Eco

Os recursos bilingues são especialmente úteis na tradução, quer na sua vertente humana quer na sua vertente automática. Este capítulo apresenta uma visão geral de algumas das abordagens usadas na tradução.

A tradução pode ser realizada com diferentes graus de automatização: desde métodos completamente manuais, métodos assistidos por computador, até aos métodos completamente automáticos. A figura 2.1 esquematiza a relação entre os vários graus de automatização e as respectivas abordagens de tradução.

Do lado direito temos a tradução completamente manual, realizada desde os primórdios da tradução. Com a crescente banalização dos computadores foram desenvolvidas aplicações para facilitar a tarefa dos tradutores: os sistemas CAT — *Computer Aided Translation* / Tradução Assistida por Computador. Estas aplicações não pretendem substituir o tradutor, mas apoiar a sua tarefa de tradução (ver secção 2.1). Embora ainda sem resultados excepcionais (excepto entre línguas próximas,

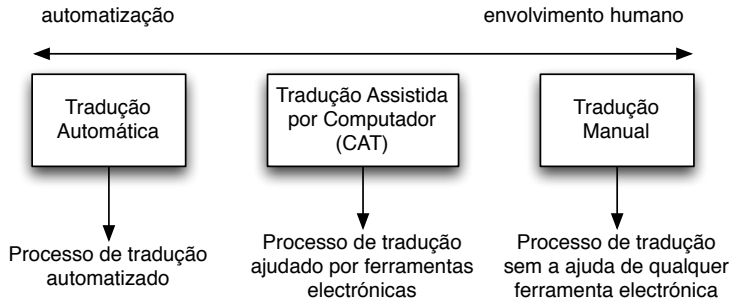


Figura 2.1: Níveis de automatização na tradução.

como o Espanhol e Português, em que os resultados têm sido bastante bons), a investigação na tradução automática (automatização máxima) tem vindo a crescer sendo esta uma área de investigação bastante activa nos últimos anos.

A secção 2.2 apresenta um breve resumo histórico da evolução desta área. A secção 2.3 detalha várias das abordagens à tradução automática actualmente usadas. A secção 2.4 descreve as principais métricas na avaliação automática da tradução automática. Por fim, a secção 2.5 descreve algumas ferramentas relevantes na área da tradução, quer na sua abordagem assistida por computador quer na sua abordagem automática.

2.1 Tradução Assistida por Computador

Alguns dos principais sistemas de tradução assistida por computador (CAT) são o SDL Trados (SDL Trados, 2006; Trados Manual, 2005), Star Transit (Transit Manual, 2006) e Déjà Vu (déjà vu Manual, 2003). Todos estes sistemas têm em comum um conjunto de funcionalidades úteis na tarefa de tradução:

- reconhecem um conjunto de formatos de documentos (como Rich Text Format, Microsoft Word, HyperText Markup Language, eXtended Markup Language) e um conjunto de formatos de recursos usados em internacionalização de software (como Xliff) o que lhes

permite abstrair o tradutor do formato específico do documento que se encontra a traduzir. O tradutor pode ignorar por completo o formato específico em que o documento original se encontra, sabendo que a tradução será gerada com o mesmo formato.

- integram-se com ferramentas de terminologia (como o Trados MultiTerm (MultiTerm, 2003)) tornando simples a pesquisa terminológica durante a tradução. Permite a construção de uma tradução termo-a-termo, sobre a qual o tradutor pode trabalhar. O uso de uma terminologia também permite que o sistema possa avisar o tradutor de que deve ter cuidado na sua tradução. Uma tradução com a terminologia mal traduzida é difícil ou impossível de entender, mas uma tradução com a terminologia bem traduzida e possíveis erros de tradução no restante texto é entendível.
- guardam todas as traduções já realizadas pelo tradutor numa base de dados (memória de tradução) para reutilização posterior. Permitem também realizar concordâncias sobre as traduções já realizadas para que o tradutor possa reutilizar manualmente determinadas traduções.

O trabalho desenvolvido durante esta dissertação é útil num sistema CAT, uma vez que foram desenvolvidos métodos para a extracção automática de terminologia e de exemplos de tradução. A secção seguinte detalha o algoritmo usado pelos sistemas de ajuda à tradução (tradução baseada em memórias de tradução) o que permitirá explicitar como estes sistemas podem tirar partido dos recursos criados.

2.1.1 Tradução baseada em Memórias de Tradução

A tradução baseada em memórias de tradução tem como principal objectivo a reutilização de traduções anteriormente realizadas. Uma memória de tradução é uma base de dados de segmentos traduzidos (unidades de tradução) que permitem ao tradutor:

- propagar no texto de destino as traduções de frases que se repetem no texto original;
- reciclar traduções que foram realizadas noutros projectos, podendo

reutilizá-las tal como armazenadas na memória de tradução, ou depois de alteradas;

- analisar um novo texto original e encontrar segmentos cujas traduções se encontram armazenadas na memória de tradução, permitindo desta forma reutilizar porções de traduções já realizadas;

O processo de tradução usando memórias de tradução realiza-se da seguinte forma (de acordo com a figura 2.2[1]):

1. O programa divide o texto original em segmentos. Esta divisão é feita tendo em conta a pontuação da língua em causa, e a marcação do formato específico em que o documento se encontra;
2. A tradução é realizada para cada segmento do texto de origem pela sua ordem natural, de acordo com os seguintes passos:
   (a) o programa verifica se o próximo segmento a ser traduzido está na memória de tradução, ou se algum segmento razoavelmente semelhante já foi traduzido;
   (b) o tradutor determina se vai usar, editar ou ignorar a tradução que o programa encontrou;
   (c) o programa guarda o segmento da língua de origem e a respectiva tradução na memória de tradução;

O uso de memórias de tradução aumenta a produtividade (quando o tipo de texto é adequado: repetitivo e com actualizações frequentes) facilitando a reutilização de traduções, e um controlo manual sobre a qualidade da tradução. Existe um conjunto de desvantagens que se deve ter em conta:

- os erros anteriores que possam ter sido inseridos na memória de tradução são propagados: o tradutor esquece-se de actualizar a memória de tradução;
- o texto traduzido pode resultar numa "salada de frases" (Bédard, 2000), um texto menos coerente ou inteligível, já que o tradutor é confinado à tradução ao nível da frase, tenta maximizar a reu-

[1]Neste esquema e seguintes, L.O. abrevia "Língua Origem" e L.D. abrevia "Língua de Destino," respectivamente "source language" e "target language" na língua inglesa.

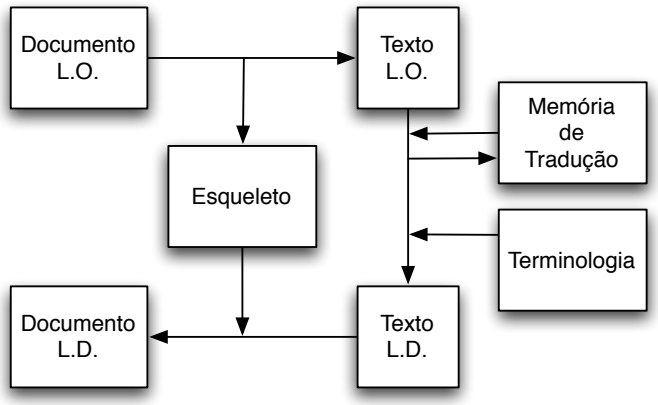


Figura 2.2: Fluxo de tradução num sistema CAT.

tilização de traduções e pode utilizar memórias de tradução com textos de várias áreas e/ou traduzidos por diferentes tradutores (Bowker and Barlow, 2004).

- as similaridades encontradas pelo sistema são na forma (escrita semelhante) e não na semântica;
- torna o tradutor menos ágil (Kenny, 2004) já que tenta reutilizar tudo o que pode.

Embora esta abordagem à tradução tenha algumas desvantagens, esta é a forma actualmente mais usada para a tradução de texto jurídico, legislativo, técnico e científico. A única área em que a tradução tem vindo a ser quase totalmente manual é a literária, já que a prosa ou poesia literária não permite tirar grande partido das ferramentas de tradução assistida por computador.

2.2 Um pouco de História da Tradução Automática

Esta secção conta um pouco da história e evolução da tradução automática. Alguns conceitos importantes foram introduzidos logo nos primeiros tempos da investigação nesta área.

A história da tradução automática foi influenciada por vários factores dos quais salientamos as limitações no poder computacional, e imposições politico-económicas. Enquanto que o primeiro factor tem permitido a evolução da investigação nos últimos tempos, o segundo decidiu especialmente as línguas em que mais se investiu na tradução automática: nos anos 50 e 60 o interesse dos Estados Unidos nos avanços tecnológicos russos levou a que se encorajasse a investigação na tradução russo–inglês; mais recentemente, o facto do Canadá ser um país bilíngue, e da União Europeia ter de gerir legislação nas várias línguas dos países aderentes, têm vindo a fomentar a investigação na tradução automática entre as línguas envolvidas.

2.2.1 Os primórdios da Tradução Automática

Em meados de 1930, o russo Petr Troyanskii (Hutchins, 2005) fez a primeira proposta para um método automático de tradução, baseada num esquema de codificação de regras gramaticais inter-linguísticas (baseadas em Esperanto), bem como uma especificação de como a análise do texto de origem, e a síntese na língua destino deveriam ser feitas. Nesta altura o computador ainda não tinha nascido pelo que o trabalho de Troyanskii ficou esquecido até há bem pouco tempo.

Em Julho de 1949, Warren Weaver fez uma das primeiras referências à tradução automática. Depois dos grandes sucessos no uso de computadores para quebrar códigos durante a segunda grande guerra, Warren via um sistema de tradução como um sistema de *codificação*:

> *"When I look at an article in Russian, I say: This is really written in English, but it has been coded in some strange symbols. I will now proceed to decode."*

Ou seja, Warren defendia que o processo de tradução podia ser visto como um processo de *codificação*, substituindo símbolos (palavras) de uma língua, em símbolos de outra língua.

Os sistemas baseados nesta premissa eram primitivos: procuravam num dicionário bilingue cada palavra da frase a traduzir, substituindo-a na frase original pela tradução. No caso de o dicionário ter mais

do que uma tradução possível, todas eram impressas. O método era insatisfatório, e depressa surgiram tentativas para re-arranjar o texto depois de traduzido.

2.2.2 A primeira conferência da área

Embora os sistemas fossem básicos, urgiram vários projectos de tradução automática em muitas universidades nos EUA. A primeira conferência sobre tradução automática remonta a 1952, realizada em Junho no Instituto de Tecnologia de Massachusetts (Hutchins, 1997).

Esta conferência foi organizada por Yehoshua Bar-Hillel que tinha sido designado pela MIT para estudar o estado-da-arte da área. Este estudo (Bar-Hillel, 1951) serviu de base à organização da conferência, e foi o primeiro trabalho genérico sobre a área. Infelizmente as actas desta conferência não chegaram a ser publicadas, nomeadamente porque apenas dois dos artigos correspondiam ao conteúdo apresentado durante a conferência.

Embora se tenha evoluído imenso desde esta altura, é interessante reparar que os problemas relatados na altura continuam a fazer parte da investigação actual.

Pré-edição

As primeiras propostas para o uso de um nível de pré-edição e pós-edição foram feitas por Erwin Reifler, no início de 1950. A tradução era vista apenas como um sistema de substituição de palavras, o que não permitia produzir uma tradução legível. Assim, ou o pós-editor iria ter de escolher a tradução correcta em determinado contexto de entre uma lista de possíveis traduções, ou seria necessário um passo de pré-edição (Reifler, 1952b) onde as palavras fossem desambiguadas morfologicamente e semanticamente.

Como o envolvimento humano era demasiado, e por vezes mais complicado do que a própria tradução, Reifler propôs várias soluções para este problema. A mais simples consistia num mecanismo de auxílio à

inserção de códigos correspondentes às categorias e significados das palavras. Soluções mais criativas incluíam o uso de uma nova forma de ortografia em que as categorias gramaticais seriam distinguidas pela sua forma gráfica: os substantivos levariam a primeira letra em maiúscula, os verbos principais a segunda, os adjectivos a terceira, e assim por diante (por exemplo, a frase alemã "*er hegt die fromme Hoffnung*" seria escrita como "*er hEgt die frOmme Hoffnung*").

Talvez tenha sido pela excentricidade das propostas de Reifler que durante os anos que se seguiram nenhum investigador fez considerações sérias sobre o uso de uma etapa de pré-edição. Muitos anos passaram até que se propusesse o uso de uma linguagem normalizada para textos de uma área contida e controlada.

Pós-edição

A tradução que era realizada palavra a palavra, resultava num conjunto de palavras na língua de destino que tinham de ser reordenadas pelo pós-editor. Esta ordenação pressupunha o conhecimento da frase original para que se pudesse manter a sua semântica. O conhecimento das palavras traduzidas (sem qualquer ordem) não era suficiente para determinar qual teria sido a frase original.

Mais tarde, e com base em técnicas de análise estatística de contextos (Kaplan, 1950), à base de análise de bigramas e trigramas, Bar-Hillel chegou à conclusão de que a pós-edição poderia ser feita por pessoas que conhecessem apenas a língua de destino (Bar-Hillel, 1952a).

Termos Multi-palavra e Expressões Idiomáticas

Além da ordem estranha das palavras traduzidas, outro dos problemas que Bar-Hillel (Bar-Hillel, 1952c) refere é a tradução de termos multi-palavra e expressões idiomáticas que, na melhor das hipóteses, seriam traduzidas palavra a palavra, e não como uma unidade. Bar-Hillel propôs três abordagens:

- a adição de novas traduções nos dicionários de tradução mono-

palavra, de forma a que os termos multi-palavra e as expressões idiomáticas acabassem por ser bem traduzidas ao realizar uma tradução palavra-por-palavra. Embora esta abordagem gerasse traduções correctas, também gerava um conjunto de outras erradas. Assim, ao traduzir uma expressão normal, as traduções de palavras para expressões idiomáticas também iriam ser usadas, pelo que o pós-editor teria de saber qual a expressão a escolher;

- a criação de um dicionário de expressões multi-palavra que pudesse ser usado para os termos multi-palavra e para as expressões idiomáticas. O pós-editor só teria de reconhecer as situações em que se tratava de uma expressão idiomática. Nesta abordagem, Bar-Hillel tinha especial receio do tamanho que estes dicionários poderiam vir a ganhar, já que não fazia ideia de quantas entradas o dicionário poderia vir a ter;
- dar toda a responsabilidade de detecção de expressões idiomáticas ao pós-editor, que sem qualquer ajuda automática deveria decidir se se tratava ou não de uma expressão idiomática. Esta abordagem esbarrava numa das ideias de Bar-Hillel: o pós-editor não deveria precisar de saber a língua original do documento.

Ao apresentar estas abordagens, Bar-Hillel estava a reconhecer a necessidade do tratamento de colocações semânticas e desambiguação contextual.

Linguagem Controlada

Se as ideias de pré-edição de Reifler eram olhadas com cepticismo, o mesmo acontecia com a sua defesa do uso de uma linguagem controlada para a escrita dos textos de origem.

(Dodd, 1952) propôs uma forma simplificada de inglês, para ser usada quer como língua de origem, quer como língua de destino. Esta simplificação consistia na regularização das formas verbais ("*She did be loved*" em vez de "*She was loved*"), o uso apenas das formas nominativas dos pronomes ("*I will send he to she*" em vez de "*I will send him to her*"), a regularização da ordem pelas quais as palavras devem ocorrer (advérbios antes de substantivos, objectos directos antes dos indirec-

tos) e, claro, o uso das palavras na sua forma (significado) mais comum ("*tank*" apenas para tanque de água, e sobre-especificar para outros significados, como "*army tank*"). Embora rígida, esta abordagem pode ser vista como o predecessor de outras abordagens usadas em diversos contextos na actualidade (Elliston, 1979; Pym, 1990; Hayes, Maxwell, and Schmandt, 1996).

Com esta abordagem, simplificava-se o sistema de tradução, que iria gerar uma linguagem também controlada. A pós-edição também seria simplificada já que em princípio o significado da expressão era mantido, sendo necessário ao revisor apenas re-escrever pequenas porções de texto.

Sistema de Tradução Universal

Bar-Hillel escreveu no seu artigo de 1951 que a tradução automática genérica, envolvendo mais do que uma língua de destino, iria precisar de uma gramática universal ou, pelo menos, bastante genérica.

(Reifler, 1952a) afirmou que, embora uma gramática universal fosse difícil de obter, deveria ser possível criar gramáticas pseudo-universais derivadas de línguas com gramáticas bastante similares. Propôs o uso de gramáticas de alinhamento que não eram mais do que mapeamentos entre marcas gramaticais que eram traduzidas juntamente com as palavras para a língua de destino desejada. Esta noção era muito próxima às ideias de gramáticas de transferência (Harris, 1954) e das propostas mais recentes de gramáticas isomórficas (Landsbergen, 1987) em sistemas baseados em *interlíngua* (ver a secção 2.3.1).

Sub-linguagens

Bar-Hillel mencionou as linguagens restritas (ou sub-linguagens, como a usada pelos pilotos de avião), como boas áreas para a aplicação de técnicas de tradução automática.

Oswald e Bull demonstraram que numa área de conhecimento restrita (no caso concreto, cirurgias ao cérebro) com um léxico diminuto,

as possíveis palavras ambíguas passam a ter um único significado. Como um resultado da sua investigação, Oswald propôs o uso de micro-glossários, em que cada palavra não deveria ter mais do que uma tradução possível na língua de destino. No seu estudo sobre frequências de palavras, Oswald reparou que não só os termos técnicos tinham uma frequência elevada, mas também que os cirurgiões escreviam os artigos usando um conjunto de construções frásicas restrito, e mesmo um número pequeno de substantivos não-técnicos.

Cedo se chegou à conclusão que o uso de micro-glossários não seria suficiente para resolver o problema da ambiguidade. (Bull, 1952) defende que não existe nenhum método de criar um vocabulário limitado, que permita traduzir uma percentagem razoável de conteúdos: um micro-glossário só servirá para um micro-assunto, uma área em que o número de entidades envolvidas e de acções possíveis seja extremamente limitado.

Actualmente sabemos que o uso de sub-linguagens só nos resolve problemas da tradução automática em que se pretende traduzir pequenos textos de domínio específico, já que poucas são as áreas de conhecimento escritas estritamente numa única sub-linguagem.

Uso de métodos estatísticos

(Bull, 1952) realça um dos problemas da tradução automática que se estende até aos dias de hoje:

> *"The limitations of machine translation which we must face are, vocabularywise, the inadequacy of a closed and rigid system operating as the medium of translation with an ever-expanding, open continuum".*

Todos os participantes chegaram à conclusão de que o estudo estatístico da língua era um dos pontos fundamentais para o sucesso da tradução automática. No entanto, a nenhum dos participantes ocorreu o facto de poderem usar os computadores para fazer a análise estatística da língua.

Análise Gramatical

(Bar-Hillel, 1952b) estava convencido de que para se avançar em relação à tradução palavra-a-palavra seria necessário a análise sintáctica, pelo que defendeu o desenvolvimento de "gramáticas operacionais" para identificar e desambiguar categorias gramaticais, bem como para analisar estruturas sintácticas.

Durante a conferência, Oswald descreveu como se podiam identificar "blocos sintácticos" (sintagmas nominais e verbais) com base em "marcadores": pontuação, artigos, substantivos, formas verbais, advérbios, pronomes relativos, etc. (Oswald, 1952). Na verdade, Oswald estava a basear-se na teoria de "análise de constituintes" já familiar aos linguistas (Harris, 1946; Wells, 1947). O que de facto era novo, era a possibilidade dos métodos de Oswald poderem ser formulados como instruções para um computador, apesar de não terem sido implementados na altura.

Durante os anos que se seguiram não houve muito desenvolvimento nas gramáticas para tradução automática (as propostas de Harris e mesmo as gramáticas transformacionais de Chomsky não foram tomadas em conta para esta área). Só mais recentemente, (Wood, 1993) voltou a falar em gramáticas de unificação, e o seu potencial uso na tradução automática.

Língua Pivot ou Interlíngua

No fim da conferência, Dostert sugeriu que a tradução automática de uma língua para várias deveria ser pensada de forma a que primeiro se traduzisse para uma língua intermédia — língua pivot (sugerindo um sub-conjunto da língua inglesa) — e dessa para as línguas desejadas. Durante a discussão foram mencionadas outras possíveis línguas pivot: o Esperanto ou línguas simplificadas (Inglês simplificado de Dodd).

No entanto, nesta altura não foi assumido de que uma língua intermédia (interlíngua) deveria ser independente de qualquer língua. Actualmente, sabemos que é impossível a criação de uma interlíngua para todas as línguas, sendo possível apenas para línguas próximas (Santos, 1996).

2.2.3 Evolução e Relatório ALPAC

A primeira demonstração pública de um sistema de tradução automática foi realizada em 1954, numa colaboração da IBM com a universidade de Georgetown. O sistema usava um vocabulário de apenas 250 palavras Russas, apenas seis regras de gramática, e um conjunto bem escolhido de frases simples em russo. Embora o sistema demonstrado não tenha valor científico, encorajou a crença de que a tradução usando um computador tinha sido resolvida, e só faltavam pormenores de natureza técnica, o que estimulou o início de vários projectos de tradução automática por todo o mundo.

Estes novos sistemas consistiam essencialmente em dicionários bilingues enormes, onde cada palavra da língua de origem era mapeada numa ou mais palavras equivalentes na língua de destino, e em algumas regras gramaticais para produzir resultados com as palavras na ordem correcta. À medida que se tentou obter melhores resultados, o número de regras tornou-se imensurável, e tornaram-se demasiado complexas, o que levou à necessidade de métodos sistemáticos para a análise sintáctica.

Durante cerca de uma década que a investigação continuou até começar a surgir alguma desilusão, quando se começou a encontrar barreiras semânticas para as quais não se viam soluções práticas. Existiam vários sistemas funcionais, mas a qualidade de tradução era desmotivante.

Em 1964 o governo dos EUA começou a preocupar-se com a falta de progresso na área da tradução automática, e a Fundação Nacional para a Ciência instituiu o Comité para o Aconselhamento do Processamento Automático da Língua (ALPAC – *Automatic Language Processing Advisory Committee*) para avaliar a falta de progresso nesta área. Este comité concluiu em 1966, num famoso relatório que se tornou conhecido como "*ALPAC Report*," (ALPAC, 1966) de que:

- a tradução automática é menos precisa e duas vezes mais cara do que a tradução realizada completamente por humanos;
- não existe prospecção de utilidade da tradução automática num futuro imediato;

- se devia investir em ferramentas para o apoio à tradução manual.

Estas conclusões levaram a que as instituições públicas perdessem o financiamento e portante, todo o interesse na investigação em tradução automática.

(Bar-Hillel, 1960) não duvidava de que os métodos de análise sintáctica poderiam ser muito melhorados com a ajuda de teoria linguística, mas também estava convicto de que os problemas semânticos nunca poderiam vir a ser completamente resolvidos pelo que tradução automática com qualidade seria impossível.

2.2.4 Investigação pós ALPAC

Embora o relatório ALPAC tenha diminuído o interesse na área, alguma investigação continuou no Canadá, França e Alemanha. Os seus objectivos tornaram-se mais realísticos: deixou-se de procurar traduções estilicamente perfeitas mas sim legibilidade e fidelidade ao original.

Foram surgindo sistemas mais avançados, baseados em abordagens indirectas, e foi aumentando a variedade de línguas envolvidas. Apareceram projectos privados como o sistema Logos (1969) (Scott, 2003), Weidner-CAT (1977) e o ALPS (1980). Também foi nessa altura que o sistema Systran(Toma, 1977a; Toma, 1977b) foi instalado para uso da Força-Aérea Norte-Americana (1970), e pouco depois para a Comissão das Comunidades Europeias para traduzir os grandes volumes de documentação (1976).

Destes sistemas, o Systran foi (e continua a ser) um dos maiores sistemas de tradução. Nos anos 70, o processo de tradução do Systran baseava-se em cinco passos básicos: entrada, pesquisa inicial no dicionário, análise, transferência e síntese. Embora com cinco etapas distintas, o Systran continuava a ser um sistema de tradução directa (ver secção 2.3.1): os programas de análise e síntese eram desenhados para pares específicos de línguas. Durante o tempo, foi adquirindo propriedades de um sistema de transferência (ver secção 2.3.1), já que os processos de Análise, Transferência e Síntese se tornaram claramente independentes.

O sistema Logos apareceu com o objectivo de traduzir manuais de aviões americanos para Vietnamita. Tal como o Systran, o Logos tem uma separação completa das etapas de análise e síntese pelo que, embora os seus procedimentos fossem desenhados para um par de línguas específico, os programas eram adaptáveis para novos pares. Em comum com quase todos os sistemas modernos, não existe confusão entre processos de programação e dados e regras linguísticas.

Os sistemas que adoptaram a abordagem "indirecta" foram bastante influenciadas por teorias linguísticas. A possibilidade de traduzir usando uma língua intermediária "universal" (sistemas *interlíngua*, ver secção 2.3.1) já tinha sido sugerida por Weaver no seu memorando mas só em 1960 é que surgiram os primeiros modelos.

Entretanto foi desenvolvida uma aplicação com grande sucesso na tradução automática, o Météo (Chandioux, 1976). Foi fundado em 1975 na Universidade de Montreal, no Canadá, com o objectivo de traduzir automaticamente previsões meteorológicas de Inglês para Francês. A especificidade da aplicação, e o pequeno conjunto de terminologia e construções gramaticais necessários ajudaram ao sucesso deste projecto.

Nos anos 80 o interesse pela investigação em tradução automática foi renovado, devendo-se especialmente à criação de instituições bilingues e multilingues (de que a União Europeia é um exemplo), bem como devido à globalização e necessidades comerciais de empresas multinacionais.

(Berger et al., 1994) publicou resultados de experiências realizadas num sistema baseado em métodos estatísticos (secção 2.3.2). Pela mesma altura começaram-se a usar métodos baseados em corpora de exemplos de tradução (corpora paralelos), usando a abordagem a que hoje se chama "tradução baseada em exemplos" (secção 2.3.2). Estas duas abordagens diferenciaram-se das anteriores no facto de não usarem regras sintácticas ou semânticas, mas apenas informação estatística obtida de grandes quantidades de corpora paralelos.

Na frente de investigação, as principais áreas de crescimento têm vindo a ser observadas na tradução automática baseada em exemplos, e na tradução baseada em estatística, e no desenvolvimento de tradução de fala para domínios específicos.

2.3 Abordagens na Tradução Automática

No desenvolvimento de *software*, as abordagens podem tender a usar estruturas de dados mais complexas com um algoritmo simples, ou estruturas de dados simples e algoritmos mais complexos. Também nas arquitecturas de sistemas de tradução automática se pode observar esta dicotomia:

- **tradução baseada em regras:** estes sistemas são os mais comuns nas ferramentas comerciais, e também os primeiros a surgir. Normalmente são classificados como: sistemas directos, sistemas baseados em *interlíngua* e sistemas de transferência[2].
- **tradução baseada em dados:** baseiam-se em textos já traduzidos (corpora paralelos e memórias de tradução) e destes extraem a informação necessária para realizar a tradução. Dividem-se em Sistemas de Tradução por informação estatística (SMT/SBMT) e em Sistemas de Tradução baseados em Exemplos (EBMT).

2.3.1 Tradução baseada em Regras

Os sistemas de tradução evoluíram de sistemas monolíticos de tradução directa, para sistemas baseados em regras de transferência, mais usados actualmente. Esta secção apresenta algumas das abordagens baseadas em regras: tradução directa, interlíngua e regras de transferência.

Tradução Directa

Os sistemas mais simples de tradução, pertencentes à primeira geração de tradutores automáticos, são os sistemas de tradução directa. A figura 2.3[3], esquematiza este tipo de tradução.

Estes sistemas são desenhados para um par específico de línguas, o que obriga à re-escrita completa do sistema para a adição de novas

[2]Conhecidos por *rule-based translation systems* ou *transfer-based systems*.

[3]Esta e as figuras seguintes sobre modelos de tradução foram adaptadas de (Hutchins, 1986).

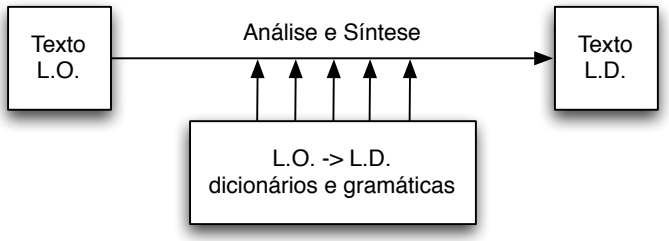


Figura 2.3: Sistemas de Tradução Directa.

línguas. A tradução é feita de forma directa, sem uma representação intermédia. O texto de origem é analisado minimamente, apenas para a resolução de ambiguidades, a identificação das traduções mais apropriadas, e a escolha da ordem de palavras no texto de destino. A análise sintáctica é desenhada de forma a fazer pouco mais do que o reconhecimento de classes de palavras (substantivos, verbos, adjectivos,...) de forma a tratar palavras homógrafas.

(Garvin, 1972) chama-lhe o método da "*força bruta*": um programa é escrito para um corpus específico, testado noutro corpus, corrigido e melhorado, testado com um corpus maior, corrigido de novo, e assim sucessivamente. O resultado é um programa monolítico complexo, sem separação clara entre as partes que analisaram o texto de origem e as partes que geraram o texto de destino. Toda a informação sobre a gramática das línguas envolvidas é incorporada na própria estrutura do programa, tornando difícil qualquer modificação ao sistema.

As vantagens deste método, para além da sua robustez, resumem-se a precisar de poucos recursos: um dicionário bilingue, e algum conhecimento rudimentar da língua de destino. Como desvantagens salientamos o facto da tradução gerada ser de fraca qualidade dado o modelo de tradução realizado quase palavra a palavra, bem como a dificuldade de manutenção e de adição de novas línguas.

Os sistemas de tradução directa tiram partidos de recursos bilingues como sejam dicionários de tradução ou terminologia bilingue.

Interlíngua

Os sistemas *interlíngua* tentam abstrair qualquer língua numa representação intermédia (a que chamam *interlíngua*). Como se pode ver na figura 2.4[4], o texto original é convertido numa representação intermédia que é posteriormente convertida em texto na língua de destino.

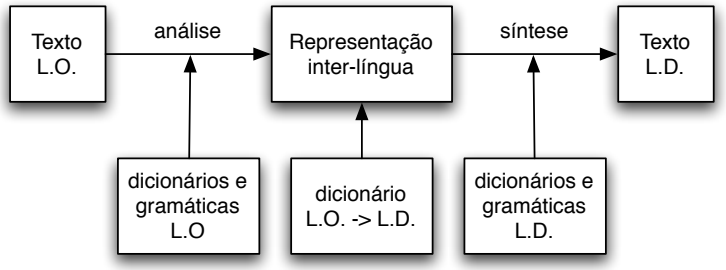


Figura 2.4: Sistemas de Tradução *interlíngua*.

Nestes sistemas, o processo de análise e síntese são completamente independentes, usando dicionários e gramáticas separados para a língua de origem e língua destino. Em teoria, este processo facilita a adição de novas línguas. Para cada nova língua de origem só é necessário adicionar um conversor para a representação intermédia e, para cada nova língua de destino, adicionar um gerador a partir da representação intermédia.

Os adeptos desta abordagem argumentam que, enquanto que as línguas diferem muito à "superfície," partilham uma estrutura interna comum: em qualquer língua duas formas que são equivalentes em significado à superfície (p. ex. paráfrases) são derivadas da mesma estrutura interna. No entanto, é muito difícil (ou mesmo impossível) de encontrar verdadeiras representações intermédias que possam ser usadas com qualquer par de línguas.

[4]Embora se possa argumentar a inexistência de um dicionário entre a língua de origem e de destino neste modelo, a grande dificuldade na criação de uma representação *interlíngua* pura leva a que muitas vezes o processo de análise seja só parcial, e portanto, exista a necessidade de mapear palavras da língua de origem na língua de destino.

> Embora em teoria os sistemas interlíngua usem dicionários separados para cada língua, as suas implementações tiram partido de dicionários de tradução e de terminologia bilingue.

Regras de Transferência

A abordagem *interlíngua* era demasiado ambiciosa. A abordagem baseada em regras de transferência é, sem dúvida, mais cautelosa, realística, flexível e adaptável. Na abordagem baseada em regras de transferência quer a língua de origem quer a língua de destino têm a sua própria representação interna (ver figura 2.5).

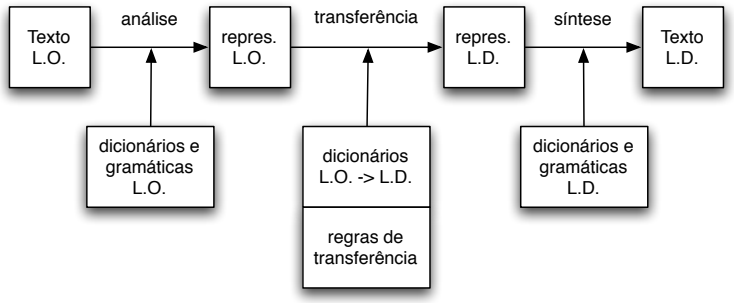


Figura 2.5: Sistemas de Tradução baseados em Transferência.

A tradução processa-se em três fases:

- **análise** do texto de origem e conversão de cada frase numa representação abstracta. Esta representação deve resolver as ambiguidades léxicas e sintácticas da língua de origem. Não é feita qualquer análise à possibilidade destas palavras poderem ter mais do que uma tradução na língua de destino;
- **transferência**: são utilizadas regras (denominadas de regras de transferência) para converter a representação abstracta da língua de origem na representação abstracta da língua de destino. São também utilizados dicionários bilingues para realizar a "transferência" entre o léxico da língua de origem para a língua de des-

tino. Esta divisão corresponde à separação ideal do módulo de transferência léxica do módulo de transferência estrutural.

- **síntese** da representação abstracta da língua de destino num texto.

A profundidade da análise sintáctica realizada nestes sistemas é bastante mais superficial, do que a dos ambiciosos sistemas *interlíngua*. A análise semântica é restrita à resolução de homógrafos e testes da coerência semântica das potenciais análises sintácticas.

Assim como a tradução *interlíngua*, este método privilegia a modularidade do sistema de tradução: abstractores, geradores e conversores. Embora os sistemas de abstracção e de geração possam ser reaproveitados para diferentes pares de línguas, o componente de transferência tem de ser dedicado a determinado par de línguas e direcção de tradução.

Os recursos necessários a um sistema de tradução baseado em regras são: gramáticas monolingues para cada uma das línguas envolvidas e de dicionários bilingues.

> Os sistemas de tradução baseados em regras de transferência tiram partido de dicionários de tradução, terminologia bilingue e de padrões de tradução.

A figura 2.6 esquematiza o processo de tradução de acordo com as várias abordagens até aqui discutidas. Se considerarmos que cada um dos vértices inferiores correspondem à língua de origem e língua de destino respectivamente, a base do triângulo pode ser vista como o processo de tradução directa, sem qualquer tipo de análise: uma tradução baseada em memórias de tradução.

Por sua vez, as duas outras arestas correspondem aos passos de análise e geração. No caso dos sistemas *interlíngua* pretendia-se que a análise fosse total, passando pelo terceiro vértice (linguagem intermédia). No entanto, a tradução por regras de transferência correspondem às setas intermédias: é feita alguma análise ao texto de origem (a quantidade de análise depende da frase e do sistema em causa), é usada uma regra de transferência, e é realizada alguma geração, correspondente à análise realizada originalmente.

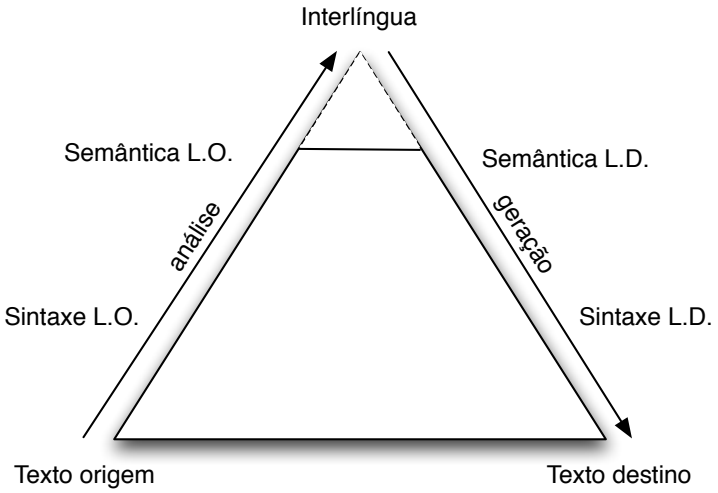


Figura 2.6: Interlíngua versus Sistemas de Transferência.

Um dos exemplos mais antigos de um sistema baseado em regras de transferência é o TAUM, um projecto da Universidade de Montreal, um sistema de tradução de Inglês para Francês, suportado pelo governo Canadiano desde meados de 1960. Existem outros sistemas baseados em regras, como o projecto de tradução de Russo para Alemão, da Universidade de Saarbrücken, que começou em 1967.

2.3.2 Tradução baseada em Dados

As abordagens baseadas em dados dão mais importância a textos paralelos e a recursos bilingues do que a regras. Originalmente surgiram dois principais métodos (que actualmente não se distinguem): a tradução automática estatística, e a tradução automática baseada em exemplos.

A tradução automática estatística (ou baseada em estatística — SMT/SBMT[5]) usa corpora paralelos para extrair factos e propriedades estatísticas sobre a tradução das várias palavras. São estes factos e propriedades que são usados posteriormente durante a tradução.

[5] *statistical based machine translation*

A tradução automática baseada em exemplos (EBMT[6]) foi inspirada numa citação de (Nagao, 1984), em que refere uma analogia entre a forma de tradução humana (*translation-memory based machine translation*) com a tradução automática:

> *"Man does not translate a simple sentence by doing deep linguistic analysis, rather, man does translation, first, by properly decomposing an input sentence into certain fragmental phrases, then by translating there phrases into other language phrases, and finally by properly composing these fragmental translations into one long sentence. The translation of each fragmental phrase will be done by the analogy translation principle with proper examples as its reference"*

Os sistemas EBMT usam corpora paralelos alinhados (ao nível da frase, ao nível do segmento e ao nível da frase) para realizar a tradução. A estes corpora são chamados bases de exemplos.

As duas secções que se seguem pretendem caracterizar cada um destes dois modelos na sua definição original. Actualmente, os sistemas estatísticos tiram partido de ambas as abordagens, pelo que já não faz sentido esta divisão.

Sistemas de Tradução Estatísticos

Os sistemas SMT extraem informação estatística de corpora paralelos (como sejam dicionários probabilísticos, cadeias de Markov, n-gramas, etc.) que é usada durante o processo de tradução das traduções obtidas, a melhor é escolhida de acordo com um modelo de língua (Knight, 2004; Knight and Koehn, 2004; Koehn, 2006).

A tradução SMT pode ser vista como a maximização de duas variáveis estatísticas: a probabilidade de uma frase ser tradução da outra, e a probabilidade da tradução fazer parte das frases válidas na língua de destino.

[6] *example based machine translation*

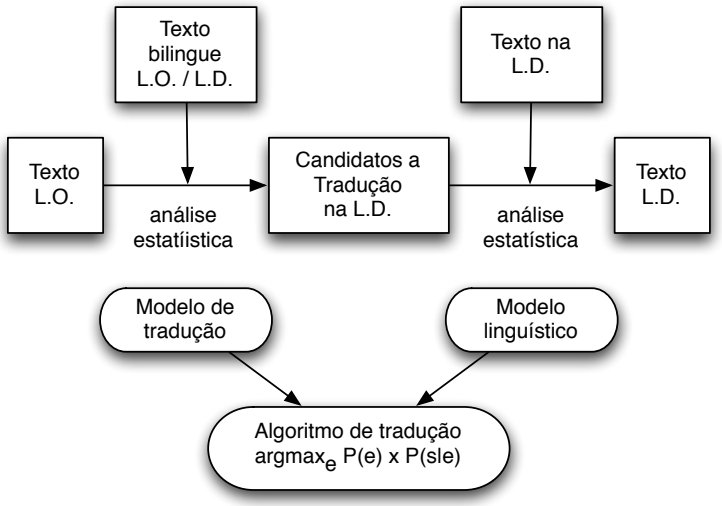


Figura 2.7: Sistema de Tradução Estatístico.

O modelo de tradução é uma variável estatística de probabilidade de, dado um par de frases $\langle f, e\rangle$, e ser tradução de f. Esta probabilidade $P(f|e)$ é elevada se e se parece com uma tradução de f, e baixa caso contrário.

O modelo da língua de destino (por exemplo, inglês) é usado para calcular a probabilidade de determinada frase pertencer a essa língua. Dada uma frase inglesa e, podemos calcular $P(e)$ tal que: se e é uma frase em inglês correcto, $P(e)$ é elevado; se e é uma frase incorrecta, $P(e)$ é baixo.

O sistema de tradução apenas precisa de, dado um modelo de língua, um modelo de tradução e uma frase f, encontrar a tradução e que maximize $P(e) \times P(f|e)$.

Consideremos o exemplo de traduzir a frase *"Que fome eu tenho"* para inglês:

- a primeira etapa passa por gerar todas[7] as traduções possíveis para

[7]Na verdade não se geram todas as traduções possíveis, tentando-se analisar quais as mais prováveis. Deste modo não necessitam de percorrer todo o espaço de frases

esta frase, quer façam ou não sentido na língua de destino:

"What hunger have I"
"Hungry I am so"
"I am so hungry"
"Have I that hunger"

Note-se que estas traduções não podem ser vistas como paráfrases, uma vez que são geradas apenas com combinações das várias possíveis traduções das palavras na língua original.

- posteriormente, usando o modelo de língua, escolhe-se a frase que mais se parece com inglês correcto, ou seja, *"I am so hungry."*

Esta abordagem pretende obter resultados fluentes já que guia a escolha de palavras e a sua ordem por um modelo de língua. Em especial, este modelo é habitualmente estimado usando corpora monolingue adicional (biliões de palavras), calculando trigramas de palavras que são posteriormente utilizados para o cálculo das probabilidades de determinada frase pertencer ao modelo de língua:

$$\begin{aligned} p(\text{A Maria chorou}) &= p(\text{A}|\text{START}) \\ &\times\ p(\text{Maria}|\text{START}, \text{A}) \\ &\times\ p(\text{chorou}|\text{A}, \text{Maria}) \end{aligned}$$

> Os sistemas de tradução estatísticos tiram partido de recursos bilingues como sejam dicionários probabilísticos de tradução, ou terminologia bilingue probabilística.
> Este sistemas também usam n-gramas (trigramas e tetragramas) para a construção de modelos de língua.

Sistemas de Tradução Baseados em Exemplos

(Somers, 1999) enuncia três critérios cada vez mais restritivos que caracterizam um sistema de EBMT:

que, em muitos casos, seria infinito.

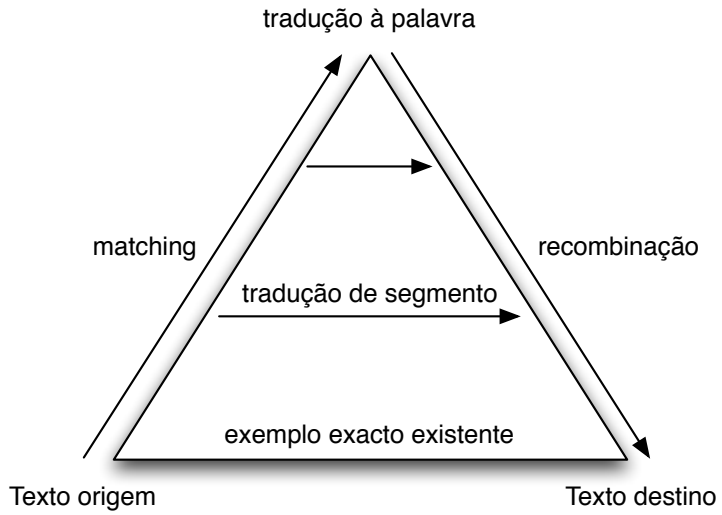


Figura 2.8: Analogia dos sistemas de transferência com os EBMT.

1. o sistema usa corpora bilingues;
2. o sistema usa corpora bilingues como principal base de conhecimento;
3. o sistema usa corpora bilingues em tempo de execução, como principal base de conhecimento;

Destes critérios, Somers considera que enquanto que os dois primeiros são demasiado abrangentes, o terceiro é demasiado restrito, já que exclui os sistemas SMT, onde toda a informação probabilística é calculada previamente.

Além destes critérios, Somers considera que um sistema EBMT usa, como principal base de conhecimento, a base de exemplos. Um corolário desta afirmação é que a precisão do sistema pode ser aumentada adicionando simplesmente mais exemplos. No entanto, outras ferramentas e recursos, como dicionários, ontologias, analisadores léxicos, reconhecedores de entidades mencionadas e outros, podem ser muito importantes num sistema EBMT.

Segundo Somers, um sistema EBMT combina três fases: *matching*, alinhamento e recombinação. Turcato e Popowich defendem que as operações essenciais num sistema EBMT são a decomposição das frases e a selecção da tradução, que se encaixam, respectivamente, nas duas primeiras fases propostas por Somers. (Brown, 2002) propõe as três fases de Somers como áreas de investigação em EBMT:

- procurar os maiores *matches* exactos de porções de texto a ser traduzido;
- combinar as peças posteriormente;
- para que isto funcione, é preciso determinar que pedaço da tradução na base de exemplos corresponde à porção de texto que foi realmente encontrada.

A decomposição das frases é uma das tarefas mais importantes, já que é pouco provável que durante a tradução de um texto se encontre exemplos na base de exemplos do sistema que correspondam a frases completas do texto a traduzir. É, portanto, importante decompor as frases (quer da frase a traduzir, quer das frases da base de exemplos). O processo de decomposição é baseado em diferentes técnicas, como sejam: divisão em frases usando a pontuação ou palavras-marca (Green, 1979) como delimitadores, realização de reconhecimento de entidades mencionadas para obter exemplos mais genéricos, uso de segmentos analisados morfologicamente como sequências de substantivos, ou o parsing de frases em árvores de dependências.

Durante a selecção da tradução, o sistema EBMT vai tentar encontrar traduções de cada um dos pequenos segmentos decompostos. As vantagens referidas em (Knight and Koehn, 2004) relativamente à tradução denominada de *Phrase-based Statistical Machine Translation* (ver próxima secção) são, na verdade, vantagens dos sistemas baseados em exemplos sobre os sistemas SMT:

- usam o contexto local durante a tradução (esse contexto cresce de acordo com o tamanho dos exemplos usados);
- permitem a tradução de frases não composicionais;
- quanto mais corpora forem usados, mais frases e frases maiores podem ser aprendidas;

Os sistemas EBMT são bastante propensos a ruído dada a sua natureza estatística, pelo que podem apresentar menor clareza sintáctica e semântica do que as abordagens de tradução mais formais. No entanto, são bastante mais robustos e escaláveis. Embora por vezes manifestem alguma falta de qualidade nos resultados, este nível não se degrada com a quantidade e qualidade das frases originais (Veale and Way, 1997). Ainda em relação a questões de qualidade, os sistemas EBMT são normalmente bem classificados de acordo com o estilo idiomático da tradução na língua em causa.

> Os sistemas de tradução baseados em exemplos usam dicionários probabilísticos de tradução, terminologia bilingue, exemplos de tradução e, nos sistemas mais evoluídos, técnicas de generalização de exemplos.

2.3.3 Convergência

Cada vez mais as abordagens à tradução automática convergem na utilização de métodos híbridos. Assim como na tradução mais convencional, em que os sistemas têm deixado de poder ser classificados claramente como sendo de tradução directa, baseada em *interlíngua* ou regras de transferência, também na tradução baseada em dados as abordagens estatística e baseada em exemplos têm vindo a convergir.

Por exemplo, a abordagem denominada de *Phrase-based Statistical Machine translation* não é mais do que o uso conjunto de técnicas entre da tradução estatística e da tradução baseada em exemplos. O principal problema na literatura continua a ser o facto de muitos autores não reconhecerem que estão a utilizar ideias que surgiram originalmente numa das outras abordagens.

> Actualmente os sistemas SMT e EBMT são bastante semelhantes e devem ser considerados como pertencentes a uma mesma classe: tradução baseada em dados.

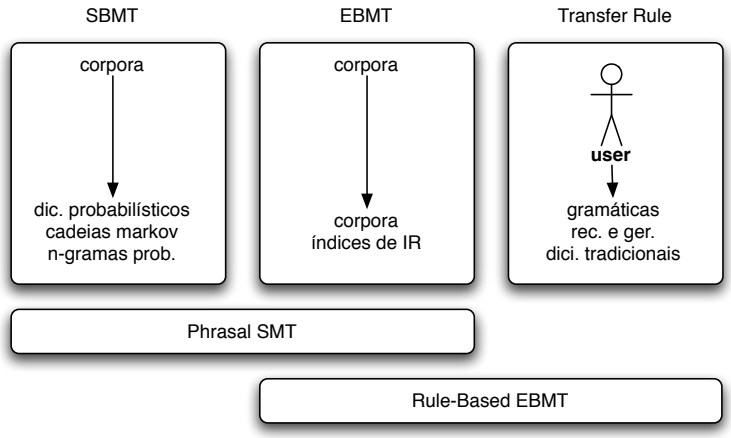


Figura 2.9: Convergência na tradução automática.

As abordagens baseadas em dados têm vindo a adoptar o uso de regras que, até certo ponto, podem ser vistas como regras de transferência. No entanto, normalmente são menos sofisticadas e mais instanciadas (exemplos genéricos ou exemplos parametrizáveis).

2.4 Avaliação Automática

Com a evolução e para a evolução da tradução automática surgiu uma área associada de investigação: a avaliação (automática ou não) da tradução automática. A avaliação, como sabemos, é importante para que se possam comparar sistemas e, em particular, se possam comparar diferentes variantes do mesmo sistema de forma a saber se houve uma evolução positiva.

A existência de métodos de avaliação automáticos é importante já que a avaliação manual é demasiado lenta, dispendiosa e difícil de reproduzir. A avaliação baseada em conjuntos de testes é mais fiável e permite que se tente melhorar automaticamente a performance das ferramentas de tradução automática.

2.4.1 Medidas de Avaliação

Esta secção apresenta duas medidas de avaliação automática de tradução: uma mais antiga, denominada de *Word Error Rate*, e uma mais recente, denominada de *BLEU*.

Word Error Rate

O *Word Error Rate*[8] (WER) é uma medida que compara uma frase (obtida por determinado programa) com uma frase de referência. Surgiu para a avaliação de sistemas de reconhecimento de voz (McCowan et al., 2004) e tem vindo a ser adoptada em diferentes campos de investigação, como sejam a avaliação da tradução automática.

Normalmente, a comparação de determinada frase f com uma frase de referência r é difícil dado que as sequências de palavras de f e r podem ter comprimentos e ordens diferentes. O WER é baseado na distância de Levenshtein mas transposta para o domínio das palavras (e não o original domínio dos caracteres).

O WER é calculado depois de comparadas as palavras de f e r usando a fórmula:

$$\text{WER} = \frac{S + R + A}{N}$$

em que

- S é o número de palavras substituídas de f para obter r;
- R é o número de palavras removidas de f para obter r;
- A é o número de palavras adicionadas a f para obter r;
- N é o número de palavras de referência, comprimento de r;

[8]Uma tradução para português seria: Taxa de Palavras Erradas.

BLEU

O BLEU[9] (proposto por (Papineni et al., 2002)) tem como base o WER mas expandido de forma a comparar não só palavras, mas sequências de palavras, e portanto, avaliar a fluência da tradução.

Os autores defendem que o BLEU é rápido, barato, independente de língua e que se correlaciona de forma elevada com a avaliação humana. O BLEU assenta em dois pontos fulcrais:

- uma métrica de proximidade da tradução;
- um corpus de referência de tradução (frases na língua original, e as respectivas traduções) com boa qualidade.

A métrica de proximidade de tradução permite diferenças legítimas na escolha entre palavras e na sua ordem, já que tipicamente existem muitas traduções correctas para uma mesma frase.

A ideia passa por classificar cada frase f comparando com n-gramas entre cada frase candidata e uma ou mais frases de referência r. A implementação consiste na realização das comparações e contagem do número de n-gramas semelhantes. Estas comparações são independentes da posição. Quantos mais n-gramas semelhantes, melhor f é. O BLEU foi especialmente desenhado para se aproximar à avaliação ao nível do corpus, e portanto não é aconselhado para avaliar a qualidade de frases isoladas.

Uma tradução que use as mesmas palavras (1-gramas) que as da frase de referência tende a satisfazer a adequabilidade. Quanto maior os n-gramas, melhor será a fluência da tradução.

Embora este método continue a ser bastante usado na avaliação de ferramentas de tradução automática há estudos que mostram que o BLEU nem sempre é adequado à tarefa que se propõe resolver.

Em (Callison-Burch, Osborne, and Koehn, 2006), comparou-se o valor do BLEU com uma avaliação manual para três sistemas: um sistema SMT bom, um sistema SMT mau e o sistema Systran (baseado em re-

[9] BiLingual Evaluation Understudy.

gras). Embora o BLEU tenha sido eficiente para diferenciar os sistemas SMT, deu os valores mais baixos para o sistema Systran que foi classificado manualmente como o melhor sistema. Os autores defendem que pode ser necessária uma re-avaliação manual cuidada sempre que o BLEU não mostre melhorias no desenvolvimento de uma ferramenta de tradução.

2.4.2 Competições e Avaliações Cooperativas

A participação em competições permite, também, uma avaliação e comparação de ferramentas. Anualmente existem competições de ferramentas de tradução automática (NIST Open MT[10], IWSLT[11]) em que os investigadores interessados podem participar com os seus sistemas. Embora não resultem valores absolutos de classificação permitem a comparação de abordagens.

Estes encontros trazem vantagens no desenvolvimento da área já que são realizadas demonstrações públicas do estado-da-arte, desenvolvidos e disponibilizados conjuntos abertos de recursos para a avaliação, dão credibilidade aos sistemas participantes, e permitem a partilha de ideias e implementações.

No entanto, se estas competições e avaliações não forem levadas com o devido espírito crítico pode levar a que a investigação se torne limitada, já que se irá tentar re-implementar os métodos vencedores.

2.5 Ferramentas de Tradução

Esta secção não pretende ser uma lista exaustiva de ferramentas de tradução, mas sim referir aquelas que de alguma forma se relacionam com o trabalho efectuado, e que motivam os diferentes tipos de recursos extraídos nesta dissertação.

[10] `http://www.nist.gov/speech/tests/mt/`

[11] *International Workshop on Spoken Language Translation.* A edição de 2007 está disponível em `http://iwslt07.itc.it/`

De acordo com o que foi discutido previamente, dividimos as ferramentas em tradução assistida por computador (de acordo com a secção 2.1), sistemas de tradução automática baseados em regras, e sistemas de tradução automática baseados em dados (de acordo com a secção 2.3).

2.5.1 Tradução baseada em Memórias de Tradução

Nas ferramentas de tradução assistida por computador pretende-se que o tradutor tenha total controlo sobre a tradução realizada. O sistema informático existe apenas para fazer sugestões e automatizar algumas tarefas básicas.

Os sistemas de tradução assistida por computador que se referem nesta secção são o SDL Trados Freelancer por ser dos mais bem cotados entre os sistemas comerciais, o OmegaT por ser o mais conhecido dos sistemas livres (juntamente com o bitext2tmx como sistema auxiliar) e o TRANSBey, uma filosofia baseada em wiki para tradução cooperativa.

SDL Trados

O SDL Trados (Trados Manual, 2005; SDL Trados, 2006) é um dos sistemas de tradução assistida mais conhecidos e também dos mais usados. Este sistema surgiu originalmente no mercado como Trados, mas foi recentemente (2005) comprado pela SDL.

Dos vários produtos vendidos pela SDL Trados, a versão Freelancer é a mais conhecida e a que foi usada para experiências durante a realização da dissertação. O SDL Trados Freelancer é constituído por várias ferramentas, das que destacamos:

- Translator's Workbench
  Este é o gestor de memórias de tradução: vai guardando as traduções à medida que o tradutor as vai realizando, e vai procurando unidades de tradução armazenadas semelhantes à frase que está a ser traduzida. O tradutor pode editar, aceitar ou rejeitar cada unidade.

- MultiTerm
  O MultiTerm é o gestor de terminologia, permitindo que o tradutor crie a sua própria terminologia. A estrutura destas bases terminológicas é definida pelo utilizador, sendo que cada registo deve ser orientado ao conceito. A terminologia pode ser exportada facilmente para diferentes formatos.
  Como ferramenta de ajuda à tradução, integra automaticamente com o TagEditor ou o Microsoft Word, permitindo acesso imediato ao conteúdo da base terminológica. Também permite que a partir de qualquer uma destas duas ferramentas se introduzam novos termos na terminologia.
- TagEditor
  Para que o tradutor se possa abstrair dos formatos específicos dos documentos que está a traduzir, o TagEditor é um editor genérico com funcionalidades para a tradução. Suporta vários formatos como sejam PowerPoint, Excel, Word, HTML, dialectos XML e outros.
  Integra com o Translator's Workbench, que é usado para a pesquisa nas memórias de tradução das frases que estão a ser traduzidas. Também permite a tradução por aplicação directa das traduções constantes na terminologia, realizando uma tradução termo por termo.
- WinAlign
  O WinAlign é um alinhador[12] de texto ao nível da frase. Permite que o tradutor re-aproveite todo o trabalho que realizou sem o uso de uma ferramenta assistida de tradução. O WinAlign usa um algoritmo para o alinhamento automático dos documentos, e permite a posterior edição manual do alinhamento. O resultado deste alinhamento pode ser exportado para formatos standard, ou integrado na base do Translator's Workbench.

[12] A definição formal de alinhamento (ao nível da palavra ou ao nível da frase) será apresentada no capítulo 3. Entretanto, e para facilitar a leitura, um alinhador no contexto da tradução assistida por computador corresponde a uma ferramenta que permite analisar corpora paralelos e fazer corresponder frases da língua original às respectivas traduções na língua de destino.

OmegaT

O OmegaT (Prior, 2002) é uma ferramenta de tradução assistida por computador livre e de código aberto. Está escrito em *Java* o que lhe permite ser independente de plataforma (ao contrário de todas as outras aplicações de tradução assistida por computador comerciais que apenas funcionam em Microsoft Windows).

Embora inferior em termos de funcionalidades quando comparado com as alternativas comerciais, o facto do OmegaT ser código livre permite que seja usado por investigadores para implementação de algoritmos e provas de conceito.

O OmegaT suporta:

- *fuzzy matching*;
- propagação de traduções;
- uso simultâneo de várias memórias de tradução;
- uso de bases terminológicas externas;
- filtros para o tratamento de documentos em texto, HTML, OpenOffice, Xliff e MediaWiki;
- suporte de Unicode para o uso de alfabetos não latinos;
- suporte de línguas com escrita da direita para a esquerda;
- memórias de tradução em formato TMX.

bitext2tmx

O software bitext2tmx é a alternativa livre e de código aberto do WinAlign. É um alinhador de textos paralelos com correcção manual.

Assim como o OmegaT, o bitext2tmx também está a ser desenvolvido em Java, é livre e de código aberto pelo que permite a sua utilização em qualquer sistema operativo. Pode ser descarregado livremente a partir de `http://bitext2tmx.sf.net/`.

TRANSBey

O TRANSBey (Bey, Boitet, and Kageura, 2006) é um sistema cooperativo de tradução baseado num sistema Wiki. A ideia primordial é a transposição da tarefa de tradução para a Internet de modo a que qualquer utilizador possa ajudar na tradução.

O processo de tradução acaba por ser semelhante aos anteriormente referidos, recorrendo ao uso de memórias de tradução. A principal diferença é o facto de existir mais do que um tradutor a traduzir ao mesmo tempo (em frases diferentes), e de a memória de tradução usada ser partilhada por todos os tradutores.

> Os sistemas de tradução assistida por computador podem tirar partido de dicionários probabilísticos de tradução e as terminologias bilingues, que podem ser usados para a criação de terminologias.
> Por sua vez, os exemplos de tradução podem ser usados como memórias de tradução.

2.5.2 Tradução baseada em Regras

Esta secção apresentada três sistemas de tradução baseados em regras:

- **Logos/OpenLogos**, um sistema próximo dos sistemas comerciais, agora em código aberto;
- **Apertium**, um sistema de tradução entre línguas aparentadas implementado sobre um sistema de transdutores;
- **Text::Translate**, uma ferramenta para a criação de protótipos de sistemas de tradução baseados em regras.

Logos/OpenLogos

A Logos Corporation e o sistema Logos (Scott, 2003) surgiram no meio do conflito entre os Estados Unidos da América e o Vietname, em res-

posta da necessidade de traduzir grandes quantidades de manuais militares americanos para vietnamita. Embora tenha surgido em 1970 (ainda muito perto do relatório ALPAC), a Logos Corporation que tinha acabado de ser criada insistiu que conseguiria obter os resultados necessários. O governo americano deu uma hipótese, pedindo que em três meses a Logos traduzisse um manual de 20 páginas sobre determinado helicóptero. Os resultados foram promissores o que levou a que o projecto fosse aprovado e financiado.

Quando a guerra terminou, milhares de páginas tinham sido traduzidas em vários dos ramos das forças militares americanas. No seu relatório anual de 1972, John Foster, director da defesa, pesquisa e engenharia, indicou que o sistema Logos tinha "demonstrado a possibilidade de tradução automática em larga-escala." Este foi o primeiro resultado positivo na tradução automática após o relatório ALPAC.

O sistema Logos continuou no mercado como um dos maiores programas comerciais de tradução automática. Muitos recursos foram desenvolvidos para este sistema, para várias línguas. Recentemente foi disponibilizado em código aberto sob o nome de OpenLogos[13] O OpenLogos (cuja arquitectura é baseada na versão anterior Logos) é um sistema de tradução baseado em regras de transferência.

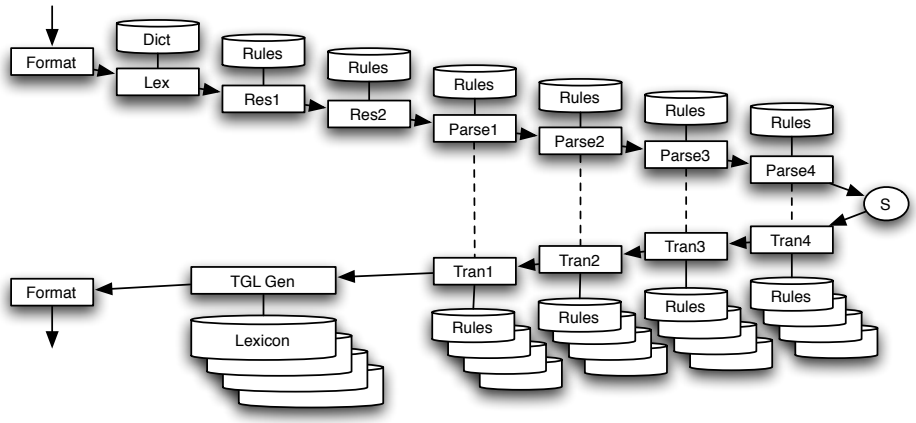


Figura 2.10: Arquitectura do sistema OpenLogos.

[13]OpenLogos — `http://logos-os.dfki.de/`.

Embora os requisitos originais pudessem levar a que o Logos tivesse sido desenvolvido de uma forma arbitrária, o seu desenvolvimento foi feito tendo sempre em vista a construção de um sistema de uso genérico, e que pudesse ser usado com qualquer combinação de línguas.

O modelo usado pelo Logos é descrito pelos seus criadores como:

1. um motor independente de língua que, com base num dicionário externo, converte uma frase numa lista de símbolos (semântico-sintácticos); item estes símbolos são confrontados com padrões existentes em bases de regras;
2. quando os símbolos estão de acordo com determinada regra, é interpretada a acção associada. Algumas propriedades que possam ser pertinentes para a geração do texto na língua de destino são guardadas como análises de cada um dos constituintes originais;
3. a língua de destino é gerada assim que seja terminada a análise à frase original.

De acordo com a figura 2.10, o texto na língua de origem entra no topo, onde a formatação é analisada e removida, e limites frásicos são identificados. Cada frase é convertida numa lista de símbolos semântico-sintácticos, usando substituição léxica. Esta lista passará pelas bases de regras, efectuando uma análise simples, *bottom-up*. As regras consistem em padrões semântico-sintácticos e, quando estão de acordo com alguma parte da lista de símbolos previamente calculados, tornam-se activas.

A transferência para a língua de destino é obtida com equivalências entra árvores usando quatro níveis de *parsing*, reflectindo uma abordagem composicional. Segue-se a geração da frase na língua de destino usando informação morfológica sobre a lista de símbolos semântico-sintácticos obtida pela transferência entre árvores.

Apertium

O Apertium (Corbí-Bellot et al., 2005; Armentano-Oller et al., 2005; Armentano-Oller et al., 2006) é um sistema de tradução automática de código aberto. É baseado nos sistemas de tradução espanhol:catalão in-

terNOSTRUM (Canals-Marote et al., 2001; Garrido et al., 1999; Garrido-Alenda and Forcada, 2001) e Traductor Universia (Garrido-Alenda et al., 2003; Gilabert-Zarco et al., 2003), ambos desenvolvidos na Universidade de Alicante.

Usa uma arquitectura de transferência sintáctica superficial bastante semelhante a alguns sistemas comerciais de tradução automática. Tem vindo a ser desenvolvido para os pares de língua galego:espanhol, espanhol:catalão, catalão:espanhol e espanhol:português.

A arquitectura segue a ideia de que, no caso de línguas próximas como o espanhol, galego e catalão, uma tradução mecânica palavra à palavra apresenta erros, mas que podem ser resolvidos com uma análise morfológica seguida de uma análise sintáctica superficial, e com um tratamento adequado das ambiguidades léxicas.

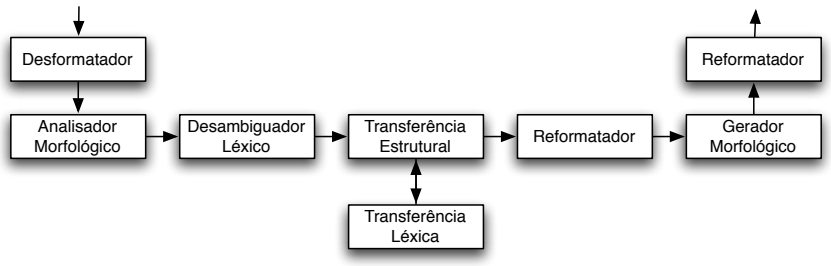


Figura 2.11: Módulos do Apertium.

O sistema é composto por oito módulos:

- *desformatador*, que separa o texto a traduzir do formato específico em que o documento se encontra;
- *analisador morfológico*, segmenta o texto e, para cada, retorna uma ou mais formas léxicas. Cada uma destas formas léxicas é composta por um lema, uma categoria morfológica e informação de flexão morfológica;
- *desambiguador léxico*, elege (usando modelos estatísticos) uma das formas léxicas de acordo com o seu contexto, já que o analisador morfológico pode ter retornado uma análise ambígua;
- *módulo de transferência estrutural*, detecta padrões de palavras

que precisem de um tratamento especial por causa das divergências estruturais entre as línguas (trocas de género e número, reordenamento, etc). Este módulo funciona com base numa base de regras de tradução.

- *módulo de transferência léxica*, funciona com base num dicionário bilingue e é invocado pelo módulo de transferência estrutural. Este módulo recebe uma forma léxica na língua original e retorna uma forma léxica na língua de destino. Pode ser visto de forma simplificada como um dicionário bilingue.
- *gerador morfológico*, pega em cada uma das formas léxicas retornadas pelo módulo de transferência léxica e constrói a forma superficial adequada na língua de destino, usando para isso um analisador morfológico.
- *pós-gerador*, realiza operações ortográficas simples na língua de destino como sejam as contracções ou a adição de apóstrofes.
- *reformatador*, reintegra a tradução no formato original.

Quatro destes módulos (analisador morfológico, módulo de transferência léxica, gerador morfológico e pós-gerador) estão implementados usando transdutores de estados finitos o que lhes confere grande eficiência.

Além de funcionar como tradutor, cada um destes módulos do pacote Apertium pode funcionar de forma independente. Assim, é possível utilizar, por exemplo, o analisador morfológico de forma independente dos outros módulos.

Text::Translate

O módulo Perl `Text::Translate` (Almeida, 2003) foi desenvolvido com base num sistema de re-escrita por camadas. Cada camada de re-escrita recebe um conjunto de padrões e um conjunto de substituições que devem ser realizadas. Quando um padrão está de acordo com o texto de origem, é efectuada uma substituição. Este processo repete-se até que não seja possível realizar-se mais substituições. É esperado que depois de todas estas substituições o texto se encontre traduzido.

O facto de estar desenvolvido em Perl permite grande flexibilidade na

construção de protótipos, tornando-se possível a integração com qualquer outra aplicação, ou mesmo a Internet, de forma simples.

Uma vez que o sistema funciona por camadas é possível que cada uma destas camadas tenha objectivos diferentes, quase que como os módulos do Apertium ou do Logos. Estas camadas de re-escrita são:

- *pré-edição*, onde determinadas palavras são substituídas ou protegidas, bem como onde as contracções são divididas;
- *tradução*, onde vários dicionários são consultados em cascata, e palavras substituídas. Normalmente funciona com uma lista de dicionários, dos mais específicos para o mais genéricos. Nesta mesma fase as palavras desconhecidas são tratadas, adicionando-lhes um marcador na tradução realizada, e é criado um dicionário auxiliar com a lista de palavras desconhecidas;
- *pós-edição*, onde são corrigidos problemas de concordância na tradução realizada, bem como outros pequenos ajustes.

Embora este sistema seja bastante simples, mostrou-se bastante útil para a prototipagem de sistemas de tradução por regras e baseados em exemplos.

Uma descrição mais detalhada do funcionamento desta ferramenta pode ser encontrada na secção 6.4.

Para além de recursos monolingues (analisadores morfológicos, p.ex), os sistemas de tradução baseados em regras tiram partido de todo o tipo de recursos bilingues que se possam extrair, desde dicionários de tradução, terminologia bilingue, exemplos de tradução ou mesmo padrões de tradução.

2.5.3 Tradução baseada em Dados

Nesta secção são apresentados cinco sistemas de tradução baseados em dados:

- **Pharaoh/Moses/Phramer**, estes são três sistemas relacionados para a tradução baseada em estatística ao nível do segmento (conhecida por Phrase-based Statistical Machine Translation — PSMT). Também como já foi referido, esta abordagem usa técnicas de sistemas SMT e EBMT.
- **Gaijin/MaTrEx**, é um sistema de tradução baseado em exemplos, que usa como conhecimento linguístico[14] apenas algumas listas de palavras (palavra-marca), para a segmentação de unidades de tradução.
- **EDGAR**, é um sistema de tradução baseado em exemplos que usa análise morfológica e *shallow parsing* para criar pequenas árvores sintácticas, que são posteriormente traduzidas utilizando exemplos.
- **ReVerb**, é um sistema de tradução baseado em exemplos que assenta numa visão de raciocínio baseado em casos.
- **Pangloss Mark III**, é um sistema híbrido: surgiu originalmente como um tradutor baseado em dados, mas dada a falta de resultados, foi desenvolvido em paralelo um sistema de tradução baseado em regras. Actualmente conjuga os resultados destes dois subsistemas.

Pharaoh / Moses / Phramer

O Pharaoh[15] (Koehn, 2004) é um sistema estatístico para a construção de ferramentas de tradução automática. Corresponde ao modelo apresentado na secção 2.3.2 mas em que a tradução não é realizada palavra

[14] É certo que os corpora paralelos incluem conhecimento linguístico. Quando nos referimos concretamente a conhecimento linguístico referimo-nos a conhecimento explícito.

[15] O sistema Pharaoh está disponível em `http://www.isi.edu/publications/licensed-sw/pharaoh/`.

a palavra, mas ao segmento: existe um modelo de tradução que associa traduções a segmentos, e existe um modelo de língua que valida a ordem das palavras e as tradução mais prováveis de existir na língua de destino. O Phramer[16] é uma implementação Java do algoritmo usado pelo Pharaoh. Por sua vês, o Moses[17] (Koehn et al., 2007) é o substituto do Pharaoh, desenvolvido pelos mesmos autores.

Estes sistemas usam um dicionário probabilístico de tradução multi-palavra que é usado para a geração de traduções de forma automática. Posteriormente, é usado um modelo estatístico de custo/benefício para avaliar as traduções e escolher a com maior relação de qualidade de tradução/legibilidade (baseado na probabilidade do dicionário de tradução e no modelo de língua, respectivamente).

> Os sistemas de tradução baseados em estatística usam dicionários (mono ou multi-palavra) com informação estatística associada, como sejam dicionários probabilísticos de tradução ou terminologia bilingue. Os próprios exemplos de tradução podem ser vistos como dicionários estatísticos de tradução ao nível do segmento.

Gaijin / MaTrEx

O Gaijin (Veale and Way, 1997) é um sistema de tradução automática baseada em exemplos. Não usa modelos de língua explícitos: retira todo o conhecimento de que necessita de corpora paralelos bilingues. O sistema usa métodos estatísticos, *matching* de segmentos, raciocínio baseado em casos, e *matching* de regras (templates), numa solução com pouco conhecimento linguístico.

Começou a ser desenvolvido com vista à tradução entre inglês e alemão, no domínio restrito de ficheiros de ajuda e de documentação de

[16] Phramer - An Open-Source Statistical Phrase-Based MT Decoder `http://www.utdallas.edu/~mgo031000/phramer/`

[17] O sistema Moses está disponível em `http://www.statmt.org/moses/`.

um pacote de desenho. Uma das premissas no seu desenvolvimento foi o uso do mínimo conhecimento linguístico possível de forma a facilitar a sua adaptação para novas línguas e domínios.

O corpus paralelo usado pelo Gaijin deve ser preparado de acordo com as seguintes etapas:

- *alinhamento do corpus bilingue*
  A estrutura do documento é analisada e alinhada, e posteriormente as frases em cada uma das partes do documento são alinhadas. Nos casos em que o alinhamento da estrutura do documento não é possível de ser realizado, o utilizador terá de o alinhar manualmente ou remover os textos do corpus.
- *construção automática do léxico*
  O alinhamento do léxico é feito usando uma abordagem similar à usada na extracção de dicionários probabilísticos de tradução, descrita no capítulo 4. No entanto, os autores do Gaijin citam os artigos (Kay and Röscheisen, 1993) e (Somers, McLean, and Jones, 1994), que usam uma matriz de co-ocorrências para o alinhamento à frase.
  Esta etapa é descrita como a criação de uma matriz que relaciona as palavras do corpus de origem e de destino. Esta matriz inclui uma medida baseada nas frequências absolutas das palavras em cada um dos corpus, e na frequência das suas ocorrências conjuntas no mesmo exemplo. Além destes valores, o Gaijin calcula um peso extra de acordo com a diferença de tamanho do exemplo em relação à média dos tamanhos de exemplos: quanto maior for o exemplo, menor a relevância da co-ocorrência, e quanto menor o exemplo, maior a sua relevância.
- *inferência de regras (templates) de transferência*
  Embora o Gaijin use estatística baseada em corpora, não a usa como uma estratégia de tradução (Brown et al., 1990), mas como base para inferir regras de transferência (mais próxima da perspectiva apresentada em (Collins, Cunningham, and Veale, 1996a)).
  Uma regra (ou *template*) de transferência é uma associação entre duas estruturas vagas de uma frase (baseada essencialmente na Hipótese das Palavras-Marca (Green, 1979)). Esta estrutura não é mais que um conjunto de *place-holders* tipados por uma ou mais

palavras-marca.

O processo de tradução começa pela pesquisa da regra de transferência a ser usada. Ao traduzir uma frase f, se f tem uma estrutura semelhante a uma destas regras, então a *template* na língua de destino é usada. Cada um dos *place-holders* são traduzidos com base em exemplos:

- *recuperação de exemplos*
  Ao desenhar um sistema de recuperação de exemplos é preciso ter em consideração se vai ser procurado um exemplo grande, que cubra toda a frase a traduzir, ou se, por outro lado, se vão tentar traduzir pequenas porções compostas posteriormente. O Gaijin usa uma estratégia entre estas duas: por uma lado usa uma única regra para traduzir toda a frase, de acordo com as regras de transferência apresentadas no item anterior, mas cada um dos sub-segmentos da regra são traduzidos independentemente.
- *adaptação de exemplos*
  Depois de encontrada a regra que se adequa à frase a traduzir, é preciso traduzir cada um dos sub-segmentos. Se possível, a tradução existente da regra original é usada. O caso mais frequente é que esta não possa ser usada directamente, mas que difira apenas na alteração de algumas palavras ("desenho" → "desenhos"). No caso de não ser possível fazer este tipo de retoque ao nível da palavra, outro exemplo terá de ser procurado.
- *aquisição de novos exemplos*
  Depois de uma tradução ter sido realizada é apresentada ao utilizador. Este, pode aprovar a tradução de forma a que este par de frases passe a ser um novo exemplo, e possa vir a ser usado em novos processos de tradução.

O Gaijin tem vindo a ser expandido. Actualmente chama-se MaTrEx e inclui chinês, árabe, italiano, basco, espanhol, alemão, japonês e francês.

O Gaijin pode tirar especial partido dos exemplos de tradução obtidos usando a hipótese das Palavras-marca, e de conjuntos de palavras parentes.

EDGAR

O Sistema EDGAR[18](Carl, 1999) é descrito pelos seus autores como um sistema de tradução baseado em exemplos mas que usa algum conhecimento linguístico. Na verdade, serve-se de um analisador morfológico e de um *shallow parser* para a criação de árvores sintácticas que são posteriormente traduzidas utilizando exemplos. Este processo de tradução tira partido de um mecanismo de inferência para a generalização de padrões de tradução a partir de um conjunto de traduções de referência.

O processo de tradução pode ser descrito como:

- decomposição da frase na língua de origem por análise morfológica e *shallow parsing*. Cada palavra ou sintagma é catalogado de acordo com a sua categoria morfológica ou sintáctica;
- é usada uma base de exemplos simples (pares de texto na língua de origem e na língua de destino) e exemplos generalizados (exemplos em que determinadas palavras foram substituídas por variáveis tipadas com uma categoria morfológica ou sintáctica) para criar uma árvore de decomposição.

  Os exemplos são etiquetados como s ou dp, se corresponderem a um exemplo de uma frase completa, ou de um sintagma, respectivamente). Os verbos são etiquetados com a sua forma (fin).

$$
\begin{aligned}
(\textit{Every handsome man})_{\mathrm{dp}} &\leftrightarrow (\textit{Jeder stattliche Mann})_{\mathrm{dp}} \\
(\textit{a pretty woman})_{\mathrm{dp}} &\leftrightarrow (\textit{eine hübsche Frau})_{\mathrm{dp}} \\
(\mathcal{X}_{\mathrm{dp}} \quad \textit{love}_{\mathrm{fin}} \quad \mathcal{Y}_{\mathrm{dp}})_s &\leftrightarrow (\mathcal{X}_{\mathrm{dp}} \quad \textit{lieben}_{\mathrm{fin}} \quad \mathcal{Y}_{\mathrm{dp}})_s
\end{aligned}
$$

  Os primeiros dois exemplos correspondem a sintagmas extraídos dos corpora de base do EDGAR. o Terceiro exemplo corresponde a uma frase generalizada, em que apenas o verbo foi preservado.

[18]EDGAR é um acrónimo de *Example-based Decomposition, Generalization And Refinement*: decomposição baseada em exemplos, generalização e refinamento.

- é realizada a redução da frase e posterior refinamento usando um conjunto de regras que alteram árvores de decomposição, removendo, alterando e adicionando nodos de acordo com um conjunto de condições.
  Por exemplo, considerando a frase "*Every handsome man loves a pretty woman*" e os três exemplos anteriores, a decomposição seria feita da seguinte forma:
  1. a frase é segmentada usando a análise morfológica e o shallow parsing em "(*Every handsome man*) *loves* (*a pretty woman*)"
  2. é possível substituir alguns dos segmentos por variáveis tipadas: "$\mathcal{X}_{dp}$ $love_{fin}$ $\mathcal{Y}_{dp}$"
  3. de acordo com o exemplo generalizado esta árvore pode ser traduzida para: "$\mathcal{X}_{dp}$ $lieben_{fin}$ $\mathcal{Y}_{dp}$"
  4. as variáveis podem ser substituídas pelas respectivas traduções: "(*Jeder stattliche Mann*) *liebt* (*eine hüsche Frau*)"

O EDGAR não usa exemplos simples directamente. Depois de etiquetados morfologicamente e sintacticamente passam a ser úteis para este sistema de tradução.

ReVerb

O ReVerb (Collins, Cunningham, and Veale, 1996a; Collins, Cunningham, and Veale, 1996b) é um sistema de tradução baseado em exemplos que usa técnicas de raciocínio baseado em casos para a adaptação de exemplos para a sua posterior aplicação.

O sistema compara listas de propriedades morfológicas e escolhe aquela que melhor unifica com a frase a traduzir. Assim como os exemplos de padrões do EDGAR, os do ReVerb também contêm variáveis que indicam que porções podem ser substituídas, e portanto, aumentando a probabilidade dos exemplos unificarem. Estas variáveis são tipadas com as funções sintácticas, e portanto não é necessário que as palavras sejam exactamente as mesmas para que o exemplo seja aplicado.

Como mecanismo de pesquisa de exemplos, o ReVerb usa dois níveis: um baseado apenas na comparação de palavras, e outro baseado em informação morfológica e sintáctica:

- comparação de palavras
  Este mecanismo não faz qualquer análise linguística à frase a traduzir: apenas palavras exactas são procuradas na base de exemplos. Nem sequer palavras vizinhas morfologicamente ("objecto" e "objectos") são consideradas. Embora esta abordagem descarte toda a informação morfológica e sintáctica, ela não é retirada, podendo vir a ser usada em caso de necessidade;
- comparação sintáctica
  Para a pesquisa baseada em informação sintáctica, a frase a traduzir é previamente processada por um *shallow parser*, de forma a que cada segmento obtido tenha uma *head-word*[19] nítida. A pesquisa é feita dando à *head-word* um maior peso. Segue-se uma comparação palavra a palavra dentro de cada segmento.

> O ReVerb é um sistema de tradução baseado em exemplos que tira partido directamente de exemplos extraídos de corpora paralelos.

Pangloss Mark III

Originalmente, o sistema Pangloss (Nirenburg, 1995) foi implementado como um sistema de tradução automática espanhol:inglês baseado em conhecimento (knowledge-based machine translation — KBMT), implementado sobre uma arquitectura *interlíngua*.

A primeira versão, o Pangloss Mark I, era um sistema puramente baseado em conhecimento, mas que não teve grandes resultados na primeira avaliação do projecto em 1992. Desta forma, foram tomadas outras direcções, e o Pangloss Mark II foi apresentado como um sistema simples baseado em transferência lexical. A avaliação dos seus

[19]Neste contexto a tradução de *head-word* seria pouco clara. Considera-se *head-word* uma palavra que explicita a função sintáctica do segmento em causa.

resultados foi melhor que a primeira. Em vez de optar apenas por melhorar uma destas abordagens, a equipa decidiu juntar as duas técnicas, e mesmo, incorporar outras. Na verdade, o PANGLOSS Mark III não usa apenas um motor de tradução, mas um conjunto de vários, cujos resultados são posteriormente integrados para um melhor resultado.

O sistema actual usa três motores de tradução:

- *o sistema original baseado em conhecimento*
  esta abordagem segue a filosofia de tradução baseada em *interlíngua* pelo que se decompõe em duas partes principais: a análise e a geração. Dado que o sistema não é um *interlíngua* puro, existe ainda um processo de transferência.
  - Análise (PANGLYZER)
    O sistema de análise funciona por níveis. Cada um dos oito níveis marca determinado tipo de informação: conversão do texto em estruturas de dados Prolog; etiquetagem do Part-of-Speech; criação de sintagmas; reconhecimento de entidades mencionadas; representação semântica dos sintagmas; criação de grupos de sintagmas e a sua etiquetagem; anotação de dependências sintácticas; e a classificação de interpretações de acordo com os seus contextos.
  - Transferência/Interlíngua (PANGLYZER-to-PENMAN)
    Esta etapa corresponde à análise da estrutura obtida pelo gerador, e a sua conversão para uma sintaxe de frases, denominada Sentence Plain Language.
  - Geração (PENMAN)
    O PENMAN é um gerador de língua orientado à frase que a partir de uma especificação não-linguística (na dita sintaxe SPL) é capaz de gerar frases inglesas. O sistema é composto por uma gramática inglesa e vários recursos auxiliares, dos quais o principal é uma taxonomia de símbolos semânticos de alto-nível.
- *um sistema de tradução baseado em exemplos*
  Assim como a maioria dos sistemas EBMT, o PANGLOSS também se baseia num corpus alinhado à frase. Para a tradução de uma frase, são realizados os seguintes passos:

  - pesquisa de segmentos da língua de origem no corpus que são parecidos com a porção de texto a traduzir. Neste processo, as frases são quebradas pela pontuação e por palavras desconhecidas (não pertencentes ao corpus). Estes segmentos são procurados no corpus, fazendo uma pesquisa difusa. Para cada um destes resultados inexactos é calculada uma penalidade, de acordo com a diferença com o segmento procurado.
  - obtenção de segmentos na língua de destino correspondentes ao segmento na língua de origem que foi encontrado:
    * cálculo, com base num dicionário, das traduções para todas as palavras da frase da língua de origem;
    * cálculo das raízes de todas as palavras da frase na língua de destino;
    * alinhamento da unidade de tradução ao nível da palavra;
    * pesquisa do maior segmento na língua de destino que pode ser tradução do segmento da língua de origem;
    * pesquisa do melhor segmento usando medidas de classificação;
- *um sistema de transferência lexical*
  O sistema de transferência lexical usado é simples e tradicional. Funciona como uma rede de segurança, para quando os outros dois métodos não dão resultados (ou são demasiado fracos). A transferência lexical é realizada usando análise morfológica e um conjunto de recursos bilingues: léxicos desenvolvidos para o sistema KBMT e um dicionário bilingue produzido manualmente. Para permitir a aplicação de regras lexicais em padrões "abertos", foram introduzidas variáveis nos glossários para representar entidades (nomes próprios, lugares, etc), números e pronomes (pessoais, possessivos, etc).

A frase a ser traduzida é cortada em segmentos utilizando um *chunker*. Cada um destes segmentos é traduzido usando os vários motores, e a cada tradução é associado um valor de fiabilidade (calculado por cada um dos motores). Segue-se um algoritmo de programação dinâmica para seleccionar as melhores traduções que melhor cobrem a frase original. No final, um conjunto de regras simples de pós-edição são aplicadas

para resolver certos problemas, como sejam a concordância de género e número.

> O Pangloss, sendo um sistema híbrido, tira partido de todo o tipo de recursos bilingues que se possam extrair.

A Título de Conclusão

Neste capítulo começámos por analisar as diferentes abordagens na tradução: quer as tecnologias da literatura, quer os sistemas que existem implementados.

Embora muitas das ferramentas que foram vistas neste capítulo também tirem partido de corpora monolingues, nesta dissertação decidiu-se abordar essencialmente os recursos resultantes do processamento de corpora paralelos (embora também se tenham extraído recursos puramente monolingues, como sejam n-gramas).

Em relação aos recursos bilingues, foi possível verificar que são cruciais à tradução nas suas diversas etapas. Nomeadamente:

- **dicionários de tradução:** qualquer que seja a metodologia de tradução é impossível de a realizar sem o conhecimento atómico de como se traduzem palavras ($\mathcal{T}(w_{\mathcal{A}}) = w_{\mathcal{B}}$). Por outro lado, nem toda a tradução é composicional. Nomeadamente, há um conjunto de terminologia e unidades multi-palavra que se traduzem de forma especial ($\mathcal{T}(w_1 \cdot w_2) \neq \mathcal{T}(w_1) \cdot \mathcal{T}(w_2)$).
- **memórias de tradução:** a tradução assistida por computador usa traduções já efectuadas para tentar poupar trabalho ao tradutor. As metodologias de tradução estatísticas precisam de muitas unidades de tradução (ou seja, de corpora paralelos em grandes quantidades) para que possam aprender e inferir conhecimento.
- **exemplos de tradução:** a tradução baseada em exemplos usa o conceito de *exemplos de tradução* que correspondem a unidades de tradução pequenas, normalmente de tamanho inferior a uma frase. No entanto, este tipo de recurso pode tam-

bém ser integrado em sistemas de tradução assistida por computador, ajudando o tradutor a traduzir porções de frases ao invés de frases completas.

- **regras de tradução:** a tradução baseada em regras usa desde sempre comandos formais para especificar como a tradução é efectuada entre línguas. Estas regras não são mais que unidades de tradução generalizadas, de acordo com o ponto de vista da tradução baseada em exemplos.
- **conjuntos de palavras:** a generalização leva à necessidade de construção de conjuntos de palavras semelhantes. Não semelhantes semanticamente, mas que pertencem a uma mesma família: dias da semana, animais, compostos químicos, etc.).

Capítulo 3

Corpora Paralelos

[...] more data is better,
and even more data is even better.

(Koehn, 2002)

Os métodos de extracção de recursos bilingues desenvolvidos durante esta dissertação têm um cariz estatístico forte, pelo que o tamanho dos corpora usados para recolha de factos estatísticos é importante. Torna-se, pois, necessária a criação ou angariação de corpora de tamanhos razoáveis.

Definição 2 *O termo* ***corpus*** *será usado para designar um grande conjunto de textos (habitualmente armazenado e processado electronicamente). Um corpus pode conter textos numa única língua (corpus monolingue) ou em várias línguas (corpus multilingue).*

Os corpora monolingue são habitualmente usados para o estudo de uma língua, embora também sejam úteis para o enriquecimento de recursos bilingues, ou para a construção de modelos de língua a serem

usados por ferramentas de tradução automática. No trabalho realizado deu-se especial atenção à criação e processamento de corpora multilingue paralelos já que são constituídos por dois corpora monolingue independentes, existe maior escassez deste tipo de corpora, e pela sua riqueza de informação multilingue.

Definição 3 *Um* ***texto paralelo*** *(ou* ***bitexto****) é um texto numa língua juntamente com a sua tradução numa outra língua. Grandes colecções de bitextos são chamadas de* ***corpora paralelos****.*

Embora a definição habitual de ***corpora paralelos*** *não implique o seu alinhamento, é nossa convicção de que estes recursos são especialmente úteis quando alinhados ao nível da frase, pelo que usaremos o termo* ***corpora paralelos*** *para designar textos paralelos alinhados ao nível da frase (ou da unidade de tradução).*

Foram criados e adoptados vários corpora paralelos de diferentes tamanhos e géneros. Esta diversidade foi importante a vários níveis:

- embora defendamos a necessidade de corpora de tamanho grande, é importante o uso de diferentes tamanhos para a análise de escalabilidade das ferramentas (de acordo com a secção 7.2), e concluir sobre a influência do tamanho dos corpora na qualidade dos resultados obtidos;
- alguns investigadores defendem que os corpora paralelos de origem literária são de pouca qualidade para a extracção automática de recursos bilingues Para se poder analisar a influência do género linguístico nos algoritmos usados, foram adoptados textos de cariz literário, legislativo e de transcrição oral.
- foram escolhidos corpora de várias línguas para analisar a sua influência nos métodos implementados. Além dos corpora enumerados na secção 7.2 (que incluem as línguas inglesa, francesa, alemã e portuguesa), foram realizadas experiências noutros corpora, de tamanho reduzido, que incluem textos em Latim, Hebreu, Grego e Alemão.

Enquanto que alguns dos corpora usados foram construídos de raiz, outros encontravam-se disponíveis para investigação. No entanto, todos precisaram de ser convertidos, filtrados e limpos de ruído. Este capítulo discute todas estas tarefas inerentes à preparação de corpora paralelo, desde a sua criação à sua disponibilização.

A secção 3.1 dedica-se aos métodos usados para a construção dos corpora paralelos criados, bem como o seu alinhamento ao nível da frase. Segue-se a secção 3.2 que caracteriza cada um dos corpora (criados e adoptados) nomeadamente em termos de tamanho e género literário.

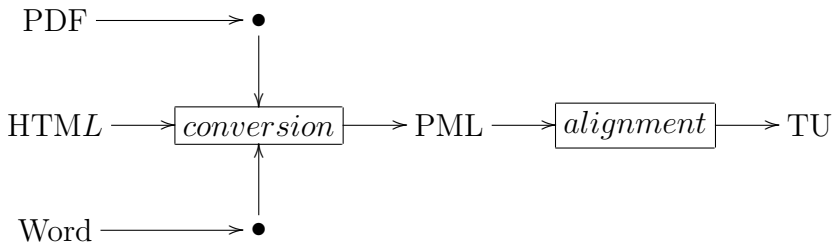
A secção 3.3 discute os problemas no processamento de corpora de grandes dimensões, e como uma abordagem incremental permite a escalabilidade deste processo. Como exemplo de processamento típico de corpora paralelos serão apresentadas algumas abordagens para a sua limpeza, como sejam a detecção e remoção de unidades de tradução repetidas, unidades de tradução não balanceadas (em que uma das línguas tem um comprimento excessivo em relação à outra) ou mesmo a remoção de ruído (entradas não textuais).

Finalmente, a tarefa de disponibilização de corpora (importante pela sua consequência imediata para outros investigadores) e a necessidade do uso de codificadores de corpora para garantir a sua consulta eficiente, são apresentadas na secção 3.4.

3.1 Criação de Corpora

A criação de corpora paralelos é difícil, especialmente no que respeita à recolha de textos paralelos. Actualmente, a forma mais simples é o uso da *Web* como corpus, aplicando técnicas de pesquisa de corpora paralelos na Internet (Resnik, 1998; Almeida, Simões, and Castro, 2002). Além do facto de nem sempre ser simples a detecção de corpora paralelos, é importante realçar os problemas legais que daí podem surgir e que não devem ser descurados.

Depois de detectados, estes documentos devem ser convertidos num formato comum e estruturado para o seu posterior alinhamento. O diagrama seguinte esquematiza este processo.



A secção 3.1.1 descreve os problemas inerentes aos conversores de formatos, e a secção 3.1.2 discute o algoritmo adoptado para o alinhamento à frase.

3.1.1 Injectores

Para que o processo de alinhamento do corpora e todo o fluxo de extracção de recursos possa ser executado de uma forma uniforme, é importante que os documentos extraídos partilhem o mesmo formato. Só assim se poderá aplicar a mesma sequência de processamento a qualquer documento, sem necessidade de duplicar ferramentas para processar tipos de documentos diferentes.

O formato escolhido, e que foi baptizado de PML (*Paragraph Markup Language*), é XML com a estrutura base de um documento: frases, parágrafos e ficheiros:

$$\begin{aligned} \text{Text} &= \text{Paragraph}^\star \\ \text{Paragraph} &= \text{Sentence}^\star \end{aligned}$$

Foram criados injectores de vários tipos de documentos para PML: ferramentas que interpretam formatos específicos, como sejam HTML, PDF ou Word, e os convertem em XML, de acordo com a estrutura do formato PML.

A estrutura do documento é estudada para o reconhecimento de parágrafos. A divisão em frases é realizada posteriormente com um

segmentador (`Lingua::PT::PLNbase`). Este mesmo módulo Perl também permite realizar a atomização das frases (divisão em átomos, ou seja, nas partes básicas que constituem uma frase, como sejam as palavras simples, abreviaturas ou elementos não-palavra como pontuação, e-mails)[1].

Injector HTML

O injector para documentos HTML[2] tem de ter em consideração que nem todas as etiquetas HTML têm texto. Por exemplo, existem zonas com definições de folhas de estilo (CSS) ou de código JavaScript que não devem ser preservadas, uma vez que não têm conteúdo textual. As restantes etiquetas foram divididas em dois grupos: estruturais e de formatação. As etiquetas de formatação como negritos ou itálicos devem ser removidas preservando apenas o seu conteúdo: não forçam o início de um novo parágrafo.

A solução passou pela definição de três conjuntos de etiquetas:

- as que devem ser removidas, bem como o seu conteúdo;
- as que devem ser removidas mas o seu conteúdo preservado;
- as que devem ser transformadas em parágrafos/segmentos;

Injector PDF

O injector de documentos PDF é um pouco mais rudimentar. Um documento PDF inclui pouca informação sobre a estrutura da informação, preocupando-se mais com a sua aparência. Os conversores de PDF para texto (cuja fiabilidade depende em grande parte da ferramenta que foi usada para a criação do PDF) conseguem extrair pouca mais informação para além da existência de alguns parágrafos.

[1] A definição de átomo pode variar de acordo com o contexto. Por exemplo, pode ser importante a divisão das contracções nas partes constituintes ($nas \rightarrow em\ as$), ou a junção das palavras que constituem termos multi-palavra ($Donald\ Knuth \rightarrow Donald + Knuth$).

[2] Em (Sánchez-Villamil et al., 2006) avalia-se o alinhamento de documentos paralelos HTML e também como processar e tirar partido da sua estrutura.

Embora na Web se encontrem outro tipo de documentos, como sejam RTF ou Microsoft Word, os documentos que foram processados para a criação dos nossos corpora limitaram-se aos formatos HTML e PDF.

3.1.2 Alinhamento à Frase

Para a extracção de recursos paralelos é necessário estabelecer pontes entre as duas línguas do corpus paralelo: extrair relacionamentos entre termos, segmentos de palavras ou frases.

Definição 4 *Dados textos paralelos U e V, um **alinhamento** é uma segmentação de U e V em n segmentos cada, tal que para cada i, $1 \leq i \leq n$, u_i e v_i são traduções mútuas.*

*Um par de **segmentos alinhados** (ou **unidade de tradução**) a_i é um par ordenado (u_i, v_i). Desta forma, um alinhamento A também pode ser definido como uma sequência de segmentos alinhados: $A \cong \langle a_1, a_2, \ldots, a_n \rangle$.*

O alinhamento mais básico e mais fácil de obter (com pouco custo computacional) é o alinhamento entre frases. Este tipo de alinhamento associa a cada frase do corpus da língua de origem uma frase no corpus da língua de destino, que corresponde à sua tradução.

A tradução nem sempre preserva o número de frases. É habitual a divisão ou junção de frases pelo tradutor, de acordo com as suas necessidades linguísticas ou estilísticas. Este tipo de liberdade (que permite a adição ou remoção de frases) leva a que nem sempre se obtenham relacionamentos entre frases únicas, mas por vezes, entre uma frase e um par de frases, ou genericamente, entre n e m frases.

Definição 5 *Dados textos paralelos U e V, um **alinhamento à frase** é uma segmentação de U e V em n segmentos cada, tal que para cada i, $1 \leq i \leq n$, u_i e v_i são traduções mútuas, u_i é uma sequência de frases de U e v_i é uma sequência de frases de V.*

As sequências u_i e v_i são tão curtas quanto possível, sendo normalmente de comprimento 1 (alinhamento de frase para frase). No entanto também podem ocorrer relações de uma para nenhuma frase (situação em que o tradutor removeu ou adicionou uma frase) ou de uma para várias frases (situação em que o tradutor subdividiu ou juntou frases).

O pacote NATools inclui um alinhador à frase, derivado do Vanilla Aligner (Danielsson and Ridings, 1997). Este alinhador é uma implementação do algoritmo de (Gale and Church, 1991). O algoritmo é baseado na quantidade de frases em cada língua, e nos comprimentos dessas mesmas frases. Com base em programação dinâmica, o algoritmo procura agrupar frases de forma a que os tamanhos destes grupos sejam equilibrados entre línguas.

Existem várias heurísticas para ajudar este algoritmo a obter melhores resultados, como seja o uso de listas de palavras bilingues, ou de palavras que se traduzem por ela mesma, que permitem estabelecer âncoras durante o alinhamento.

O EasyAlign[3] é um alinhador à frase que usa este tipo de heurísticas, pelo que deverá ser mais robusto. Para mais discussão sobre a avaliação de alinhadores à frase sugere-se a consulta de (Caseli and Nunes, 2003) e (Simões and Almeida, 2007).

Esta diferença de robustez foi a principal razão para o termos usado no alinhamento dos corpora paralelos construídos.

[3]O EasyAlign faz parte do IMS Corpus Workbench (Christ et al., 1999).

O alinhamento à frase pode ser melhorado com o uso de dicionários bilingues. Estes dicionários podem ser obtidos a partir de dicionários probabilísticos de tradução (de acordo com a secção 4.3.1).

3.2 Corpora Paralelos Utilizados

A tabela 3.2 apresenta um resumo dos vários corpora coleccionados e relaciona-os de acordo com a quantidade de unidades de tradução existente. A tabela 3.2 detalha esta informação comparativa ao nível do número de palavras e de *tokens*.

É importante salientar que, com excepção do corpus da Constituição Portuguesa, de um par de corpora, todos os outros têm uma evolução temporal bastante forte. As próximas secções apresentam algumas características destes corpora, nomeadamente em termos de conteúdo, tamanho e diversidade.

Corpus	PT–EN	PT–ES	PT–FR
Constituição	2 013	2 011	2 013
COMPARA	97 215	—	—
Le Monde Diplomatique	—	—	68 231
JRC	286 008	281 185	277 754
EuroParl	998 830	1 006 895	1 023 841
EurLex	10 394 893	1 111 068	1 710 760

Tabela 3.1: Número de unidades de tradução por corpus paralelo.

3.2.1 Constituição Portuguesa

A Constituição da República Portuguesa está disponível na Internet em várias línguas. Neste caso concreto, o processo de construção do corpus correspondeu à procura de uma versão em mais do que uma língua (quarta revisão constitucional), a cópia local dos documentos e a sua conversão para o formato PML. A sua estrutura por partes, títulos,

Corpus		Número Tokens		Número Formas	
		Origem	Destino	Origem	Destino
Constituição	PT:EN	38 024	40 984	3 761	3 113
	PT:ES	38 024	41 855	3 761	3 817
	PT:FR	38 024	42 484	3 761	3 916
Compara	PT:EN	1 714 049	1 797 976	71 759	45 429
L.M.D.	FR:PT	1 730 166	1 887 250	66 950	59 009
JRC-Acquis	PT:EN	8 248 333	7 797 133	68 325	55 797
	PT:ES	8 005 805	8 333 518	67 314	64 471
	PT:FR	7 934 385	8 134 116	66 939	59 453
EuroParl	PT:EN	29 232 417	28 366 649	137 607	87 511
	PT:ES	29 331 905	29 736 743	142 189	135 126
	PT:FR	29 826 035	33 286 644	148 259	108 356
EurLex	PT:EN	226 600 339	213 832 551	658 601	608 921
	PT:ES	22 904 057	23 724 321	161 804	158 942
	PT:FR	36 589 842	39 799 740	206 467	184 405

Tabela 3.2: Comparação do número de tokens e formas entre corpora.

capítulos e artigos aumentou a qualidade do alinhamento ao nível da frase.

Embora este corpus seja limitado pelo seu tamanho reduzido e género linguístico, tornou-se bastante útil para a realização de experiências rápidas: é um texto pequeno, com pouco ruído e um bom alinhamento.

3.2.2 COMPARA

O corpus paralelo COMPARA (Frankenberg-Garcia and Santos, 2001; Frankenberg-Garcia and Santos, 2003) contém uma colecção de textos literários paralelos português:inglês. Este corpus está a ser desenvolvido no âmbito da Linguateca[4].

O COMPARA é um corpus em crescimento: tem vindo a incorporar novos textos sempre que tal se torna possível. Em Janeiro de 2008 o

[4]O COMPARA está disponível para pesquisa em `http://www.linguateca.pt/COMPARA/`, e acessível a partir da página principal da Linguateca, `http://www.linguateca.pt/`.

COMPARA incluía 72 pares de textos de ficção contemporânea e não contemporânea, de autores e tradutores da África do Sul, Angola, Brasil, Estados Unidos, Moçambique, Portugal e Reino Unido.

Dado o cariz literário deste corpus torna-se difícil a extracção de recursos bilingues bons, já que é menos rico terminologicamente, e as traduções são menos genéricas: são realizadas especificamente para aquele texto, aquele autor, e aquela situação em concreto do enredo.

3.2.3 Le Monde Diplomatique

O *Le Monde Diplomatique* é um prestigiado jornal com mais de 28 anos de origem francesa focado na discussão política internacional. Embora bastante mais jovem, existe uma versão portuguesa deste jornal do qual cerca de 70% do conteúdo corresponde à tradução de artigos da sua versão francesa. Deste modo, torna-se possível extrair dos arquivos existentes bitextos de artigos publicados.

Num trabalho de colaboração com os detentores destes jornais e o Instituto de Letras e Ciências Humanas da Universidade do Minho (Correia, 2006), foi criado um corpus francês:português.

O processo de criação consistiu na reunião de artigos franceses e portugueses do seguinte modo:

- extracção dos textos em língua francesa a partir da base de dados do sistema de publicação electrónica usado (SPIP[5]);
- extracção dos textos em língua portuguesa a partir de um conjunto de documentos em formato HTML;
- extracção dos nomes dos autores dos vários artigos, e criação de relacionamentos brutos entre artigos (classes de artigos por autor), usando os tamanhos dos artigos para a obtenção de um relacionamento mais detalhado entre artigos;
- validação manual das correspondências propostas entre os artigos em cada classe;

[5]Informações sobre o sistema de publicação electrónica SPIP podem ser encontradas em `http://www.spip.net/`.

- processamento do relacionamento obtido, atomização e segmentação dos textos, e posterior alinhamento ao nível da frase (de acordo com o descrito na secção 3.1.2);
- disponibilização em vários formatos (TMX, NATools) para uso local, e na Internet, para pesquisa de concordâncias.

Este corpus tem um tamanho muito próximo do COMPARA, mas um género (jornalístico) completamente diferente e também uma qualidade de tradução inferior. Embora o facto de as línguas envolvidas não serem as mesmas e portanto não permitir comparações directas, é interessante para a comparação de rendimento de determinados algoritmos.

3.2.4 JRC-Acquis Multilingual Parallel Corpus

Para se juntar à União Europeia, os novos Estados Membros precisam de traduzir e aprovar a legislação actual da UE, que consiste em textos escritos entre 1950 e 2005. Este corpo de texto legislativo, que é composto por aproximadamente 800 documentos que cobrem uma gama variada de domínios, é chamado *Acquis Communautaire*.

No início de 2005 existiam 20 línguas oficiais na União Europeia pelo que este corpo legislativo existe como texto paralelo em 20 línguas: checo, dinamarquês, alemão, grego, inglês, espanhol, estónio, finlandês, francês, húngaro, italiano, lituano, letão, maltês, holandês, polaco, português, eslovaco, esloveno e sueco.

O *Acquis Communautaire* e outra legislação comunitária existe disponível publicamente nas páginas na Internet da Comissão Europeia. A equipa de Tecnologia da Língua, do Joint Research Centre (JRC) em Ispre, Itália, identificou os documentos que fazem parte do *Acquis Communautaire* e converteu-os para um formato XML. Em passos subsequentes, os textos foram limpos dos seus cabeçalhos e anexos, e foram alinhados ao parágrafo usando dois alinhadores: Vanilla Aligner e o HunAlign (Varga et al., 2005). Deste processo resultou um corpus paralelo multilingue JRC-Acquis (Steinberger et al., 2006) que tem vindo a ser continuamente expandido e melhorado.

O JRC-Acquis está disponível como um ficheiro TEI (Text Encoding

Initiative) diferente por língua, e um ficheiro para cada alinhamento, também em formato TEI. Na prática, cada ficheiro XML de texto em determinada língua contém o corpus dividido em frases anotadas com um identificador único. O ficheiro de alinhamento inclui correspondências entre conjuntos de identificadores. Foi implementada uma pequena ferramenta para a conversão deste formato em TMX[6].

No trabalho realizado durante esta dissertação foram utilizados os pares português:inglês, português:espanhol e português:francês.

3.2.5 EuroParl: European Parliament Proceedings

O EuroParl[7] (Koehn, 2002) foi compilado com base nas actas do Parlamento Europeu de 1996 a 2003, com supervisão de Philipp Koehn. Inclui versões em 11 línguas europeias (francês, italiano, espanhol, português, inglês, holandês, alemão, dinamarquês, sueco, grego e finlandês). É distribuído em ficheiros separados, um por língua, juntamente com um pequeno programa para realizar os alinhamentos. A partir da página web do corpus também é possível descarregar 10 corpora paralelos já alinhados (inglês alinhado com cada uma das outras línguas).

Este corpus tem vindo a crescer, tendo sido disponibilizada em Setembro de 2007 a sua versão 3. No caso concreto do trabalho realizado durante esta dissertação, foi utilizada a versão 2 e foi usado o programa de alinhamento para criar os corpora correspondentes aos pares português:inglês, português:espanhol e português:francês. Estes corpora foram posteriormente filtrados de algum ruído resultante do alinhamento.

3.2.6 EurLex

O EurLex é constituído por vários corpora paralelos que foram construídos com base no Jornal Das Comunidades, disponibilizado pela Comunidade Europeia na Internet.

[6] A secção 3.3.1 fará uma pequena comparação destes dois formatos, e porque é que no nosso trabalho optamos por usar o TMX.

[7] O EuroParl está disponível em `http://www.statmt.org/europarl/`.

O processo de construção passa pela recolha dos textos paralelos na Internet (Almeida, Simões, and Castro, 2002), de onde resulta um conjunto de documentos em formato HTML que são posteriormente convertidos e alinhados (ver secção 3.1). Durante o alinhamento é feita a contagem dos vários tipos de alinhamento (1:1, 1:2, 2:1, etc). O alinhamento de um par de ficheiros é rejeitado se a percentagem de alinhamentos 1:1 for demasiado baixa.

Estes corpora são bastante maiores do que os restantes (especialmente o português:inglês), e bastante ricos em terminologia.

Os corpora contêm algum ruído resultante do alinhamento e conversão do HTML. Na secção 3.3.4 são discutidos vários métodos para a limpeza de corpora paralelos, métodos esses que foram aplicados ao EurLex. Nessa mesma secção serão apresentadas várias medidas relativas a esta limpeza, como sejam a taxa de repetição de unidades de tradução, ou a percentagem de unidades de tradução desequilibradas.

3.3 Processamento de Corpora Paralelos

Para os nossos objectivos interessa-nos o processamento de corpora paralelos alinhados ao nível da frase. Este processamento deve ser realizado de forma uniforme, escalável, e que permita abstrair o formato concreto em que o corpus se encontra.

3.3.1 Formatos de Corpora Paralelos

É habitual que cada investigador use o seu próprio formato para a codificação dos seus corpora. Embora exista a tentativa de definição de um standard (XCES[8]) a falta de ferramentas para o seu processamento tem limitado a sua globalização.

No caso concreto dos corpora paralelos existem duas outras abordagens comuns: o uso dos esquema do Text Encoding Initiative (TEI[9]) e

[8]Corpus Encoding Standard for XML — `http://www.xml-ces.org/`
[9]Text Encoding Initiative — `http://www.tei-c.org/index.xml`

o uso do formato de intercâmbio de memórias de tradução (TMX[10]).

O TEI tem vindo a ser usado especialmente em corpora multilingue (com mais de duas línguas) uma vez que permite poupar espaço em disco, reutilizando cada um dos corpora de cada língua: cada corpus é codificado num ficheiro XML, em que cada frase ($s_{\mathcal{A}}$ de um corpus $c_{\mathcal{A}}$ na língua $\mathcal{A}$) é etiquetada com um identificador único: $\binom{\mathrm{id}\,(s_{\mathcal{A}})}{s_{\mathcal{A}}}_{s_{\mathcal{A}} \in c_{\mathcal{A}}}$. Para cada alinhamento (para cada par de línguas $\mathcal{A}$ e $\mathcal{B}$) existe um relacionamento entre sequências de identificadores. Se $\mathrm{ID}_{\mathcal{A}}$ corresponder ao identificador de um segmento na língua A, então o alinhamento pode ser visto como um elemento do tipo $(\mathrm{ID}^{\star}_{\mathcal{A}} \times \mathrm{ID}^{\star}_{\mathcal{B}})^{\star}$.

O TMX é especialmente usado no mundo da tradução assistida por computador, para a codificação e intercâmbio de memórias de tradução entre ferramentas. Este formato tem a vantagem de ser mais simples de processar: é armazenado num único ficheiro e tem uma notação XML muito simples. Além disso, permite maior facilidade no intercâmbio com a comunidade de tradutores, pelo que se adoptou o formato TMX e se implementou conversores entre TEI e TMX.

> O formato TMX é mais simples e rápido de processar do que o TEI. Permite a utilização de corpora paralelos como memórias de tradução, e o uso de memórias de tradução como corpora paralelos.

A figura 3.1 apresenta um pequeno documento TMX. O TMX é um formato estruturado de acordo com a gramática que se segue:

$$\begin{aligned} \mathrm{TMX} &= \mathrm{Head} \times \mathrm{Body} \\ \mathrm{Body} &= \mathrm{TU}^{\star} \\ \mathrm{TU} &= \mathrm{TUV}^{\star} \times \mathrm{Note}^{\star} \times \mathrm{Prop}^{\star} \\ \mathrm{TUV} &= \mathrm{Seg} \times \mathrm{Note}^{\star} \times \mathrm{Prop}^{\star} \end{aligned}$$

Um documento TMX é composto por dois grandes blocos: o cabeçalho de meta informação, e o corpo. Esta segunda parte — a principal destes

[10]Translation Memory eXchange — `http://www.lisa.org/standards/tmx/`

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<!DOCTYPE tmx SYSTEM "tmx14.dtd">
<tmx version="version 1.4">
<header  creationtool="cwb-utils"
         creationtoolversion="1.0"
         segtype="sentence"
         adminlang="EN-US"
         srclang="fr"
         o-tmf="CQP-corpora" />
<body>

<tu>
 <tuv lang='pt'>
    <seg>Praticamente ausente dos mapas de fluxo de dados, a
    África não contabiliza mais linhas telefónicas do que Tóquio
    ou Manhattan, nem mais computadores ligados à Internet do
    que a Lituânia.</seg>
 </tuv>
 <tuv lang='fr'>
    <seg>Quasi absente des cartes de flux de données, l'Afrique
    ne compte pas plus de lignes téléphoniques que Tokyo ou
    Manhattan, pas plus d'ordinateurs connectés à Internet que
    la Lituanie.</seg>
 </tuv>
</tu>

<tu>
 <tuv lang='pt'>
    <seg>Todavia, o continente não escapa às transformações
    nas telecomunicações, onde se lêem, mais do que em qualquer
    outro sítio, as recomposições inéditas impostas pela
    mundialização.</seg>
 </tuv>
 <tuv lang='fr'>
    <seg>Pourtant, le continent n'échappe pas au bouleversement
    des télécommunications, dans lequel se donnent à lire, là
    plus qu'ailleurs, les recompositions inédites qu'impose la
    mondialisation.</seg>
 </tuv>
</tu>
   [...]
</body>
</tmx>
```

Figura 3.1: Extracto de um documento TMX.

documentos — é composta por pequenas entradas, correspondentes às unidades de tradução[11]. Cada unidade de tradução (representada pela etiqueta `tu`) contém uma ou mais variantes da unidade de tradução por língua (etiquetas `tuv`). Dentro destas encontra-se o segmento de texto que compõe a unidade de tradução, juntamente com alguma meta-informação opcional (propriedades e notas).

O *standard* permite o uso de várias etiquetas dentro dos segmentos de texto, de forma a preservar a formatação original do documento. Permite também o uso da etiqueta `hi` para sublinhar (*highlight*) secções de texto especiais. De acordo com o standard, é usado para delimitar unidades terminológicas, nomes próprios, palavras que não devem ser traduzidas, etc. Suporta um atributo `type` para especificar o tipo da secção de texto marcada.

É importante realçar que uma memória de tradução e um corpus paralelo podem ser vistos como isomórficos, sempre e quando se considere que a ordem das memórias de tradução é preservada (ordem esta que não é garantida de acordo com o *standard*).

3.3.2 Necessidade de Processamento de Corpora Paralelos

Durante o processo de construção de um corpus paralelo é necessário realizar várias tarefas sobre um corpus, como sejam:

- **anotação** do corpus:
  - adição de lemas por palavra (numa ou ambas as línguas);
  - detecção de entidades mencionadas;
  - cálculo de Part-Of-Speech por palavra;
- a **limpeza** de Corpora Paralelos, removendo unidades de tradução anómalas;
- a **conversão** de formatos (TMX para TEI, TMX para o formato

[11] No final de escrita desta dissertação a associação LISA colocou disponível a versão 2.0 do *standard* do formato TMX para discussão pública. No entanto, todos os exemplos aqui apresentados correspondem à versão 1.4.

usado pelo NATools, etc.);

- a **extracção de sub-corpora**, limitando o número de unidades de tradução, ou procurando e extraindo apenas unidades de tradução com determinados padrões linguísticos;
- a adição de **propriedades e notas** com meta-informação às unidades de tradução, como sejam a área temática em que se insere ou uma medida de qualidade (ver figura 3.2);

Estas e outras tarefas são úteis quer por si só, quer como forma de enriquecer o corpus para tarefas subsequentes.

O processamento de um corpus paralelo, essencialmente depois de ter sido escolhido um formato único para os armazenar, deve ser realizado tentando abstrair o mais possível do formato em causa, permitindo ao programador concentrar-se na tarefa que pretende resolver.

3.3.3 Processamento de Ordem Superior

Para permitir que o programador se abstraia do formato concreto em que o corpus está codificado, foi desenvolvida uma API de ordem superior (Dominus, 2005).

De acordo com as várias tarefas que foram propostas, cada unidade de tradução pode ser processada de forma independente, pelo que a função de processamento poderá ser invocada para cada unidade de tradução existente. Ou seja, é possível invocar um processador de ordem superior, com uma função específica que irá processar cada uma das unidades de tradução. Esta função tem a seguinte assinatura:

$$proc : \mathrm{TU} \times \mathrm{Prop}^\star \times \mathrm{Note}^\star \longrightarrow (\mathrm{TU} \times \mathrm{Prop}^\star \times \mathrm{Note}^\star) + \bot$$

Quando o valor retornado é $\bot$, a unidade de tradução será removida. A função *proc* pode:

- transformar unidades de tradução: $\mathrm{TU} \rightarrow \mathrm{TU}$
  o processador não é mais que um *map* funcional que aplica a cada unidade de tradução uma função de processamento que devolve a unidade de tradução depois de processada (e/ou produz efeitos laterais: $tu \times state \rightarrow tu \times state$);

- remover unidades de tradução: $\mathrm{TU} \rightarrow \perp$
  no caso da função de processamento devolver um objecto vazio, a unidade é retirada da memória de tradução gerada gerada.
- alterar propriedades: $\mathrm{TU} \times \mathrm{Prop}^{\star} \times \mathrm{Note}^{\star} \rightarrow \mathrm{TU} \times \mathrm{Prop}^{\star} \times \mathrm{Note}^{\star}$
  além do texto e respectiva tradução o *standard* TMX permite definir propriedades (etiqueta `prop`) e notas (etiqueta `note`) sobre cada unidade de tradução. A função de processamento recebe não só o texto correspondente à unidade de tradução mas também a lista de propriedades e de notas associadas, podendo alterá-las, removê-las ou adicionar novas.

O processador permite ainda receber um conjunto de opções que controlam como, quantas e quais unidades de tradução são processadas:

- indicar o ficheiro de saída pretendido:
  por omissão a função escreve a nova memória de tradução para o *standard output*. No entanto este comportamento pode ser alterado indicando o nome do ficheiro para onde a nova memória deve ser escrita.
- a criação ou processamento de sub-corpora:
  - a definir um número máximo de TU a processar:
    em algumas ferramentas, como as que funcionam sobre a web, é importante limitar o número de unidades de tradução a processar de forma a aliviar o processamento. Este número pode ser definido ao invocar o processador, que parará após a n-ésima unidade de tradução.
  - definir o número máximo de TU a obter:
    funciona de forma semelhante à anterior, mas em vez de limitar o número de unidades de tradução a processar, processa unidades de tradução até que seja retornado o número de unidades pretendido.
- indicar um padrão de activação:
  permite especificar uma expressão regular de pesquisa, de forma a que apenas as unidades de tradução que façam *matching* sejam processadas.

> O uso de uma API de alto nível permite que o programador se posso concentrar na tarefa a realizar e não nos pormenores intrínsecos ao formato em que o corpus se encontra.

Antes de apresentarmos exemplos reais, é aqui discutido um exemplo trivial, que usa este processador de ordem superior para contar o número de unidades de tradução existentes numa TMX.

```
1  use  XML::TMX::Reader;
2  my $mem = XML::TMX::Reader->new('sample.tmx');

3  my $count = 0;
4  $mem->for_tu(
5      sub { $count++; }
6    );

7  print $count;
```

linha 1: carregar o módulo para leitura de TMX;
linha 2: criar um objecto com a TMX em causa;
linha 4: iterar com `for_tu` sobre todas as unidades de tradução;
linha 5: definir a função de processamento da unidade de tradução que se limita a contar o número de unidades encontradas.

3.3.4 Exemplos de uso: Limpeza de Corpora Paralelos

Em todo o trabalho de extracção de informação a partir de corpora paralelos há uma grande dependência da qualidade das unidades de tradução da TMX de partida. Neste sentido, há necessidade de um conjunto de estratégias para a avaliação de memórias de tradução e a sua remoção ou tratamento automático.

Esta secção serve dois propósitos: exemplificar o uso da API de ordem superior para o processamento de corpora paralelos, e apresentar um conjunto de heurísticas e métricas para o aumento de qualidade de um corpus paralelo.

Remoção de entradas duplicadas

Ao criar e juntar memórias de tradução acabam por existir unidades de tradução repetidas. Embora a eliminação de entradas duplicadas seja discutível dadas as diferenças obtidas nos recursos extraídos, a sua contabilização é imprescindível. No caso concreto de corpora paralelos criados automaticamente por extracção de informação a partir da Internet a remoção de entradas duplicadas acaba por ser benéfica.

O exemplo abaixo apresentado mostra uma forma rápida de as remover, usando para isso o valor de *hashing* MD5 de cada unidade de tradução[12].

```
1  tie %dic, 'DB_File', "mydbfile.db",
2      O_RDWR|O_CREAT|O_TRUNC , 0640, $DB_BTREE;

3  my $tm = XML::TMX::Reader->new($filename);

4  $tm->for_tu(
5      sub {
6          my $tu = shift;
7          my $digest = md5(normaliza("$tu->{en},$tu->{pt}"));

8          if ($dic{$digest}) {
9                 return undef
10         } else {
11                $dic{$digest} = 1;
12                return {%$tu} ;
13         }
14     }
15 );
```

linha 1: criar uma base de dados (em disco) de valores MD5 para consulta rápida;
linha 5: iterar todas as memórias de tradução;
linha 7: calcular o valor MD5 da unidade de tradução depois de normalizada;

[12]Não é possível o uso directo das unidades de tradução em vez do seu MD5, já que levaria à criação de uma base de dados demasiado grande (e mais lenta, devido às comparações de grandes sequências de palavras).

linha 8: se o valor MD5 está na base de dados, a unidade é repetida pelo que é ignorada;

linha 10: se o valor não existe, é guardado na base de dados e a unidade de tradução é devolvda.

Aplicando este algoritmo ao corpus EurLex português:inglês foram removidas mais de quatro milhões de unidades de tradução (40% das unidades de tradução). Este processo demorou cerca de 24 horas[13] e foi criada uma base de dados de valores MD5 com mais de 600 MB.

Remoção de unidades anómalas

Ao criar corpora paralelos de forma automática, é habitual existirem maus alinhamentos (unidades de tradução cujo texto não corresponde, ou corresponde parcialmente, à tradução correcta). Uma heurística simples que permite a remoção automática de várias destas unidades de tradução passa pela comparação dos tamanhos dos segmentos: se uma unidade de tradução tiver segmentos com tamanhos muito díspares deve ser removida.

Outra heurística para a detecção de unidades de tradução anómalas é a comparação dos elementos não textuais, como sejam os números presentes no texto (tipicamente o conjunto de números são comuns entre línguas).

Para a limpeza dos vários corpora usados, além da normalização de entradas e posterior remoção de entradas duplicadas, utilizaram-se as seguintes heurísticas:

- remoção de unidades sem elementos textuais;
- remoção de unidades com tamanho superior a 50 caracteres e em que o tamanho do segmento numa língua seja superior ao dobro do tamanho do outro;
- cálculo da quantidade de números contidos em cada segmento da unidade de tradução, e remoção daquelas em que a diferença seja superior a 3 números. Esta abordagem não pode ser mais restritiva

[13] Limpeza realizada num Pentium IV a 3GHz, com 3GB de RAM.

(como obrigar a que os números fossem exactamente os mesmos ou que a sua quantidade fosse exactamente a mesma), já que muitas unidades de tradução contêm certos números em notação arábica numa das línguas, e por extenso na outra língua.

O algoritmo 1 mostra o uso destas heurísticas para a limpeza de corpora.

$tu_{pt} \longleftarrow normaliza(tu_{pt})$
$tu_{en} \longleftarrow normaliza(tu_{en})$
$aceitar \longleftarrow$ **True**
if $\neg contém\_letras(tu_{pt}) \vee \neg contém\_letras(tu_{en})$ **then**
  $aceitar \longleftarrow$ **False**
if $tamanho(tu_{pt}) > 50 \wedge tamanho(tu_{en}) > 50 \wedge$
  $(tamanho(tu_{pt}) > 2 \times tamanho(tu_{en}) \vee$
  $tamanho(tu_{en}) > 2 \times tamanho(tu_{pt}))$ **then**
  $aceitar \longleftarrow$ **False**
$núm_{pt} \longleftarrow extrai\_números(tu_{pt})$
$núm_{en} \longleftarrow extrai\_números(tu_{en})$
if $|\#núm_{pt} - \#núms_{en}| > 3$ **then**
  $aceitar \longleftarrow$ **False**
if $|\#(núm_{pt} \bigcap núm_{en}) - max(\#núm_{pt}, \#núm_{en})| > 2$ **then**
  $aceitar \longleftarrow$ **False**
if $aceitar$ **then**
  return tu
else
  return undef

Algoritmo 1: Detecção de unidades de tradução anómalas.

A aplicação destas heurísticas ao corpus EurLex português:inglês resultou na eliminação de 124 mil unidades sem letras, 43 mil unidades com tamanhos díspares, e 37 mil com uma quantidade de números (muito) desequilibrada. Este processo demorou cerca de hora e meia. Após a remoção de entradas duplicadas e de entradas anómalas o corpus EurLex reduziu 40% (passou de 10 394 893 a 6 021 642 unidades de tradução).

3.3.5 Implementação e Escalabilidade

A possibilidade de processamento de corpora paralelos de forma independente do formato, e com funções de ordem superior é bastante prática. Este facto é especialmente verdade se o processador de ordem superior estiver preparado para escalar para tamanhos *reais* de corpora. Foi necessária a implementação de uma abordagem híbrida para o processamento de TMX (Almeida and Simões, 2007) a usar dois métodos para processamento de documentos XML: SAX e DOM.

Processamento Híbrido de TMX

A abordagem para o processamento de memórias de tradução de grandes dimensões aqui apresentada, baseia-se na na grande repetição de certos elementos XML. O corpo de um documento TMX não é mais que uma sequência de unidades de tradução em que cada uma é um documento XML perfeitamente válido: as etiquetas encontram-se correctamente aninhadas e existe um bloco (`tu`) que alberga todas as outras etiquetas, pelo que é possível usar um processador típico de documentos XML passando-lhe apenas uma unidade de tradução.

A implementação do algoritmo usou como base a facilidade da linguagem Perl na definição de um separador de registo que é usado pelos métodos de leitura de ficheiros para a divisão do documento em porções (registos) de informação. Definindo como separador de registo a etiqueta de término da unidade de tradução (`</tu>`) todos os registos (com excepção do primeiro e do último) contêm unidades de tradução completas.

Cada um destes registos é processado pelo módulo XML::DT (Almeida and Ramalho, 1999) que constrói uma árvore DOM para cada uma destas unidades de tradução. Esta abordagem obriga à inicialização de um *parser* XML para cada uma das unidades de tradução o que o torna o processo lento, mas escalável já que não é necessária a criação da totalidade da árvore DOM em memória.

Este algoritmo não é mais do que o processamento SAX do documento TMX (etiquetas `tu`, que delimitam unidades de tradução), e o

posterior processamento DOM (conteúdo dessas etiquetas).

> Uma abordagem híbrida SAX e DOM permite processar documentos XML com uma estrutura repetitiva de forma eficaz e escalável.

Esta abordagem, embora tenha sido implementada com vista à resolução do problema no processamento de memórias de tradução é facilmente generalizável para outros esquemas de documentos XML.

Considerações referentes a desempenho

A tabela 3.3 mostra uma comparação de tempos[14] do processador de ordem superior **for_tu**, implementado com base na construção da árvore DOM completa ou usando o processamento incremental por *chunks*. Foi construído um exemplo de teste que conta o número de unidades de tradução (ver secção 3.3.3), que foi testado com memórias de tradução com diferentes quantidades de unidades de tradução.

Enquanto o DOM do documento cabe em memória, esta abordagem é mais eficiente. Assim que o DOM deixa de caber em memória, esta abordagem deixa de ser exequível. Por outro lado, a abordagem de processamento incremental por *chunks* tem um crescimento linear. Embora possa demorar mais tempo consegue dar uma resposta. Note-se que são normais memórias de tradução com mais de um milhão de unidades de tradução.

Considerando um exemplo mais complexo como seja a remoção de unidades de tradução repetidas (ver secção 3.3.4), a abordagem de processamento incremental demorou cerca de 35 minutos e 25 segundos para uma memória de tradução com 1 784 164 unidades (removendo 47% de unidades repetidas).

[14] As medidas apresentadas nesta secção foram obtidas num Pentium IV, 3GHz, com 3GB de RAM, Linux.

		DOM		Chunks	
TUs	Tamanho	tempo	memória	tempo	memória
53 500	18 MB	38s	108 MB	50s	10 MB
68 000	25 MB	41s	145 MB	61s	10 MB
380 500	83 MB	230s	637 MB	343s	10 MB
1 110 000	353 MB	—	—	1003s	10 MB

Tabela 3.3: Comparação de tempos de parsing de memórias de tradução.

3.4 Indexação e Disponibilização

Depois de estabilizados, limpos e etiquetados, os corpora paralelos podem ser utilizados para diversas tarefas, como sejam a consulta de concordâncias via Web, o acesso de forma programática para a extracção de recursos, ou a sua integração num sistema de tradução automática. Todas estas e outras tarefas precisam de consultar os corpora de forma eficiente, pesquisando unidades de tradução específicas, ou com determinados padrões.

Quando os corpora começam a crescer a eficiência na pesquisa torna-se relevante. Enquanto que para a pesquisa num corpus pequeno uma aplicação pode ler e consultar o corpus de cada vez que o utilizador faz uma pesquisa, para a pesquisa num corpus médio/grande esta mesma abordagem não é possível.

É importante a disponibilização eficaz de corpora paralelos:

- com uma API simples e eficiente, que permita a uma aplicação consultar corpora paralelos sem que para isso precise de se fazer passar por um utilizador comum (como por exemplo, usando uma Interface Web desenhada especialmente para utilizadores humanos);
- que permita a consulta por utilizadores pouco ou nada familiarizados com a programação, utilizando uma interface intuitiva especialmente desenhada para eles.

Para ambas as situações, é importante a indexação dos corpora para permitir pesquisa eficiente de concordâncias. Os corpora utilizados pelas

ferramentas do NATools para a extracção de recursos bilingues devem ser pré-processados e indexados previamente. A secção 3.4.2 aborda este processo de indexação, começando por analisar outras ferramentas já existentes para a indexação e disponibilização de corpora.

> A indexação de corpora é imprescindível para que se possam consultar de forma eficiente.

3.4.1 Gestores de Corpora

Quando estamos em presença de corpora de dimensões médio/grande, a pesquisa em texto livre não é eficiente e por isso, cedo se sente necessidade de criar sistemas de indexação de texto. A indexação básica de texto, habitual em sistemas de recolha de informação como o Glimpse[15] ou o `ht://Dig`[16], não se mostraram versáteis para as necessidades no armazenamento e indexação de corpora.

Em (Bernardini, Baroni, and Evert, 2006) são apontadas quatro características importantes dos sistemas de indexação de corpora:

- **expressividade:** o sistema deve permitir realizar pesquisas complexas, não apenas pesquisas booleanas de palavras, mas também pesquisas sobre anotações específicas como sejam o *Part-Of-Speech* de determinada palavra, ou a sua função sintáctica;
- **facilidade de uso:** não deve ser preciso mais do que cinco minutos para que o utilizador consiga aprender a linguagem de pesquisa, e consiga fazer pesquisas razoavelmente complexas;
- **desempenho:** embora muitas tarefas de PLN possam ser executadas *durante a noite*, a maior parte dos utilizadores querem o resultado das suas pesquisas imediatamente. O sistema deve

[15] O motor de indexação Glimpse, e o software de indexação de páginas Web WebGlimpse estão disponíveis em `http://webglimpse.net/`.

[16] O `ht://Dig` é um pacote de software para a indexação de sites Web, permitindo uma ordenação de resultados com base em métricas de relevância. Está disponível em `http://www.htdig.org/`.

ser rápido a responder à generalidade das expressões de pesquisa independentemente da sua complexidade;

- **escalabilidade:** os corpora existentes são cada vez maiores, e cada vez os seus utilizadores procuram que eles cresçam. O sistema deve ser robusto para conseguir gerir corpora com milhões de palavras.

Adicionalmente, para investigação em Processamento de Linguagem Natural, existem outras características importantes quando não se pretende apenas disponibilizar corpora mas também utilizar esse corpora em ferramentas automáticas:

- **disponibilidade:** a ferramenta deve estar disponível livremente para qualquer utilizador. A disponibilidade do código-fonte da aplicação torna mais simples a análise, melhoramento ou adaptação da aplicação, permitindo mesmo que sirva de ponto de partida para novas ferramentas;
- **programabilidade:** a interface de um programa com uma aplicação desenvolvida tendo em vista o utilizador final não é trivial e, na grande maioria dos casos, é lenta. É importante a existência de uma API versátil.

Estas ferramentas de gestão de corpora estão habitualmente divididas em três módulos

- **indexador:** processa o corpus, codifica-o e cria índices;
- **servidor:** consulta os índices, e responde às pesquisas efectuadas;
- **clientes:** fazem a interacção entre o utilizador e o servidor.

Segue-se a discussão de alguns sistemas que têm vindo a ser utilizados para a indexação, pesquisa e disponibilização de corpora.

Sara e Xaira

O XAIRA[17] (*XML Aware Indexing and Retrieval Architecture*) é o substituto do SARA, o sistema de indexação desenvolvido originalmente para o *British National Corpus*[18]. Foi desenvolvido tendo em consideração as seguintes premissas:

- permitir indexar qualquer corpus codificado em XML, embora tenha sido desenvolvido para usar documentos codificados em TEI;
- suportar completamente o uso de Unicode;
- estar disponível em código aberto, encontrando-se sob a GNU General Public License;
- permitir a escrita de clientes que acedam ao servidor Xaira usando várias API (C++ e APIs web: XML-RPC e SOAP);

O Xaira foi desenvolvido a pensar unicamente em corpora monolingues o que tornaria o seu uso difícil para corpora paralelos.

IMS Corpus Workbench

O IMS Corpus WorkBench (Christ et al., 1999) é um dos sistemas mais conhecidos e usados. É também conhecido por CQP (Corpus Query Processor), o nome da linguagem de *query*. Embora não seja de código aberto[19], nem disponível livremente para instalação local, é gratuito para investigação mediante a assinatura de um pequeno contrato.

Foi desenvolvido numa abordagem por camadas (ou *layers*), em que sobre a camada base que contém o texto se colocam novas camadas com informação adicional, como sejam a análise morfológica e sintáctica. Esta abordagem por camadas permitiu que facilmente se estendesse o sistema inicial de processamento de corpora monolingue para suportar texto paralelo: dois corpora monolingues em que a cada um se adiciona uma camada com a informação de alinhamento.

[17] Projecto disponível em `http://xaira.sf.net/`.

[18] O BNC (*British National Corpus*) está em `http://www.natcorp.ox.ac.uk/`.

[19] Existe um projecto de uma versão aberta do CWB, mas que disponibilizou a sua primeira versão livre no final de escrita desta dissertação.

Os principais problemas do IMS Corpus WorkBench prendiam-se com a falta de flexibilidade no que se refere à sua instalação (nomeadamente pela necessidade de se arranjar um binário para a arquitectura em causa), quer no que se refere à possibilidade de novas experiências, já que se tratava de um pacote de software fechado, sem facilidade de evolução por terceiros.

Uma das grandes vantagens do CWB é a sua linguagem de pesquisa que pode ser considerada uma linguagem de programação dado o seu poder expressivo.

Emdros

Os autores descrevem o Emdros (Petersen, 2004) como um motor de base de dados[20] para texto analisado ou anotado. É um sistema de código aberto[21], baseado numa abordagem por camadas bastante versátil e em XML. Embora tecnicamente seja possível utilizar o Emdros como ferramenta para a indexação de corpora paralelos aplicando uma abordagem semelhante à do IMS Corpus WorkBench, o Emdros não tem suporte nativo para este tipo de corpora. A sua grande mais-valia é a abordagem por camadas e o suporte de uma linguagem de *query* versátil.

3.4.2 Codificação de Corpora Paralelos

Além dos gestores de corpora apresentados, existem muitos outros. Optamos por desenvolver o nosso próprio, já que nos interessa um sistema adaptável que permita realizar experiências e compor com novas aplicações. No entanto, é verdade que o sistema não consegue competir contra todos os detalhes suportados pelas outras ferramentas.

O tratamento de cada corpus paralelo começa por ser a sua codificação: representar cada átomo (palavra, número ou símbolo) por um inteiro. Para cada um dos corpus $c_{\mathcal{A}}$ e $c_{\mathcal{B}}$ (língua original e língua de

[20] E na verdade, é implementado sobre um sistema relacional de base de dados.
[21] A página oficial do Emdros está em `http://emdros.org/`.

destino), e de forma independente, são criados:

- um mapeamento bidireccional de palavra para identificador:

$$\text{Lexicon}\,(c) = \left\langle \begin{pmatrix} word \\ \text{id}\,(word) \end{pmatrix}_{word \in c}, \begin{pmatrix} \text{id}\,(word) \\ word \end{pmatrix}_{word \in c} \right\rangle$$

- considerando a função "wordid" que dado o léxico $l = \text{Lexicon}(c)$ e uma palavra retorna o identificador dessa palavra, então o processo de codificação do corpus é definido por:

$$\text{EncodeCorpus}\,(c) = \begin{pmatrix} \text{id}\,(sent) \\ \langle \text{wordid}\,(l, word)\,|word \in sent \rangle \end{pmatrix}_{sent \in c}$$

Estes índices permitem aceder a cada unidade de tradução a partir do seu identificador. Além desta informação básica, são criados índices para cada camada de informação, que especificam palavras ou zonas de segmento usando o identificador da unidade da tradução e o *offset* em causa. São também criados índices de pesquisa por palavra.

Todo o processamento posterior sobre os corpora é realizado com base nos corpora codificados para maior rapidez.

3.4.3 Concordâncias

O cálculo de concordâncias sobre um corpus paralelo codificado não é mais do que a conversão da expressão de pesquisa para os respectivos identificadores numéricos, e a sua pesquisa utilizando os índices construídos. Esta função recebe o identificador do corpus (um inteiro) e a expressão de pesquisa[22]:

$$Conc : \mathbb{N} \times \mathrm{W}^{\star}_{\mathcal{A}} \times \mathrm{W}^{\star}_{\mathcal{B}} \longrightarrow set\,(S_{\mathcal{A}} \times S_{\mathcal{B}})$$

É possível procurar uma expressão $s = word^{\star}$ na língua $\mathcal{A}$ ou na língua $\mathcal{B}$, ou ainda um par de expressões (s_α, s_β), procurando s_α em $\mathcal{A}$ e s_β em $\mathcal{B}$. Estas expressões de pesquisa podem ainda conter um símbolo especial (asterisco) que corresponde a uma posição onde pode ocorrer qualquer palavra. O resultado desta pesquisa é um conjunto de unidades de tradução que satisfazem a expressão de pesquisa.

[22] $\mathrm{W}_{\mathcal{A}}$ corresponde às palavras do corpus $C_{\mathcal{A}}$ e S_A às frases do corpus $C_{\mathcal{A}}$.

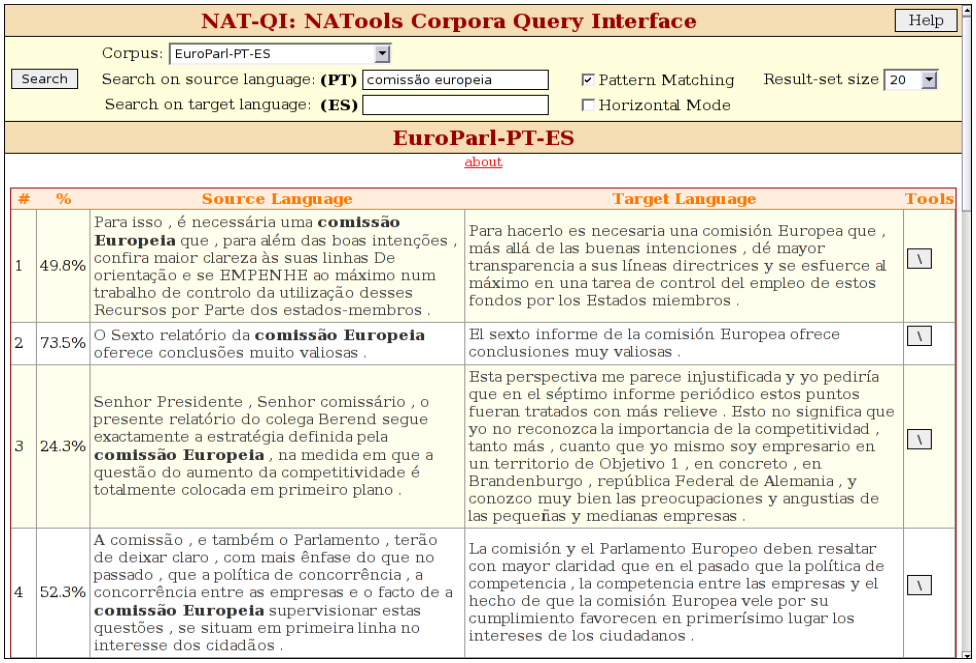


Figura 3.2: NatSearch: consulta de concordâncias em corpora paralelos via Web.

A figura 3.2 mostra a pesquisa de concordâncias usando uma interface Web, bastante útil para utilizadores finais. Esta interface é composta por uma barra onde o utilizador pode colocar as expressões de pesquisa e limitar a quantidade de respostas obtidas. O resultado é apresentado numa tabela com o número do resultado, e as unidades de tradução encontradas, lado a lado. Para os corpora que tenham essa informação calculada, a tabela inclui uma segunda coluna com uma medida de qualidade da unidade de tradução.

A interface Web para cálculo de concordâncias é similar às interfaces habituais para pesquisa de corpora, como sejam o TransSearch (RALI Laboratory, 2006) e o COMPARA (Frankenberg-Garcia and Santos, 2003). A principal diferença corresponde à integração da nossa interface com outras ferramentas, de acordo com a secção 6.1.

Além da interface Web também foi desenvolvida uma API para permitir a consulta eficiente de corpora por outras aplicações. Segue-se um

extracto de código que mostra o uso desta API para interagir com o servidor de corpora.

```
1   use NAT::Client;
2   $server = NAT::Client->new( PeerAddr => 'localhost' );
3   $concs  = $server->conc(join(" ",@ARGV));
4   for my $tu (@$concs) {
5          print "$tu->[0]\n";
6          print "$tu->[1]\n";
7          print "\n"
8   }
```

linha 1: carregar o módulo com a API para interacção com o servidor;
linha 2: criar um novo cliente, indicando-lhe o endereço onde se encontra o servidor;
linha 3: calcular as concordâncias, de acordo com a expressão indicada na linha de comando;
linha 4: iterar as concordâncias e imprimir cada língua da unidade de tradução numa linha;

Durante o resto do documento esta API será usada noutros exemplos. Será também expandida de forma a incluir métodos para a consulta de outros recursos.

3.4.4 Cálculo de n-gramas

Embora seja de cariz monolingue, existe outro tipo de informação estatística bastante usada em linguística de corpora corresponde às frequências de n-gramas de palavras, ou seja, o número de vezes que determinada sequência de n palavras ocorre.

No caso do NATools é calculado o número de vezes que cada par de palavras (w_1, w_2) ocorre (bigramas, $n = 2$), o número de vezes que três palavras (w_1, w_2, w_3) ocorrem (trigramas, $n = 3$) e o número de vezes que quatro palavras (w_1, w_2, w_3, w_4) ocorrem (tetragramas, $n = 4$).

Por exemplo, na frase "*o gato comeu o rato*" correspondem a bigramas *(o, gato)*, *(gato, comeu)*, *(comeu, o)* e assim por diante. Os

Corpus		Bigramas	Trigramas	Tetragramas
Constituição	PT	15 333	25 936	31 514
	EN	14 945	26 749	33 194
	ES	14 677	26 064	32 919
	FR	15 576	27 508	34 183
Compara	PT	544 404	1 243 195	1 590 800
	EN	456 262	1 141 322	1 558 686
L.M.D.	FR	512 694	1 146 103	1 472 700
	PT	479 452	1 104 721	1 491 293
JRC-Acquis	PT	625 033	1 894 326	3 157 634
	EN	544 686	1 681 498	2 847 163
	ES	569 499	1 684 436	2 885 807
	FR	533 226	1 621 974	2 801 385
EuroParl	PT	2 443 512	9 839 617	18 397 532
	EN	1 976 473	8 598 533	16 842 394
	ES	2 324 120	9 153 448	17 607 643
	FR	2 056 042	8 468 080	16 820 695

Tabela 3.4: Contagens de n-gramas.

trigramas são calculados como *(o, gato, comeu)*, *(gato, comeu, o)* e *(comeu, o, rato)*. Por sua vez os tetragramas são *(o, gato, comeu, o)* e *(gato, comeu, o, rato)*.

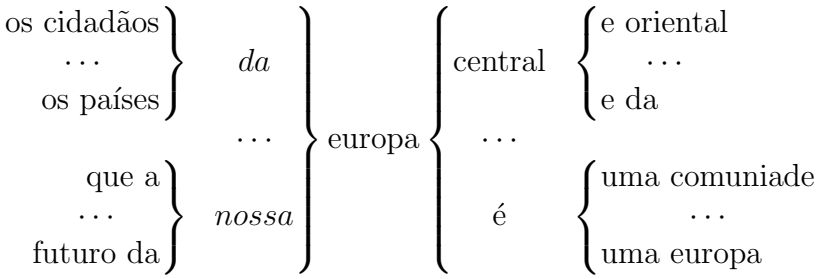
O uso de n-gramas é útil para o estudo de contexto de palavras e construção de classes de palavras (ver secção 5.4.3), bem como para a construção de modelos de língua, bastante usados para a avaliação/classificação de traduções como pertencentes ou não a determinada língua (ver por exemplo a secção 2.3.2).

Um dos principais problemas na geração de n-gramas é o seu armazenamento eficiente, isto porque a quantidade de tuplos diferentes aumenta com o tamanho dos n-gramas. A tabela 3.4[23] apresenta contagens de n-gramas para os corpora apresentados previamente.

A tabela 3.5 permite analisar o contexto esquerdo e direito de uma

[23]Note-se que os números de n-gramas para a língua portuguesa não são necessariamente iguais entre corpora paralelos para línguas diferentes. No entanto os valores são muito semelhantes.

palavra. No caso concreto, foi escolhida a palavra "*Europa*" e o corpus EuroParl. A tabela mostra de forma condensada a contagem de trigramas à esquerda e à direita da palavra, de acordo com o seguinte esquema:



Cada grupo de n-gramas é apresentado juntamente com o número de ocorrências, o que permite um estudo estatístico do contexto das palavras, e dessa forma, a construção de um modelo estatístico de língua.

3.4.5 Memórias de Tradução Distribuídas

A indexação e disponibilização de corpora permite que vários clientes os possam consultar eficientemente em paralelo. Ao pretender-se disponibilizar muitos ou grandes corpora, a possibilidade de distribuir a carga ao nível dos servidores também é relevante, pelo que é importante a problemática de disponibilização de corpora paralelos de forma distribuída.

Também para a área da tradução assistida por computador, a disponibilização e partilha de memórias de tradução do trabalho realizado por vários tradutores é importante.

Uma abordagem para a resolução deste problema passa pela implementação de um sistema de tradução cooperativo baseado na Internet, como descrito em (Bey, Boitet, and Kageura, 2006). No entanto, os tradutores estão demasiado habituados a trabalhar com as suas aplicações tornando-se difícil a sua adaptação a sistemas diferentes. Nesse sentido, o uso de servidores de memórias de tradução distribuídas (Simões, Guinovart, and Almeida, 2004; Simões, Almeida, and Guinovart, 2004) permite colmatar este problema.

o futuro	do conselho	os países	os cidadãos	dos países	12870	da
em toda	de toda	união para	de que	por toda	12463	a
que ,	nós ,	de trabalho	do emprego	o emprego	8504	na
construção de	, de	criação de	sentido de	favor de	2595	uma
em relação	o alargamento	em direcção	caminho rumo	de dar	1119	à
dos independentes	não só	se interessem	caminho percorrido	esforços envidados	457	pela
que ,	isso acreditamos	que acreditam	colegas ,	é inaceitável	327	numa
precisamos de	necessidade de	é preciso	que querem	. ”	192	mais
que a	, na	. a	para a	futuro da	182	nossa
o tipo	a ideia	o conceito	própria ideia	do tipo	67	de

6971	,	mas também	o que	bem como	e não	que é
6853	.	senhor presidente	senhora presidente	no entanto	por isso	penso que
2497	e	os estados	no mundo	a américa	para o	, em
1752	central	e oriental	e de	e do	e da	, oriental
998	que	estamos a	queremos construir	está a	não seja	tenha uma
962	não	pode ser	é apenas	pode continuar	é uma	pode ficar
808	de	leste ,	leste .	hoje ,	amanhã .	hoje .
755	dos	cidadãos .	cidadãos ,	seus cidadãos	cidadãos e	quinze ,
732	é	uma comunidade	capaz de	mais do	o maior	uma europa
631	do	século xxi	futuro .	sudeste .	conhecimento ,	sudeste ,

Tabela 3.5: Análise do contexto direito e esquerdo da palavra “*europa*” usando tetragramas.

Embora o trabalho realizado para a disponibilização de corpora não resolva o problema de partilha de memórias de tradução, ajuda na sua disponibilização eficiente. A integração de uma API de consulta sobre o servidor de corpora num sistema de tradução assistida por computador seria completamente trivial.

A Título de Conclusão

A existência de corpora paralelos é imprescindível para que se possam extrair recursos de tradução: são a matéria prima sem a qual nada se pode fazer. No entanto, nem sempre este corpora existe em quantidade suficiente, ou com a qualidade desejada.

A criação de corpora obriga à conversão de formatos, definindo injectores de vários tipos de documentos para um mesmo formato textual e estruturado, e leva também à necessidade de alinhamento destes textos ao nível da frase.

Estes corpora são depois processados de acordo com as necessidades e fins em vista. Algum deste processamento pode ser feito de forma linear, processando unidades de tradução, uma de cada vez. Este é o exemplo de cálculos parciais, como medidas de qualidade de tradução ou a limpeza de corpora.

Existe outro tipo de processamento que obriga à pesquisa e acesso aleatório aos corpora e que não pode ser realizado directamente sobre as memórias de tradução. Nestes casos, e depois de o corpus ser limpo e ter estabilizado, procede-se à sua indexação: criação de mecanismos eficientes para a pesquisa em corpora paralelos.

A definição de disponibilização eficiente depende dos objectivos em causa. Para um linguista a estudar determinado fenómeno linguístico, a interface Web pode ser suficiente. Um tradutor tirará partido imediato destes corpora se estiverem disponíveis como memórias de tradução convencionais ou distribuídas. Finalmente, um investigador em Processamento de Linguagem Natural quererá uma API para a consulta e processamento de corpora.

Capítulo 4

Dicionários Probabilísticos de Tradução

Learning French is trivial: the word for horse is cheval, and everything else follows in the same way.

Alan J. Perlis

Os dicionários de tradução são recursos cruciais para a tradução, seja ela manual, semi-automática ou completamente automática. Permitem associar (de várias maneiras) palavras entre duas ou mais línguas diferentes.

Embora existam dicionários de tradução livres para vários pares de língua (por exemplo, o FreeDict[1]), a maioria são demasiado pequenos e pouco específicos, pelo que acabam por não cobrir áreas técnicas. Além disso, a compra de dicionários de tradução ou a sua criação são dispendiosas.

Assim, torna-se imprescindível o desenvolvimento de uma ferramenta para a extracção automática de dicionários de tradução a partir de corpora paralelos.

[1] O Projecto FreeDict está disponível em `http://www.freedict.org/en/`.

Durante a dissertação de mestrado (Simões, 2004; Simões and Almeida, 2003) foi estudado um algoritmo para extracção automática de dicionários de tradução, e desenvolvida uma ferramenta para a sua extracção.

Estes dicionários são denominados por Dicionários Probabilísticos de Tradução (PTD — Probabilistic Translation Dictionaries), uma vez que a sua componente estatística é demasiado grande para que possa ser ignorada. O facto de serem extraídos usando métodos estatísticos sobre corpora paralelos e sem o uso de qualquer outro recurso, leva a que determinados resultados possam ser errados. Um nome mais correcto para estes recursos poderia ser o de *tabelas de associação entre palavras de duas línguas*, já que estes PTD mapeiam para cada palavra de uma língua um conjunto de possíveis traduções (ou palavras associadas) e a respectiva confiança dessa tradução (ou associação). A definição formal destes dicionários[2] é apresentada na secção 4.1.

Segue-se a entrada da palavra "*codificada*" de um PTD extraído a partir do corpus EuroParl.

$$\mathcal{T}\left(\text{codificada}\right) = \begin{cases} \text{codified} & 62.83\% \\ \text{uncoded} & 13.16\% \\ \text{coded} & 6.47\% \\ \dots & \end{cases}$$

Este exemplo deve ser entendido como: no corpus EuroParl, a palavra "*codificada*" tem uma grande co-relação com as palavras "*codified*", "*uncoded*", "*coded*" e outras. Esta co-relação tem um grau de certeza de 63% para a primeira tradução, 13% para a segunda, e 6% para a terceira. Como se trata de um dicionário probabilístico de tradução, este exemplo é visto como: a probabilidade da palavra "*codificada*" ser traduzida por "*codified*" é de 63%.

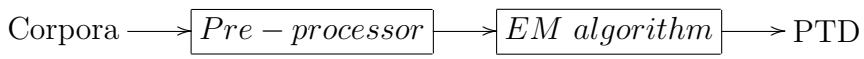
[2]Durante este capítulo falaremos essencialmente sobre PTD pelo que, para facilitar a escrita e leitura, a palavra "*dicionário*" deverá ser entendida como "*dicionário probabilístico de tradução*". Na necessidade de referir um dicionário convencional esse facto será realçado.

Este capítulo descreve uma álgebra heterogénea de dicionários probabilísticos de tradução.

Os dicionários são criados com base em corpora paralelos alinhados ao nível da frase. O construtor dos dicionários pode ser formalizado como:

$$createPTD : \mathrm{TU}^{\star} \longrightarrow \mathrm{PTD}$$

Este processo pode ser dividido em dois módulos, um pré-processador de corpora, e o processo estatístico (*Expectation-Maximization*) que realiza a extracção dos dicionários.



A secção 4.1 discute a construção de dicionários sem qualquer tipo de pré-processamento do corpus paralelo. Também inclui uma secção com uma análise detalhada de situações pouco intuitivas que podem ocorrer num dicionário probabilístico de tradução, e comparação do algoritmo usado com outras abordagens existentes.

Antes de se aplicar algum tipo de pré-processamento ao corpus é importante ter uma noção das características e da qualidade dos dicionários extraídos com o algoritmo base. Só depois de ter esse tipo de informação é que se poderá concluir sobre as vantagens ou inconvenientes de abordagens de pré-processamento. A secção 4.2 apresenta métodos de caracterização, comparação e avaliação dos dicionários probabilísticos de tradução.

A secção 4.3 apresenta novos operadores na álgebra dos PTD, bem como várias abordagens de pré-processamento, com o intuito de observar de que forma se podem obter melhores dicionários probabilísticos.

- A primeira abordagem no melhoramento de PTD é a sua filtragem com base num predicado (Predicate) sobre entradas do PTD:

  $$filter : \mathrm{PTD} \times \mathrm{Predicate} \longrightarrow \mathrm{PTD}$$

  Esta filtragem pode basear-se em várias heurísticas, mas todas

com o mesmo objectivo: encontrar entradas no dicionário que aparentem pouca confiança.

- Segue-se a discussão da adição de dicionários probabilísticos de tradução:

$$+ : \text{PTD} \times \text{PTD} \longrightarrow \text{PTD}$$

  Esta adição é usada internamente para permitir a escalabilidade na extracção de dicionários, mas o que aqui se pretende estudar é se existe vantagem em somar dicionários probabilísticos obtidos de diferentes corpora, e de diferentes áreas.

- Na extracção de recursos precisamos, como já foi discutido, de corpora de tamanho razoável, para que os resultados possam ser considerados válidos. Em algumas situações interessa-nos extrair um PTD de um corpus pequeno, mas nesta situação esperamos um PTD com pouca qualidade. A abordagem proposta para solucionar este problema, consiste em adicionar unidades de tradução fictícias com base num PTD:

$$smallCorpusPTDExtractor : \text{TU}^{\star} \times \text{PTD} \longrightarrow \text{PTD}$$

- Segue-se um conjunto de experiências que se baseiam no pré-processamento de corpora para melhorar entradas nos PTD. O que se pretende é anotar o corpus para ajudar o processo de extracção dos PTD.

$$PreProcessor : \text{TU}^{\star} \times \text{Processor} \longrightarrow \text{TU}^{\star}$$

  A função "Processor" recebe uma unidade de tradução e conhecimento externo e anota a unidade de tradução. Este conhecimento externo pode ser qualquer tipo de informação, como sejam uma lista de nomes, um reconhecedor de entidades mencionadas ou um analisador morfológico. O resultado é um corpus paralelo anotado que é posteriormente processado da forma habitual.
  Os pré-processadores podem ser tão simples como o tratamento das contracções (separando-as) ou mais complicados como a detecção de entidades mencionadas ou lematização dos corpora.

Finalmente, a secção 4.4 apresenta a API desenvolvida para o manuseamento de dicionários probabilísticos de tradução, apresentando exem-

plos para a construção eficiente de recursos genéricos de processamento de linguagem natural.

4.1 Extracção de Dicionários

Um dicionário probabilístico de tradução relaciona palavras de uma língua com um conjunto de possíveis traduções numa outra língua e, a cada uma destes relacionamentos associa uma medida de certeza.

Definição 6 *Um **dicionário probabilístico de tradução** entre duas línguas $\mathcal{A}$ e $\mathcal{B}$ é composto por um par de dicionários:*

$$\mathrm{PTD}_{\mathcal{A},\mathcal{B}} = \mathrm{D}_{\mathcal{A},\mathcal{B}} \times \mathrm{D}_{\mathcal{B},\mathcal{A}}$$

Cada um dos dicionários extraídos tem a seguinte estrutura formal:

$$\begin{aligned} \mathrm{D}_{\mathcal{A},\mathcal{B}} &= \mathrm{W}_{\mathcal{A}} \rightharpoonup (\mathrm{Occs} \times \mathrm{Trads}) \\ \mathrm{Occs} &= \mathbb{N} \\ \mathrm{Trads} &= \mathrm{W}_{\mathcal{B}} \rightharpoonup [0..1] \end{aligned}$$

Cada entrada do dicionário contém o número de ocorrências da palavra no corpus em causa, bem como a lista das suas possíveis traduções. Esta lista tem n traduções, em que $n \in [0, N]$, sendo N um valor configurável em tempo de compilação do NATools (por omissão o número máximo de traduções armazenadas é 8). A figura 4.1 mostra um extracto de um PTD obtido pelo do processamento do EuroParl.

Antes de prosseguir vamos definir uma notação ligada a PTD:

- um dicionário probabilístico de tradução $ptd_{\mathcal{A},\mathcal{B}}$ é um par de dicionários $d_{\mathcal{A},\mathcal{B}}$ e $d_{\mathcal{B},\mathcal{A}}$;
- na necessidade de referir mais do que um dicionário de tradução usaremos um identificador em índice: $ptd_{\mathcal{A},\mathcal{B}_1}$, $d_{\mathcal{A},\mathcal{B}_1}$ e $d_{\mathcal{B},\mathcal{A}_1}$;

```
1       europe => {    ocorr => 42853,
2                      trans => {      europa    => 0.9471,
3                                      europeus  => 0.0339,
4                                      europeu   => 0.0081,
5                                      europeia  => 0.0011,
6                                 },
7                  },
8       stupid => {    ocorr => 180,
9                      trans => {      estúpido  => 0.1755,
10                                     estúpida  => 0.1099,
11                                     estúpidos => 0.0741,
12                                     avisada   => 0.0565,
13                                     direita   => 0.0558,
14                                     impasse   => 0.0448,
15                                  },
16                 },
```

Figura 4.1: Extracto de um Dicionário Probabilístico de Tradução extraído do EuroParl PT:EN.

- sempre que as línguas envolvidas estejam inequivocamente definidas serão removidas: ptd, d, ptd_1 e d_1;
- o conjunto das traduções de determinada palavra $w_{\mathcal{A}}$ utilizando o dicionário $d_{\mathcal{A},\mathcal{B}}$ é representado por $\mathcal{T}_{d_{\mathcal{A},\mathcal{B}}}(w_{\mathcal{A}})$;
- a probabilidade da palavra $w_{\mathcal{A}}$ ser traduzida por $w_{\mathcal{B}}$ no dicionário $d_{\mathcal{A},\mathcal{B}}$ é representada por $\mathcal{P}\left(w_{\mathcal{B}} \in \mathcal{T}_{d_{\mathcal{A},\mathcal{B}}}(w_{\mathcal{A}})\right)$;
- o número de palavras existente no corpus que deu origem ao dicionário será denotado por $\text{size}(d_{\mathcal{A},\mathcal{B}})$;
- $\text{occs}_{d_{\mathcal{A},\mathcal{B}}}(w_{\mathcal{A}})$ corresponde ao número de ocorrências da palavra $w_{\mathcal{A}}$ no corpus da língua $\mathcal{A}$ que deu origem ao dicionário $d_{\mathcal{A},\mathcal{B}}$.

Esta secção descreve informalmente o algoritmo de extracção de dicionários, analisa entradas típicas de PTD e, finalmente, compara a extracção de dicionários probabilísticos de tradução com o alinhamento de corpora paralelos ao nível da palavra (ou do termo).

4.1.1 Algoritmo de Extracção

O algoritmo de extracção de dicionários probabilísticos de tradução é completamente estatístico usando apenas como informação um corpus paralelo alinhado ao nível da frase.

Descrição Informal

O processo de extracção é iniciado com a contagem de co-ocorrência entre palavras, e a sua análise estatística. Intuitivamente é fácil de perceber o algoritmo: se determinada palavra $w_{\mathcal{A}}$ co-ocorre quase sempre com a palavra $w_{\mathcal{B}}$, e bastante menos com outras palavras, então é provável que $w_{\mathcal{A}}$ se traduza por $w_{\mathcal{B}}$.

Consideremos o seguinte exemplo composto por três frases simples:

- *a flor cresce / a casa é grande / a casa azul tem flores*
- *the flower grows / the house is big / the blue house has flowers*

A tabela 4.1 mostra as co-ocorrências: cada célula $M_{i,j}$ contém o número de vezes que cada par de palavras $w_{\mathcal{A}}$ e $w_{\mathcal{B}}$ aparece na mesma unidade de tradução $(s_{\mathcal{A}}, s_{\mathcal{B}})$.

	a	flor	cresce	casa	é	grande	azul	tem	flores
the	3	1	1	2	1	1	1	1	1
flower	1	1	1	0	0	0	0	0	0
grows	1	1	1	0	0	0	0	0	0
house	2	0	0	2	1	1	1	1	1
is	1	0	0	1	1	1	0	0	0
big	1	0	0	1	1	1	0	0	0
blue	1	0	0	1	0	0	1	1	1
has	1	0	0	1	0	0	1	1	1
flowers	1	0	0	1	0	0	1	1	1

Tabela 4.1: Contagem de co-ocorrências.

Esta matriz é processada com um algoritmo estatístico (Expectation-Maximization (Dempster, Laird, and Rubin, 1977)), mas beste exemplo

iremos ignorar esse passo, e passar à interpretação da matriz.

Ao procurar o valor mais elevado na matriz encontramos a relação entre a palavra "*a*" e "*the*". Uma vez que não há qualquer outro valor tão alto, esta relação pode ser dada como correcta, e portanto, remover (ou atenuar) a primeira linha e coluna na matriz. Procurando o valor máximo na nova matriz iremos encontrar um novo relacionamento entre as palavras "*casa*" e "*house*". Mais uma vez esta linha e coluna podem ser removidas. A figura 4.2 mostra a matriz depois de removidas essas linhas e colunas.

	flor	cresce	é	grande	azul	tem	flores
flower	1	1	0	0	0	0	0
grows	1	1	0	0	0	0	0
is	0	0	1	1	0	0	0
big	0	0	1	1	0	0	0
blue	0	0	0	0	1	1	1
has	0	0	0	0	1	1	1
flowers	0	0	0	0	1	1	1

Tabela 4.2: Contagem de co-ocorrências depois de removidas as relações mais fortes.

A partir desta nova matriz não conseguimos tirar mais relacionamentos inequívocos. No entanto, podemos retirar conclusões probabilísticas. Por exemplo, a palavra "*flor*" estará associada a "*flower*" com 50% de certeza, e a "*grows*" com outros 50% de certeza. Do mesmo modo, "*azul*" estará associada a cada uma das palavras "*blue*", "*has*" e "*flowers*" com 33% de certeza. Note-se que as matrizes não são sempre simétricas, pelo que são extraídos dois dicionários probabilísticos de tradução, um da língua de origem para a língua de destino e vice-versa.

Escalabilidade do Algoritmo

O tamanho das matrizes de co-ocorrências (se considerarmos um corpus como o EuroParl, a matriz tem um tamanho de cerca de $130\,000 \times 90\,000$ elementos) levam a que a extracção de dicionários seja um processo de consumo intensivo de memória. Embora estas matrizes sejam esparsas,

não cabem na memória central de uma máquina comum actual.

Para resolver este problema o processo de extracção de PTD foi dividido de modo a processar de forma independente fatias do corpus, ao invés de o tentar processar de uma só vez. Esta abordagem corresponde à defendida na secção 7.2 para a escalabilidade de processos com grandes requisitos de memória.

Em vez de um único dicionário, este processo constrói um conjunto de dicionários (um par por fatia) que têm de ser adicionados. Para a soma de dois dicionários d_1 e d_2 (na verdade $d_{\mathcal{A},\mathcal{B}_1}$ e $d_{\mathcal{A},\mathcal{B}_2}$), são percorridas todas as palavras correspondentes à união dos domínios dos dicionários:

$$w_{\mathcal{A}} \in \mathrm{dom}\,(d_1) \cup \mathrm{dom}\,(d_2)$$

e, para cada entrada, é calculado:

- o número de ocorrências que corresponde à soma das ocorrências dos dois dicionários

$$\mathrm{occs}_{d_1+d_2}(w_{\mathcal{A}}) = \mathrm{occs}_{d_1}(w_{\mathcal{A}}) + \mathrm{occs}_{d_2}(w_{\mathcal{A}})$$

- o conjunto das possíveis traduções, que corresponde à união das traduções dos dois dicionários

$$\mathcal{T}_{d_1+d_2}\,(w_{\mathcal{A}}) = \mathcal{T}_{d_1}\,(w_{\mathcal{A}}) \cup \mathcal{T}_{d_2}\,(w_{\mathcal{A}})$$

- a probabilidade de tradução para cada uma destas possíveis traduções deve ter em conta o tamanho do corpus que lhe deu origem para manter a representatividade dos resultados de acordo com o discutido em (Simões, 2004). Esta probabilidade é calculada com:

$$\frac{\mathcal{P}\,(w_{\mathcal{B}} \in \mathcal{T}_{d_1}\,(w_{\mathcal{A}}))\,\mathrm{occs}_{d_1}(w_{\mathcal{A}})\,\mathrm{size}(d_2) + \mathcal{P}\,(w_{\mathcal{B}} \in \mathcal{T}_{d_2}\,(w_{\mathcal{A}}))\,\mathrm{occs}_{d_2}(w_{\mathcal{A}})\,\mathrm{size}(d_1)}{\mathrm{occs}_{d_1}(w_{\mathcal{A}})\,\mathrm{size}(d_2) + \mathrm{occs}_{d_2}(w_{\mathcal{A}})\,\mathrm{size}(d_1)}$$

A possibilidade de somar dicionários é especialmente importante dada o seu uso na acumulação de dicionários (descrita na secção 4.3.2).

4.1.2 Análise de Casos

Para se melhor perceber as características dos dicionários probabilísticos de tradução, são aqui apresentados alguns exemplos de resultados típicos, e nem sempre intuitivos.

Entradas típicas

As entradas típicas de um PTD apresentam possíveis traduções correctas com medida de confiança elevada, e traduções menos prováveis ou incorrectas com confiança baixa.

```
1          Palavra: europa
2          Ocorrências: 39 917
3          Traduções:
4                 88.50%  europe
5                  5.73%  european
6                  2.37%  europa
7                  1.16%  (none)
8                  0.57%  eu
9                  0.23%  unece
```

Neste exemplo as primeiras três traduções são relacionadas com a palavra em causa, embora o algoritmo tenha atribuído maior probabilidade à primeira. A pseudo-palavra "*(none)*" indica a supressão da tradução. Este fenómeno é explicado com mais detalhe no próximo exemplo. A palavra "*eu*" corresponde à abreviatura de "*European Union*", pelo que também é uma tradução válida.

Entradas com supressão de tradução

Este exemplo (da língua inglesa para a portuguesa) mostra a supressão de palavras na tradução. Em determinadas situações o algoritmo pode determinar que a tradução da palavra foi suprimida. Para representar a supressão de tradução, os PTD sugerem como tradução mais provável a pseudo-palavra *(none)*.

```
1          Palavra:we
2          Ocorrências: 300431
3          Traduções:
4                 17.81%  (none)
5                  8.25%  que
6                  6.02%  temos
```

A maioria deste tipo de relacionamento resulta do facto de na língua portuguesa o pronome pessoal ser muitas vezes omitido (sujeito omisso). Enquanto que em inglês encontramos frases como "*We have to...*", na versão portuguesa iremos encontrar "*Temos de...*" e não "*Nós temos de...*"

Entradas com traduções com variante morfológica

Embora os dois exemplos aqui apresentados sejam de verbos, convém salientar que este fenómeno não acontece apenas para esta categoria morfológica. No entanto, dado que em inglês existem no máximo quatro formas verbais e que em português esse número ultrapassa as setenta formas, os verbos são os exemplos mais evidentes do fenómeno que interessa aqui discutir.

Dado que uma forma verbal em inglês pode ser traduzida por diferentes formas em português, o PTD vai apresentar probabilidades diferentes para cada uma delas. Isto leva a que o número de relações seja bastante elevado, e portanto as probabilidades se encontrem *diluídas*. Além disso, o facto de (por omissão) o extractor armazenar apenas as oito traduções mais prováveis leva a que se percam traduções com probabilidades baixas[3].

[3]Embora este facto seja descrito mais à frente, repare-se que a palavra "*represento*" tem como principal tradução (88% de certeza) a palavra "*represent*". Ou seja, a certeza associada à relação é baixa da língua inglesa para a portuguesa, mas forte no sentido inverso.

```
1   Palavra: read                Palavra: represent
2   Ocorrências: 2435            Ocorrências: 2538
3   Traduções:                   Traduções:
4          29.32%  ler                  17.87%  representam
5          13.75%  li                   11.57%  representar
6   *       8.36%  read                  8.93%  represento
7           5.96%  lido                  7.54%  representamos
8           3.54%  lemos                 4.93%  constituem
9           1.60%  leio                  3.63%  representa
10          1.46%  estar                 3.37%  (none)
11          1.45%  leu                   2.35%  representante
```

Para a palavra "*read*" aparece a própria palavra como possível tradução, que resulta do facto do corpus ter sido normalizado para letras minúsculas e existir uma deputada chamada "*Read*".

Na secção 4.3.8 é apresentada uma abordagem que com base num analisador morfológico junta as formas verbais lematizando-as (ou genericamente as formas de uma qualquer palavra) de modo a que o alinhamento não se disperse por tantas possíveis traduções, aumentando as respectivas probabilidades de tradução.

Entradas com antónimos como traduções

Outro tipo de entradas que faz com que estes dicionários não possam ser vistos como verdadeiros dicionários de tradução, são as entradas em que, para além de uma tradução certa, surgem traduções que correspondem a antónimos da palavra original.

```
1              Palavra: aceitável
2              Ocorrências: 1713
3              Traduções:
4                     71.48%  acceptable
5                      8.56%  unacceptable
```

Esta entrada aparece no dicionário pelo uso frequente de "*não aceitável*" na língua portuguesa em vez da tradução directa de "*unacceptable*"

("*inaceitável*"). Isto leva a que existam muitas co-ocorrências de "*unacceptable*" com "*não aceitável*" e, dado que a palavra "*não*" irá ter uma maior co-ocorrência com a palavra "*not*", o algoritmo irá dar maior peso à relação com a palavra "*aceitável*."

Entradas com traduções de Expressões Idiomáticas

Em algumas situações, a palavra e respectiva tradução mais provável aparentam não ter qualquer tipo de relação.

```
Palavra: palavra
Ocorrências: 6337
Traduções:
       35.75%  floor
       16.88%  word
       13.57%  (none)
        9.28%  speak
```

Estas entradas resultam de expressões idiomáticas (ou idiomáticas em determinado contexto) cuja tradução não é a convencional. No exemplo anterior, retirado de um dicionário do EuroParl (ligado às sessões do Parlamento Europeu), aparecem como traduções prováveis da palavra "*palavra*" as palavras "*floor*" e "*speak*".

Embora à primeira vista sejamos tentados a dizer que o algoritmo não funciona, depois de procurar evidências no corpus chega-se à conclusão de que existe um conjunto de duas ou três expressões idiomáticas muito semelhantes e muito usadas, pelo que existe uma grande ligação entre estas palavras. Não se pode dizer que estas palavras sejam traduções mútuas, mas que pertencem a uma expressão "*tem a palavra*" que se traduz, pelo menos no contexto deste corpus, pela expressão "*has the floor*."[4]

```
Tem a palavra, em nome da comissão, o senhor comissário...
Mr Barnier has the floor on behalf of the Commission.
```

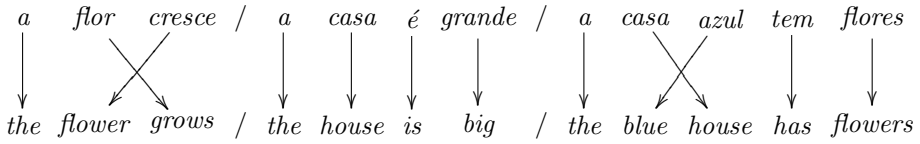
[4]O dicionário Oxford define a expressão "*the floor*" como a parte da casa onde os seus membros se sentam e do qual falam. Em particular, define "*have the floor*" como o direito de ser o próximo orador num debate.

4.1.3 Trabalho Relacionado

Na literatura não se encontram referências à extracção de dicionários probabilísticos de tradução já que, habitualmente, os autores consideram que este processo é o de alinhamento ao nível da palavra. Defendemos que, embora os métodos sejam muito semelhantes, devem ser considerados processos diferentes:

- o *alinhamento à palavra* obtém um relacionamento entre todas as palavras de cada frase. Ou seja, o sistema irá indicar, para cada palavra (instância) do corpus, qual a palavra que é a sua tradução;
- a extracção de *dicionários probabilísticos de tradução* obtém relacionamentos entre palavras de duas línguas para todo o corpus. Pode ser visto como um resumo do alinhamento à palavra.

Em relação ao alinhamento à palavra, a ferramenta mais usada é o GIZA++ (Och and Ney, 2003). Embora o processo de alinhamento do GIZA++ também passe pela construção de uma matriz de co-ocorrências e pelo algoritmo de *expectation-maximization*, o seu resultado final não é exactamente um dicionário probabilístico, mas um alinhamento de todas as ocorrências (tokens) de cada palavra com uma ou mais palavras na língua de destino. Ou seja, no exemplo apresentado anteriormente o GIZA++ teria um alinhamento óptimo representado por:



Os dicionários probabilísticos de tradução obtidos com o NATools seguem a abordagem do *Twente-Aligner* (Hiemstra, August 1996; Hiemstra, 1998) que, embora seja descrito como um alinhador ao nível da palavra, é um extractor de dicionários probabilísticos de tradução.

Os resultados destas duas abordagens são bastante diferentes em forma, mas não é complicada a sua conversão (é bastante simples ob-

ter dicionários probabilísticos a partir do alinhamento à palavra do GIZA++).

Neste trabalho optou-se pelo uso de dicionários probabilísticos de tradução por herança do trabalho realizado na dissertação de mestrado, e pela facilidade na alteração do seu extractor.

4.2 Avaliação e Caracterização de PTD

A avaliação de um dicionário é complicada, e a avaliação de um dicionário probabilístico de tradução não é mais simples.

É possível extrair de forma automática características de dicionários que nos permitam comparar dicionários em termos do seu tamanho e forma (que tipo de traduções compreende, quais as probabilidades de tradução médias, etc.). Embora permitam uma comparação básica, estas características não nos permitem concluir acerca da qualidade dos dicionários. Na secção 4.2.1 é apresentada uma ferramenta que calcula de forma automática um conjunto de métricas e características de um PTD.

Ao avaliar o conteúdo de um dicionário probabilístico, e não apenas a sua forma, deparamo-nos com um dilema, já que é possível realizar avaliações a diferentes níveis:

- avaliar o *dicionário todo*, comparando todas as palavras, todas as traduções e respectivas probabilidades de tradução;
- avaliar uma (ou um conjunto) de *entradas* do dicionário;
- avaliar o *processo de extracção*, e de que forma o algoritmo afecta os resultados obtidos;
- avaliar o *corpus de partida*, e de que forma afecta o algoritmo;
- avaliar ou validar por *utilização dos recursos*.

Nesta dissertação optou-se pela:

- avaliação manual de um conjunto de entradas aleatórias do dicionário, de acordo com a secção 4.2.2;

- avaliação ou validação por utilização e disponibilização de recursos, tornando os dicionários públicos e acessíveis na web (secção 4.4.1), e utilizando-os para a extracção de recursos mais ricos (capítulo 5).

A avaliação é importante mas complicada de ser realizada de forma eficaz. Defendemos que, na impossibilidade de realizar uma avaliação cuidada automaticamente, se definam métricas de comparação. Estas métricas devem permitir analisar a evolução de uma ferramenta (analisar o comportamento de determinado algoritmo) e, se possível, indicar onde se encontram as maiores diferenças (apontar as diferenças mais significativas a um avaliador manual). A secção 4.2.3 apresenta a definição de uma diferença entre entradas com esta finalidade.

Mesmo antes de uma avaliação cuidada podemos constatar que:

- *A qualidade e abrangência do dicionário crescem com o tamanho do corpus.*

  De acordo com a lei de Zipf, quanto maior o corpus, maior o número de palavras cobertas. Dado que análise estatística conta ocorrências de factos; quantas mais vezes determinado facto ocorrer, maior será a probabilidade de esse facto ser significativo.

  Estas conclusões justificam o esforço investido na criação dos novos corpora apresentados no capítulo 3 para além dos já existentes. Justificam também a necessidade de uma ferramenta que permita adicionar (ou acumular) PTD para aumentar a abrangência e qualidade do dicionário resultante (ver secção 4.3.2).

- *A existência de ruído diminui a qualidade dos dicionários.*

  O algoritmo de extracção conta ocorrências de factos sem qualquer tipo de informação sobre se o facto é ou não correcto. Desta forma, a existência de muitas unidades de tradução com ruído, sejam símbolos estranhos ou simplesmente traduções erradas, levara a que factos errados sejam contados e contabilizados para a criação do dicionário, esbatendo a realidade.

  Para minorar este problema seguiram-se duas abordagens (compatíveis): um esforço de aumentar os tamanhos dos corpora, na

esperança que o número de unidades de tradução anómalas e com ruído não cresça à mesma velocidade, e por outro lado, um esforço de analisar unidades de tradução, removendo unidades completas ou ruído localizado (discutido na secção 3.3.4). Foi também aplicada a remoção de unidades de tradução duplicadas.

Neste contexto convém reflectir até que ponto a remoção de unidades de tradução duplicadas é benéfica ou não para a melhoria dos dicionários: enquanto que a repetição de unidades correctas acabariam por melhorar o dicionário no que respeita às palavras constantes nessas unidades, a verdade é que corremos o risco inverso de a unidade repetida ser incorrecta ou usar determinadas palavras num contexto pouco habitual.

- *O comprimento excessivo das unidades de tradução prejudica a qualidade dos dicionários obtidos.*

  Como vimos na secção 4.1, a falta de evidências leva a que entradas para determinada palavra $w_{\mathcal{A}}$ contenham a mesma probabilidade para todas as suas traduções. Logo, numa unidade de tradução grande, cada palavra da língua $\mathcal{A}$ irá co-ocorrer com todas as palavras da língua $\mathcal{B}$, pelo que as evidências de tradução serão muito fracas.

- *A criatividade na tradução prejudica a qualidade dos dicionários.*

  A tradução de texto literário obriga muitas vezes a que o tradutor seja um outro escritor: um romance que seja uma tradução literal acaba por ser uma má tradução. Ao dar liberdade ao tradutor, determinadas frases podem não ser traduzidas da forma mais natural. Dois exemplos típicos onde um tradutor terá de usar toda a sua imaginação é na tradução de humor ou de um ditado popular, onde a tradução literal é desastrosa.

  Embora este tipo de tradução não possa ser considerado errado, é desfavorável para a extracção de dicionários probabilísticos de tradução: leva a que existam evidências no corpus que não são as mais esperadas, e que portanto, os dicionários resultantes acabem por incluir relacionamentos menos óbvios ou mesmo um tanto ou quanto disparatados.

- *O pré-processamento de corpora pode melhorar os dicionários obtidos.*

  Algum tipo de pré-processamento dos corpora pode levar a que os dicionários extraídos tenham mais qualidade, ou tentem realçar diferentes tipos de relacionamentos. A simples lematização (ver secção 4.3.8) das palavras na língua portuguesa levará a que existam menos relacionamentos entre as palavras inglesas e as respectivas traduções, e por isso que as suas probabilidades aumentem.

  Na secção 4.3 são apresentado alguns pré-processadores, e as respectivas melhorias alcançadas.

4.2.1 Caracterização de Dicionários

Embora não sirvam de avaliação, o cálculo de algumas métricas sobre dicionários permite-nos ter uma ideia da sua abrangência e da certeza das suas traduções (embora o facto de um dicionário ter probabilidades mais elevadas não corresponda a maior qualidade).

Neste sentido foram calculadas diferentes medidas sobre os dicionários (d) obtidos (ver tabela 4.3):

- número de entradas do dicionário;
- média das probabilidades de tradução contidas no dicionário, ou seja, média dos valores t_i que correspondem, para cada entrada $w_{\mathcal{A}i}$ com n traduções, ao valor $t_i = \sum_{j=1}^{n} \mathcal{P}\left(w_{\mathcal{B}j} \in \mathcal{T}_d\left(w_{\mathcal{A}i}\right)\right)$
- número de entradas do dicionário em que a tradução com maior probabilidade está acima dos 80%, dos 60% e dos 40%, e respectiva média do número de ocorrências dessas palavras;
- distribuição de entradas por quantidade de traduções: número de entradas do dicionário com uma tradução, com duas traduções, com três traduções, etc., ou mesmo sem traduções.

Estas medidas permitem a constatação de que:

- quanto maior o número de entradas de um dicionário, maior é a sua cobertura — no entanto também é habitual que aumente a

	PT → EN		EN → PT	
número de entradas	24202		18395	
média do total de probabilidades		88.45%		86.56%
nº entradas $\mathcal{P}(1^{a}$ tradução$) \geq 80\%$	6098	25.20%	4992	27.10%
nº médio de ocorrências	342.35		339.90	
nº entradas $\mathcal{P}(1^{a}$ tradução$) \geq 60\%$	10462	43.20%	8199	44.60%
nº médio de ocorrências	278.73		284.47	
nº entradas $\mathcal{P}(1^{a}$ tradução$) \geq 40\%$	15878	65.60%	12563	68.30%
nº médio de ocorrências	312.83		287.16	
nº entradas com 0 traduções	7	0.03%	113	0.61%
nº entradas com 1 tradução	4426	18.29%	4198	22.82%
nº entradas com 2 traduções	4470	18.47%	4056	22.05%
nº entradas com 3 traduções	4014	16.59%	3432	18.66%
nº entradas com 4 traduções	3437	14.20%	2642	14.36%
nº entradas com 5 traduções	2826	11.68%	1802	9.80%
nº entradas com 6 traduções	2153	8.90%	1067	5.80%
nº entradas com 7 traduções	1505	6.22%	565	3.07%
nº entradas com $n \geq 8$ traduções	1364	5.64%	520	2.83%

Tabela 4.3: Medidas dos dicionários obtidos a partir do corpus JRC-Acquis PT:EN.

quantidade de ruído presente no dicionário (não palavras, números);

- o valor da média total de probabilidades de tradução permite concluir sobre a cobertura das traduções — como o algoritmo armazena apenas as n traduções mais frequentes (com $n = 8$ por omissão), é provável que outras traduções possivelmente relevantes não apareçam se este valor for baixo;
- o número médio de ocorrências das palavras que têm uma primeira tradução com probabilidade acima de determinada percentagem, permite ter uma ideia do número de ocorrências necessário para que o algoritmo consiga associar essa mesma probabilidade a determinada palavra;
- o número de entradas com $n > 0$ traduções permite concluir sobre a dispersão de traduções — quantas mais entradas o dicionário incluir apenas com uma tradução, mais certeiro deverá ser.

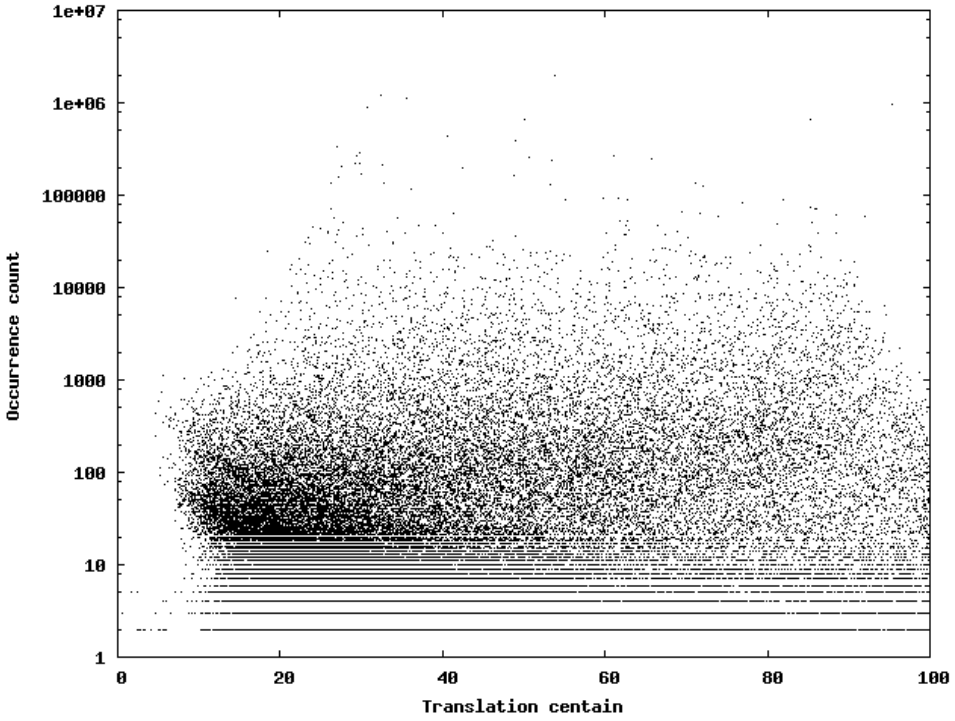


Figura 4.2: Distribuição da melhor tradução de acordo com a sua probabilidade e número de ocorrências.

O gráfico da figura 4.2 mostra a distribuição de entradas do dicionário probabilístico extraído do EuroParl PT:EN de acordo com o seu número de ocorrências e certeza (probabilidade de tradução) da sua melhor tradução. Uma análise à nuvem de pontos permite concluir que a maior parte das palavras do corpus têm menos de 100 ocorrências, e probabilidades de tradução abaixo dos 30%.

4.2.2 Avaliação Manual

Esta secção apresenta três métodos e respectivos resultados, para a avaliação manual de entradas de um dicionário. O maior problema na avaliação é a definição do que é a uma tradução correcta: devem a tradução de "*palavra*" por "*floor*" ser considerada correcta? Decidiu-se avaliar cada uma das traduções de acordo o com contexto geral em que

se usa essa tradução[5].

Foram realizadas avaliações tomando como ponto de partida diferentes objectos de avaliação:

1. obter aleatoriamente 1000 traduções de um dicionário probabilístico de tradução (em que a probabilidade de uma palavra ser escolhida é proporcional ao seu número de ocorrências), com probabilidades de tradução superior a 20%;
2. obter aleatoriamente 1000 traduções como descrito no ponto 1, mas com a restrição de que existam pelo menos 50 ocorrências dessa palavra;
3. obter aleatoriamente 1000 traduções como descrito no ponto 1, mas em que a única restrição é a existência de reflexividade (a tradução da tradução incluir a própria palavra).

Avaliação 1

Para esta avaliação tomou-se como ponto de partida o dicionário português:inglês obtido do EuroParl. Retiraram-se todas as traduções com probabilidades inferiores a 20%, e todos os *tokens* que não são palavras. Criou-se uma lista com todas as traduções. Cada tradução foi repetida na lista de acordo com o seu número de ocorrências no corpus ($1 + \ln(ocur)$ vezes[6]). Esta lista foi ordenada por probabilidades de tradução, e retiradas 1000 traduções de forma aleatória.

A seguinte tabela caracteriza a amostra obtida. É interessante reparar que a amostra contém elementos com probabilidades e número de ocorrência em todo o domínio de valores.

[5]No caso do par "*palavra/floor*", este seria marcado como errado.

[6]A tentativa de multiplicar cada entrada apenas pelo seu número de ocorrências levou a que apenas as entradas com muitas ocorrências fossem realmente avaliadas. O uso do logaritmo permite que as entradas com mais ocorrências tenham um pouco mais de probabilidade de serem avaliadas mas não afoguem por completo as restantes entradas. Ao resultado do logaritmo é somada uma unidade para permitir que as entradas com apenas uma ocorrência apareçam na lista final, e portanto, possam vir a ser avaliadas.

	Probabilidades	Ocorrências
valor mínimo	1.00	1
valor máximo	0.20	1 103 267
média	0.45	1 642
desvio padrão	0.23	35 221

Embora o método para obtenção dos elementos de teste tenha sido pensado para manter alguma aleatoriedade dos elementos, a verdade é que como se pode constatar pela média das probabilidades, a maior parte dos elementos tem probabilidades abaixo dos 50% (de notar que a média deveria ser 0.6).

A tabela 4.4 resume os resultados obtidos nesta avaliação. Embora não se possam definir limites a partir do qual se saiba seguramente se uma palavra é ou não uma boa tradução, estas medidas permitem concluir que o valor de probabilidade do dicionário é um indicador de qualidade de tradução. Por outro lado, é possível concluir que o número de ocorrências de uma palavra no corpus é relevante para a qualidade das suas traduções, já que a média de ocorrências das entradas erradas é de 63 (contra 3699 para as entradas correctas).

	Erradas		Correctas	
nº de entradas	566 (56.6%)		434 (43.4%)	
	Prob	Ocur	Prob	Ocur
valor mínimo	0.20	1	0.20	1
valor máximo	1.00	6 755	1.00	1 103 267
média	0.40	63	0.52	3 699
desvio padrão	0.21	418	0.24	53 376

Tabela 4.4: Resultados da avaliação manual de um PTD (probabilidades superiores a 20%).

Apenas 43% das entradas têm tradução correcta mas, como se verifica nos resultados de avaliação de outros recursos em capítulos seguintes, a possível falta de qualidade dos dicionários não é um factor limitativo nos métodos apresentados.

Esta avaliação é pessimista, já que considera erradas muitas traduções pertencentes a expressões multi-palavra que, embora erradas indivi-

	Total		Erradas		Correctas	
nº de entradas	1000		150 (15%)		850 (85%)	
	Prob	Ocur	Prob	Ocur	Prob	Ocur
valor mínimo	0.20	50	0.20	50	0.20	50
valor máximo	0.99	435 374	0.82	24 022	0.99	435 374
média	0.48	2 476	0.36	1 097	0.50	2 719
desvio padrão	0.21	16 894	0.14	2 720	0.21	18 278

Tabela 4.5: Resultados da avaliação manual de um PTD (probabilidades superiores a 20%, e com mais de 50 ocorrências).

dualmente, são correctas do ponto de vista de correspondência pontual frásica.

Avaliação 2

Para a segunda avaliação tomou-se como ponto de partida uma lista de traduções bastante semelhante à da avaliação anterior, apenas com uma grande diferença: só foram seleccionadas traduções para palavras com mais de 50 ocorrências.

O simples facto de se remover entradas com poucas ocorrências levou a que a média das probabilidades também subisse. É também curioso reparar que o valor máximo de ocorrências da amostra é inferior a metade do valor máximo da amostra anterior, o que é normal devido à lei de Zipf (poucas palavras com muitas ocorrências, muitas palavras com poucas ocorrências).

Em relação à avaliação desta amostra, a percentagem de entradas correctas subiu para 85%, praticamente o dobro do valor anterior. Em relação às probabilidades médias de tradução para as entradas correctas e erradas, pode-se constatar que não mudaram muito em relação ao teste anterior, embora o valor máximo tenha baixado.

	Total		Erradas		Correctas	
nº de entr.	1000		447 (44.7%)		553 (55.3%)	
	Prob	Ocur	Prob	Ocur	Prob	Ocur
v.mínimo	0.00	1	0.00	1	0.00	1
v.máximo	1.00	895 679	1.00	650 477	1.00	895 679
média	0.21	3 522	0.11	2 044	0.29	4 717
desv.padr.	0.24	41 123	0.14	30 762	0.28	47 854

Tabela 4.6: Resultados da avaliação manual de um PTD (entradas com traduções reflexivas).

Avaliação 3

Como terceiro método de avaliação (de notar que existem imensas abordagens possíveis para a avaliação de dicionários probabilísticos de tradução) propomos a avaliação de entradas reflexivas de um dicionário probabilístico, ou seja, entradas referentes a uma palavra $w_{\mathcal{A}}$ tal que

$$w_{\mathcal{A}} \in \mathcal{T}_{d_{\mathcal{B},\mathcal{A}}}\left(\mathcal{T}_{d_{\mathcal{A},\mathcal{B}}}\left(w_{\mathcal{A}}\right)\right).$$

O conjunto de teste de onde foram retiradas 1000 traduções para avaliação manual foi construído com todas as palavras e respectivas traduções em que a reflexividade apresentada anteriormente se verificava. Além disso, também foram duplicadas as entradas de acordo com o número de ocorrências da palavra no corpus (como descrito anteriormente).

Esta avaliação resultou nas medidas apresentadas na tabela 4.6. Como as entradas não foram filtradas, nem em termos de número de ocorrências, nem em termos de probabilidades, é de notar que o conjunto de teste tem probabilidades muito baixas (0.0001%), bem como número de ocorrências mínimo (1 ocorrência). No entanto, o facto de se obrigar à existência de traduções reflexivas leva a que a qualidade do dicionário seja por volta dos 55%.

Os conjuntos de traduções correctas e erradas têm também uma abrangência grande em termos de probabilidades e de ocorrências, pelo que a utilização de um valor-limite mínimo para estes valores levará a

uma melhoria significativa dos resultados.

> A filtragem de dicionários probabilísticos de tradução restringindo-os às entradas com tradução reflexiva, número de ocorrências e de probabilidade de tradução mínimas, a percentagem de traduções correctas sobe para muito próximo dos 95%.

4.2.3 Comparação de Dicionários

Nem sempre é possível definir uma boa métrica de avaliação. No entanto é possível calcular um conjunto de métricas para cada dicionário e, com base nessas métricas, discernir sobre a provável qualidade relativa dos dicionários. Também é possível definir uma medida de distância entre dicionários, que permita evidenciar quais as entradas com maiores diferenças, e que devem ser avaliadas ou comparadas manualmente.

Comparação com base em Métricas

O pacote NATools inclui uma ferramenta (`nat-compareDicts`) para a comparação de dicionários, com base num conjunto de medidas estatísticas:

- o *número de entradas* permite relacionar quantitativamente os dicionários: no entanto deve-se ter em conta que o facto de um dicionário conter mais entradas do que outro não implica que a sua abrangência seja necessariamente maior, já que pode conter ruído (*tokens* que não são palavras);
- o *número médio de traduções por entrada*, que terá como valor máximo 8 (já que é o número máximo de entradas que o NATools calcula por omissão) permite ter uma ideia da dispersão das traduções. Um valor elevado significa que existem muitas traduções ambíguas, enquanto que um valor baixo implica um dicionário mais focado. Note-se que há alguns casos, como as entradas de

	d_1	d_2
nº entradas	137 607	646 106
nº médio de traduções por entrada	5.54	4.37
número mínimo de ocorrências	1	1
número máximo de ocorrências	2 000 857	9 949 231
média de ocorrências	212	280
probabilidade mínima (1ª tradução)	0.04	0.00
probabilidade máxima (1ª tradução)	1.00	1.00
probabilidade média (1ª tradução)	0.50	0.60
entradas com 0 traduções	1	2 907
entradas com 1 tradução	14 584	141 438
entradas com 2 traduções	12 687	90 765
entradas com 3 traduções	12 934	70 104
entradas com 4 traduções	11 560	55 445
entradas com 5 traduções	9 509	42 018
entradas com 6 traduções	7 347	31 786
entradas com 7 traduções	5 750	24 008
entradas com $n \geq 8$ traduções	63 235	187 635
entradas com *(none)* como 1ª tradução	2 044	7 417
entradas com *(none)* como 2ª tradução	2 669	6 861
entradas com *(none)* como 3ª tradução	1 818	6 875
entradas com *(none)* como 4ª tradução	1 214	6 373
entradas com *(none)* como 5ª tradução	1 032	5 866
entradas com *(none)* como 6ª tradução	766	4 934
entradas com *(none)* como 7ª tradução	757	3 989
entradas com *(none)* como 8ª tradução	571	3 669
entradas *iguais* em d_1 e d_2		571 (0.41%)
entradas x tais que $\mathcal{T}_{d_1}(x) = \mathcal{T}_{d_2}(x)$		599 (0.44%)
entradas de d_1 e d_2 com 1ª tradução igual		18 110 (13.16%)
entradas x tais que $\mathcal{T}_{d_2}(x) \subset \mathcal{T}_{d_1}(x)$		1 000
entradas x tais que $\mathcal{T}_{d_1}(x) \subset \mathcal{T}_{d_2}(x)$		1 684
palavras x tais que $x \in d_1$ e $x \notin d_2$		49 057
palavras x tais que $x \in d_2$ e $x \notin d_1$		557 556

Tabela 4.7: Comparação das características dos dicionários do EuroParl (d_1) e EurLex (d_2) para o par PT:EN.

tempos verbais, em que a dispersão não implica uma real ambiguidade.

- como já foi discutido, existe possibilidade de certas traduções serem eventualmente omitidas, o que leva a que no dicionário existam entradas em que *uma das traduções é a pseudo-palavra (none)*. No entanto, a existência de muitas entradas com este tipo de tradução é um mau indicador em relação à qualidade do dicionário.
- especialmente no caso de se estar a comparar dicionários obtidos a partir do mesmo corpus mas com pré-processamentos diferentes, é importante saber:
  - que *entradas são completamente iguais*, ou seja, aquelas cujo conjunto de traduções é o mesmo, e as probabilidades de traduções são semelhantes. Duas entradas são consideradas iguais se contêm a mesma sequência de traduções (traduções pela mesma ordem).
  - que *entradas têm as mesmas traduções*, ou seja, entradas com conjuntos de traduções iguais, mas não necessariamente pela mesma ordem;
  - que *entradas têm a melhor tradução igual*, ou seja, aquelas cujos conjuntos de traduções são ou não iguais, mas cuja melhor tradução (tradução com maior probabilidade) é a mesma.
- o conjunto de entradas cujas *traduções por um dicionário estão contidas nas traduções pelo outro dicionário* permite concluir sobre a possibilidade de um dos dicionários estar contido no outro;
- *o número de palavras que existe apenas num dos dicionários* permite concluir sobre a sobreposição ou não dos dicionários. Ou seja, embora os dicionários possam ter tamanhos semelhantes, nada implica que não tenham uma taxa de sobreposição pequena.

A tabela 4.7 mostra estas medidas na comparação dos dicionários PT:EN obtidos a partir do EuroParl e do EurLex. Note-se que para a comparação de dicionários de tamanhos tão díspares faria sentido apresentar algumas das medidas como valores relativos e não absolutos. No entanto, esta ferramenta foi desenvolvida tendo em vista a comparação

de métodos para a melhoria de dicionários (ver secção 4.3), em que os dicionários têm tamanhos muito semelhantes.

Distância entre Entradas

Além das medidas estatísticas sobre os dicionários é possível calcular medidas de comparação sobre pares de entradas, de forma a que dados dois dicionários se possam mostrar as entradas que mais diferem entre si.

1 **difPTD:** $(\mathrm{D}_1 \times \mathrm{D}_2) \longrightarrow (\mathrm{D}_{\mathcal{A}} \rightharpoonup \mathbb{R})$

2 $entradas \longleftarrow \mathrm{dom}(d_1) \cup \mathrm{dom}(d_2)$
3 **for** $e \in entradas$ **do**
4 $\quad T \longleftarrow \mathcal{T}_{d_1}(e) \cup \mathcal{T}_{d_2}(e)$
5 $\quad diferença \longleftarrow \sum_{t \in T} |\mathcal{P}(t \in \mathcal{T}_{d_1}(e)) - \mathcal{P}(t \in \mathcal{T}_{d_2}(e))|$
6 $\quad medida\_diferença[e] \longleftarrow diferença \times \ln\left(1 + \frac{\mathrm{occs}_{d_1}(e) + \mathrm{occs}_{d_2}(e)}{2}\right)$
7 **return** $medida\_diferença$

Algoritmo 2: Cálculo de uma medida de diferença entre entradas de dois dicionários d_1 e d_2 ($d_{\mathcal{A},\mathcal{B}_1}$ e $d_{\mathcal{A},\mathcal{B}_2}$).

O algoritmo 2 calcula a diferença entre entradas: as distâncias entre as probabilidades das várias possíveis traduções. Para duas entradas iguais, as probabilidades de tradução de cada palavra serão a mesma nos dois dicionários, pelo que a distância será zero. Por outro lado, se duas entradas têm traduções completamente diferentes, teremos um somatório de distâncias máximo de 200 (que corresponde à soma das probabilidades de tradução de ambas as entradas e portanto, no pior dos casos, será 200%). Este valor é posteriormente multiplicado pela média de ocorrências da palavra nos corpora[7]. Desta forma damos mais peso a diferenças em palavras que ocorrem mais vezes, mas ao não multiplicar directamente pelo número de ocorrências leva a que o valor não aumente

[7] A este valor é somada uma unidade para que a medida não se anule no caso de haver apenas uma ocorrência em cada um dos corpora.

linearmente, mas logaritmicamente, e portanto os valores sejam mais comparáveis.

Como exemplo prático consideremos as seguintes entradas de dois dicionários probabilísticos (EuroParl e EurLex, respectivamente):

Palavra: *requisitos*		Palavra: *requisitos*
Ocorrências: *1891*		Ocorrências: *40598*
Traduções:		Traduções:
59.18%	*requirements*	80.63% *requirements*
12.97%	*(none)*	16.49% *conditions*
7.76%	*demands*	1.25% *(none)*
6.07%	*conditions*	0.18% *watercraft*
2.10%	*requirement*	0.15% *requirement*
1.59%	*standards*	0.10% *criteria*
0.95%	*prerequisites*	0.01% *standards*
0.60%	*criteria*	

Para o cálculo das distâncias é necessário calcular o conjunto de traduções dos dois dicionários e calcular as distâncias entre probabilidades:

	EuroParl	*EurLex*	*Distância*
requirements	59.18	80.63	21.45
(none)	12.97	1.25	11.72
demands	7.76		7.76
conditions	6.07	16.49	10.42
requirement	2.10	0.15	1.95
standards	1.59	0.01	1.58
prerequisites	0.95		0.95
criteria	0.60	0.10	0.50
watercraft		0.18	0.18

O somatório das distâncias é 56.51 que, multiplicado pelo logaritmo da média das ocorrências, é 5.63. Segue-se um exemplo com uma diferença mais elevada:

Palavra (w)	Distância	$\mathcal{T}_{d_1}(w)$	$\mathcal{P}(\mathcal{T}_{d_1}(w))$	$\mathcal{T}_{d_2}(w)$	$\mathcal{P}(\mathcal{T}_{d_2}(w))$
senhor	18.96	mr	70.93%	member	67.92%
		(none)	7.57%	honourable	7.10%
reenvio	16.73	back	53.32%	referring	32.14%
		referral	13.44%	national	20.17%
câmara	15.77	house	52.54%	board	93.81%
		chamber	18.23%	chamber	1.96%
prejudicial	15.44	harmful	36.68%	preliminary	75.72%
		damaging	23.16%	ruling	15.78%
obrigado	15.29	thank	84.09%	required	62.88%
		thanks	3.73%	obliged	19.31%
petição	15.29	petition	73.04%	application	96.15%
		(none)	7.58%	has	2.78%
assembleia	15.26	house	62.94%	assembly	69.83%
		assembly	11.80%	meeting	25.96%
recorrente	15.22	recurring	10.06%	applicant	91.94%
		process	7.88%	appellant	5.90%
despacho	15.22	stood	31.79%	order	86.69%
		presence	11.96%	klagenfurt	4.60%

Tabela 4.8: Entradas com grande distância. d_1 corresponde ao EuroParl, e d_2 ao Eurlex (PT:EN).

```
1  Palavra: assembleia          Palavra: assembleia
2  Ocorrências: 11340           Ocorrências: 4451
3  Traduções:                   Traduções:
4       62.94%  house                69.83%  assembly
5       11.80%  assembly             25.96%  meeting
6        8.76%  parliament            1.09%  who
7        7.10%  (none)               0.72%  contributor
8        4.41%  chamber              0.42%  s
9        0.57%  you                  0.37%  house
10       0.20%  I                    0.34%  diekirch
11       0.19%  qualified            0.29%  (none)
```

Realizando o cálculo das distâncias de modo semelhante, obtém-se 170.07 que multiplicado pelo logaritmo da média das ocorrências é de 16.44.

A tabela 4.8 é um extracto do conjunto de palavras com maiores diferenças nas suas entradas do dicionário. Por sua vez, a tabela 4.9 mostra um extracto do conjunto de palavras com menores diferenças.

Palavra (w)	Distância	$\mathcal{T}_{d_1}(w)$	$\mathcal{P}(\mathcal{T}_{d_1}(w))$	$\mathcal{T}_{d_2}(w)$	$\mathcal{P}(\mathcal{T}_{d_2}(w))$
roleta	0.607	roulette	96.02%	roulette	90.75%
		figurines	1.60%	poker	3.35%
burundi	0.58	burundi	94.47%	burundi	95.16%
		enables	2.00%	united	2.59%
monóxido	0.58	monoxide	94.73%	monoxide	93.35%
		poisoning	1.22%	n20	0.67%
empregadores	0.55	employers	89.98%	employers	90.92%
		employer	4.83%	employer	7.06%
singapura	0.54	singapore	95.58%	singapore	98.56%
		ought	1.76%	sgd	1.33%
genebra	0.54	geneva	94.88%	geneva	96.38%
		rejecting	1.27%	genève	1.08%
latina	0.52	latin	95.27%	latin	97.09%
		emphasized	1.36%	eu-latin	0.74%
dopagem	0.52	doping	90.64%	doping	92.15%
		drugs	4.05%	drugs	2.76%
aduaneira	0.48	customs	96.31%	customs	97.88%
		(none)	0.70%	office	0.74%

Tabela 4.9: Entradas com menor distância. d_1 corresponde ao EuroParl, e d_2 ao Eurlex (PT:EN).

A comparação directa de distâncias permite a análise dos resultados quando se altera o algoritmo. A sua ordenação permite que se possam encontrar rapidamente as entradas com maiores diferenças. Por fim, o somatório destas distâncias para todas as entradas do dicionário permite avaliar proximidades entre dicionários (e, por exemplo, calcular o que se encontra mais próximo de um dicionário de referência).

> A comparação de dicionários não permite a sua avaliação automática, mas permite que o avaliador humano possa ser dirigido para as alterações relevantes.

4.3 Melhoria de Dicionários

Durante todo o processo de construção, avaliação e uso de dicionários probabilísticos de tradução, foi-se encontrando problemas localizados.

Esta secção apresenta várias abordagens no intento de melhorar (pelo menos de forma localizada) a qualidade de dicionários probabilísticos de tradução. As primeiras duas tomam como ponto de partida os próprios dicionários, enquanto que as seguintes alteram a forma como os dicionários são calculados:

- *filtragem de dicionários:* uma solução para a melhoria de dicionários passa por remover aquelas entradas com probabilidades baixas ou com um número baixo de ocorrências;
- *acumulação de dicionários:* é possível acumular os dicionários extraídos de vários corpora obtendo dicionários com maior abrangência e maiores certezas de tradução;
- *extracção de dicionários a partir de corpora pequenos:* para a extracção de dicionários técnicos é necessário o uso de corpora específico de determinada área, que nem sempre existe em quantidades suficientes para obter bons resultados. A abordagem apresentada usa um dicionários probabilísticos de tradução externo para o enriquecimento do corpus pequeno, e posterior extracção do dicionário;
- *extracção de dicionários a partir de expressões terminológicas:* dada a existência de métodos para acumulação de dicionários, é possível realizar a extracção de dicionários sobre terminologia bilingue (mono ou multi-palavra) para a extracção de dicionários mais fortes que possam vir a ser adicionados aos dicionários originais;
- *entidades mencionadas:* a detecção e protecção de entidades permite que as suas partes constituintes não sejam consideradas palavras diferentes durante a extracção do dicionário;
- *expansão de contracções:* algumas contracções na língua portuguesa são associadas a duas palavras na língua de destino, como sejam o "*dos*" e "*of the*." Neste sentido, a separação das contracções nas suas partes constituintes pode ajudar neste tipo de relacionamentos;
- *tratamento de locuções:* assim como o referido acerca das entidades mencionadas, as locuções devem ser vistas como objectos que não devem ser divididos. As locuções podem ser anotadas e

protegidas para que sejam consideradas como uma única palavra durante a extracção do dicionário;

- *lematização:* como já foi mostrado num dos exemplos de entradas dos dicionários, a extracção de dicionários entre línguas com níveis de flexão muito diferentes leva a que existam entradas com traduções muito dispersas, pelo que a lematização poderá resolver este problema;
- *tratamento de tempos compostos:* embora a lematização defendida no ponto anterior resolva grande parte da dispersão entre formas verbais, não soluciona todos os problemas, já que os tempos compostos são constituídos por mais do que uma palavra. A detecção e anotação destes tempos compostos pode complementar a lematização para a extracção de dicionários probabilísticos de tradução de verbos;
- *tratamento de termos multi-palavra:* com base em listas de termos multi-palavra podemos anotar o corpus de forma a extrair relacionamentos entre estes termos e não entre as palavras que os constituem;

Para cada uma destas abordagens é apresentada a metodologia, exemplos de resultados e uma reflexão sobre a melhoria obtida. As abordagens descritas não melhoram necessariamente o dicionário como um todo. Muitas delas melhoram determinado tipo de entradas (por exemplo, verbos) e as restantes entradas mantêm ou perdem qualidade.

Estes exemplos poderiam ter sido mais explorados do que o que aqui se apresenta. Estas secções pretendem ser apenas a motivação para o estudo de diferentes abordagens para a extracção de dicionários probabilísticos de tradução.

4.3.1 Filtragem de Dicionários

Como vimos na secção 4.1.2, um PTD não pode ser visto como um dicionário de tradução convencional. No entanto, é possível realizar um conjunto de filtragens com base num conjunto de heurísticas configuráveis, de forma a aproximá-lo de um dicionário de tradução.

	d_1	$\mathcal{F}(d_1)$
nº entradas	137 607	63 402
nº médio de traduções por entrada	5.54	4.27
número mínimo de ocorrências	1	3
número máximo de ocorrências	2 000 857	1 214 672
média de ocorrências	212	404
probabilidade mínima (1ª tradução)	0.04	0.05
probabilidade máxima (1ª tradução)	1.00	1.00
probabilidade média (1ª tradução)	0.50	0.41
entradas com 0 traduções	1	0
entradas com 1 tradução	14 584	4 181
entradas com 2 traduções	12 687	7 883
entradas com 3 traduções	12 934	11 121
entradas com 4 traduções	11 560	12 279
entradas com 5 traduções	9 509	11 121
entradas com 6 traduções	7 347	8 339
entradas com 7 traduções	5 750	4 982
entradas com 8 traduções	63 235	3 496
entradas iguais em d_1 e d_2	6 258 (4.55%)	
entradas x tais que $\mathcal{T}_{d_1}(x) = \mathcal{T}_{d_2}(x)$	6 421 (4.67%)	
entradas de d_1 e d_2 com 1ª tradução igual	62 870 (45.69%)	
entradas x tais que $\mathcal{T}_{d_2}(x) \subset \mathcal{T}_{d_1}(x)$	63 402	
entradas x tais que $\mathcal{T}_{d_1}(x) \subset \mathcal{T}_{d_2}(x)$	6 421	
palavras x tais que $x \in d_1$ e $x \notin d_2$	74 205	
palavras x tais que $x \in d_2$ e $x \notin d_1$	0	

Tabela 4.10: Comparação estatística entre um dicionário d_1 (EuroParl PT:EN) antes e depois de filtrado.

> Os dicionários probabilísticos de tradução são úteis para a construção de forma manual ou automática, de dicionários bilingues convencionais (Guinovart and Fontenla, 2005).

Para a filtragem de dicionários foram usadas as seguintes heurísticas:

- *remoção de números:* embora grande parte das entradas com números sejam correctas, existem algumas que abreviam determina-

das palavras (como "*6*" em vez de "*sexta*" ou "*sexto*") e que portanto não fazem sentido num dicionário de tradução. Por outro lado, as próprias entradas puramente numéricas, embora correctas, não devem fazer parte de um dicionário de tradução;

- *remoção de não-palavras:* em quase todos os corpora existem não-palavras: sequências de caracteres alfanuméricos (*CO2*, *E314*) que fazem sentido no corpus em questão mas que não são úteis para a tradução (até porque na maioria dos casos têm como tradução a própria sequência);
- *remoção de probabilidades baixas:* se definirmos um determinado patamar (que nem sempre é fácil de calcular) nas probabilidades de tradução a partir da qual se considere que as traduções estão correctas, é possível obter entradas que, em princípio, correspondem realmente a entradas de um dicionário de tradução. No entanto a definição de um limiar a partir do qual as entradas passam a ser válidas é complicada e obriga muitas vezes à análise manual do dicionário em causa (ver secção 4.2.2);
- *remoção de entradas com poucas ocorrências:* embora esta heurística remova muitas entradas correctas, por vezes é útil. Permite remover entradas que ocorrem poucas vezes. No entanto não é seguro que as entradas com poucas ocorrências correspondam a más traduções (como se pode ver na figura 4.2, existem entradas com poucas ocorrências e probabilidade de tradução elevada);
- *remoção da tradução "vazia":* nos dicionários aparecem traduções que correspondem à remoção ou adição de palavras, como foi visto na secção 4.1.2. Embora estas entradas tenham a sua utilidade, não são úteis para dicionários de tradução convencionais;
- *remoção de entradas vazias:* algumas entradas dos PTD aparecem sem traduções, como já foi mostrado. Por outro lado, depois de aplicar as heurísticas descritas acima, é de esperar que o número de entradas sem traduções aumente. Como estas entradas não são úteis num dicionário de tradução devem ser removidas.

Outras heurísticas podiam ser implementadas, como por exemplo, remover traduções em que uma palavra em determinada língua tenha um número de ocorrências muito maior (ou menor) do que a respectiva tradução. No entanto, esta abordagem obriga ao processamento paralelo

dos dois dicionários, o que não é estritamente necessário nas heurísticas descritas.

Esta operação foi automatizada com o `nat-PTDfilter` que permite activar ou desactivar cada uma destas heurísticas, bem como indicar valores limite (probabilidade e número de ocorrências mínimos).

Esta ferramenta foi aplicada ao dicionário extraído do EuroParl PT:EN activando os filtros com os seguintes valores exemplo:

- número mínimo de ocorrências: 3;
- probabilidade mínima de tradução: 0.05 (5%);
- remoção de entradas numéricas;
- remoção de entradas não textuais;

A tabela 4.10 mostra algumas medidas comparativas do dicionário antes e depois de filtrado. Note-se que o número máximo de ocorrências é diferente porque foram removidas entradas não textuais, como a pontuação, que têm um número de ocorrências bastante elevado.

As entradas com maiores diferenças entre estes dois dicionários correspondem a numerais que tinham relacionamentos com dígitos. No entanto, devido à filtragem de todas as entradas não textuais estas traduções desaparecem:

Palavra (w)	Distância	$\mathcal{T}_{d_1}(w)$	$\mathcal{P}(\mathcal{T}_{d_1}(w))$	$\mathcal{T}_{d_2}(w)$	$\mathcal{P}(\mathcal{T}_{d_2}(w))$
vinte	3.31	twenty	42.03%	twenty	42.03%
		20	32.69%		
quinze	2.98	fifteen	52.80%	fifteen	52.80%
		15	36.66%	(none)	5.68%
trinta	2.93	30	36.43%	thirty	34.21%
		thirty	34.21%	(none)	6.33%

Continuando a descer na tabela de medidas encontram-se diferenças mais interessantes, nomeadamente de entradas cuja melhor tradução não estava correcta e que passa a estar:

Palavra (w)	Distância	$\mathcal{T}_{d_1}(w)$	$\mathcal{P}(\mathcal{T}_{d_1}(w))$	$\mathcal{T}_{d_2}(w)$	$\mathcal{P}(\mathcal{T}_{d_2}(w))$
necessite	1.99	1938	41.68%	needs	25.65%
		needs	25.65%		
revoltante	1.99	45	24.03%	revolting	9.53%
		revolting	9.53%		
representavam	1.92	19.3	18.33%	accounted	18.13%
		a2	18.22%	represented	7.53%

De acordo com os resultados obtidos pode-se concluir que a filtragem de dicionários permite melhorar a qualidade dos mesmos, obtendo relacionamentos mais ricos. No entanto, as probabilidades dos novos dicionários devem ser recalculadas no novo universo para ser possível uma mais correcta adição com outros dicionários (ver secção 4.3.2).

> A possibilidade de filtrar dicionários probabilísticos de tradução permite a criação de dicionários de tradução bilingues de qualidade.

4.3.2 Acumulação de Dicionários

A existência de uma função para a adição de dicionários permite que se acumulem dicionários provenientes de diferentes fontes.

É certo que cada corpus tem um contexto no qual foi criado, e portanto, uma linguagem muito própria. Também é sabido pela lei de Zipf, que se aumentarmos a quantidade de texto em determinado corpus, novas palavras irão aparecer. Embora isto seja verdade, não implica que as palavras novas que vão aparecendo sejam realmente úteis. Um exemplo simples corresponde a um corpus de texto jornalístico onde (a não ser que se incluam secções de opinião) é muito pouco usada a primeira pessoa, pelo que ao adicionar mais texto do mesmo género irá aumentar a cobertura do dicionário obtido, mas não irá contemplar verbos na primeira pessoa.

Por outro lado, normalmente não há interesse em juntar corpora de diferentes tipos (ou há interesse em não o fazer). Surge a necessidade de arranjar um método para a junção dos PTD obtidos de corpora diferentes para que se consiga aumentar a cobertura de forma mais abrangente.

A fórmula apresentada na secção 4.1 para o cálculo de probabilidades de tradução na soma de dois dicionários garante que a representatividade das palavras nos corpus de onde os dicionários foram extraídos é preservada. Assim, uma palavra que ocorre muitas vezes num corpus pequeno terá as suas traduções preservadas ao contrário de uma palavra que ocorre muitas poucas vezes num corpus muito grande.

	d_1	d_2	$d_1 + d_2$
Tamanho do dicionário	137 607	646 106	695 163
Nº Traduções por entrada	5.54	4.37	4.46
número mínimo de ocorrências	1	1	1
número máximo de ocorrências	2 000 857	9 949 231	11 611 733
média de ocorrências	212	280	302
probabilidade mínima (1ª tradução)	0.04	0.00	0.00
probabilidade máxima (1ª tradução)	1.00	1.00	1.00
probabilidade média (1ª tradução)	0.50	0.60	0.58
entradas com 0 traduções	1	2 907	2 899
entradas com 1 tradução	14 584	141 438	146 308
entradas com 2 traduções	12 687	90 765	95 454
entradas com 3 traduções	12 934	70 104	74 955
entradas com 4 traduções	11 560	55 445	59 654
entradas com 5 traduções	9 509	42 018	45 207
entradas com 6 traduções	7 347	31 786	34 372
entradas com 7 traduções	5 750	24 008	25 926
entradas com 8 traduções	63 235	187 634	210 388

Tabela 4.11: Comparação dos dicionários português:inglês dos corpora EuroParl, EurLex e do resultado da sua soma.

Sendo trivial de se verificar que a cobertura do dicionário aumenta com a sua soma (a não ser que se somem corpus exactamente com as mesmas palavras), é necessário verificar se a qualidade do dicionário também aumenta. Uma vez que se pressupõe que a existência de corpora grandes permite extrair dicionários melhores, e esta extracção se baseia na soma de dicionários extraídos em fatias (portanto, de vários corpora pequenos), então o mesmo se deverá poder concluir em relação à soma de dois dicionários obtidos por processamento de corpora diferentes.

A tabela 4.11 sumariza a comparação dos dicionários português:inglês dos corpora EuroParl, EurLex e do resultado da sua soma.

Algumas das medidas apresentadas são esperadas: correspondem

	d_1	d_2
entradas iguais em d_i e $d_1 + d_2$	58 980	520 526
entradas x tais que $\mathcal{T}_{d_i}(x) = \mathcal{T}_{d_1+d_2}(x)$	61 595	561 941
ent. de d_i e $d_1 + d_2$ com 1ª tradução igual	112 173	588 782
entradas x tais que $\mathcal{T}_{d_1+d_2}(x) \subset \mathcal{T}_{d_i}(x)$	61 267	561 945
entradas x tais que $\mathcal{T}_{d_i}(x) \subset \mathcal{T}_{d_1+d_2}(x)$	86 473	572 855
palavras x tais que $x \in d_i$ e $x \notin d_1 + d_2$	0	0
palavras x tais que $x \in d_1 + d_2$ e $x \notin d_i$	557 556	49 057

Tabela 4.12: Caracterização dos dicionários português:inglês dos corpora EuroParl, EurLex em relação ao resultado da sua soma.

à soma de ocorrências e ao facto de existirem mais palavras na soma do que em cada um dos dicionários (o que acaba por demonstrar a lei de Zipf: embora o corpus correspondente a d_2 seja quase seis vezes maior do que o de d_1, existem cerca de 49 mil novas palavras). O valor médio de ocorrências também aumenta como esperado, já que embora existam algumas palavras novas a sua grande maioria são comuns aos dois dicionários.

Olhando para o número de entradas sem traduções é interessante verificar que baixou (embora uma quantidade insignificante).

> A soma de dicionários probabilísticos de tradução permite aumentar a cobertura do dicionário, bem como salientar as traduções frequentes.

4.3.3 Extracção de Dicionários a partir de Corpora pequenos

Em determinadas situações pretende-se realizar a extracção de um dicionário probabilístico de tradução a partir de um corpus pequeno. Por exemplo, se dispomos de um pequeno corpus de uma área específica como a medicina, e o queremos processar para obter um dicionário bilingue de termos médicos.

Ao processar este corpus o algoritmo poderá não ter informação suficiente para desambiguar todas as relações possíveis. Nestes casos, é

habitual encontrar unidades de tradução com várias possíveis traduções, todas com a mesma probabilidade:

$$\mathcal{T}\,(\text{sódio}) = \begin{cases} \text{sodium} & 25\% \\ \text{chloride} & 25\% \\ \text{salt} & 25\% \\ \text{pure} & 25\% \end{cases}$$

Para resolver este problema propomos o uso de um dicionário probabilístico de tradução extraído de outro (ou outros) corpus, de tamanho razoável, para expandir o corpus pequeno e melhorar a qualidade do dicionário extraído.

O processo de expansão é realizado de acordo com:

- cada unidade de tradução $tu = (s_{\mathcal{A}}, s_{\mathcal{B}})$ é analisada, e obtidas as suas palavras;
- para cada palavra $w_{\mathcal{A}} \in s_{\mathcal{A}}$ é calculado o seu conjunto de traduções $\mathcal{T}(w_{\mathcal{A}})$ usando o dicionário probabilístico externo, e verificado se existe $w_{\mathcal{B}}$ tal que $w_{\mathcal{B}} \in s_{\mathcal{B}} \wedge w_{\mathcal{B}} \in \mathcal{T}(w_{\mathcal{A}})$. Se esta condição se verificar, é criada uma unidade de tradução artificial constituída por $(w_{\mathcal{A}}, w_{\mathcal{B}})$.
- segue-se o mesmo processo da língua $\mathcal{B}$ para a língua $\mathcal{A}$.

Consideremos o seguinte exemplo de uma unidade de tradução:

a eucaristia é ao domingo . / the eucharist is on sunday .

Depois de processada, obtém-se uma entrada do dicionário probabilístico de tradução com:

$$\mathcal{T}\,(\text{eucaristia}) = \begin{cases} \text{sunday} & 20\% \\ \text{is} & 20\% \\ \text{eucharist} & 20\% \\ \text{the} & 20\% \\ \text{on} & 20\% \end{cases}$$

Depois de aplicar o processo de expansão ao corpus com um dicionário obtido do EuroParl (em que a palavra "*eucaristia*" não existe), a

tradução é a esperada:

$$\mathcal{T}\,(\text{eucaristia}) = \{\text{eucharist} \quad 100\%$$

Esta abordagem é bastante útil para a extracção de terminologia específica a partir de corpora pequenos. No entanto, os resultados não serão bons se o corpus contiver muitas palavras desconhecidas nos dicionários externos usados.

> A expansão de um corpus pequeno, adicionando unidades de tradução básicas, é um método eficiente para melhorar a qualidade dos dicionários probabilísticos extraídos, especialmente no que respeita a terminologia específica.

4.3.4 Extracção de Dicionários a partir de Expressões Terminológicas

Em determinados recursos, como ontologias multilingues (como por exemplo o projecto MegaThesaurus (Almeida and Simões, 2006; Almeida and Simões, 2006)) ou bases terminológicas, existem entradas paralelas de pequeno comprimento. Embora uma parte seja constituída por unidades de uma palavra, as ontologias técnicas são constituídas essencialmente por termos multi-palavra. Deste modo, constituem um corpus paralelo de terminologia bilingue que pode ser alinhado para a extracção de PTD. A vantagem no uso de terminologia em relação a corpora paralelos clássicos é que as unidades terminológicas são bastante mais pequenas (uma média de 3 palavras) do que as unidades de tradução típicas de um corpus paralelo.

Os dicionários probabilísticos obtidos são bons para serem somados a outros dicionários obtidos de corpora clássicos, para a extracção de dicionários temáticos e técnicos ou mesmo para a extracção de subterminologia.

4.3.5 Reconhecimento de Entidades Mencionadas

A extracção de dicionários a partir de texto com entidades é problemática especialmente no caso das entidades que são traduzidas entre línguas, e das que são compostas por mais do que uma palavra. Existe muito trabalho na área de reconhecimento de entidades (Mota, Santos, and Ranchhod, 2007; Cardoso, 2006) que pode ser aproveitado para pré-processar o corpus. As entidades são protegidas e enviadas ao extractor de dicionários como se fossem apenas uma palavra.

Para realizar experiências em relação ao reconhecimento de entidades mencionadas foi usado o módulo Perl `Lingua::PT::ProperNames`[8] que permite de forma eficaz encontrar nomes próprios em corpora. Embora o módulo tenha sido construído a pensar em entidades portuguesas, tem um comportamento razoável para outras línguas. Em todo o caso o propósito deste documento não é a discussão relativa à qualidade de ferramentas de reconhecimento de entidades mencionadas.

A abordagem para reconhecimento de entidades e posterior alinhamento pode dividir-se nas seguintes tarefas:

1. detecção de entidades em cada um dos corpora que constituem o corpus paralelo a alinhar;
2. marcação das entidades de forma a que o atomizador não divida a entidade em mais do que um átomo;
3. extracção do dicionário probabilístico a partir do novo corpus paralelo.

Os primeiros dois passos podem ser feitos de forma elegante com a função `forPN` do módulo `Lingua::PT::ProperNames`. Esta função detecta entidades e, sempre que encontra uma, invoca uma função recebida como parâmetro para a processar. Esta função de ordem superior pode marcar imediatamente as entidades encontradas.

A figura 4.3 compara duas entradas (que fazem parte de uma enti-

[8] Informação sobre este módulo, incluindo documentação e possibilidade de *download* pode ser encontrada em `http://search.cpan.org/~ambs/Lingua-PT-ProperNames/`.

```
Palavra: comunidades                Palavra: comunidades
Ocorrências: 2044                   Ocorrências: 1373
Traduções:                          Traduções:
     71.68%  communities                 80.11%  communities
      8.96%  (none)                       7.76%  (none)
      3.46%  community                    3.55%  community

Palavra: europeias                  Palavra: europeias
Ocorrências: 7009                   Ocorrências: 6259
Traduções:                          Traduções:
     86.73%  european                    85.25%  european
      4.73%  (none)                       5.49%  europe
      4.68%  europe                       4.68%  (none)
```

Figura 4.3: Comparação de duas entradas entre um dicionário obtido pelo método tradicional (esquerda) e de um dicionário obtido após detecção de entidades mencionadas (direita).

```
Palavra: Comunidades Europeias
Ocorrências: 188
Traduções:
     60.46%  European Communities
     10.45%  accession
      3.41%  European Community
      3.35%  Community Law

Palavra: Comissão das Relações Económicas Externas
Ocorrências: 298
Traduções:
     79.09%  External Economic Relations
      6.24%  (none)
      2.51%  transparency
      1.77%  committee
```

Figura 4.4: Duas entradas correspondentes a entidades mencionadas obtidas após detecção de entidades mencionadas.

dade) extraídas de um corpus sem qualquer tipo de anotação, e de um corpus com entidades mencionadas anotadas. A parte importante nesta comparação é verificar que as probabilidades de tradução são idênticas embora o número de ocorrências tenha diminuído. A figura 4.4, por sua vez, mostra que a tradução da entidade mencionada que as contém também foi bem detectada e a tradução bem calculada.

Por sua vez, o segundo exemplo da figura 4.4 mostra que o algoritmo de detecção de entidades mencionadas nem sempre funciona como devia (já que em inglês o termo *External Economic Relations committee* não tem uma letra maiúscula na última palavra). No entanto, o algoritmo conseguiu associar a palavra "*committee*" à entidade. Este problema poderia ser minorado com o recurso a um reconhecedor de entidades específico para a língua em causa.

Ainda em relação à extracção de dicionários bilingues sobre entidades mencionadas, é possível extrair uma lista de entidades a partir de um corpus e realizar um alinhamento sobre esta lista com base no seu número de co-ocorrências.

Entidade em português	Entidade em inglês	#
Comissão	Commission	5363
Presidente	President	2445
União Europeia	European Union	2143
Conselho	Council	2077
Parlamento	Parliament	2041
Europa	Europe	1883
Estados-Membros	Member States	1528
Parlamento Europeu	European Parliament	986
Estado-Membro	Member State	250
Comissão Europeia	European Commission	210
Conferência Intergovernamental	Intergovernmental Conference	206
Estados Unidos	United States	202
Senhor Presidente	Mr President	179
Fundos Estruturais	Structural Funds	145
Livro Branco	White Paper	144
Carta dos Direitos Fundamentais	Fundamental Rights	98
Cimeira de Lisboa	Lisbon Summit	71

Tabela 4.13: Extracto do alinhamento entre Entidades.

Os resultados desta abordagem (ver tabela 4.13) têm o mesmo problema da abordagem anterior, de depender de um reconhecedor de entidades mencionadas que tem problemas com a inexistência de letras maiúsculas.

> A detecção de entidades mencionadas em texto paralelo permite que se possam extrair dicionários onomásticos ou semi-terminológicos.

4.3.6 Expansão de Contracções

Enquanto que na língua portuguesa as preposições seguidas de artigos podem ser contraídas (e.g. "*dos*" em vez de "*de os*"), no caso da língua inglesa este fenómeno não acontece (mantendo-se "*of the*"). A expansão de contracção antes da extracção de dicionários tem como principal objectivo melhorar a qualidade das relações entre estas palavras e, indirectamente, entre as restantes. Para realizar esta tarefa foi construída uma correspondência entre contracções e a sua forma expandida com base numa lista[9]. A expansão foi aplicada a todas as palavras incluindo os clíticos pertencentes à lista de contracções.

Sem a expansão de contracções a palavra correspondente à contracção vai ter uma correlação com o par (ou triplo) de palavras que lhe correspondem. Por exemplo, procurando as entradas das palavras "*dos*" e "*deste*," encontramos:

```
1      Palavra: dos                Palavra: deste
2      Ocorrências: 209 942        Ocorrências: 21 383
3      Traduções:                  Traduções:
4           29%  of                     67%  this
5           28%  the                     7%  of
6           ...                         ...
```

[9] A lista usada inclui: *à ao àquele àquilo às comigo connosco consigo contigo convosco daí dalém dalgo dalguém dalgum dalgures dali daquele daquém daqui daquilo dele dentre desse deste disso disto do donde doutrem doutro doutrora dum essoutro estoutro há-de hão-de lho mo nalgum naquele naqueloutro naquilo nele nesse neste nisso nisto no noutro num pelo*, e respectivos femininos e plurais.

```
Palavra: de                          Palavra: of
Ocorrências: 1 214 672               Ocorrências: 930 638
Traduções:                           Traduções:
     32%  (none)                          33%  de
     20%  of                              17%  (none)
      7%  to                              12%  da
      7%  the                              9%  do
     ...                                  ...
Palavra: os                          Palavra: the
Ocorrências: 284 087                 Ocorrências: 1 991 837
Traduções:                           Traduções:
     27%  the                             20%  a
     21%  (none)                          16%  o
      7%  to                               9%  da
     ...                                  ...
Palavra: este                        Palavra: this
Ocorrências: 66 117                  Ocorrências: 282 115
Traduções:                           Traduções:
     68%  this                            14%  este
      9%  (none)                          14%  esta
      3%  that                             5%  deste
     ...                                  ...
```

Depois da expansão das contracções as duas primeiras palavras deixam de existir no dicionário, e é esperado que as traduções e respectivas probabilidades das palavras "*de*", "*os*" e "*este*" sejam mais elevadas:

```
Palavra: de                          Palavra: of
Ocorrências: 2 481 472               Ocorrências: 930 513
Traduções:                           Traduções:
     40%  (none)                          81%  de
     25%  of                               6%  (none)
      8%  the                              4%  a
      3%  to                               3%  o
     ...                                 ...
```

```
1   Palavra: os                      Palavra: the
2   Ocorrências: 656 521             Ocorrências: 1 991 897
3   Traduções:                       Traduções:
4        29%  (none)                      35%  a
5        20%  the                         28%  o
6         6%  to                          12%  de
7        ...                              ...
8   Palavra: este                    Palavra: this
9   Ocorrências: 123 391             Ocorrências: 282 136
10  Traduções:                       Traduções:
11       63%  this                        25%  este
12       10%  (none)                      21%  esta
13        3%  that                         7%  (none)
14       ...                              ...
```

Embora os resultados não tenham sido muito interessantes do ponto de vista da tradução da língua portuguesa para a inglesa, já o inverso mostra uma melhoria significativa. Não só as contracções desapareceram das possíveis traduções, como as traduções correctas tiveram um aumento na sua probabilidade de tradução.

4.3.7 Tratamento de Locuções

Designaremos por classes fechadas de palavras[10] aquelas cuja enumeração dos seus elementos é finita, como sejam pronomes, artigos ou preposições. Por sua vez, verbos, nomes, adjectivos e alguns advérbios são consideradas classes abertas de palavras.

O que se pretende neste exercício é tratar as sequências de palavras de classes fechadas (em ambas as línguas) como uma única entidade. Esta abordagem faz sentido especialmente porque na tradução o número de palavras de classes abertas é habitualmente mantido, enquanto que o número de palavras de classes fechadas varia (até devido à própria estrutura da língua).

Para a realização desta experiência foi usado o analisador morfológico jSpell (Simões and Almeida, 2001; Almeida and Pinto, 1994) com

[10] Ver também o conceito de palavra-marca, na secção 5.1.

os respectivos dicionários para a língua portuguesa e inglesa. Foram consideradas classes fechadas de palavras as seguintes categorias gramaticais: *pronomes (possessivos, interrogativos, demonstrativos, pessoais, relativos e indefinidos), artigos, preposições, conjunções, advérbios de negação, tempo, quantidade e contracções preposicionais.*

Esta abordagem pretende por um lado melhorar (ou pelo menos manter) a qualidade de tradução entre palavras pertencentes a classes abertas, e por outro lado, extrair relacionamentos entre sequências de palavras pertencentes a classes fechadas que sejam úteis em tradução automática.

O primeiro passo na comparação dos resultados compreende a verificação de que a junção de palavras de classes fechadas não piora o resultado para as restantes palavras.

```
1   Palavra: sabiam                 Palavra: sabiam
2   Ocorrências: 99                 Ocorrências: 99
3   Traduções:                      Traduções:
4        31%  knew                       52%  knew
5         8%  did                         8%  were
6         6%  were                        6%  freed
7         3%  initiated                   3%  because_a

8   Palavra: parlamento             Palavra: parlamento
9   Ocorrências: 71 071             Ocorrências: 71 071
10  Traduções:                      Traduções:
11       86%  parliament                 85%  parliament
12        7%  (none)                      7%  (none)
13        4%  house                       4%  house
```

Torna-se também importante a análise dos termos correspondentes a palavras de classes fechadas no sentido de analisar a sua usabilidade na tradução automática. A tabela 4.14 mostra uma lista de alguns destes termos juntamente com as suas duas melhores traduções. Embora uma avaliação cuidada de correcção obrigue à análise do contexto destes termos e respectivas traduções, é possível verificar de forma superficial que os resultados de tradução são interessantes.

Termo t	Ocor.	1ª Tradução		2ª Tradução	
que a	52475	that the	33%	that	12%
de uma	44097	*(none)*	19%	a	18%
que o	42107	that the	33%	that	11%
de um	39204	*(none)*	21%	a	19%
para o	32984	*(none)*	22%	for the	19%
com a	32205	with the	26%	*(none)*	20%
sobre a	26397	on the	33%	on	28%
e de	26285	and	74%	*(none)*	9%
e o	26021	and the	40%	and	38%
o que	24561	*(none)*	24%	what	17%
e da	20578	and	55%	*(none)*	17%
que os	19584	that the	23%	that	21%
sobre o	19239	on the	32%	on	22%
a sua	19104	its	28%	their	18%
e que	17569	*(none)*	19%	and which	12%
de que	15036	*(none)*	25%	that	10%
e os	14923	and	43%	and the	35%
para os	14123	for	31%	*(none)*	18%
de que a	4971	that the	44%	that	21%
de todos os	3811	of all	29%	of all the	18%
entre o	3753	between the	50%	between	33%
e uma	3719	and a	34%	and	32%
de que o	3659	that the	54%	that	13%
em todo o	2985	in any	22%	*(none)*	21%
sem um	336	without a	26%	without	22%
de um dos	334	one of the	31%	of one of the	12%

Tabela 4.14: Exemplo de algumas das melhores traduções resultantes da extracção de dicionários probabilísticos a partir de corpora pré-processado aglutinando palavras pertencentes a classes fechadas.

4.3.8 Lematização

O facto de duas línguas terem níveis de flexão muito diferentes (como o inglês e o português, em que este último tem a flexão bastante mais rica) leva a que algumas entradas nos dicionários (especialmente entradas referentes a verbos) tenham muitas traduções potencialmente correctas, com probabilidade bastante baixa.

Uma primeira experiência para a resolução deste problema foi a tentativa de lematizar verbos, inicialmente na língua portuguesa e posteriormente também para a língua inglesa.

Para a lematização foi utilizado o analisador morfológico jSpell. Sendo certo que existe ambiguidade no processo de lematização, para esta experiência ignorou-se este problema, não realizando a lematização nas palavras que podem ter mais do que um lema. Desta forma, um texto como,

> Senhora Presidente, *gostaria* de saber se esta semana o Parlamento *terá* oportunidade de manifestar a sua inequívoca posição de descontentamento face à decisão, hoje *tomada*, de não renovar o embargo de armas *destinadas* à Indonésia, tendo em atenção que a grande maioria da assembleia *apoiou* o *referido* embargo quando este foi *decretado*.

seria transformado para[11]:

> Senhora Presidente, *gostar* de saber se esta semana o Parlamento *ter* oportunidade de manifestar a sua inequívoca posição de descontentamento face à decisão, hoje *tomar*, de não renovar o embargo de armas *destinar* à Indonésia, tendo em atenção que a grande maioria da assembleia *apoiar* o *referir* embargo quando este foi *decretar*.

A figura 4.5 mostra de forma gráfica as probabilidades fictícias para a tradução de algumas formas do verbo "*to define*/*definir*," em que as pro-

[11] Que seria a versão esperada se a frase tivesse sido proferida pelo Deputado Tarzan!

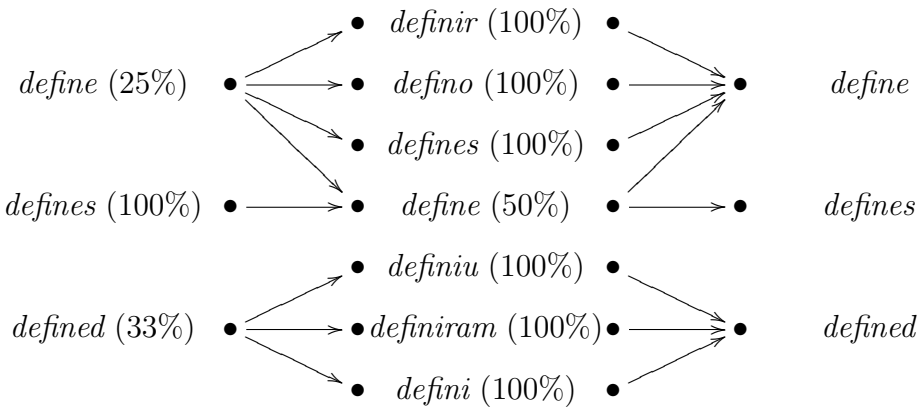


Figura 4.5: Probabilidades fictícias de tradução entre algumas formas verbais do verbo "*to define/definir*" entre a língua portuguesa e inglesa.

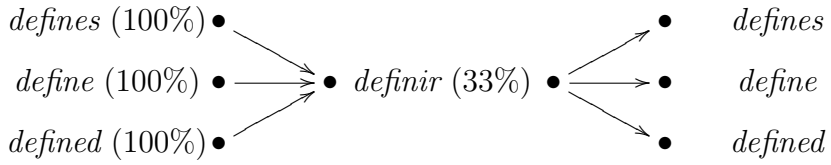


Figura 4.6: Probabilidades fictícias de tradução entre formas verbais do verbo "*to define/definir*" entre a língua portuguesa e inglesa após lematização do lado português.

babilidades de tradução de português para inglês são bastante superiores às das traduções de inglês para português.

Realizando a lematização na língua portuguesa obtemos um relacionamento semelhante ao mostrado na figura 4.6: um aumento das probabilidades da língua inglesa para a portuguesa, mas uma diminuição das probabilidades inversas.

Uma solução simples seria a lematização nas duas línguas, o que resultaria em probabilidades óptimas de 100% em qualquer direcção entre as duas línguas. No entanto, esta abordagem iria destruir bastante

informação que o corpus nos dá.

Para melhorar o dicionário sem perder informação sobre o tempo verbal optou-se por fazer uma lematização selectiva para lemas artificiais: lemas que representem determinado tempo verbal correspondente a um existente na língua inglesa (por exemplo, a concatenação do lema com um POS parcial).

O seguinte extracto mostra as probabilidades de traduções para o verbo procurar (*to find*) num corpus sem lematização:

```
1       Palavra: find
2       Ocorrências: 3 242
3       Traduções:
4            63%  encontrar
5            13%  procurar
6             4%  (null)
7             2%  de
8             1%  procura
9             1%  encontrei
```

Compare-se com o extracto em que se usou um corpus lematizado:

```
1       Palavra: find
2       Ocorrências: 4 785
3       Traduções:
4            79%  encontrar
5             9%  procurar
6             1%  procura
7             1%  descobrir
8             1%  considerar
```

Ao lematizar a língua portuguesa e inglesa o número de ocorrências de verbos aumenta. Do mesmo modo, a probabilidade de tradução do lema irá aumentar. Embora a probabilidade das restantes traduções tenha baixado, essas traduções não desapareceram completamente.

> A lematização de corpora antes da extracção de dicionários probabilísticos de tradução permite obter relacionamentos mais fortes entre palavras. No entanto é perdida informação, como os tempos verbais, género ou número.

A lematização de verbos com uma posterior extracção e filtragem de PTD permite obter dicionários bilingues de verbos.

Um tratamento semelhante poderia ser feito sobre palavras de outras classes morfológicas, por exemplo transformando todos os substantivos e adjectivos para a sua forma normalizada (masculina singular). Como as probabilidades de tradução destas classes morfológicas não é tão baixa como a dos verbos optou-se por não realizar esta experiência.

4.3.9 Tratamento de Tempos Compostos

Outro problema relacionado com a extracção de relacionamento entre verbos são os tempos compostos. Ao extrair relacionamentos entre português e inglês é natural que o verbo em português vá ter uma grande co-ocorrência com o verbo auxiliar e o verbo principal na língua inglesa, mas não um relacionamento com a construção completa. Ou seja, num caso como "*extrairei*" que se traduz por "*will extract*", o dicionário probabilístico irá associar como tradução a palavra "*extract*", já que o "*will*" irá co-ocorrer com várias outras palavras[12].

A abordagem neste caso passa, mais uma vez, pela concatenação de palavras. Uma vez que não é prático construir uma lista com toda as formas compostas, a solução passou pelo uso de um sistema de reescrita textual (`Text::RewriteRules`) que, de acordo com um conjunto de regras de padrões e algumas restrições, realiza substituições em texto.

[12]O caso particular dos tempos compostos na língua portuguesa é ligeiramente diferente, já que não se associa uma palavra de uma língua a um termo composto na outra, mas sim um termo composto em cada língua. Em todo o caso, uma abordagem semelhante seria possível para obter relacionamentos entre verbos compostos.

Por exemplo, um conjunto básico de regras para lidar com o *futuro* pode ser escrito como:

```
1 will ($wrd)  ==>  will_$1  !!  ok({CAT=>'v'} ,$dic->fea($1))

2 'll ($wrd)   ==>  will_$1  !!  ok({CAT=>'v'} ,$dic->fea($1))

3 will not ($wrd) ==> not will_$1 !! ok({CAT=>'v'},$dic->fea($1))

4 won't ($wrd) ==>  not will_$1  !!  ok({CAT=>'v'},$dic->fea($1))
```

Estas regras são divididas em três partes: o padrão a encontrar, a expressão a substituir, e o predicado a validar.

Consideremos a primeira regra: o padrão tenta encontrar o verbo auxiliar "*will*" seguido de uma qualquer palavra. O predicado verifica se a categoria gramatical[13] da palavra é *verbo*. Se assim for, a regra é activada, e as palavras são substituídas pela sua concatenação.

Na terceira e quarta regra o padrão tenta encontrar as formas negativas do verbo. Nestes dois casos estamos explicitamente a separar a palavra "*not*" uma vez que na língua portuguesa também irá existir (em princípio) a palavra "*não*".

Segue-se um extracto com alguns resultados interessantes obtidos usando esta abordagem[14].

```
1                    Palavra: gostava
2                    Ocorrências: 258
3                    Traduções:
4                         20%  would_like
5                         19%  like
6                         10%  wanted
```

[13]Usando a API disponibilizada pelo analisador morfológico jSpell.

[14]De notar que a tradução entre tempos e modos de português para inglês não é única. Dependendo do contexto o tempo e modo escolhido na língua de destino pode ser diferente.

```
Palavra: tivesse
Ocorrências: 179
Traduções:
     24%  had
     17%  would_have
      7%  it
      7%  would_prefer

Palavra: seria
Ocorrências: 3 180
Traduções:
     42%  would_be
      9%  would
      5%  it
```

A qualidade dos resultados desta abordagem irão crescer de acordo com o número de tempos compostos previstos pelo sistema de re-escrita.

> O tratamento de tempos compostos é crucial para a extracção cuidada de dicionários probabilísticos de tradução de verbos.

4.3.10 Tratamento de Termos Multi-Palavra

Como já foi sendo referido, os dicionários extraídos usando o NATools inclui apenas relacionamentos de uma palavra para uma palavra. No entanto, é sabido que existem palavras que se traduzem como termos multi-palavra.

Esta abordagem usa uma lista de termos multi-palavra extraída da junção de vários thesaurus que estão disponíveis na Internet (p.ex. o thesaurus da UNESCO). Esta lista inclui mais de 90 mil entrada para cada língua.

Usando esta lista de termos multi-palavra realizaram-se duas experiências:

1. substituir todos os termos multi-palavra por um único token (concatenando as palavras constituintes do termo multi-palavra)[15];
2. substituir todos os termos multi-palavra por um único token, mas também manter as palavras originais.

O seguinte exemplo mostra os resultados para a primeira abordagem. Note-se que os termos multi-palavra foram considerados palavras simples.

```
Palavra: jovem
Ocorrências: 133
Traduções:
     46%  young
     19%  young_person
      1%  young_woman
      1%  experienced

Palavra: rapidamente
Ocorrências: 1 521
Traduções:
     37%  quickly
     14%  wheeled
     14%  suddenly
     9%  as_soon_as_possible
     5%  rapid
     3%  rapidly

Palavra: again
Ocorrências: 2 608
Traduções:
     31%  novamente
     13%  de_novo
      7%  mais_uma_vez
```

A segunda abordagem deu resultados que consideramos de qualidade inferior. Como as palavras são mantidas como termos separados,

[15]Esta substituição é realizada pela ordem dos termos na lista. Em particular, colocaram-se os termos maiores (com mais palavras) no topo. No caso de colisão do tamanho, será usada a que aparecer primeiro.

a quantidade de palavras na matriz e a quantidade de co-ocorrências aumenta, o que leva a um aumento significativo da entropia na matriz de alinhamento. O resultado não é mais do que a união do resultado anterior com o dicionário probabilístico de tradução original.

```
1  Palavra: jovem
2  Ocorrências: 137
3  Traduções:
4       68%  young
5        2%  numbers
6        1%  systems

7  Palavra: rapidamente
8  Ocorrências: 1 527
9  Traduções:
10      33%  quickly
11      27%  wheeled
12      12%  soon
13       9%  rapidly
14       4%  rapid
15       3%  as_soon_as_possible

16 Palavra: again
17 Ocorrências: 3 995
18 Traduções:
19      31%  novamente
20      11%  mais_uma_vez
21       8%  mais
22       8%  novo
23       7%  de_novo
```

O Pré-processamento do corpus paralelo permite que se extraiam dicionários probabilísticos de tradução com diferentes tipos de resultados, que podem ser posteriormente processados e integrados, obtendo um dicionário bastante mais rico do que o obtido pelo processamento *standard* do corpus.

4.4 Programação orientada aos PTD

Esta secção demonstra a API disponibilizada para o manuseamento de dicionários probabilísticos de tradução, e a sua aplicação em diferentes tarefas no processamento de linguagem natural.

Um dicionário probabilístico de tradução atinge facilmente grandes proporções (em formato ASCII os dicionários do EuroParl ocupam 30 MB e 40 MB para cada uma das línguas). O seu carregamento não é, por isso, eficiente, especialmente para ferramentas interactivas. Foi adicionado um módulo ao NatServer (servidor de corpora e n-gramas) para a consulta eficiente de PTD.

A API do cliente Perl para o NatServer disponibiliza essencialmente duas funções para a consulta de dicionários probabilísticos:

`ptd:` para determinado corpus, língua e palavra, obter o seu número de ocorrências, e lista de possíveis traduções juntamente com a respectiva probabilidade;

`iterate:` para determinado corpus e língua, iterar sobre todas as palavras do dicionário probabilístico usando uma função de ordem superior;

Detalhes sobre esta API podem ser encontrados na secção 7.3. Esta secção inclui exemplos de uso desta API para:

- a navegação num dicionário probabilísticos de tradução usando uma interface web;
- a detecção de classes de *palavras aparentadas*: sinónimos, pertencentes ao mesmo domínio ou simplesmente aparentadas;
- a construção de dicionários bilingues *off-line* para consulta interactiva usando a aplicação StarDict.

> O uso de uma API para o manuseamento de dicionários probabilísticos de tradução permite a escrita compacta de ferramentas úteis.

4.4.1 Disponibilização de Dicionários

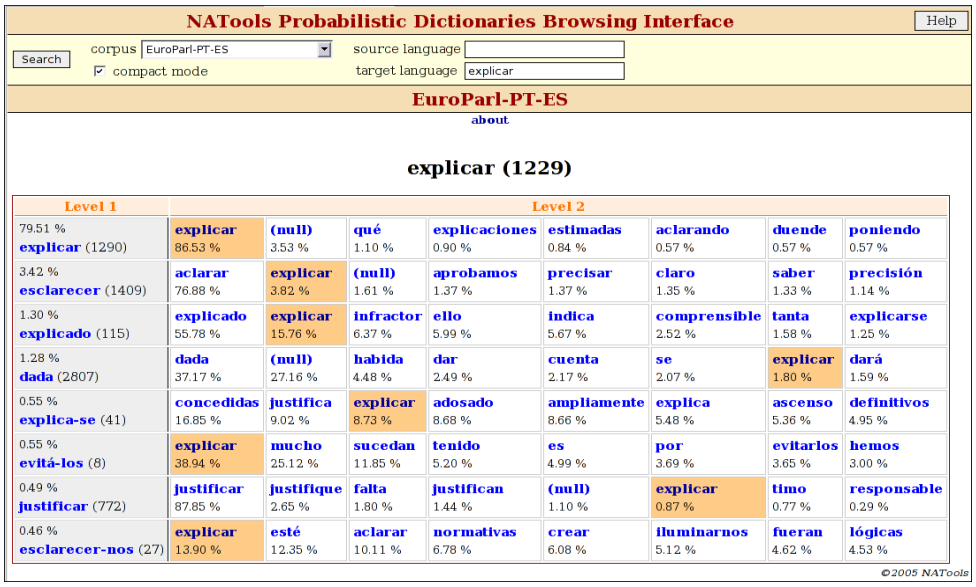


Figura 4.7: Interface web em modo compacto para a consulta e navegação em dicionários probabilísticos de tradução.

Assim como para os corpora paralelos, parece-nos crucial que estes dicionários não sejam utilizados apenas para o desenvolvimento de novas ferramentas, mas que possam desde logo ser consultados por utilizadores finais. Com base nesta premissa foi desenvolvido um interface Web para a consulta e navegação em dicionários disponíveis no NatServer. Esta interface está integrada com as restantes interfaces web, como descrito na secção 6.1.

A figura 4.7 mostra a forma compacta desta interface. A tabela apresenta na primeira coluna as traduções da palavra procurada com a respectiva probabilidade de tradução. Cada uma das linhas corresponde às traduções da primeira palavra dessa mesma linha (portanto, traduções das traduções da palavra procurada). As células sombreadas correspondem àquelas traduções que contam com a palavra original como possível tradução, ou seja, com a tradução reflexiva:

$$w_{\mathcal{A}} \in \mathcal{T}_{d_{\mathcal{B},\mathcal{A}}} \left(\mathcal{T}_{d_{\mathcal{A},\mathcal{B}}} \left(w_{\mathcal{A}} \right) \right)$$

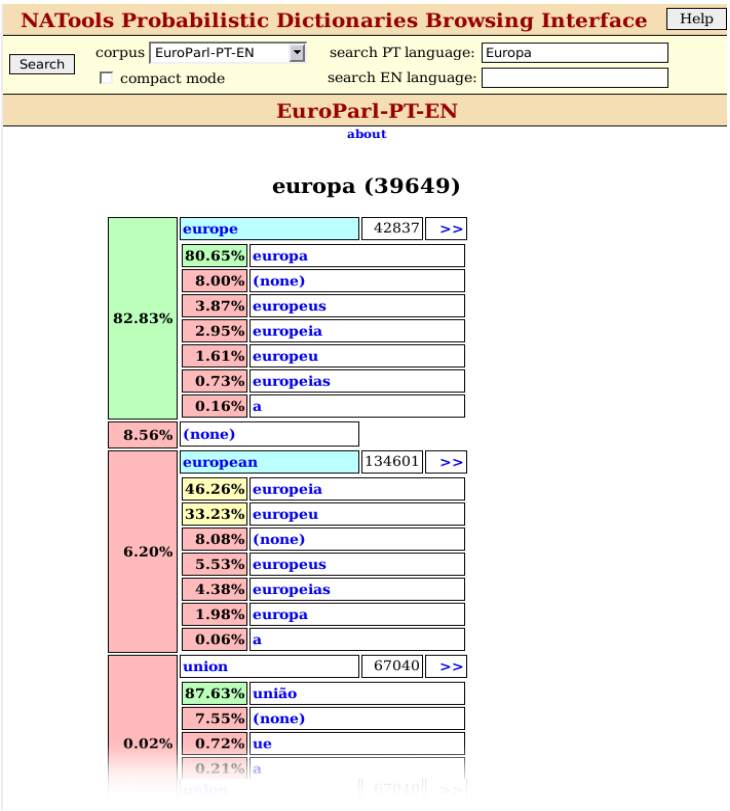


Figura 4.8: Interface web em modo expandido para a consulta e navegação em dicionários probabilísticos de tradução.

Existe ainda a possibilidade de mudar da forma compacta para a expandida, onde se consegue ter uma noção visual por cores das probabilidades de tradução, de acordo com a figura 4.8.

A interface permite comutar entre estes modos usando para isso uma opção na barra no topo da interface, onde também é possível escolher o corpus/dicionário e a língua para consulta.

Nos dois modos, as palavras são clicáveis de modo a ser possível ir navegando no dicionário, consultando traduções de palavras em ambas as línguas.

Também é possível seguir uma ligação para a pesquisa de concordâncias no corpus que está a ser consultado. Esta concordância é realizada

com a palavra a ser visualizada, e a tradução escolhida. Esta funcionalidade é especialmente útil na compreensão de traduções inesperadas (ver exemplo da página 117).

Do mesmo modo, é possível a partir da interface de concordâncias saltar automaticamente para a consulta do dicionário probabilístico de tradução bastando para isso fazer duplo-clique sobre a palavra a consultar.

> O interface de consulta de recursos deve ser rico em informação e, sempre que possível, integrado e ligado.

4.4.2 Palavras Aparentadas

Num dicionário probabilístico de tradução, é de esperar que as traduções de determinada palavra estejam de alguma forma relacionadas com essa palavra. Se esta relação for transitiva, é possível calcular um conjunto de palavras relacionadas com uma palavra x a partir do cálculo das traduções das suas traduções, ou seja, a composição de um dicionários com o seu inverso, $\mathcal{T}_{d_{\mathcal{B},\mathcal{A}}}\left(\mathcal{T}_{d_{\mathcal{A},\mathcal{B}}}\left(w_{\mathcal{A}}\right)\right)$, como esquematizado na figura 4.9. O algoritmo 3 apresenta com maior detalhe esta abordagem.

Parentes: $\mathrm{W}_{\mathcal{A}} \longrightarrow \mathrm{W}^{\star}_{\mathcal{A}}$

for $w_{\mathcal{A}} \in dom\left(d_{\mathcal{A},\mathcal{B}}\right)$ **do**
    $Parentes_{w_{\mathcal{A}}} \longleftarrow \{\}$
    $T_{w_{\mathcal{A}}} \longleftarrow \mathcal{T}_{d_{\mathcal{A},\mathcal{B}}}\left(w_{\mathcal{A}}\right)$
    for $w_{\mathcal{B}} \in T_{w_{\mathcal{A}}}$ **do**
        $T_{w_{\mathcal{B}}} \longleftarrow \mathcal{T}_{d_{\mathcal{B},\mathcal{A}}}\left(w_{\mathcal{B}}\right)$
        $Parentes_{w_{\mathcal{A}}} \longleftarrow Parentes_{w_{\mathcal{A}}} \cup T_{w_{\mathcal{B}}}$

Algoritmo 3: Cálculo de palavras aparentadas de $w_{\mathcal{A}}$ usando um $ptd_{\mathcal{A},\mathcal{B}}$.

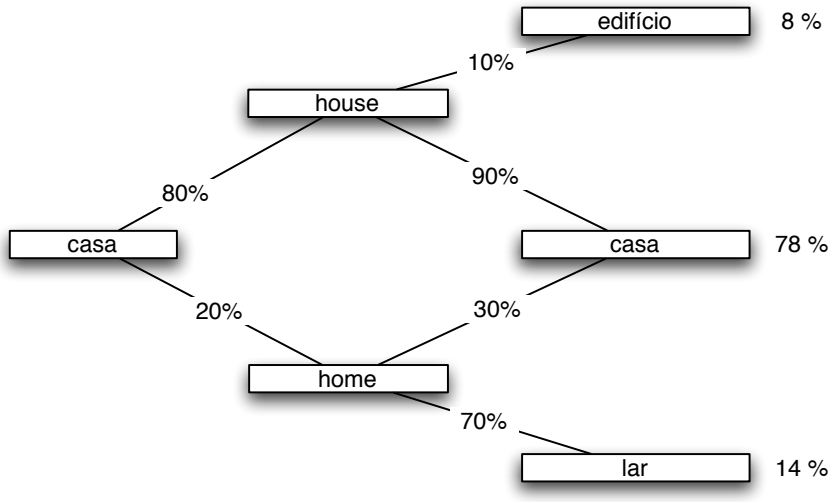


Figura 4.9: Esquema de cálculo de palavras aparentadas.

Tirando partido do facto de estarmos perante um dicionário probabilístico e não apenas de um dicionário de tradução, é-nos possível calcular uma probabilidade de determinada palavra pertencer ao conjunto de palavras aparentadas. Para isso é usada a seguinte fórmula[16]:

$$\mathcal{P}\left(v_{\mathcal{A}} \in Parentes_{w_{\mathcal{A}}}\right) = \sum_{\substack{w_{\mathcal{B}} \in \mathcal{T}_{d_{\mathcal{A},\mathcal{B}}}(w_{\mathcal{A}}) \\ v_{\mathcal{A}} \in \mathcal{T}_{d_{\mathcal{B},\mathcal{A}}}(w_{\mathcal{B}})}} \mathcal{P}\left(w_{\mathcal{B}} \in \mathcal{T}_{d_{\mathcal{A},\mathcal{B}}}\left(w_{\mathcal{A}}\right)\right) \mathcal{P}\left(v_{\mathcal{A}} \in \mathcal{T}_{d_{\mathcal{B},\mathcal{A}}}\left(w_{\mathcal{B}}\right)\right)$$

Segue-se um pequeno extracto dos conjuntos de palavras semelhantes a "*país*" e "*povo*," juntamente com a confiança associada.

	país		*povo*	
	país	62.511	pessoas	36.158
	países	8.153	*povo*	9.914
	estado	0.453	cidadãos	5.934

[16]Esta fórmula é uma aproximação à probabilidade: uma medida de parentesco. Não pode ser considerada uma probabilidade já que os eventos $\mathcal{P}\left(w_{\mathcal{B}} \in \mathcal{T}_{d_{\mathcal{A},\mathcal{B}}}\left(w_{\mathcal{A}}\right)\right)$ e $\mathcal{P}\left(v_{\mathcal{A}} \in \mathcal{T}_{d_{\mathcal{B},\mathcal{A}}}\left(w_{\mathcal{B}}\right)\right)$ não são propriamente independentes.

```
5        território  0.427          população    5.321
6        turquia     0.412          popular      3.872
7    *   de          0.332          povos        3.237
8        nacionais   0.277          nação        1.830
9    *   há          0.170      *   os           1.748
```

Embora nem todas as respostas sejam correctas ou úteis, as probabilidades associadas são relevantes já que permitem tirar conclusões sobre a confiança do sinónimo encontrado. As palavras encontradas que correspondem a respostas pouco úteis (e que foram marcadas com um asterisco) poderiam ter sido facilmente removidas usando uma lista de *stop-words*.

Segue-se a implementação do algoritmo em Perl, como forma de exemplificar o uso da API disponibilizada para manuseamento de PTD.

```
1    use NAT::Client;

2    my $client = NAT::Client->new( crp => "EuroParl-PT-EN" );
3    my %r = ();

4    my $a1 = $client->ptd( "europa" );
5    for my $b1 (keys %{$a1->[1]}) {
6      my $c = $client->ptd( { from => 'target' }, $b1);
7      for my $d (keys %{$c->[1]}) {
8        $r{$d} +=  $a1->[1]{$b1} * $c->[1]{$d};
9      }
10   }
11   for((sort {$r{$b} <=> $r{$a}} keys %r)[0..9]) {
12       printf "  %15s %.3f ", $_, $r{$_}*100
13   }
```

linha 4: calcular todas as traduções para a palavra *europa*;
linha 5: iterar sobre as traduções;
linha 6: calcular as traduções para cada tradução (composição);
linha 7: iterar sobre as traduções das traduções;
linha 8: calcular as medidas de confiança;
linha 11–12: imprimir resultados;

4.4.3 Dicionários StarDict

Apesar da generalização do acesso à Internet, ainda existe vantagem na consulta de dicionários e de outros recursos em modo local (*off-line*), pelo que se considera útil a criação de dicionários que possam ser instalados e usados num computador pessoal para ajuda na tradução.

Tomando como ponto de partida os PTD e tendo também como fonte de informação a pesquisa de concordâncias, desenvolveu-se um programa para a criação de dicionários StarDict[17].

Os dicionários StarDict implementam correspondências entre palavras e informação associada:

$$\mathrm{W}_{\mathcal{A}} \rightharpoonup \mathrm{Info}$$

Com base num dicionário probabilístico de tradução d e no corpus que lhe deu origem é possível criar um dicionário de tradução que, para cada palavra $w_{\mathcal{A}}$, mostre:

- as traduções $w_{\mathcal{B}} \in \mathcal{T}_d(w_{\mathcal{A}})$, juntamente com a sua probabilidade $\mathcal{P}(w_{\mathcal{B}} \in \mathcal{T}_d(w_{\mathcal{A}}))$;
- para cada uma das possíveis traduções $w_{\mathcal{B}} \in \mathcal{T}_d(w_{\mathcal{A}})$, algumas entradas de concordâncias extraídas do corpus que deu origem ao dicionário, de forma a explicitar em que situações a palavra $w_{\mathcal{A}}$ se traduz por $w_{\mathcal{B}}$.

Com a API disponibilizada pelo módulo de acesso ao servidor NatServer é possível construir este dicionário com pouquíssimas linhas:

```
1  use NAT::Client;
2  $client = NAT::Client -> new ( crp => "EuroParl-PT-EN" );

3  $client -> iterate ( { Language => "PT" },
4      sub {
5          my %param = @_;
6          for my $trans (keys %{$param{trans}}) {
```

[17] O StarDict foi desenvolvido por Hu Zheng e é uma ferramenta gráfica livre para a consulta de dicionários. A página oficial do projecto é `http://stardict.sourceforge.net/`.

```
7               if ($param{trans}{$trans} > 0.2) {
8                   my $concs = $client->conc({concordance => 1},
9                                             $param{word}, $trans);
10                  $stardict{$param{word}}{$trans} = $concs -> [0];
11              }
12          }
13         });
14  print StarDict($stardict);
```

linha 3 iterar por todas as palavras do dicionário;
linha 4 definição da função para processar cada entrada;
linha 6 iterar sobre as traduções de cada palavra;
linha 7 se a tradução tiver uma certeza acima de 20% é colocada no dicionário;
linha 8 calcular as concordâncias para aquele par (palavra,tradução);

A figura 4.10 mostra a interface da aplicação StarDict a consultar um destes dicionários. A secção 6.2 apresenta um exercício semelhante ao aqui apresentado mas em que o dicionário foi enriquecido com n-gramas e entradas terminológicas.

Os dicionários StarDict são muito úteis para a tarefa de tradução assistida por computador, uma vez que incluem o contexto em que as traduções são aplicadas.

A Título de Conclusão

A extracção automática de dicionários de tradução (mesmo que probabilísticos) permite a criação rápida e eficaz de recursos de tradução que obrigariam a um grande investimento se criados manualmente.

A avaliação deste tipo de recursos não é simples. Uma avaliação manual cuidada permite obter uma noção de qualidade para determinado fim (normalmente, como um dicionário de tradução convencional). No entanto, os recursos que são obtidos são dicionários referentes a determinado corpus, e portanto em determinado contexto.

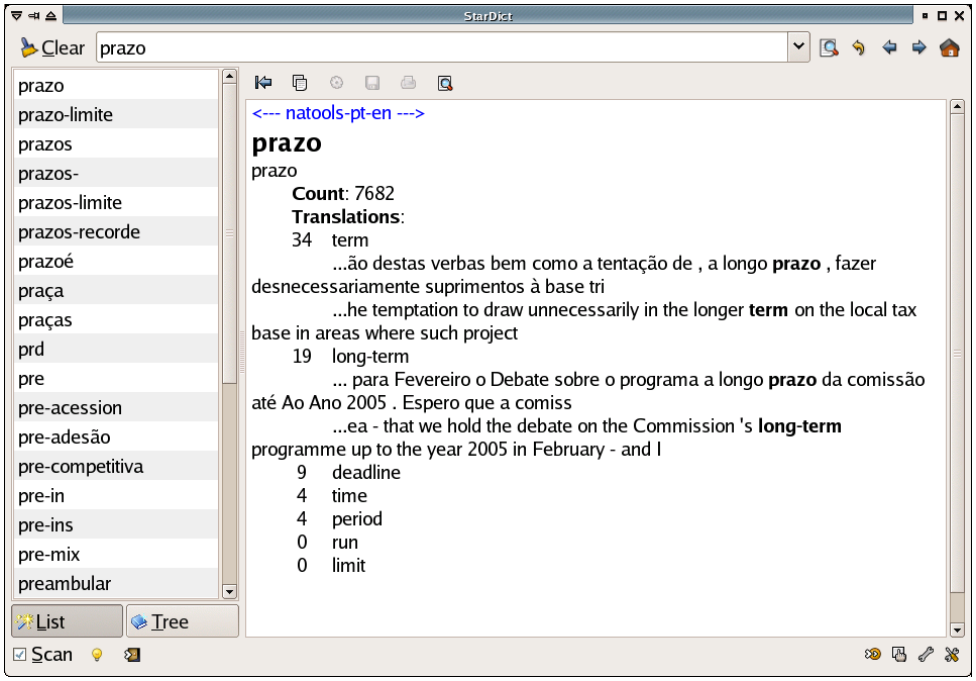


Figura 4.10: StarDict com um dicionário baseado em PTD.

Foram realizadas três abordagens de avaliação manual que demonstram a qualidade dos dicionários, não na sua forma bruta, mas depois de definidas restrições nas suas entradas, como sendo patamares de número de ocorrências ou de probabilidades de tradução.

Como a avaliação manual é morosa e dispendiosa, foram apresentados métodos para a comparação de dicionários e medidas para a detecção das entradas com maiores diferenças.

Embora os dicionários como um todo não possam ser considerados dicionários de tradução, foram apresentadas abordagens para melhorar a sua qualidade e de certa forma aproximar o resultado obtido a um dicionário de tradução convencional. Alguns dos métodos apresentados não melhoram um dicionário como um todo, mas melhoram traduções entre determinados conjuntos ou tipos de palavras. Destes métodos, a maior parte baseia-se no pré-processamento dos corpora, com a anotação de determinado tipo de palavras.

Finalmente, foram apresentados alguns recursos que podem ser ob-

tidos directamente a partir de dicionários probabilísticos de tradução, como sejam a criação de classes de palavras ou a criação de dicionários StarDict para a consulta em *offline* de dicionários e concordâncias bilingues.

Capítulo 5

Extracção de Exemplos de Tradução

What makes literature interesting is that it does not survive its translation. The characters in a novel are made out of the sentences. That's what their substance is.

Jonathan Miller

Como foi sendo introduzido no Capítulo 2, a tradução, seja ela automática, semi-automática ou manual, tira partido de traduções realizadas anteriormente de forma a re-aproveitar trabalho já realizado.

O nível de reutilização depende do tipo de recurso consultado. A tabela 5.1 resume o nível de reutilização de diferentes tipos de recursos bilingues e qual a confiança na sua reutilização.

Os sistemas de tradução baseados em memórias de tradução reutilizam frases. Esta reutilização pode ser realizada com confiança já que as frases incluem uma grande parte do contexto (uma mesma frase em sítios diferentes pode ser traduzida quase sempre da mesma forma). No entanto, normalmente só são reutilizáveis para traduzir exactamente a mesma frase (ou com alguns sistemas de *fuzzy matching*, uma frase bastante semelhante).

Recurso	Fronteira	Reutilização	Confiança
texto	clara	muito baixa	muito boa
frase	razoável	pequena	boa
exemplo/segmento	difícil	razoável	razoável
exemplo com padrões	difícil	boa	razoável
palavras	clara	muito boa	muito baixa

Tabela 5.1: Níveis de reutilização de diferentes tipos de recursos.

Por outro lado, a reutilização de palavras é muito alta, mas a sua confiança de reutilização é bastante baixa dada a grande ambiguidade na tradução de uma única palavra.

Os sistemas de tradução automática baseados em dados (de acordo com a secção 2.3.2) usam um compromisso entre a reutilização de frases e de palavras. O objectivo é dividir uma frase em segmentos (que são normalmente designados por *exemplos de tradução*) que tenham um nível de reutilização razoável (acima do nível da frase mas, infelizmente, abaixo do nível da palavra), e com uma confiança de reutilização aceitável.

Definição 7 *Um* ***Exemplo de Tradução*** *é um par de segmentos de palavras* $\langle s_{\mathcal{A}}, s_{\mathcal{B}} \rangle$ *do tipo* $\mathrm{W}_{\mathcal{A}}{}^{\star} \times \mathrm{W}_{\mathcal{B}}{}^{\star}$*, tal que* $\mathcal{T}(s_{\mathcal{A}}) = s_{\mathcal{B}}$*.*

Não existe qualquer restrição relativa ao número de palavras de cada um dos segmentos, sendo que habitualmente os exemplos de tradução têm duas ou mais palavras, e raramente excedem as 8 a 10 palavras.

O conceito de *exemplo de tradução* é especialmente usado na abordagem à tradução automática denominada por Tradução Automática Baseada em Exemplos. Usamos o termo *exemplo de tradução* como um objecto próximo da unidade de tradução mas com tamanho médio mais pequeno. Estes *exemplos*, por serem mais pequenos do que as frases completas existentes na memória de tradução são mais facilmente reutilizáveis: não se tenta encontrar a frase completa na memória de tradução, mas sim traduzir segmentos de acordo com os exemplos existentes.

Um tradutor, enquanto utilizador de uma ferramenta de tradução assistida, consegue gerir de forma mais ou menos controlada o tamanho das unidades das suas memórias de tradução. Quando se extrai unidades de tradução de forma automática isto não é possível. Basta analisar alguns dos corpora paralelos mais usados actualmente para investigação em tradução automática, como sejam o EuroParl ou o JRC-Acquis, para se verificar que as unidades de tradução são muito grandes (acima das 20 palavras).

Assim, têm vindo a ser estudados métodos para a segmentação de unidades de tradução construindo exemplos de tradução. Sendo este recurso útil à comunidade de tradutores e à comunidade de investigadores em tradução automática, investiu-se na construção de algoritmos para a *Extracção Automática de Exemplos de Tradução* tomando como base *Corpora Paralelos* e *Dicionários Probabilísticos de Tradução*.

Este capítulo apresenta duas abordagens para a extracção de exemplos:

- A primeira abordagem (*hipótese das palavras-marca*) é baseada em trabalho semelhante (Armstrong et al., 2006), embora neste trabalho se use o conhecimento obtido no cálculo de PTD para facilitar a tarefa de alinhamento entre exemplos (secção 5.1);
- A segunda abordagem baseia-se num re-alinhamento ao nível do segmento. Esta abordagem será apresentada em duas secções distintas:
  - detecção de âncoras de alinhamento usando probabilidades de tradução de um PTD (secção 5.2);
  - semelhante à anterior, mas tirando partido de padrões de tradução definidos pelo utilizador (secção 5.3).

Para aumentar a reutilização dos exemplos de tradução, tem-se vindo a aplicar técnicas de *generalização*. Estas técnicas têm como principal objectivo a substituição de determinadas palavras por *place-holders*, ou variáveis tipadas. Estas regras (segmentos paramétricos) podem ser compostas com diferentes palavras e padrões, aumentando assim a sua re-utilização. A secção 5.4 apresenta uma abordagem para a generalização de exemplos utilizando padrões de tradução.

5.1 Hipótese das Palavras-Marca

Com base em trabalho realizado por (Veale and Way, 1997) decidiu-se analisar a *Hipótese das palavras-marca* (na sua versão inglesa, *Marker Hypothesis*) para a segmentação de texto bilingue, tendo especial atenção os casos em que uma das línguas é o português. Esta segmentação foi usada para a extracção de exemplos de tradução.

5.1.1 Segmentação Monolingue

Em (Green, 1979) é definida a *Hipótese das palavras-marca*, uma restrição psico-linguística na estrutura gramatical, que foi usada posteriormente por (Juola, 1995) para a segmentação em tradução automática. Esta *hipótese* conjectura que as línguas naturais têm a sua estrutura gramatical marcada (ou delimitada) por um conjunto fechado de lexemas ou morfemas (*palavras-marca*).

Isto significa que um sistema pode obter uma segmentação básica de uma frase explorando uma lista fechada de palavras-marca que delimitam o início e fim de cada segmento.

Estas palavras-marcas pertencem habitualmente a classes fechadas de palavras (ver também a secção 4.3.7), como sejam preposições, pronomes, locuções, artigos, determinantes e alguns advérbios.

Para explicar o algoritmo de segmentação baseado na hipótese das palavras-marca consideremos a seguinte frase:

O João passou toda a tarde a brincar com os colegas.

As marcas presentes nesta frase são os artigos "*o*", "*a*" e "*os*", a preposição "*com*" e o pronome "*toda*":

O João passou toda a tarde a brincar com os colegas.

Se considerarmos todos os segmentos que começam por uma ou mais marcas, e terminam antes do próximo conjunto de marcas, obtemos os seguintes segmentos:

(O João passou) (toda a tarde) (a brincar) (com os colegas.)

Embora estes segmentos não correspondam aos verdadeiros sintagmas da frase, constituem uma boa aproximação.

A lista de marcadores para a língua portuguesa foi construída com base na lista inglesa usada no projecto MaTrEx (Armstrong et al., 2006). A tabela 5.2 mostra um excerto desta lista. É interessante reparar que a lista portuguesa é razoavelmente maior devido à flexão de género e número que não é tão comum em inglês (um total de 398 marcas para a língua inglesa e de 596 marcas para a língua portuguesa).

O algoritmo de segmentação de uma frase de acordo com a hipótese das palavras-marca corresponde a, dada uma frase e uma lista de marcas:

1. encontrar todas as marcas existentes na frase;
2. considerar qualquer sequência de marcas como uma única marca, que corresponde ao início de um segmento (obviamente que este algoritmo/hipótese não terá uma aplicação directa nas línguas em que se marcam os finais dos segmentos, como o japonês, turco ou basco);
3. cada segmento termina na palavra imediatamente antes do próximo conjunto de marcas (ou no final da frase);

A tabela 5.3 mostra alguns dos segmentos mais comuns no corpus EuroParl PT:EN para ambas as línguas. Por sua vez, a tabela 5.4 resume as marcas mais produtivas em cada uma das línguas.

> A hipótese das palavras-marca constitui um método simples e eficaz para uma segmentação básica de frases na língua portuguesa.

5.1.2 Segmentação Bilingue e Alinhamento

Como vimos, a segmentação monolingue usando a hipótese das palavras-marca tem um algoritmo simples. Embora a sua aplicação a uma uni-

Marca em inglês	Marca em português
most	maior; maioria
much	muito
my	meu; minha; meus; minhas
near; nearby	perto; próximo; quase
neither	tão-pouco; também não
next	seguinte; próximo; próxima
nigh	próximo
no	não
nor	nem
now	agora; uma vez que; considerando que
of	de; por; em
off; out	fora; de fora
on	em; sobre; em cima de; de; relativa
once	desde que; uma vez que; se
one	um; uma
only	apenas; todavia; mas; contudo
or	ou; se não
other	outro; outra; outras; outros
our	nosso; nossa; nossos; nossas
over	sobre; em cima de; por cima de
owing to	devido a: por consequência de; por causa de
own	próprio; ser proprietário
past	por; para além disso; fora de
per	por; através de; por meio de; devido a acção de
plus	mais; a acrescentar a; a adicionar a
round	em torno de; à volta de
same	mesmo; mesma
several	vários
sort of	espécie de; género de; tipo de; de certo modo
since	desde; desde que; depois que
so	portanto; por isso
some	algum; alguns; alguma; algumas
such	este; esse; aquele; isto; aquilo
supposing	supondo; se; no caso de; dada a hipótese de
than	de; que; do que; que não
that	aquele; aquela; aquilo; esse; essa; isso; ...
the	o; a; os; as

Tabela 5.2: Excerto de marcadores EN:PT.

34 137	**da**	comissão
17 277	**do**	conselho
16 891	**da**	união europeia
11 379	**em**	matéria
9 880	**de**	trabalho
9 850	**da**	união
9 479	**no**	sentido
8 465	**da**	europa
8 454	**da**	ue
8 004	**do**	parlamento
Total de 3 070K segmentos		

13 566	**and**	gentlemen
11 466	**the**	commission
11 079	**in**	order
9 182	**to**	make
8 712	**to**	be
8 356	**to**	do
7 992	**of the**	european union
7 941	**of the**	committee
7 814	**to**	say
7 574	**with**	regard
Total de 3 103K segmentos		

Tabela 5.3: Alguns segmentos extraídos do EuroParl (PT e EN).

dade de tradução seja igualmente simples, o alinhamento entre os segmentos obtidos não é trivial.

O primeiro problema surge em relação ao número de segmentos existentes em cada uma das frases. Embora se pudesse esperar que uma frase e a sua tradução tivessem o mesmo número de segmentos, a verdade é que tal não acontece. Mesmo no caso de traduções literais a própria estrutura da língua leva a que a quantidade de palavras-marca, e portanto a quantidade de segmentos, seja diferente. Veja-se como exemplo a seguinte unidade de tradução[1]:

- (this decision shall take effect) (on 16 september 1999)
- (a presente decisão produz efeitos) (em 16) (de setembro) (de 1999)

A segmentação de uma unidade de tradução resulta numa sequência de segmentos com uma cardinalidade diferente para cada língua. O alinhamento entre estes segmentos pode ser visto como um caso particular do alinhamento de frases, e portanto com uma solução conhecida baseada em programação dinâmica (Gale and Church, 1991). Como dispomos de dicionários probabilísticos de tradução, a nossa abordagem

[1]Embora este exemplo seja realmente extraído de um corpus, não é um dos melhores exemplos da dificuldade de alinhamento entre segmentos extraídos pela hipótese das palavras-marca. Um pré-processador que detectasse a anotasse datas permitiria um alinhamento mais simples.

815815	de
557697	,
468409	a
352064	da
297634	do
232629	e
197922	que
196801	o
178537	em
156299	dos
[...]	
35394	para a
33079	que o
32213	de um
31539	nos
31492	muito
30805	às
Total de 243K marcas dif.	

541197	to
471332	the
440903	of
400417	,
370161	and
252298	of the
214191	in
152164	a
131225	in the
112446	for
105992	that
92180	on
91033	to the
78264	we
70578	on the
67805	this
65092	that the
Total de 198K marcas dif.	

Tabela 5.4: Contagens das marcas mais produtivas (extraídas do Euro-Parl PT:EN).

usa-os, associando a cada par de segmentos um valor de probabilidade ou certeza de tradução mútua.

Um dos problemas na realização do alinhamento é a baixa probabilidade de tradução que existe habitualmente entre palavras-marca. Por exemplo, a profusa flexão da língua portuguesa leva a que as probabilidades associadas à tradução de um artigo da língua inglesa para a portuguesa sejam bastante baixas (considerando um caso óptimo de "*the*" traduzido por "*a*", "*o*", "*as*" e "*os*" teríamos 25% de probabilidade para cada uma destas traduções). Existe ainda a problemática da variância de locuções, do sujeito omisso, das contracções e, genericamente, de toda a ambiguidade de tradução entre palavras-marca.

Para minorar este problema optou-se por dar maior peso à probabilidade de tradução das palavras que não são marcas do que à probabilidade de tradução entre palavras-marca.

Outras abordagens podiam ter sido tomadas, como o cálculo prévio de todos os segmentos existentes no corpus para se poder tirar partido do seu número de co-ocorrências. Esta abordagem não foi experimentada por se preferir um método que processe cada unidade de tradução de forma independente.

Também foi experimentada uma abordagem semelhante à proposta na secção 4.3.7 mas em que se aglutinaram todas as sequências de palavras marca, para obter um dicionário probabilístico de tradução entre segmentos de marcas. No entanto, os resultados obtidos foram inferiores aos aqui apresentados.

No cálculo das probabilidades de tradução deve-se ter atenção que um segmento em determinada língua ($s_{\mathcal{A}}$) pode corresponder a vários segmentos noutra língua ($s_{\mathcal{B}_i}$). Neste caso, só algumas palavras do primeiro segmento ($s_{\mathcal{A}}$) vão ter uma correspondência em cada um dos segmentos da outra língua, pelo que a probabilidade de tradução não deve ser vista como "*a probabilidade de* $s_{\mathcal{A}}$ *e* $s_{\mathcal{B}}$ *serem traduções mútuas*" mas antes (considerando que $s_{\mathcal{A}} > s_{\mathcal{B}}$) como "*a probabilidade de a tradução de* $s_{\mathcal{B}}$ *estar contida em* $s_{\mathcal{A}}$."

O algoritmo 4 mostra de forma simplificada o processo de cálculo de uma medida probabilística da tradução entre dois segmentos utilizando um PTD, e dando um peso de apenas 10% à probabilidade de tradução entre marcas. Utilizando este método em cada combinação de dois segmentos é preenchida uma matriz de alinhamento como a apresentada na tabela 5.5.

Esta matriz é percorrida procurando-se as células com maior valores que correspondem aos alinhamentos mais prováveis. Estes alinhamentos são extraídos na forma de pares de segmentos. Por exemplo, da matriz apresentada poderiam ser extraídos os seguintes exemplos de tradução: "*a presente decisão produz efeitos / this decision shall take effect*" e "*em 16 de setembro de 1999 / on 16 september 1999.*" Estes segmentos são posteriormente ordenados e as suas ocorrências calculadas por tradução, de forma a que se possam estudar estatisticamente.

Data: Sejam $s_{\mathcal{A}}$ e $s_{\mathcal{B}}$ dois segmentos, na língua $\mathcal{A}$ e $\mathcal{B}$ respectivamente, tal que $s_{\mathcal{A}} < s_{\mathcal{B}}$ e, $d_{\mathcal{A},\mathcal{B}}$ o dicionário probabilístico de tradução entre essas línguas.

function quality(Dic, Set_1, Set_2)
  $Soma \longleftarrow 0$
  for $w_{\mathcal{A}} \in Set_1$ **do**
    for $w_{\mathcal{B}} \in dom\left(\mathcal{T}_{dic}\left(w_{\mathcal{A}}\right)\right)$ **do**
      if $w_{\mathcal{B}} \in Set_2$ **then**
        $Soma \longleftarrow Soma + \mathcal{P}\left(w_{\mathcal{B}} \in \mathcal{T}_{dic}\left(w_{\mathcal{A}}\right)\right)$
  return $\frac{Soma}{\text{size}(Set_1)}$
end
$MedidaMarcas \longleftarrow \text{quality}(d_{\mathcal{A},\mathcal{B}}, \text{marcas}\left(s_{\mathcal{A}}\right), \text{marcas}\left(s_{\mathcal{B}}\right))$
$MedidaTexto \longleftarrow \text{quality}(d_{\mathcal{A},\mathcal{B}}, \text{texto}\left(s_{\mathcal{A}}\right), \text{texto}\left(s_{\mathcal{B}}\right))$
$Medida \longleftarrow 0.1 \times MedidaMarcas + 0.9 \times MedidaTexto$

Algoritmo 4: Cálculo de uma medida de certeza da tradução entre dois segmentos $s_{\mathcal{A}}$ e $s_{\mathcal{B}}$.

> O uso de dicionários probabilísticos de tradução facilita o alinhamento dos segmentos extraídos com a hipótese das palavras-marca.

A tabela 5.6 apresenta alguns dos exemplos com mais ocorrências[2] em que o alinhamento foi de um para um segmento.

O exemplo 12 mostra que a hipótese das palavras-marca tem alguns problemas quando se considera que os parêntesis são marcas, e estes não aparecem em ambos os segmentos da unidade de tradução processada. A nível linguístico é interessante reparar na omissão do sujeito na língua portuguesa.

Por sua vez, as tabelas 5.7 e 5.8 mostram alguns exemplos com alinhamento de um para dois segmentos. Na tabela 5.7 o exemplo 12

[2]Foram excluídos todos aqueles que continham pontuação por serem pouco interessantes.

	this decision shall take effect	*on* 16 september 1999
a presente decisão produz efeitos	23.18	5.86
em 16	0.00	76.41
de setembro	0.00	85.60
de 1999	0.00	84.10

Tabela 5.5: Matriz de alinhamento.

encontra-se errado, que deriva do facto da palavra inglesa "*much*" ser um marcador que neste contexto aparece no final de um segmento e não no início como defende a hipótese das palavras-marca. Em relação aos alinhamentos de dois para um segmento, de salientar que o exemplo 13 é um alinhamento correcto no caso concreto do Parlamento Europeu, em que a palavra "*hemiciclo*" é omitida na língua inglesa.

Finalmente, a tabela 5.9 é a que apresenta piores resultados. A quantidade de segmentos aumenta, e a ordem das traduções também, o que leva a que o algoritmo tenha maiores problemas no alinhamento.

> O uso da hipótese das palavras-marca permite a extracção de exemplos de tradução com alguma qualidade. No entanto, à medida que o alinhamento é realizado entre um maior número de segmentos, a qualidade dos exemplos baixa.

5.1.3 Discussão de Resultados

Embora estas traduções sejam correctas não podem ser vistas como única fonte para a tradução automática. O problema conhecido como *Boundary Friction* (Brown et al., 2003) não é de todo resolvido. Ou seja, estes exemplos não podem ser concatenados sem a existência de um pós-processador que trate de homogeneizar os exemplos, como seja a concordância de género e número. Neste sentido, a avaliação deste recurso deve ser feita não como um recurso isolado mas como parte integrante de um sistema de tradução automática.

		Ocorrências	Português	Inglês
1		36886	senhor presidente	mr president
2		8633	senhora presidente	madam president
3		3152	espero	i hope
4		2930	gostaria	i would like
5		2572	o debate	the debate
6		2511	penso	i think
7		2356	está encerrado	is closed
8		1939	penso	i believe
9		1932	muito obrigado	thank
10		1854	em segundo lugar	secondly
11		1809	gostaria	i should like
12	⋆	1638	) senhor presidente	mr president
13		1423	infelizmente	unfortunately
14		1345	creio	i believe
		$\bar{x} = 1.6654$	Total: 1 507 225	

Tabela 5.6: Alguns dos exemplos (1:1) mais ocorrentes extraídos do EuroParl PT:EN com base na Hipótese das Palavras-Marca.

		Ocorrências	Português	Inglês
1		253	caros colegas	ladies and gentlemen
2		147	senhores deputados	ladies and gentlemen
3		143	devo dizer	i have to say
4		142	lamento	i am sorry
5		105	congratulo-me	i am pleased
6		95	estou convencido	i am convinced
7		90	vamos agora proceder	we shall now proceed
8	⋆	90	e senhores deputados	ladies and gentlemen
9		90	agradeço	i am grateful
10		79	e outros , em nome	and others , on behalf
11		76	refiro-me	i am referring
12	⋆	72	muito obrigado	thank you very
13		71	congratulo-me	i am glad
14		70	passamos agora	we shall now proceed
15		66	não há dúvida	there is no doubt
		$\bar{x} = 1.0464$	Total: 350 065	

Tabela 5.7: Alguns dos exemplos (1:2) mais ocorrentes extraídos do EuroParl PT:EN com base na Hipótese das Palavras-Marca.

		Ocs.	Português	Inglês
1		986	segue-se na ordem	the next item
2		222	(a sessão é suspensa	(the sitting was closed
3		169	senhor presidente em exercício	mr president-in-office
4		148	da sessão de ontem	of yesterday 's sitting
5		142	(o parlamento aprova a acta	(the minutes were approved
6	⋆	138	dos assuntos económicos e monetários	and monetary affairs
7		113	a proposta da comissão	the commission 's proposal
8		110	a proposta da comissão	the commission proposal
9		106	período de perguntas	question time
10	⋆	101	, em nome , sobre a proposta	, on behalf
11		100	dos direitos do homem	of human rights
12		84	dos direitos da mulher	on women 's rights
13	⋆	72	da direita do hemiciclo	from the right
14		67	por interrompida do parlamento europeu	of the european parliament adjourned
15		67	é muito importante	it is very important
		$\bar{x} = 1.0385$	Total: 542 671	

Tabela 5.8: Alguns dos exemplos (2:1) mais ocorrentes extraídos do EuroParl PT:EN com base na Hipótese das Palavras-Marca.

	Ocs		Português	Inglês
1	⋆	363	segue-se na ordem a discussão conjunta	the next item
2	⋆	83	(o presidente retira a palavra à oradora	(the president cut
3	⋆	59	segue-se na ordem do dia	the next item
4	⋆	42	que recebi de resolução , apresentadas	have received
5	⋆	39	de aplicação do processo de urgência	for urgent procedure
6	⋆	36	, de pé um minuto de silêncio	a minute 's silence
7		32	está encerrado o período de perguntas	that concludes question time
8	⋆	31	nos termos do artigo 37 º do regimento	pursuant to rule 37
9	⋆	29	segue-se na ordem o período	the next item
10		28	está encerrado o período de votações	that concludes voting time
11		26	está encerrado o período de votação	that concludes voting time
12	⋆	23	ao comité de conciliação de conciliação	to the conciliation committee
13	⋆	19	segue-se na ordem da discussão conjunta	the next item
14	⋆	19	ao senhor presidente em exercício do conselho	the president-in-office
15	⋆	17	de aplicação do processo de urgência	to urgent procedure
	$\bar{x} = 1.0086$		Total: 285 913	

Tabela 5.9: Alguns dos exemplos (3:1) mais ocorrentes extraídos do EuroParl PT:EN com base na Hipótese das Palavras-Marca.

Um pré-processamento adequado poderia resolver vários dos problemas, como sejam a utilização de determinadas palavras-marca no fim dos segmentos (e não no início como é defendido na hipótese das palavras-marca) ou mesmo a utilização de determinada pontuação como os parêntesis que não funcionam como marcas convencionais. Do mesmo modo, alguns dos problemas encontrados podem ser minorados com um pós-processador que rejeite grande parte dos pares errados.

5.2 Extracção Combinatória de Exemplos

O principal algoritmo (uma abordagem semelhante é descrita em (Melamed, 2001)) usado para extrair exemplos e que foi um dos pontos centrais desta dissertação usa apenas o conhecimento de dicionários probabilísticos de tradução para o alinhamento de unidades de tradução ao nível do segmento[3].

Definição 8 *Dados textos paralelos U e V alinhados à frase (um conjunto de pares ordenados (u_i, v_i), em que u_i e v_i são traduções mútuas), um* ***alinhamento ao segmento*** *é uma segmentação de u_i e v_i em n segmentos cada, tal que para cada j, $1 \leq j \leq n$, u_{ij} e v_{ij} são traduções mútuas.*

O algoritmo aqui apresentado tira partido especialmente do facto de que as línguas ocidentais se escrevem da esquerda para a direita, e de que a tradução de texto técnico é habitualmente linear. Portanto, é de esperar que numa unidade de tradução $(s_{\mathcal{A}}, s_{\mathcal{B}})$, a distância entre o início de $s_{\mathcal{A}}$ e determinada palavra $w_{\mathcal{A}}$ seja muito semelhante à distância

[3]O que na literatura é habitualmente designado por *alinhamento à palavra* (ou *word alignment* (Melamed, 2000)) será aqui chamado de *Alinhamento ao Segmento*. É certo que o termo de alinhamento à palavra é amplamente conhecido, e que o uso de terminologia diferente pode levantar algumas confusões. No entanto, parece-nos preferível correr esse risco, dando preferência à enfase de que realmente não se conseguem definir relacionamentos entre todas e cada uma das palavras de uma frase, mas sim relacionamentos entre sequências de palavras.

entre o início de $s_{\mathcal{B}}$ e a sua tradução $w_{\mathcal{B}}$. Ou seja, se construirmos uma matriz (Carl, 2001) em que colocamos em cada linha uma palavra $w_{\mathcal{A}i}$ de $s_{\mathcal{A}}$, em cada coluna uma palavra $w_{\mathcal{B}j}$ de $s_{\mathcal{B}}$, e em cada célula (i, j) a probabilidade de tradução mútua de $w_{\mathcal{A}i}$ por $w_{\mathcal{B}i}$, obteremos uma matriz em que as células que correspondem a traduções correctas terão valores elevados. O algoritmo usa esta assunção para extrair relacionamentos entre segmentos.

O algoritmo pode ser aplicado a qualquer unidade de tradução, seja ela pertencente ou não ao corpus que deu origem ao PTD usado. No entanto, a qualidade do alinhamento obtido é muito dependente do conhecimento que o dicionário tem em relação às palavras de cada unidade de tradução processada.

5.2.1 Matriz de Alinhamento

O processo de criação da matriz de alinhamento já descrito sucintamente, é agora detalhado para uma unidade de tradução $(s_{\mathcal{A}}, s_{\mathcal{B}})$. A figura 5.1 mostra uma exemplo de uma matriz de alinhamento (correspondente ao segundo passo do algoritmo).

	discussion	about	alternative	sources	of	financing	for	the	european	radical	alliance	.
discussão	**44**	0	0	0	0	0	0	0	0	0	0	0
sobre	0	**11**	0	0	0	0	0	0	0	0	0	0
fontes	0	0	0	**74**	0	0	0	0	0	0	0	0
de	0	3	0	0	**27**	0	6	3	0	0	0	0
financiamento	0	0	0	0	0	**56**	0	0	0	0	0	0
alternativas	0	0	**23**	0	0	0	0	0	0	0	0	0
para	0	0	0	0	0	0	**28**	0	0	0	0	0
a	0	1	0	0	1	0	4	**33**	0	0	0	0
aliança	0	0	0	0	0	0	0	0	0	0	**65**	0
radical	0	0	0	0	0	0	0	0	0	**80**	0	0
europeia	0	0	0	0	0	0	0	0	**59**	0	0	0
.	0	0	0	0	0	0	0	0	0	0	0	**80**

Figura 5.1: Matriz de alinhamento depois de preenchida.

As dimensões da matriz correspondem ao número de palavras da frase $s_{\mathcal{A}}$ e da frase $s_{\mathcal{B}}$. Ou seja, cada um dos índices i e j de uma célula

$M_{i,j}$ da matriz corresponde a uma palavra.

O processo de preenchimento da matriz de tradução e de extracção de exemplos segue os seguintes passos:

1. Cada célula $M_{i,j}$ da matriz é **preenchida com a probabilidade de tradução** mútua entre $w_{\mathcal{A}i}$ e $w_{\mathcal{B}j}$, calculada com:

$$\frac{\mathcal{P}\left(w_{\mathcal{A}i} \in \mathcal{T}_{d_{\mathcal{B},\mathcal{A}}}\left(w_{\mathcal{B}j}\right)\right) + \mathcal{P}\left(w_{\mathcal{B}j} \in \mathcal{T}_{d_{\mathcal{A},\mathcal{B}}}\left(w_{\mathcal{A}i}\right)\right)}{2}$$

2. Quando se realiza o alinhamento ao segmento de uma unidade de tradução usando um PTD que não o obtido a partir do corpus que a contém, irão aparecer palavras novas (Lei de Zipf). Muitas dessas palavras acabam por ser nomes próprios (ou entidades numéricas) que não são traduzidas entre línguas. Por isso, a todas as **palavras escritas da mesma forma em ambas as línguas** (palavras com mais de três caracteres), é dada uma probabilidade de 80%.
3. Como já foi discutido, como as línguas com que estamos a trabalhar são ocidentais e escritas da esquerda para a direita, podemos assumir que as traduções correctas se encontram perto da diagonal principal. Para que estas traduções tenham probabilidades mais elevadas é usado um **algoritmo de suavização dos valores**, que diminui os valores de acordo com a sua distância à diagonal principal.
4. A parte mais importante do algoritmo é a pesquisa da *diagonal de tradução* correspondente às células de traduções correctas. Este passo do algoritmo começa na primeira célula da matriz, tentando chegar à do canto inferior direito, passando pelo maior número de células com probabilidades altas.
   A diagonal de tradução não é necessariamente a diagonal principal[4] da matriz, já que é normal (como se viu no exemplo) que algumas palavras, ou mesmo segmentos grandes, mudem de ordem.
   Para encontrar a diagonal, o algoritmo baseia-se na **definição de pontos âncora**. Um ponto $x_{i,j}$ é considerado um ponto âncora se

[4] Alias, raramente a matriz é quadrada.

o seu valor é 20% superior a todos os outros elementos na coluna i e na linha j.
Quando nenhum ponto âncora é encontrado o algoritmo procede aumentando uma área rectangular, linha a linha, e coluna a coluna, até encontrar um ponto âncora, definindo **blocos de tradução**. Estes blocos incluem nos seus cantos (superior esquerdo, e inferior direito) um ponto âncora, excepto se corresponderem ao início ou fim da frase.
A figura 5.2 mostra o resultado de aplicar este método ao exemplo anterior.

	discussion	about	alternative	sources	of	financing	for	the	european	radical	alliance	.
discussão	**44**	0	0	0	0	0	0	0	0	0	0	0
sobre	0	**11**	0	0	0	0	0	0	0	0	0	0
fontes	0	0	0	**74**	0	0	0	0	0	0	0	0
de	0	3	0	0	**27**	0	6	3	0	0	0	0
financiamento	0	0	0	0	0	**56**	0	0	0	0	0	0
alternativas	0	0	**23**	0	0	0	0	0	0	0	0	0
para	0	0	0	0	0	0	**28**	0	0	0	0	0
a	0	1	0	0	1	0	4	**33**	0	0	0	0
aliança	0	0	0	0	0	0	0	0	0	0	**65**	0
radical	0	0	0	0	0	0	0	0	0	**80**	0	0
europeia	0	0	0	0	0	0	0	0	**59**	0	0	0
.	0	0	0	0	0	0	0	0	0	0	0	**80**

Figura 5.2: Matriz final de alinhamento ao segmento.

A partir da matriz apresentada na figura 5.2, é possível extrair relacionamentos bilingues:

```
1  discussão                          discussion
2  sobre fontes                       about alternative sources
3  de                                 of
4  financiamento alternativas para    financing for
5  a aliança radical                  the european radical
6  radical europeia .                 radical alliance .
```

Como se pode ver no exemplo, este algoritmo tem alguns problemas:

- existem várias traduções com níveis de confiança demasiado baixos (este problema só poderá ser resolvido com a criação de um PTD melhor);
- algumas das âncoras definidas não são aproveitadas, o que mostra que o algoritmo não está a encontrar a diagonal de tradução correcta;
- existem mudanças na ordem durante a tradução, o que leva a que não exista sempre uma diagonal de tradução nítida. A solução para este problema passa pelo uso de uma linguagem de definição de padrões de tradução, para especificar mudanças de ordem sistemáticas, como será discutido na secção 5.3.

Considerando os exemplos obtidos "*financiamento para*" e "*financing for*," é possível verificar que correspondem à concatenação de *sintagmas incompletos* (não seguem as fronteiras clássicas das árvores de *parsing*: fronteiras linguísticas), ao contrário dos obtidos pela hipótese das palavras-marca. No entanto, isto não implica a falta de qualidade ou usabilidade dos exemplos aqui obtidos. O facto de se obterem exemplos mais pequenos permite a sua maior reutilização (não existe um conceito de "*fronteira ideal*" para exemplos de tradução).

5.2.2 Combinação de Exemplos

A extracção de exemplos apresentada anteriormente encontrou alguns relacionamentos que já eram conhecidos: pertencentes ao PTD. Ou seja, todas as âncoras simples são resultado de conhecimento prévio contido no dicionário.

A existência de informação sobre as palavras soltas é importante, mas não traz nada de novo. Considerando de forma independente os exemplos (não necessariamente correctos) "*discussão/discussion*" e "*sobre fontes/about alternative sources*," não temos informação sobre como se compõem durante a tradução.

A solução proposta é a criação artificial de exemplos usando combinatória sobre os exemplos extraídos (Simões and Almeida, 2006a). Ou seja, se concatenarmos os dois primeiros exemplos, obtemos um novo

exemplo com mais informação que o anterior. Se concatenarmos o obtido com o seguinte, obtemos um exemplo ainda mais rico. Se continuarmos a concatenar, chegamos à unidade de tradução original, pelo que este método pode parecer um retroceder na extracção realizada.

No entanto, interessa-nos armazenar todas as combinações, de todos os níveis. Assim, obtemos exemplos com diferentes granularidades e com contextos de diferentes tamanhos. Ou seja, para além dos pares extraídos directamente, podemos construir de forma combinatória todos os possíveis pares: Por exemplo:

```
1 discussão sobre fontes - discussion about alternative sources
2 sobre fontes de - about alternative sources of
3 de financiamento alternativas para - of financing for
4 financiamento alternativas para a aliança radical - financ...
5 a aliança radical europeia - the european radical alliance
```

Estes pares podem voltar a ser concatenados, construindo um conjunto com exemplos de tradução ainda maiores. Uma abordagem semelhante seria o armazenamento da matriz de alinhamento, para que em tempo de execução os exemplos pudessem ser calculados dinamicamente.

O armazenamento de todos estes exemplos é importante: uma vez que a tradução é realizada procurando-se inicialmente exemplos maiores, e caminhando para exemplos mais pequenos. Sempre que possível o exemplo maior e com maior contexto (e portanto, maior confiança) será usado.

Depois de extraídos todos os exemplos, são ordenados e contados. Estes exemplos constituem um tipo de dicionário de tradução ao nível do segmento. Para cada segmento na língua $\mathcal{A}$, são calculadas todas as traduções na língua $\mathcal{B}$ e o respectivo número de ocorrências:

$$\mathrm{S}_{\mathcal{A}} \rightharpoonup (\mathrm{S}_{\mathcal{B}} \rightharpoonup \mathbb{N})$$

Este dicionário tem o seguinte aspecto[5]:

[5]Os exemplos aqui apresentados são extraídos do EuroParl PT:ES. A razão da escolha da língua espanhola em favor da língua inglesa prende-se com o facto de

```
é certo que
          es cierto que (25)
          es verdad que (6)
          cierto es que (2)
          es evidente que (2)

todos os problemas
          todos los problemas (18)
          problemas (1)

nórdica verde
          verde nórdica (13)

confederal da esquerda unitária europeia
          confederal de la izquierda unitaria europea (11)
          confederal de la izquierda unitaria europa  (1)
```

O número de ocorrências permite concluir sobre a confiança das traduções. Esta medida pode ainda ser fortalecida com o cálculo da qualidade de tradução com base num PTD. Esta medida de confiança é imprescindível para que um sistema de tradução automática possa decidir sobre que exemplo aplicar.

5.2.3 Discussão de Resultados

Como se pode ver na matriz 5.1, a tradução pode envolver a troca de ordem de palavras. Embora estas trocas possam ser realizadas de livre vontade por um tradutor, há outras que são impostas pela sintaxe das línguas envolvidas.

Dado que estas regras estão directamente relacionadas com a sintaxe das línguas, é imprescindível que o algoritmo de extracção de exemplos tenha essas trocas em consideração. Deste modo, foi definida uma linguagem para a especificação de padrões de alinhamento que será apresentada na próxima secção.

existirem muitas trocas de ordem entre palavras na tradução entre português e inglês. Este facto motivou a definição de padrões de tradução que serão apresentados na próxima secção, onde se voltará a apresentar exemplos PT:EN.

A avaliação de resultados será realizada sobre o algoritmo completo, incluindo a manipulação de padrões de tradução (secção 5.3.4).

5.3 Extracção com base em Padrões de Alinhamento

Como foi explicado na secção anterior, a tradução entre duas línguas nem sempre preserva a ordem das palavras. Embora se considere que a tradução técnica é quase sempre realizada literalmente, existem regras gramaticais que obrigam a que algumas palavras troquem de ordem durante a tradução.

O exemplo típico destas regras gramaticais é a troca de ordem entre substantivo e adjectivo na tradução entre português ou espanhol e inglês: enquanto que em português o adjectivo segue o substantivo, em inglês o adjectivo precede o substantivo. Esta regra, bem como outras semelhantes, podem ser formalizadas de modo a que o algoritmo de extracção de exemplos as possa ter em consideração.

Esta secção discute uma linguagem de domínio específico (DSL) a que chamamos *Linguagem de Descrição de Padrões — Pattern Description Language* (PDL). Esta linguagem permite a especificação numa sintaxe legível mas compacta dos padrões de tradução. O uso de linguagens para a detecção e extracção de recursos não é novo. Por exemplo, (Sánchez-Martínez and Ney, 2006) e (Sánchez-Martínez and Forcada, 2007) usam padrões para inferir regras de tradução.

A PDL é uma linguagem simples, com uma notação formal (secção 5.3.1). Esta linguagem especifica de que forma as palavras trocam de ordem, e é com base nesta especificação que o compilador constrói uma matriz padrão que será usada durante o processo de alinhamento.

Esta linguagem mostrou-se útil não só para ajudar o algoritmo de extracção de exemplos, mas também como uma ferramenta por si só para a extracção de terminologia bilingue.

5.3.1 Linguagem de Descrição de Padrões

A linguagem para descrição de padrões de alinhamento foi desenhada com a preocupação de ser compacta mas simples de ler e interpretar. Optamos por apresentar a linguagem partindo de exemplos simples, e apresentando gradualmente a motivação para as várias funcionalidades que a linguagem incorpora.

Padrões Simples

Na sua forma mais simples, um padrão de alinhamento é um triplo: o nome do padrão, a ordem das palavras na língua $\mathcal{A}$, e a ordem das palavras na língua $\mathcal{B}$. Para que as regras sejam genéricas, não explicitam a ordem de palavras específicas, mas a ordem de buracos ou variáveis (*place-holders*) que são substituídos por palavras.

A noção de padrão de alinhamento fica mais clara com alguns exemplos. Consideremos inicialmente a definição do padrão de troca de ordem entre substantivo e adjectivo. Este padrão pretende especificar que duas palavras, A e B, numa língua, terão as suas traduções pela ordem inversa. Ou seja, que

$$\mathcal{T}(A \cdot B) = \mathcal{T}(B) \cdot \mathcal{T}(A)$$

Para simplificar esta notação optamos por remover a função de tradução, e adicionar antes da regra o seu identificador, entre parêntesis rectos:

```
[ABBA] A B = B A
```

Esta regra corresponde à matriz padrão representada na tabela 5.10.

Este padrão é procurado na matriz de alinhamento que foi apresentada na secção anterior. Cada um dos X corresponde a uma célula com um valor alto: uma âncora. As restantes células têm de conter um valor próximo de zero para que o padrão possa ser aplicado.

As tabelas 5.11 a 5.14 mostram quatro padrões bastante comuns na tradução entre português e inglês[6].

[6]Embora o identificador de regra possa ser qualquer sequência de caracteres,

	Olimpic	Games
Jogos		X
Olímpicos	X	

`[ABBA] A B = B A`

Tabela 5.10: Padrão de Alinhamento ABBA.

Padrões Instanciados

A linguagem de padrões permite, para além do uso de variáveis, o uso de palavras específicas que têm de existir para que a regra possa ser aplicada. Os exemplos anteriores foram apresentados na sua forma simplificada, já que deviam contemplar todas as variantes do uso da preposição e artigo. Por exemplo, o padrão para a regra HDI deveria ser:

```
[HDI] I "de"|"da"|"do"|"dos"|"das" D H = H D I
```

Além deste pormenor da linguagem, existe um Δ numa das tabelas, que corresponde a uma célula que pode ter qualquer probabilidade (uma vez que o X obriga a uma probabilidade alta e a inexistência de um símbolo obriga a uma probabilidade baixa). Estas células têm habitualmente valores baixos já que correspondem a relações entre palavras pertencentes a classes fechadas, mas não são fáceis de prever, pelo que se optou pela definição das relações do tipo Δ.

Integração no Algoritmo

Os padrões são definidos pelo utilizador num ficheiro de texto que é passado como parâmetro ao extractor de exemplos. O ficheiro é compilado, e os padrões são aplicados[7] no algoritmo apresentado na secção 5.2.1,

optou-se por usar um exemplo paradigmático que recorde a regra em causa.

[7]Os padrões são procurados na ordem pela qual foram definidos. Deste modo, se para determinada situação existem dois possíveis padrões e não há uma ordenação

	Human	Rights
Direitos		X
do		
Homem	X	

```
[HR] A "de" B = B A
```

Tabela 5.11: Padrão de Alinhamento HR.

	neutral	point	of	view
ponto		X		
de			Δ	
vista				X
neutro	X			

```
[POV] P "de"V N = N P "of"V
```

Tabela 5.12: Padrão de Alinhamento POV.

	file	transfer	protocol
protocolo			X
de			
transferência		X	
de			
ficheiros	X		

```
[FTP] P "de" T "de" F = F T P
```

Tabela 5.13: Padrão de Alinhamento FTP.

	human	develop ment	index
índice			X
de			
desenvolvimento		X	
humano	X		

```
[HDI] I "de" D H = H D I
```

Tabela 5.14: Padrão de Alinhamento HDI.

	discussion	about	alternative	sources	of	financing	for	the	european	radical	alliance	.
discussão	**44**	0	0	0	0	0	0	0	0	0	0	0
sobre	0	**11**	0	0	0	0	0	0	0	0	0	0
fontes	0	0	0	**74**	0	0	0	0	0	0	0	0
de	0	3	0	0	**27**	0	6	3	0	0	0	0
financiamento	0	0	0	0	0	**56**	0	0	0	0	0	0
alternativas	0	0	**23**	0	0	0	0	0	0	0	0	0
para	0	0	0	0	0	0	**28**	0	0	0	0	0
a	0	1	0	0	1	0	4	**33**	0	0	0	0
aliança	0	0	0	0	0	0	0	0	0	0	**65**	0
radical	0	0	0	0	0	0	0	0	0	**80**	0	0
europeia	0	0	0	0	0	0	0	0	**59**	0	0	0
.	0	0	0	0	0	0	0	0	0	0	0	**80**

Figura 5.3: Matriz de alinhamento usando padrões.

entre o terceiro e quarto passo. Ou seja, depois de marcadas as âncoras, e da matriz ser suavizada de acordo com a distância entre células e a diagonal de tradução. Os padrões são aplicados obrigando a que cada célula com um X no padrão corresponda a uma âncora. Depois de aplicado, todo o rectângulo do padrão é transformado numa âncora para a etapa seguinte.

A figura 5.3 mostra o exemplo da secção anterior utilizando padrões. As duas zonas em que os elementos âncora fogem da diagonal principal correspondem a padrões, e por isso, todo o bloco deve ser considerado como uma única âncora. Os exemplos extraídos desta matriz são bastante mais interessantes do que os extraídos sem o uso de padrões:

```
1  discussion                          | discussão
2  about                               | sobre
3  alternative sources of financing    | fontes de financ. alternativas
4  for                                 | para
5  the                                 | a
6  european radical alliance           | aliança radical europeia
```

Estes exemplos também são concatenados combinatoriamente, tal como defendido previamente.

fácil dos mesmos, a solução passará por definir predicados (secção 5.3.2) sobre os padrões, para limitar a sua aplicabilidade.

5.3.2 Restrições sobre Padrões de Alinhamento

A PDL, tal como foi apresentada, é útil mas pouco configurável. É importante adicionar restrições à aplicabilidade de uma regra, de acordo com propriedades das palavras em causa.

Por exemplo, o padrão ABBA é aplicado correctamente em 90% das situações, mas por vezes é aplicado em situações que nada têm que ver com a troca entre substantivo e adjectivo. Nestes casos, uma restrição sobre a categoria morfológica das palavras que fazem *matching* com as variáveis permite que o padrão seja aplicado correctamente em 99% das situações.

A PDL foi expandida para suportar predicados sobre variáveis ou zonas de regras de acordo com:

- *predicados genéricos*, que permitem restringir a aplicabilidade do padrão de acordo com um conjunto de predicados definidos em Perl;
- *predicados morfológicos*, que permitem restringir a aplicabilidade do padrão de acordo com um conjunto de restrições sobre as categorias e propriedades morfológicas das palavras em causa;
- *predicados para inferência* que permitem inferir propriedades a partir de corpora.

Predicados Genéricos

Foram adicionados predicados genéricos sobre variáveis ou zonas de padrões. Estes predicados são definidos como funções Perl sobre as palavras em causa. Estas funções recebem uma sequência de palavras (de acordo com a zona afecta ao predicado) e retornam um valor booleano indicativo da sua validade.

Para permitir a definição de predicados em Perl, e seguindo uma abordagem semelhante à usada no Lex e Yacc, foi definida uma zona na qual o utilizador deve implementar os predicados. Estes predicados são definidos no fim do ficheiro de regras, sendo precedidos por um separador

(dois símbolos de percentagem), de acordo com o seguinte exemplo[8]:

```
1        [ABBA]  A B.not_comma = B.not_comma A

2        %%
3        sub not_comma {
4            my $word = shift;
5            return $word != ','
6        }
```

Antes de aplicar o padrão, o interpretador irá invocar o predicado sobre a palavra no lugar da variável, e apenas se o predicado retornar um valor verdadeiro é que o padrão será aplicado.

> O uso da linguagem Perl para a definição de predicados permite que se possam executar todo o tipo de validações, incluindo acessos a bases de dados ou aplicações externas.

Predicados Morfológicos

As restrições mais típicas correspondem à definição de que categorias (adjectivo, substantivo, etc.) ou propriedades (género, número, etc.) morfológicas as palavras devem ter para que determinado padrão possa ser aplicado. Para facilitar a escrita deste tipo de predicados, a PDL foi enriquecida com açúcar sintáctico:

```
[ABBA] A B[CAT<-adj] = B[CAT<-adj] A
```

Ou seja, cada variável pode ser seguida de um conjunto de restrições entre parêntesis rectos. Estas restrições são compostas por uma chave (nome da categoria ou propriedade morfológica) e o valor requerido para que o padrão possa ser aplicado.

[8] Note-se que este é um exemplo muito simples, apenas para ilustração da sintaxe da linguagem.

Note-se que para que estas regras funcionem é preciso ter acesso a um analisador morfológico. No caso das nossas experiências com a língua portuguesa e inglesa foi usado o analisado morfológico jSpell (Almeida and Pinto, 1994).

> Embora os predicados genéricos permitam a escrita de restrições sobre propriedades morfológicas, a integração destas restrições na própria linguagem permite que se possam escrever de forma mais legível.

Predicados para Inferência

Para além das restrições na aplicação de regras, chegou-se à conclusão que estas mesmas regras podiam ser usadas com alguma segurança para a inferência de propriedades sobre palavras.

Consideremos de novo o exemplo anterior da regra ABBA:

```
[ABBA] A B[CAT<-adj] = B[CAT<-adj] A
```

Sempre que esta regra for aplicada, estamos à espera que as palavras que façam *matching* com a variável A sejam substantivos. É, então, possível definir uma regra de modo a inferir um dicionário de substantivos, extraindo todas as palavras encontradas na posição A:

```
[ABBA] A[CAT->n] B[CAT<-adj] = B[CAT<-adj] A[CAT->n]
```

Deste modo as regras podem ser usadas para enriquecer dicionários morfológicos com alguma facilidade. Ou seja, a lista de palavras extraídas na posição A será catalogada com a categoria morfológica inferida: *nome*.

> Os padrões de tradução podem ser usados para outras tarefas que não as originalmente pensadas, nomeadamente para o enriquecimento de dicionários morfológicos.

39214	comunidades europeias	european communities
32850	jornal oficial	official journal
32832	parlamento europeu	european parliament
32730	união europeia	european union
15602	países terceiros	third countries
[...]	[...]	[...]
1	órgãos orçamentais	budgetary organs
1	órgãos relevantes	relevant bodies
1	óvulos de equino	equine ova
1	óxido de cádmio	cadmium oxide
1	óxido de estireno	styrene oxide

Tabela 5.15: Extracto das contagens de unidades nominais.

5.3.3 Extracção de Segmentos Nominais

As regras apresentadas (que foram definidas originalmente com o intuito de melhorar o algoritmo de extracção de exemplos) correspondem, na sua maioria, a componentes nominais adjectivados ou a sintagmas nominais seguidos de sintagmas preposicionais (frases nominais sem o determinante). Ao extrair estes segmentos nominais a partir de corpora paralelos técnicos, as instâncias encontradas são, na sua maioria, boas candidatas para incorporarem uma base terminológica, pelo que a sua extracção e análise é bastante importante.

Durante o processo de extracção de exemplos apresentado previamente, todos os segmentos bilingues que estão de acordo com um padrão são anotados com o identificador do padrão aplicado. Após o processamento de todo um corpus é possível obter uma lista de entradas terminológicas bilingues que podem ser ordenadas e acumuladas, de forma a obter informação estatística sobre a sua confiança.

A tabela 5.15 apresenta algumas das entradas mais e menos ocorrentes, extraídas do corpus EuroParl PT:EN. Numa visão superficial, salienta-se a qualidade quer dos elementos mais ocorrentes, quer dos menos ocorrentes (uma avaliação mais cuidadosa será apresentada em 5.3.4).

As tabelas 5.16 a 5.22 correspondem às 15 entradas mais ocorrentes, para diferentes padrões (sem uso de restrições morfológicas). Nestas

32832	parlamento europeu	european parliament
32730	união europeia	european union
4171	direitos humanos	human rights
3504	estados unidos	united states
2353	mercado interno	internal market
1911	posição comum	common position
1826	países candidatos	candidate countries
1776	comissão europeia	european commission
1708	conselho europeu	european council
1629	saúde pública	public health
1558	direitos fundamentais	fundamental rights
1546	nações unidas	united nations
1337	países terceiros	third countries
1294	conferência intergovernamental	intergovernmental conference
1258	fundos estruturais	structural funds

Tabela 5.16: Extracto de unidades nominais (`A B = B A`).

tabelas, as entradas com tradução correcta mas que não podem ser consideradas como entradas nominais estão marcadas com um $\diamond$. Por sua vez, aquelas entradas com tradução incorrecta estão marcadas com $\star$.

Em relação aos resultados obtidos nestas tabelas, salientamos que alguns dos maus resultados podiam ser facilmente corrigidos usando um predicado genérico que não permitisse, por exemplo, a aplicação do padrão a palavras pertencentes ao conjunto das *palavras-marca*.

5.3.4 Avaliação de Resultados

Ao ter uma taxa de correcção elevada, os padrões permitem formar âncoras de excelente qualidade, levando a um substancial melhoramento do algoritmo de extracção combinatória de exemplos da secção 5.2.

Esta secção pretende avaliar os padrões como método de extracção de unidades nominais.

Para a avaliação das unidades nominais extraídas foram processadas cerca de 700 000 unidades de tradução do EuroParl PT:EN. Depois de

729	plano de acção	action plan
722	conselho de segurança	security council
680	processo de paz	peace process
582	mercado de trabalho	labour market
580	pena de morte	death penalty
492	pacto de estabilidade	stability pact
431	política de defesa	defence policy
353	acordo de associação	association agreement
348	protocolo de quioto	kyoto protocol
343	programa de acção	action programme
259	branqueamento de capitais	money laundering
258	comité de conciliação	conciliation committee
241	política de concorrência	competition policy
226	processo de conciliação	conciliation procedure
217	requerentes de asilo	asylum seekers

Tabela 5.17: Extracto de unidades nominais (`A "de" B = B A`).

531	política agrícola comum	common agricultural policy
418	banco central europeu	european central bank
329	tribunal penal internacional	international criminal court
166	aliança livre europeia	european free alliance
156	modelo social europeu	european social model
153	partidos políticos europeus	european political parties
83	fundo monetário internacional	international monetary fund
75	política externa comum	common foreign policy
66	organização marítima internacional	international maritime organisation
65	◇ própria união europeia	european union itself
65	fundo social europeu	european social fund
55	direitos humanos fundamentais	fundamental human rights
45	relações económicas externas	external economic relations
45	◇ homens e mulheres	women and men
45	agência espacial europeia	european space agency

Tabela 5.18: Extracto de unidades nominais (`A B C = C B A`).

95	mandato de captura europeu	european arrest warrant
85	fontes de energia renováveis	renewable energy sources
80	mandado de captura europeu	european arrest warrant
67	sistemas de segurança social	social security systems
64	zona de comércio livre	free trade area
55	força de reacção rápida	rapid reaction force
54	orientações de política económica	economic policy guidelines
46	planos de acção nacionais	national action plans
46	direitos de propriedade intelectual	intellectual property rights
33	sistema de alerta rápido	rapid alert system
29	política de defesa comum	common defence policy
29	método de coordenação aberta	open coordination method
27	método de coordenação aberto	open coordination method
27	conselho de empresa europeu	european works council
25	acordo de comércio livre	free trade agreement

Tabela 5.19: Extracto de unidades nominais (`I "de" D H = H D I`).

39	◇	penso que não	not think that
12	◇	penso que não	not believe that
12	◇	creio que não	not think that
10	⋆	dia a discussão	debate on the
8		primeiro passo importante	important first step
7	◇	mais importante ainda	even more important
6		supremo tribunal espanhol	spanish supreme court
6	◇	nem sempre foram	were not always
5	◇	são necessárias reformas	reforms are needed
5	⋆	países em desenvolvimento	developing countries in
5	⋆	dotações para pagamentos	payment appropriations for
5	⋆	comigo e com	with me and
4	◇	são tomadas decisões	decisions are taken
4	◇	sejam tomadas medidas	measures are taken
4	◇	penso que também	also believe that

Tabela 5.20: Extracto de unidades nominais (`A B C = C A B`).

93	tribunal de justiça europeu	european court of justice
51	tribunal de contas europeu	european court of auditors
33	fontes de energia renováveis	renewable sources of energy
27	ponto de vista ambiental	environmental point of view
26	ponto de vista económico	economic point of view
21	ponto de vista jurídico	legal point of view
20	declaração de fiabilidade positiva	positive statement of assurance
18	ponto de vista político	political point of view
13	ponto de vista técnico	technical point of view
10	ponto de vista institucional	institutional point of view
9	ponto de vista orçamental	budgetary point of view
8	sistema de preferências generalizadas	generalised system of preferences
8	método de coordenação aberto	open method of coordination
7	ponto de vista social	social point of view
7	ponto de vista democrático	democratic point of view

Tabela 5.21: Extracto de unidades nominais (`P de V N = N P of V`).

41	emissões de dióxido de carbono	carbon dioxide emissions
22	sistema de informação de schengen	schengen information system
8	sistema de comércio de emissões	emissions trading system
8	plano de acção de viena	vienna action plan
8	cartão de prestação de serviços	service provision card
8	agenda de desenvolvimento de doha	doha development agenda
7	política de espectro de radiofrequências	radio spectrum policy
6	sistema de transporte de mercadorias	freight transport system
6	dispositivos de limitação de velocidade	speed limitation devices
5	plataforma de acção de pequim	beijing action platform
5	operações de gestão de crises	crisis management operations
5	critérios de convergência de maastricht	maastricht convergence criteria
4	política de mercado de trabalho	labour market policy
4	normas de protecção de dados	data protection rules
4	⋆ grupo de trabalho de alto	high-level working group

Tabela 5.22: Extracto de unidades nominais (`P de T de F = F T P`).

calculadas as unidades nominais, e de consolidados os resultados, foram obtidas 139 781 unidades diferentes. A avaliação destas unidades foi feita separadamente por cada padrão, de forma a se poder medir quais os padrões mais produtivos e com maior qualidade[9].

Padrão	Total	Máx.	Mediana	Min.	Precisão
A B = B A	77 497	938	2	1	86 %
A "de" B = B A	12 694	204	2	1	95 %
A B C = C B A	7 700	40	1	1	93 %
I "de" D H = H D I	3 336	21	1	1	100 %
A B C = C A B	1 466	4	1	1	40 %
P "de" V N = N P "of" V	564	6	1	1	98 %
P "de" T "de" F = F T P	360	3	1	1	96 %

Tabela 5.23: Avaliação de unidades nominais extraídas.

A tabela 5.23 sintetiza os resultados obtidos. Para cada padrão, foram criados três conjuntos para análise, cada um com o tamanho de 20 unidades nominais. Estes três conjuntos são constituídos pelas 20 unidades mais ocorrentes, as 20 menos ocorrentes, e um outro conjunto de 20 unidades retiradas do centro da lista. A coluna "máximo" corresponde ao número mínimo de ocorrências do conjunto de 20 unidades mais ocorrentes. A coluna "mínimo" corresponde ao número mínimo de ocorrências do conjunto de 20 unidades menos ocorrentes. Por sua vez, a coluna "mediana" corresponde ao número mínimo de ocorrências do conjunto de 20 unidades retirado do centro da lista.

É importante salientar que o conjunto das unidades menos ocorrentes bem como o conjunto de unidades retiradas do centro da lista, têm um número de ocorrências extremamente baixo, pelo que o teste é especialmente desfavorável. No entanto, a generalidade dos padrões tiveram resultados acima dos 90%. Na avaliação só foram consideradas entradas correctas aquelas que, além de serem traduções mútuas, também correspondiam a unidades nominais.

[9]Os padrões usados nesta avaliação são exactamente os apresentados na tabela (sem variações de contracções de preposições com artigos nem com predicados morfológicos).

O uso da Pattern Description Language permite a extracção de terminologia bilingue de grande qualidade.

5.4 Generalização

A *generalização* (Brown, 2001) é uma abordagem crucial para aumentar a aplicabilidade de exemplos de tradução. Consiste na substituição de palavras num exemplo de tradução por variáveis tipadas. O exemplo paramétrico obtido é uma regra de tradução que permite a tradução de frases semelhantes à que lhe deu origem, mas em que as únicas diferenças são as palavras na posição de variáveis. No entanto, se a palavra corresponder ao tipo da variável, o exemplo de tradução pode ser aplicado, sendo necessário apenas a posterior tradução da palavra em causa. Como exemplo, consideremos a unidade de tradução:

eu vi um porco gordo.
I saw a fat pig.

Se existir uma classe de animais[10] é possível criar o exemplo paramétrico de tradução:

eu vi um $\{A.animal\}$ *gordo.*
I saw a fat $\{T(A.animal)\}$.

Com este exemplo torna-se possível a tradução de novas frases, como "*eu vi um gato gordo*", frase essa que não precisa de existir como exemplo de tradução. Para a tradução desta frase pode ser aplicada a regra anterior, e gerada a tradução: "*I saw a fat* $\{\mathcal{T}(gato)\}$". Consultando um dicionário externo é possível terminar a tradução: "*I saw a fat cat*".

A generalização pode ser vista como duas funções3 independentes:

[10]Possivelmente haveria interesse em diferenciar animais de acordo com o seu género.

- uma função de detecção de determinado tipo de objecto (a que chamaremos de classe). Por exemplo, um detector de URLs, entidades mencionadas, valores, datas, horas, ou então palavras pertencentes a um conjunto pré-definido (como cores, animais, etc.).
- uma função de tradução de objectos dessa classe para a língua de destino. Esta função pode ser tão simples como a função identidade (para entidades que não se traduzem), funções matemáticas (como a conversão de medidas entre unidades imperiais e unidades métricas) ou funções de tradução com base num dicionário bilingue.

Esta secção apresenta a criação de regras para três tipos de classes: não textuais (números, datas, horas, valores monetários, URL, email, etc), entidades mencionadas, e palavras comuns.

> A detecção de classes bilingues é imprescindível para a generalização de exemplos de tradução.

5.4.1 Classes Não Textuais

A forma mais simples de generalizar é a substituição de entidades não textuais por classes. Uma determinada frase é válida com qualquer ano, ou valor monetário. Basta a substituição do número para se obter uma tradução correcta.

No entanto, é importante a definição de classes diferentes para os vários tipos de valores. A experiência realizada com base na terminologia extraída pela PDL levou à criação das seguintes classes não textuais: anos, datas, horas, valores monetários, URLs, e-mails, inteiros e decimais. Seguem-se alguns exemplos dos resultados obtidos para as classes:

- **horaA:** `\d{2}h\d{2}`
- **horaB:** `\d{2}:\d{2}`
- **ano:** `\d{4}`
- **int:** `\d+`

399	às { *horaA* }	{ *horaB* }
187	orçamento de { *ano* }	{ *ano* } budget
136	{ *int* } euros	eur { *int* }
127	directiva de { *ano* }	{ *ano* } directive
51	orçamento { *ano* }	{ *ano* } budget
46	{ *int* } de setembro	september { *int* }
31	partir de { *ano* }	{ *ano* } onwards
29	convenção de { *ano* }	{ *ano* } convention
26	eleições de { *ano* }	{ *ano* } elections
25	período { *ano* }-{ *ano* }	{ *ano* }-{ *ano* } period
25	{ *int* } dólares	usd { *int* }
24	relatório de { *ano* }	{ *ano* } report
21	convenção de genebra de { *ano* }	{ *ano* } geneva convention
17	período de { *ano* }-{ *ano* }	{ *ano* }-{ *ano* } period

Tabela 5.24: Extracto de regras nominais generalizadas usando classes não textuais.

Embora estas classes, e as regras que as usam, sejam úteis, constituem apenas uma pequena parte da generalização possível em exemplos de tradução.

5.4.2 Classes de Entidades Mencionadas

Um problema semelhante ao anterior corresponde à tradução de frases que contêm entidades mencionadas. Na generalidade dos casos a entidade não é traduzida (e em muitos casos, embora exista uma entidade equivalente na língua de destino, o uso da original não é problema), pelo que são úteis exemplos de tradução em que as entidades mencionadas foram substituídas por variáveis.

Esta generalização não é tão útil em exemplos pequenos, já que normalmente as entidades mencionadas são um exemplo por si só. No entanto, em unidades de tradução maiores, é possível encontrarem-se entidades mencionadas.

O processo de generalização passa pela detecção da entidade em ambas as línguas, pela sua extracção para um dicionário de tradução

específico, e a sua substituição por uma variável que represente a classe de entidades mencionadas. Para esta tarefa poderá ser utilizada a abordagem descrita na secção 4.3.5 para a extracção de dicionários bilingues de entidades mencionadas.

5.4.3 Classes de Palavras

A generalização torna-se mais interessante quando se criam classes semânticas de palavras. Um exemplo típico é a construção de classes de gentílicos. As palavras "*português*", "*nigeriano*", "*norueguês*" ou "*mexicano*" correspondem a uma mesma classe e podem ser substituídas numa unidade de tradução sem alterar a correcção sintáctica da frase.

Uma abordagem comum para a criação de classes de palavras é a sua análise em contexto: para cada palavra de um corpus calcular o bigrama de palavras que a precede, e o bigrama de palavras que a sucede. Indexando a cada par de bigramas as palavras que ocorrem nesse mesmo contexto, obtém-se um conjunto de palavras de uma mesma classe.

Este método é completamente monolingue: é possível extrair classes de palavras para cada uma das línguas, mas é necessário um outro método que alinhe as classes e, que dentro de cada uma, alinhe as palavras constituintes.

A abordagem aqui proposta baseia-se no uso dos padrões de alinhamento para a extracção de classes paralelas de palavras, de duas formas distintas:

- o uso de entradas terminológicas extraídas com base em padrões para a construção de palavras;
- o uso de um padrão específico para a construção de classes de palavras;

Classes de Palavras a partir de Terminologia Bilingue

Consideremos todas as entradas terminológicas extraídas pelo padrão "`A B = B A`". De acordo com as línguas a que aplicamos o padrão, sabemos

que B corresponderá a adjectivos. Se escolhermos determinada palavra em A e procurarmos todos os adjectivos que co-ocorrem em B, obtemos uma classe de adjectivos usados num mesmo contexto (uma classe de palavras).

Por exemplo, se fixarmos em A a palavra "ácido", obtemos a seguinte lista de adjectivos:

```
1          ácido =>  clorídrico  | hydrochloric
2                    sulfúrico   |    sulphuric
3                    acético     |       acetic
4                    fólico      |        folic
5                    cítrico     |       citric
6                    nítrico     |       nitric
7                    tartárico   |     tartaric
8                    benzóico    |      benzoic
9                    fórmico     |       formic
10                   málico      |        malic
11                   sulfúrico   |     sulfuric
12                   erúcico     |       erucic  <= acid
```

No entanto é necessário ter algum cuidado com as classes obtidas: no exemplo seguinte não temos uma classe de cores como poderia parecer numa análise superficial.

```
1             livro =>  verde    |  green
2                       branco   |  white
3                       azul     |   blue
4                       aberto   |   open
5                       azul     |   blue
6                       branco   |  white
7                       vermelho |    red
8                       laranja  | orange  <= book
```

Embora esta classe não possa ser generalizada para uma classe de cores, pode ser criada uma classe específica para *tipos* de livros.

Classes de Palavras a partir de Padrões Específicos

Os padrões definidos pela PDL foram definidos com principal objectivo de ajudar o processo de extracção de exemplos e de terminologia. Estes mesmos padrões podem ser usados para outros fins, como sejam a criação semi-automática de classes de palavras.

Por exemplo, a classe de gentílicos que foi proposta como motivação para a necessidade de generalização, pode ser obtida aplicando a seguinte regra:

```
[G] "povo" X = X "people"
```

O uso de predicados genéricos permite que se possam executar efeitos laterais, como seja a adição directa de todas as palavras candidatas numa base de dados.

> Os padrões de tradução podem ser usados para a pesquisa de expressões bilingues e aprendizagem.

5.4.4 Discussão da Abordagem

A definição de classes de palavras ou de entidades permite a conversão de exemplos em regras de tradução. Estas regras não são aplicáveis apenas a determinado segmento de palavras, mas são genéricas de modo a que possam ser aplicadas a um conjunto de palavras (definido em compreensão com uma expressão regular, por exemplo, ou definido em extensão com uma lista de palavras).

Além da regra, é necessária a definição de um mapeamento entre as palavras ou entidades na língua de origem para a língua de destino. Depois de aplicada a regra, as palavras pertencentes às classes têm de ser traduzidas usando este mapeamento.

É ainda possível a definição de regras em cascata, definindo classes de regras (ao invés de classes de palavras).

A Título de Conclusão

A *Hipótese das Palavras-Marca* tem resultados igualmente interessantes na língua inglesa e portuguesa. Existindo um maior número de marcadores e de uso bastante mais intensivo para a língua portuguesa, a quantidade de segmentos extraídos por unidade de tradução é maior do que a quantidade de segmentos extraídos da língua inglesa. Esta desproporção leva a que o alinhamento entre segmentos não seja trivial. O uso de dicionários probabilísticos de tradução mostrou-se imprescindível para o alinhamento eficaz destes segmentos. Os exemplos extraídos usando a *Hipótese das Palavras-Marca* são linguisticamente completos, e têm uma confiança elevada para relações entre poucos segmentos (1 : 1, 1 : 2 e 2 : 1).

A *extracção combinatória de exemplos* tem como principal vantagem a sua independência em relação a conhecimento da língua. Para que funcione é apenas necessário um dicionário probabilístico de tradução, que pode ser extraído do mesmo corpus de onde os exemplos vão ser obtidos. No entanto, as diferenças sintácticas entre línguas podem levar à troca de ordem de palavras durante a tradução. Estas trocas tornam o algoritmo menos eficaz, pelo que se definiu uma linguagem para a especificação de padrões de tradução.

Os *padrões de tradução* mostraram-se eficazes não só para a extracção de exemplos entre línguas que obrigam a troca de ordem de palavras durante a tradução, mas também para a extracção de terminologia bilingue de qualidade.

Para permitir o uso generalizado de exemplos de tradução e da terminologia bilingue extraídos optou-se pela *generalização de exemplos*, usando para isso classes de palavras e entidades. Estas classes são facilmente obtidas usando a mesma linguagem de *padrões de tradução*.

Capítulo 6

Aplicação de Recursos de Tradução

Someone who cannot speak a language idiomatically either uses the idioms of his own language translated word by word or else he simply uses foreign words according to their literal meaning.

Isaac Asimov
"The Talking Stone"

Nos capítulos anteriores foram apresentados diversos métodos para a extracção de vários tipos de recursos de tradução, mas a sua aplicação foi pouco discutida. Esta secção discute algumas formas para aplicação dos recursos extraídos:

- a disponibilização de recursos via Web, usando uma interface ligada que permita a sua validação por consulta (secção 6.1);
- a criação de dicionários StarDict para consulta *off-line* de contextos de palavras (baeado em n-gramas) e de dicionários de tradução com concordâncias e entradas terminológicas (secção 6.2);
- permitir a consulta de uma forma programática (usando *web-services*) de modo a que outras aplicações possam tirar partido

dos recursos disponíveis (secção 6.3);

- integrar os recursos extraídos num ambiente de prototipagem para a criação de sistemas de tradução automática, usando o módulo Perl `Text::Translate` (secção 6.4).

6.1 Ambiente integrado Web

Como já referido anteriormente, foi criada uma interface Web para a validação, disponibilização e difusão dos recursos bilingues criados. Com a criação de uma aplicação Web, sem necessidade de instalação nem de requisitos de plataforma e simples de utilizar, permite-se que pessoas de várias áreas de investigação, e em diferentes etapas na sua formação, possam consultar os recursos extraídos e exprimir opiniões qualitativas sobre os mesmos: aumenta-se o impacto e alarga-se o leque de comentários e sugestões vindas de diversas áreas.

Neste sentido, acreditamos que a disponibilização de recursos através de uma aplicação Web é um ponto crucial, pelo que esta secção apresenta de forma detalhada as várias interfaces Web desenvolvidas, e algumas considerações a elas ligada.

O desenvolvimento desta aplicação Web teve os seguintes requisitos:

- suporte a multi-corpora, com diferentes pares de língua e grandes dimensões;
- suporte de vários tipos de recursos;
- apresentar o máximo de informação possível sobre cada um dos elementos pesquisados;
- permitir interligação entre os vários recursos disponibilizados;
- permitir a análise de algoritmos de uma forma interactiva e visual;

A ferramenta Web desenvolvida funciona com base no servidor de recursos desenvolvido durante a dissertação: o NatServer (ver secção 7.3). É constituída por um conjunto de interfaces Web integrados que permitem a consulta de diferentes tipos de recursos:

- concordâncias (monolingues e bilingues, orientadas ou não ao padrão);
- dicionários probabilísticos de tradução;
- contexto com base em n-gramas;
- meta-informação referente aos corpora disponíveis.

Além destes recursos directamente disponíveis no servidor, a aplicação web também permite a detecção e representação da diagonal de tradução de uma unidade de tradução (de acordo com o algoritmo definido em 5.2) tendo como base dicionários probabilísticos de tradução de determinado corpus.

As imagens que se seguem para ilustrar as funcionalidades das várias interfaces desenvolvidas incluem:

- setas do topo para a imagem que ilustra os vários tipos de informação apresentada (já que as interfaces não correspondem apenas à apresentação de o resultado de uma função, mas a apresentação da aplicação de várias funções ao recurso consultado);
- setas que partem da imagem, e que correspondem a ligações da interface para outras (ou para a mesma, consultando informação diferente);

A interligação entre as interfaces foi feita tendo em conta os vários tipos de dados envolvidos. Tudo começa com a escolha do corpus em causa e, dado que qualquer uma das ferramentas usa como base um corpus, permite a consulta directa da sua informação associada (metadata).

$$\text{Corpus} \longrightarrow (\text{Property} \rightharpoonup \text{Value})$$

Como interface principal foi escolhida a de concordâncias porque é a aquela que dá acesso ao corpus como um todo. Ao realizar-se a pesquisa de concordâncias, é retornado um conjunto de unidades de tradução.

$$\text{Corpus} \times (\text{W}^{\star}_{\mathcal{A}} + \text{W}^{\star}_{\mathcal{B}}) \longrightarrow (\text{S}_{\mathcal{A}} \times \text{S}_{\mathcal{B}})^{\star}$$

A cada memória de tradução ($tu_{\mathcal{A},\mathcal{B}} = \langle s_{\mathcal{A}} \times s_{\mathcal{B}} \rangle$) foi associada a possibilidades de saltar para as ferramentas que processam unidades de

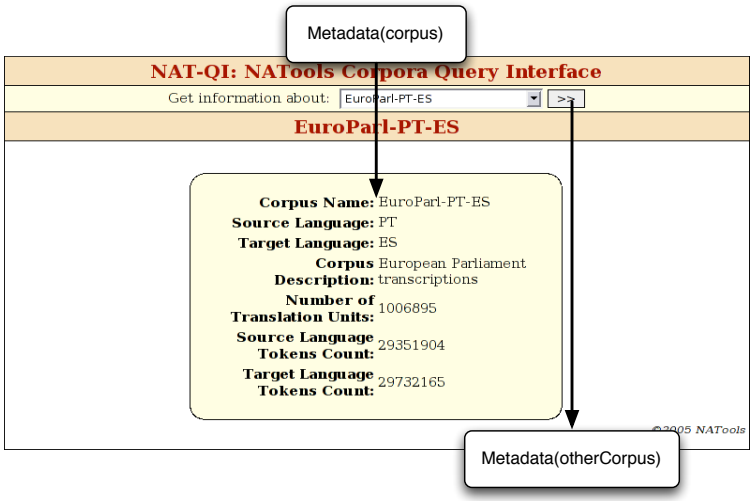


Figura 6.1: Informação sobre o corpus escolhido.

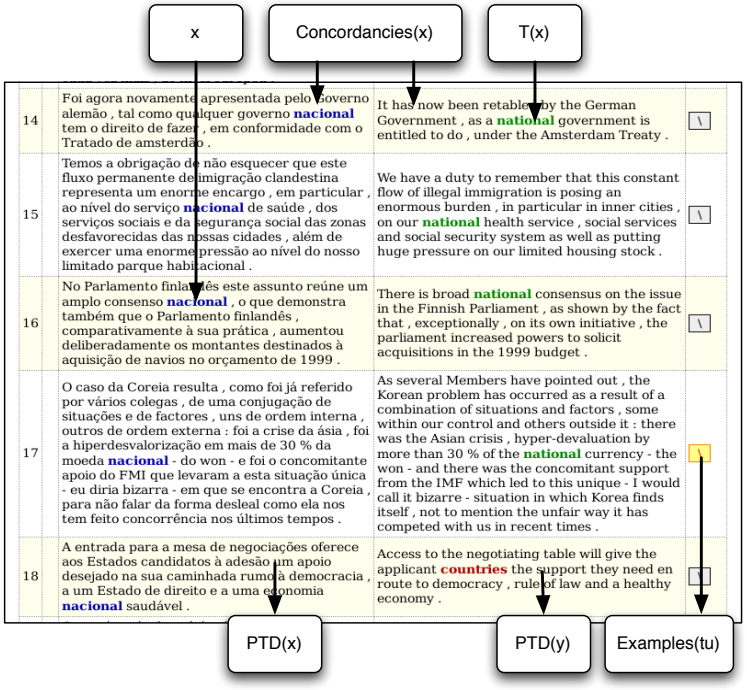


Figura 6.2: Resultado e ligações na pesquisa de concordâncias.

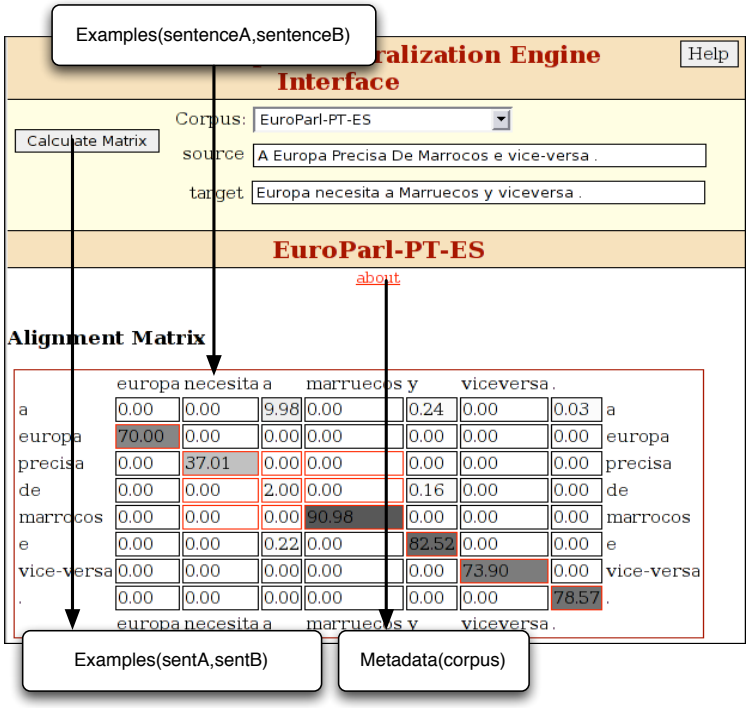


Figura 6.3: Extracção de Exemplos.

tradução: actualmente a interface para análise do algoritmo de extracção de exemplos de tradução com base na detecção da diagonal principal da matriz de tradução:

$$\text{Corpus} \times (\mathrm{S}_{\mathcal{A}} \times \mathrm{S}_{\mathcal{B}}) \longrightarrow (\mathrm{S}_{\mathcal{A}} \times \mathrm{S}_{\mathcal{B}})^{\star}$$

Cada concordância (unidade de tradução) é composta por sequências de palavras em duas línguas, pelo que é natural permitir o acesso às propriedades relativas às palavras que a constituem. Embora quer a consulta dos dicionários probabilísticos de tradução quer a consulta de n-gramas se refiram a propriedades de determinada palavra, optamos por dar prioridade à interface de consulta dos dicionários de tradução, já que associam informação multilingue (a dois níveis) a cada palavra.

Por sua vez, a consulta de n-gramas (bigramas e tetragramas) foi associada à interface de consulta dos dicionários. Ao consultar a entrada

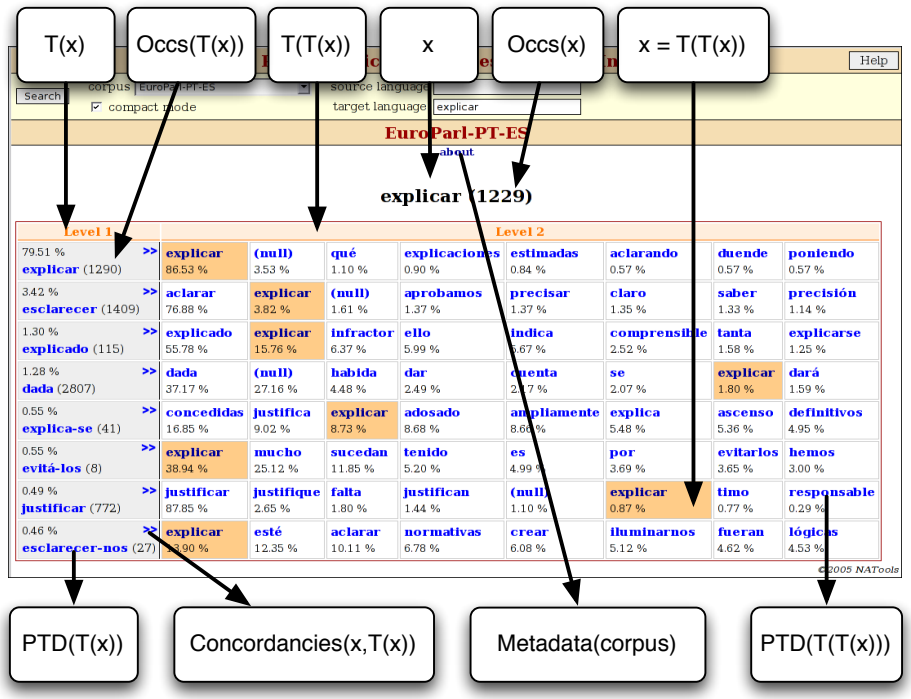


Figura 6.4: Resultado e ligações na navegação em PTD.

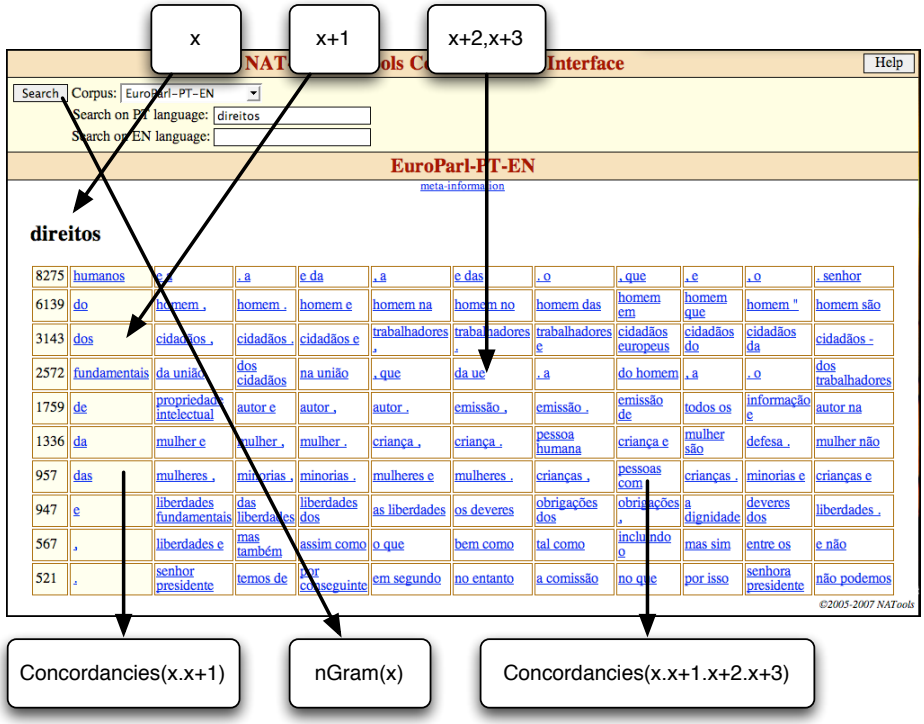


Figura 6.5: Consulta de n-gramas.

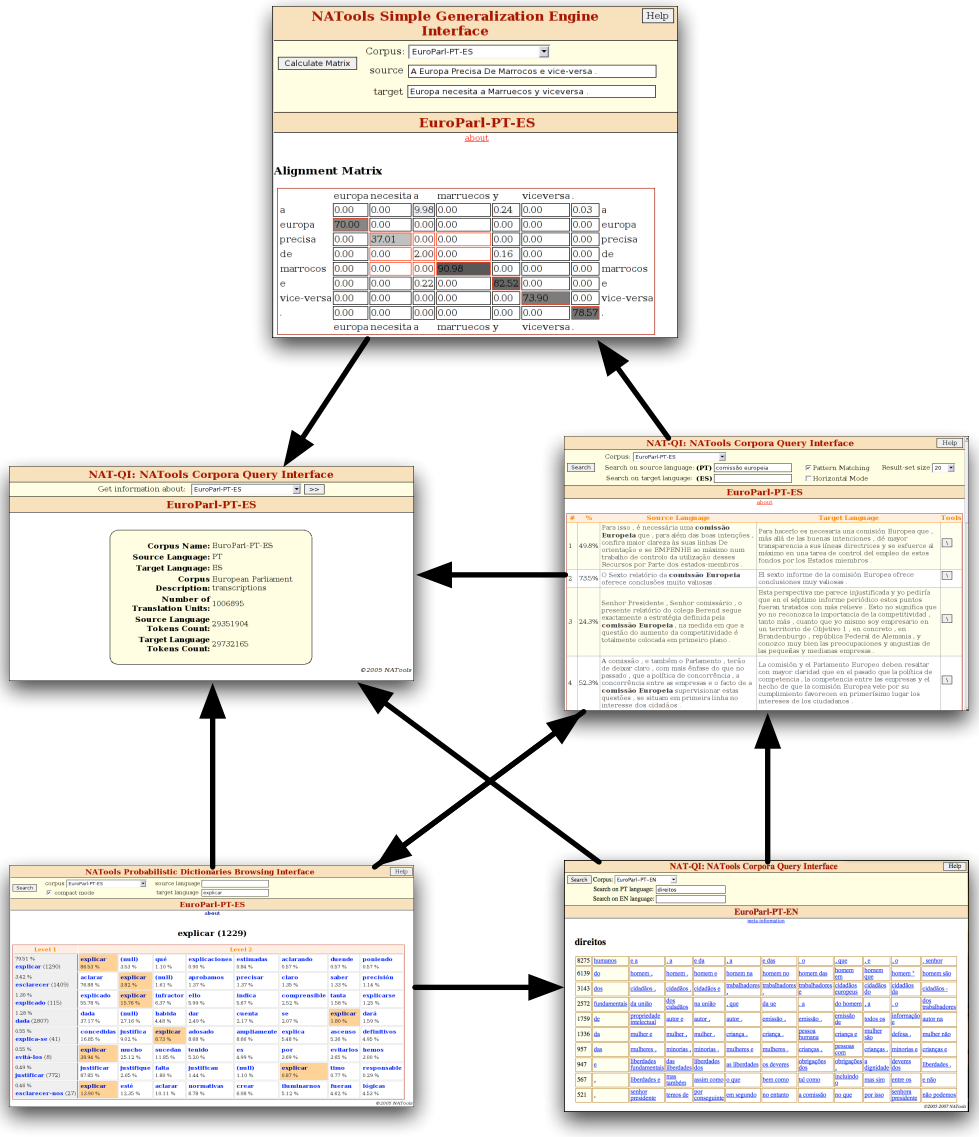


Figura 6.6: Interligação das várias interfaces web NATools.

do dicionário para determinada palavra é possível aceder às contagens de n-gramas respectivas. A figura 6.6 sumariza a integração destas várias ferramentas entre si.

6.2 Geração de Dicionários *off-line*

Aquando da apresentação e dicionários probabilísticos de tradução, na secção 4.4.3, foi apresentada sucintamente a ferramenta StarDict, e a criação de dicionários de tradução com base nos dicionários probabilísticos de tradução e em concordâncias (como exemplos de uso das respectivas traduções propostas).

Os dicionários criados previamente restringiam-se ao uso de PTD e de concordâncias. No entanto, existe uma grande quantidade de outros recursos que podem (e devem) ser incorporados em dicionários StarDict para uso em *off-line*.

Nesta secção aprofundaremos este problema apresentando algumas expressões que combinam recursos bilingues, definindo dicionários StarDict para uso geral em trabalhos de tradução ou estudos ou aprendizagem de línguas.

6.2.1 Dicionário de Contexto

O dicionário de contexto é construído com base em n-gramas e contém informação monolingue. Estes dicionários permitem consultar quais os contextos habituais para determinada palavra. Formalmente, este dicionário pode ser visto como um mapeamento entre determinada palavra e os contextos mais frequentes (à esquerda e à direita) de tamanho três,

dois e um:

$$\begin{aligned}
\text{StarDict} &= \text{W} \rightharpoonup \mathit{LeftContext} \times \mathit{RightContext} \\
\text{LeftContext} &= (\text{W} \times \text{W} \times \text{W}) \rightharpoonup \mathbb{N} \\
&\times \ (\text{W} \times \text{W}) \rightharpoonup \mathbb{N} \\
&\times \ \text{W} \rightharpoonup \mathbb{N} \\
\text{RightContext} &= (\text{W} \times \text{W} \times \text{W}) \rightharpoonup \mathbb{N} \\
&\times \ (\text{W} \times \text{W}) \rightharpoonup \mathbb{N} \\
&\times \ \text{W} \rightharpoonup \mathbb{N}
\end{aligned}$$

Cada uma das entradas para uma palavra w é construída por:

$$\begin{aligned}
\mathit{conc}(\quad & \mathit{ngrams4}(\star, \star, \star, w), \\
& \mathit{ngrams4}(w, \star, \star, \star), \\
& \mathit{ngrams3}(\star, \star, w), \\
& \mathit{ngrams3}(w, \star, \star), \\
& \mathit{ngrams2}(\star, w), \\
& \mathit{ngrams2}(w, \star) \quad)
\end{aligned}$$

em que as funções *ngrams4*, *ngrams3* e *ngrams2* calculam n-gramas dado um padrão (uma ou mais palavras, e alguns *placeholders*).

Dada a grande quantidade de contextos diferentes em que cada palavra ocorre, o dicionário inclui apenas os contextos mais frequentes[1]. São apresentados tetragramas, trigramas e bigramas uma vez que que os bigramas e trigramas mais frequentes não fazem necessariamente parte dos tetragramas mais frequentes.

A figura 6.7 mostra o StarDict com o dicionário de contextos apenas para tetragramas. Este dicionário, gerado a partir do corpus EuroParl PT:EN, tem cerca de 137 mil entradas, e ocupa mais de 50MB em disco. A criação do dicionário demora cerca de 20 minutos e terá realizado 822 000 acessos à base de dados de n-gramas. Estes dicionários permitem o estudo das palavras que mais co-ocorrem com determinada palavra.

[1]Em alternativa aos critérios de frequência podíamos usar outras medidas estatísticas mais complexas.

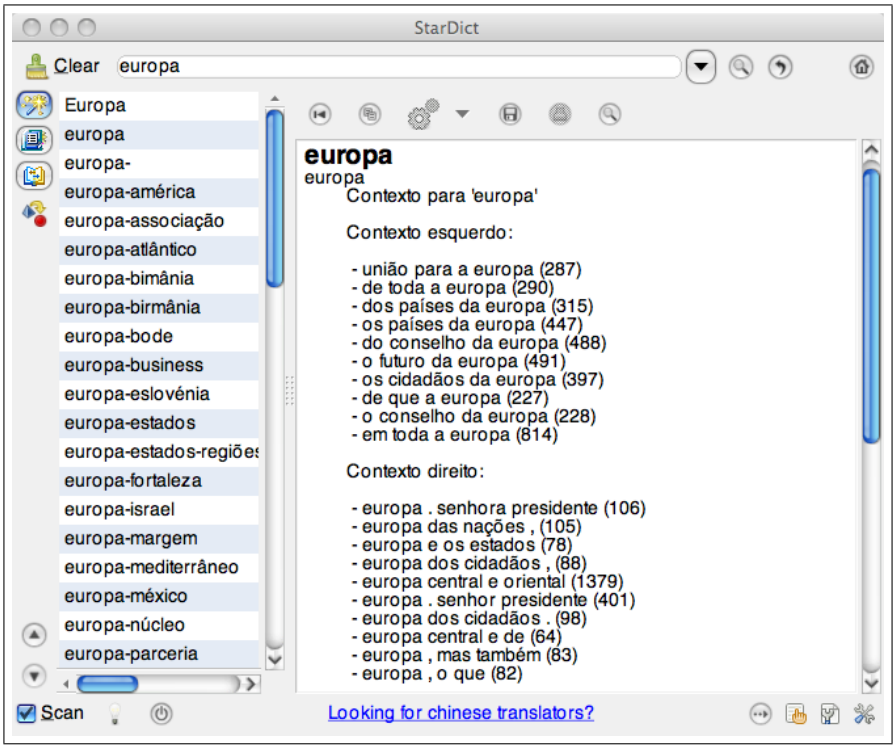


Figura 6.7: StarDict com um dicionário de contextos para a palavra "*europa*."

6.2.2 Dicionário Automático de Tradução

Para além da informação obtida dos PTD e das concordâncias em corpora, a terminologia bilingue extraída de acordo com os padrões de tradução (ver secção 5.3.3) é muito importante para um tradutor. Deste modo, construiu-se um dicionário de tradução com a seguinte estrutura:

$$\begin{aligned} \mathrm{W}_{\mathcal{A}} &= \text{Translations} \times \text{Examples} \times \text{Terminology} \\ \text{Translations} &= \mathrm{W}_{\mathcal{B}} \rightharpoonup [0..1] \\ \text{Examples} &= (\mathrm{S}_{\mathcal{A}} \times \mathrm{S}_{\mathcal{B}})^{\star} \\ \text{Terminology} &= (\mathrm{W}^{\star}_{\mathcal{A}} \times \mathrm{W}^{\star}_{\mathcal{B}}) \rightharpoonup \mathbb{N} \end{aligned}$$

Estes dicionários incluem as traduções mais prováveis, bem como três exemplos de uso para cada uma delas, e a lista de todas as entradas

terminológicas com mais de k ocorrências que incluem essa palavra e tradução.

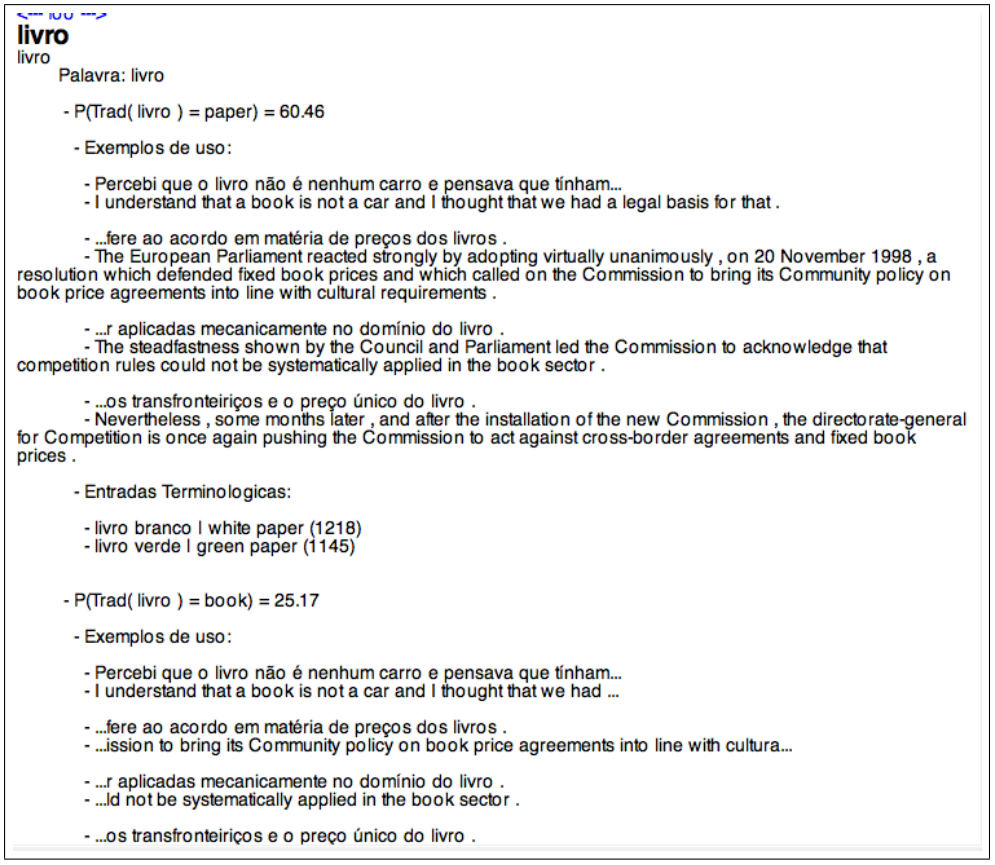


Figura 6.8: StarDict com um dicionário automático de tradução e terminologia para a palavra "*livro*"

A figura 6.8 mostra uma entrada de um destes dicionários automáticos de tradução para a palavra "*livro*". De realçar a zona com as entradas terminológicas que incluem a palavra em causa.

Os dicionários automáticos de tradução são muito úteis por apresentarem não só a tradução mais comum, mas um conjunto das mais prováveis, bem como em que contexto elas são usadas. A lista de terminologia permite analisar o comportamento da tradução da palavra dentro de expressões multi-palavra.

> Mais do que os recursos individuais, é importante a construção de dicionários envolvendo funções sobre recursos de modo a permitir uma análise mais completa em relação a cada palavra.

6.3 Recursos de Tradução Distribuídos

Na secção 3.4.5 foi apresentado o conceito de memórias de tradução distribuídas: a disponibilização via servidores dispersos pela Internet de unidades de tradução, e a sua possível integração em sistemas de tradução. Esta pequena secção pretende alargar o conceito a dicionários probabilísticos de tradução, exemplos de tradução e terminologia.

Como tem vindo a ser descrito de uma forma ortogonal à sequência lógica de extracção de recursos (e sistematizado na secção 7.3), durante a dissertação foi desenvolvido um servidor de recursos.

Este servidor usa um protocolo específico para a comunicação via *sockets* com os seus clientes. Esta abordagem, conhecida por RPC (*Remote Procedure Call*), tem vindo a perder adeptos em favor dos serviços Web, baseados em XML. No entanto, nada impede a criação de um *proxy* que proceda ao empacotamento e desempacotamento de pedidos SOAP[2]. A figura 6.9 esquematiza a padrão de uma *proxy* e como esta pode ser aplicada ao servidor NATools. A importância da abordagem SOAP em relação à tradicional RPC corresponde ao protocolo usado. Uma vez que o SOAP usa protocolo HTTP é simples de colocar serviços acessíveis por trás de *proxies* e de *firewalls*. Embora esta secção não volte a referir a abordagem SOAP, é importante salientar que a sua implementação é trivial: o comportamento obtido com o servidor RPC pode ser imitado facilmente usando a tecnologia SOAP.

Considerando a API descrita na secção 7.3, e os recursos que foram apresentados, é possível a integração de servidores NATools distribuídos em ferramentas de tradução assistida por computador, de forma a

[2]Originalmente SOAP significava "*Simple Object Access Protocol*", passou a "*Service Oriented Architecture Protocol*" e actualmente tornou-se um termo por si só.

Figura 6.9: Proxy SOAP para o servidor NatServer.

permitir:

- consultar a cada momento unidades de tradução completas ou exemplos de tradução, de forma a permitir ao tradutor reutilizar porções de traduções realizadas e não apenas traduções completas;
- consultar as possíveis traduções de uma palavra tendo informação estatística sobre qual a mais provável, e para cada uma, um conjunto de unidades de tradução em que essa tradução exista;
- consultar qual o contexto mais habitual para determinada palavra, para de uma forma simples saber qual a concordância de género e número, bem como quais as palavras vizinhas mais comuns;

Embora o NatServer actual não o permita (e talvez não o venha a permitir por não ser essa a sua finalidade), é ainda possível que um servidor de recursos permita a colaboração dos seus utilizadores para melhorar os seus recursos. A princípio a possibilidade de colaboração externa pode levar a que se pretendam implementar sistemas de controlo de utilizadores para que não sejam introduzidas más traduções, transformando o servidor de *state-less* a *state-full*. No entanto, e uma vez que a cada recurso associamos um valor estatístico do seu uso, este controlo não é de todo necessário, bastando que o servidor mantenha um contador do número de vezes que cada tradução foi usada em relação às suas alternativas. Desta forma, sempre que uma má tradução tenha sido submetida ao servidor, esta nunca terá uma marca de qualidade, a não ser que seja usada várias vezes.

Em relação à disponibilização distribuída de recursos de tradução foram publicados dois artigos sobre os conceitos técnicos envolvidos (Simões, Guinovart, and Almeida, 2004; Simões, Almeida, and Guinovart, 2004). Actualmente é necessária a implementação das funcionalidades de consulta remota ao nível dos clientes de tradução, e a possível criação

de um novo servidor para permitir colaboração externa. O desenvolvimento destas funcionalidades não faz parte do objectivo desta dissertação, pelo que se apresenta como trabalho futuro na área da tradução assistida por computador.

6.4 Adaptação de Recursos Bilingues para Tradução Automática

A validação e avaliação de recursos tem muito que ver com o contexto em que vão ser aplicados. Nesse sentido, optou-se por realizar testes de uso dos recursos obtidos no `Text::Translate`, uma ferramenta para a prototipagem rápida de sistemas de tradução. As experiências realizadas centram-se apenas na tradução de segmentos nominais.

6.4.1 Ambiente de teste

Como foi referido na secção 2.5.2, o `Text::Translate` é um módulo Perl que permite a prototipagem de sistemas de tradução automáticos essencialmente baseados em regras. Funciona com uma hierarquia de dicionários (hierarquia esta que especifica a prioridade de tradução) e um conjunto de regras de pós-processamento. Os dicionários incluem mapeamentos entre palavras, termos ou expressões multi-palavra, e as regras mudanças de ordem entre palavras.

O primeiro passo na inclusão de recursos obtidos com o NATools no `Text::Translate` foi o de definir quais e em que circunstâncias se pretendem usar:

- as entradas terminológicas extraídas com base em padrões foram usadas de duas formas distintas:
  - como exemplos de tradução, e portanto aplicadas directamente sempre que um segmento igual precise de ser traduzido;
  - como fonte para a extracção de dicionários de tradução eti-

quetados com uma categoria morfológica;

- a base de n-gramas foi usada como modelo de língua, para permitir sempre que possível escolher entre várias traduções com base na sua frequência em corpora;
- os dicionários probabilísticos de tradução foram usados directamente para a tradução de palavras desconhecidas.

Segue-se uma descrição mais detalhada da preparação destes recursos, e de como foram integrados no `Text::Translate`.

Embora as entradas terminológicas estejam a ser usadas integralmente como exemplos de tradução, o facto de serem extraídas usando padrões leva a que se possa inferir algum relacionamento entre as palavras constituintes (como foi referido na secção 5.3.1).

No contexto da tradução de inglês para português, consideremos a regra "$A\ B\ =\ B\ A$". De um modo simplificado, podemos inferir com um grau de certeza bastante elevado[3] que as palavras na posição A são traduções mútuas, e que também o são as palavras na posição B. Além disso, também é possível inferir que as palavras na posição A são adjectivos e na B são substantivos.

Da mesma forma, na regra "$B\ A\ =\ A\ de\ B$" é possível associar os substantivos na posição A, e inferir uma regra que descreve que a tradução do adjectivo B da língua inglesa é realizada mediante uma frase preposicional sobre o substantivo B na língua portuguesa.

É possível inferir propriedades sobre as palavras que façam *matching* às várias regras definidas. Neste sentido, a lista de entradas terminológicas é processada do seguinte modo:

- são extraídos todos os relacionamentos possíveis entre as palavras constituintes, de acordo com o padrão que lhe deu origem (note-se que as entradas terminológicas extraídas são anotadas com o nome do padrão). Durante este processo e sempre que tal faça sentido, as palavras são lematizadas para a sua forma masculino singular e, sempre que possível, é adicionada uma etiqueta que permita saber

[3]Relembre-se que para que o padrão ser aplicado foi necessário que as células correspondentes às traduções incluíssem uma certeza de tradução mútua elevada.

a categoria gramatical da palavra em causa para facilitar o uso de regras durante a tradução. Esta etiqueta é importante para que o pós-processador possa trocar palavras de ordem e corrigir sempre que necessário as concordâncias de género e número.

- as entradas do dicionário de tradução extraídas são contadas de forma a determinar uma medida de probabilidade, de acordo com a sua ocorrência. Esta medida será usada posteriormente para classificar qualitativamente cada uma das traduções possíveis.

Este processo permite obter três tipos de dicionários:

- $\mathcal{D}_1 = \mathrm{W}^{\star}_{\mathcal{A}} \rightharpoonup \mathrm{W}^{\star}_{\mathcal{B}}$
  um dicionário de tradução entre entre segmentos de palavras, criado automaticamente a partir das entradas terminológicas. Em caso de ambiguidade (ou seja, se um segmento $s_{\mathcal{A}}$ pode ser traduzido pelos segmentos $s'_{\mathcal{B}}$ e $s''_{\mathcal{B}}$), o algoritmo de tradução apenas considerada a tradução mais frequente, removendo assim ambiguidade na tradução de terminologia multi-palavra[4].
- $\mathcal{D}_2 = \mathrm{W}_{\mathcal{A}} \rightharpoonup \mathrm{W}_{\mathcal{B}}$
  um dicionário de tradução entre palavras, obtido a partir da terminologia bilingue, composto pelas palavras que não têm uma tradução ambígua.
- $\mathcal{D}_3 = \mathrm{W}_{\mathcal{A}} \rightharpoonup (\mathrm{W}_{\mathcal{B}} \rightharpoonup [0..1])$
  um dicionário de tradução com ambiguidade, em que a cada tradução é associada uma confiança probabilística. Este dicionário é obtido a partir dos padrões, como o dicionário anterior, mas só inclui entradas ambíguas.

Para além destes dicionários é usado um conjunto de dicionários base e regras gerais do `Text::Translate`, construídos manualmente.

O processo de tradução é baseado numa cascata de dicionários, seguido de um pós-processador baseado em regras de reescrita. A cascata de dicionários corresponde a uma lista de dicionários que vão ser consultados sequencialmente. Note-se que a ordem dos dicionários indicada ao `Text::Translate` é importante, já que em primeiro lugar devem ser

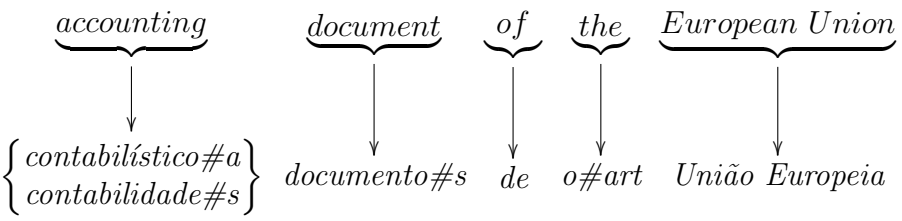
[4]Esta foi uma decisão de implementação para simplificar o algoritmo de tradução.

usadas as entradas com maior certeza de tradução.

O algoritmo de tradução pode ser considerado híbrido simples, entre os sistemas de tradução baseados em exemplos (EBMT), os sistemas de tradução estatísticos (SMT), e os sistemas baseados em regras.

Segue-se uma descrição simplificada do algoritmo, usando como exemplo a tradução do segmento nominal "*accounting documents of the European Union*":

1. a cascata de dicionários é consultada, tentando sempre que possível traduzir a maior sequência de palavras (usando o dicionário $\mathcal{D}_1$). Quando duas sequências de palavras do mesmo comprimento se sobrepõem parcialmente é usada (de forma *naif*) a que aparece primeiro.
2. sempre que tal não for possível, será necessário realizar uma tradução palavra a palavra:
   (a) é consultada a cascata de dicionários para obter a tradução da palavra em causa (através do dicionário $\mathcal{D}_2$ ou dos dicionários genéricos do `Text::Translate`), ou das várias alternativas de tradução no caso de existência de ambiguidade na tradução (usando o dicionário $\mathcal{D}_3$). Neste caso, a cada hipótese de tradução é associada uma medida de confiança (probabilidade);
   (b) se a palavra a traduzir não é encontrada na cascata de dicionários, então é consultado um dicionário probabilístico de tradução. São obtidas as melhores k traduções, e associada a cada uma a sua probabilidade;
   (c) se a palavra é completamente desconhecida, é marcada para que possa posteriormente ser analisada manualmente;



3. frequentemente, a tradução obtida é ambígua, pelo que são geradas todas as traduções possíveis mediante combinação das várias hipóteses de tradução;

   contabilístico#a documento#s de o#art União Europeia
   contabilidade#s documento#s de o#art União Europeia

4. a cada uma das traduções geradas são aplicadas regras para a re-organização de palavras e a correcção de concordâncias;

   documento contabilístico da União Europeia
   documento de contabilidade da União Europeia

5. as várias traduções devem ser avaliadas, para que se possa escolher a mais fluente (ou com maior suavidade contextual). Esta selecção é realizada usando o modelo de língua. Dada a facilidade do `Text::Translate` em usar regras condicionais baseadas em recursos externos, é possível consultar os n-gramas disponíveis localmente, e também outras bases externas como sejam o BACO (Sarmento, 2006), ou mesmo a consulta através de um motor de pesquisa na Internet, como o Google.

   documento contabilístico da União Europeia

A regras definidas estão directamente relacionadas com os padrões de extracção de terminologia (uma vez que pretendemos traduzir apenas segmentos nominais). Seguem-se dois exemplos de regras de reorganização frásica:

- na tradução de inglês para português a ordem relativa entre substantivos e adjectivos muda (como vimos no padrão ABBA). Para além da mudança de ordem, o adjectivo deve ser alterado de forma a concordar em género e em número com o substantivo em causa. Deste modo, os adjectivos são etiquetados com uma marca, do seguinte modo:

```
1    abusive=abusivo#a
2    dynamic=dinâmico#a
```

  A regra de troca de ordem durante a tradução deve ser dividida em quatro, para contemplar as várias combinações de género e

número. Seguem-se dois exemplos destas regras (para o masculino singular e para o feminino plural):[5]

```
($w)#a ($w)#sms ==> $2+$1#sms
($w)#a ($w)#sfp ==> $2+($1#TO#fp)#sfp
```

A aplicação desta regra deverá permitir a tradução dos seguintes segmentos:

```
    abusive aid         -> auxílio abusivo
    abusive alteration  -> alteração abusiva
    dynamic access      -> acesso dinâmico
    dynamic adaptations -> adaptações dinâmicas
```

- do mesmo modo, existem substantivos na língua inglesa que funcionam como adjectivos e que, na língua portuguesa, dão origem a um sintagma preposicional. Esta regra deve ser aplicada sempre que surjam, depois da tradução, dois substantivos consecutivos na língua portuguesa. A regra (simplificada) corresponde a:

```
($w)#s ($w)#s ==> $2#s+de+$1
```

e permitiria a tradução dos seguintes segmentos:

```
    embarkation areas     -> zonas de embarque
    embarkation deck      -> pavimento de embarque
    abandonment measures  -> medidas de abandono
    abandonment programme -> programa de abandono
```

6.4.2 Experiência de Tradução: Thesaurus da Academia Sueca

A primeira experiência realizada com o `Text::Translate` centrou-se na tradução de entradas semi-terminológicas de uma ontologia classificativa da Academia Sueca. Nesta experiência não foram usados os dicionários probabilísticos de tradução nem os *n*-gramas (ou seja, foi utilizada a

[5] As regras são apresentadas como apontamento meramente indicativo já que não constituem o centro da nossa intervenção.

terminologia bilingue extraída do corpus EuroParl e os dicionários de tradução dela extraídos).

Foi usado um corpus de uma área completamente diferente da do texto a traduzir por se pretender realizar uma tradução orientada à palavra e não orientada à terminologia cristalizada.

A ontologia é constituída por 666 termos. Destas entradas, 179 contêm palavras que não constam nos dicionários e terminologias usadas (como "*bioorganic*" e "*sedimentology*"). Das restantes entradas, foram seleccionadas e avaliadas manualmente 100, das quais 29 entradas foram classificadas como erradas (com problemas de má tradução, de concordâncias e de ordenação de palavras).

Seguem-se alguns exemplos de tradução (correctas e erradas) desta avaliação:

```
1          History of technology and industry
2             História de tecnologia e indústria
3          Classical archaeology and ancient history
4             Arqueologia clássica e história secular
5          Spanish language
6      *      Língua espanhol
7          Library and information science
8      *      Biblioteca e informações ciência
```

Em relação aos exemplos apresentados, o segundo exemplo foi considerado correcto embora a tradução mais esperada correspondesse a "*história antiga*" e não a "*história secular*". Mas, como foi referido previamente, a experiência tinha como principal objectivo analisar o comportamento da tradução orientada à palavra, e não a tradução usando directamente terminologia.

Embora a taxa de entradas correctas (71%) já seja aceitável, a incorporação dos dicionários probabilísticos de tradução e o uso de n-gramas para a escolha de traduções irá ajudar a melhorar a taxa de sucesso.

Note-se que mais uma vez o contexto desta experiência não foi favorável uma vez que uma quantidade razoável de termos usados nesta ontologia não fazem parte do léxico habitual do Parlamento Europeu.

6.4.3 Análise de Resultados

O objectivo da experiência realizada não era a construção de um tradutor completo, mas a demonstração da utilidade dos recursos bilingues extraídos. Pela experiência realizada parece-nos correcto dizer que os recursos bilingues extraídos podem ser usados directamente na construção de sistemas de tradução, e como recurso fonte para a extracção de novos recursos bilingues.

> Os recursos bilingues de tradução são facilmente adaptáveis para o uso em tradução automática.

A definição de regras de reordenação de palavras e adaptação de concordâncias estão fortemente ligadas aos padrões de extracção de terminologia, pelo menos no que se refere à tradução de segmentos nominais.

Para uma experiência mais séria seria necessário processar mais corpora, e de diferentes géneros, o que permitiria aumentar a cobertura de todos os dicionários usados.

Embora se tenha planeado o uso da ferramenta Apertium para a tradução inglês:português, não existiam recursos léxicos preparados para este par de línguas, pelo que se optou por realizar experiências apenas com o `Text::Translate`.

A Título de Conclusão

Existe uma grande aplicabilidade de recursos de tradução. Nesta secção foram apresentadas algumas áreas onde os recursos criados podem ser cruciais.

Nas duas primeiras secções foram apresentadas formas de disponibilização dos recursos obtidos para o uso directo pelo utilizador final: através de uma aplicação Web integrada, e usando dicionários *off-line*. Ambas as abordagens não se cingem à apresentação de recursos, mas à integração dos vários tipos obtidos, apresentando sempre que possível a maior quantidade possível de informação relacionada.

A terceira secção apresentou genericamente as abordagens possíveis para o uso de recursos de forma programática por aplicações, utilizando serviços Web ou comunicação por *sockets*.

Finalmente, foi apresentada uma metodologia para a adaptação dos recursos bilingues para uso em ferramentas de tradução automática. Esta experiência demonstrou que com um pouco de processamento é possível preparar recursos específicos para a tarefa em causa a partir de recursos já existentes.

Capítulo 7

Estratégias de Desenvolvimento e Teste

Divide and conquer was a successful military strategy. Generals observed that it was easier to defeat one army of 50,000 men, followed by another army of 50,000 men than it was to beat a single 100,000 man army. Thus the wise general would attack so as to divide the enemy army into two forces and then mop up one after the other.

Steven S. Skiena
"The Algorithm Design Manual"

Para além das contribuições referentes aos algoritmos, recursos obtidos e ferramentas disponibilizadas, esta dissertação pretende também discutir um conjunto de estratégias de desenvolvimento, que se tornaram como que directivas ou guias de estilo.

Um dos grandes problemas no desenvolvimento de aplicações escaláveis em processamento de linguagem natural tem que ver com os tamanhos dos recursos a serem processados. Por exemplo, o processamento de um corpus como o EurLex, com mais de 3 GB de texto, obriga a uma estratégia de escalabilidade sensata e independente da quantidade

de memória disponível.

O desenvolvimento das aplicações referidas neste documento teve um conjunto de requisitos de base, como já salientado na secção 1.1. Este capítulo apresenta as estratégias de desenvolvimento que permitiram cumprir os requisitos estipulados:

- **Decomposição Estrutural:** pretende-se que uma ferramenta seja decomposta estruturalmente em pequenas ferramentas (visão modular). Esta abordagem permite uma maior flexibilidade durante o desenvolvimento: não só se torna mais simples o debug, como se torna possível a reutilização e execução incremental das aplicações. A secção 7.1 detalha as vantagens desta estratégia de desenvolvimento e teste.
- **Decomposição por Partição:** interessa-nos que as ferramentas desenvolvidas sejam capazes de lidar com corpora de tamanhos reais. A estratégia usada baseia-se na partição dos corpora, a replicação das funções de processamento e a posterior junção dos resultados. Esta estratégia é descrita na secção 7.2.
- **Descomposição por Distribuição:** as aplicações devem permitir sempre que possível a distribuição de processamento. Deste modo, usou-se uma arquitectura cliente/servidor como meio para a possível paralelização na disponibilização de corpora e no seu processamento (distribuição ao nível do servidor e ao nível do cliente). A secção 7.3 detalha o servidor NatServer, e o desenvolvimento de aplicações numa arquitectura Cliente/Servidor.
- **Programabilidade:** as aplicações devem ser genéricas, de forma a que possam ser aplicadas em situações diversas, e que possam ser facilmente extendidas com novas funcionalidades. Foi disponibilizada uma API de ordem superior que para o desenvolvimento de protótipos e aplicações de forma simples e rápida. A secção 7.3 descreve a API disponibilizada pelo NatServer.

A secção 7.4 apresenta uma estratégia de paralelização e escalonamento (bem como uma ferramenta que as implementa) que tira partido da´ decomposição estrutural e da decomposição por partição para a execução de aplicações num *cluster* de computadores.

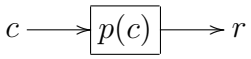
7.1 Decomposição Estrutural

Existem duas abordagens possíveis no desenvolvimento de aplicações de tamanho real: o desenvolvimento de uma única aplicação, que funciona como um todo, ou o desenvolvimento de várias aplicações ou módulos, que podem funcionar de forma independente entre si, ou como um todo de forma composicional.

Nesta dissertação defende-se a subdivisão de uma aplicação num conjunto de pequenas tarefas: aplicações pequenas, independentes e composicionais. Considere-se o processo $p(c)$ que é definido como a composição de quatro funções f, g, h e q:

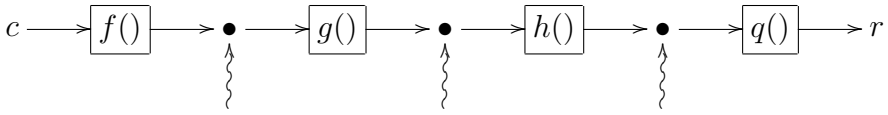
$$p(c) = q\left(h\left(g\left(f\left(c\right)\right)\right)\right)$$

Se este processo p for implementado como um único programa, o que se obtém é uma caixa negra:



No caso deste processo não funcionar ou houver necessidade de optimização, todo o código da aplicação terá de ser analisado. Da mesma forma, se ocorrer uma interrupção de serviço (como um corte de energia) durante o seu processamento, será necessário executar de novo toda a tarefa.

Se, por sua vez, p for implementado como a verdadeira composição das quatro funções, obteremos quatro caixas negras, e três pontos de teste e sincronização:



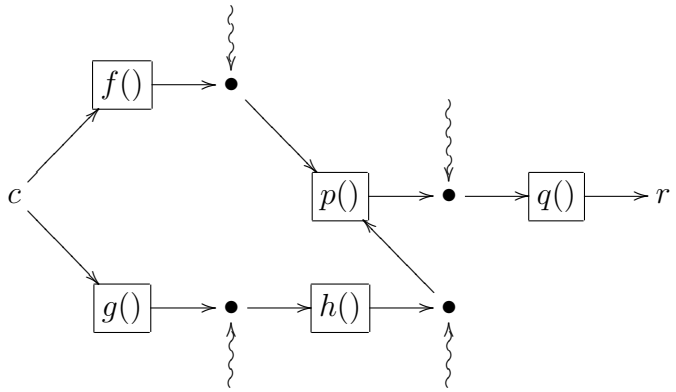
Se o resultado não for o esperado, é possível analisar os resultados intermédios, tornando-se mais simples e rápida a detecção da tarefa em erro. Do mesmo modo, se ocorrer uma falha eléctrica, é possível recuperar o processamento a partir do último ponto de sincronização.

> A decomposição estrutural permite maior facilidade na análise de código e detecção de erros. Permite ainda o processamento incremental.

Consideremos outro exemplo, um pouco mais complexo, de decomposição estrutural: se a nossa tarefa consiste na seguinte composição de funções:

$$r = q\left(p\left(f\left(c\right), h\left(g\left(c\right)\right)\right)\right)$$

podemos decompo-la em cinco tarefas, e obter quatro pontos de sincronização:



Esta estratégia de decomposição e composicionalidade de sub-tarefas foi amplamente usada durante o desenvolvimento das aplicações do NA-Tools. Por exemplo, o processo de extracção de dicionários probabilísticos de tradução é composto por quatro pequenas aplicações: codificação dos corpora, construção de uma matriz de co-ocorrências, iteração do Algoritmo EM sobre a matriz e a extracção de resultados. Desta forma é possível afinar as ferramentas de forma independente, e os resultados incrementais podem ser reutilizados em caso de necessidade.

7.2 Decomposição por Partição

O processamento típico de corpora não necessita de *ver* um corpus como um todo. Habitualmente o processamento frase a frase ou parágrafo a parágrafo é suficiente. São raras as ocasiões em que se precisa de processar fatias maiores de texto.

Há algoritmos que levam à criação de estruturas de dados complexas que crescem em memória, embora sejam preenchidas à medida que se vão processando diferentes unidades de tradução. Por exemplo, na extracção de dicionários probabilísticos de tradução é necessária a construção de uma matriz esparsa de co-ocorrências que, no caso do corpus EurLex, tem 658601×608921 células[1].

A estratégia de desenvolvimento usada para garantir a escalabilidade de um processo f, corresponde à partição, processamento independente das partes, e posterior junção dos resultados. Para que isto seja possível, é necessário a definição de uma função de partição ($\mathcal{P} : \mathcal{C} \longrightarrow \mathcal{C}^\star$), uma função de processamento das partes ($f' \cong f$) e uma função de junção ($g : \mathcal{R}^\star \longrightarrow \mathcal{R}$).

Ou seja, a aplicação de uma função $f()$ a um corpus c:

$$c \longrightarrow \boxed{f(c)} \longrightarrow r$$

é realizada pela partição do corpus em fatias (c_i), que são processadas de forma independente como nos mostra a figura 7.1. Isto significa que podemos definir $f()$ como:

$$f(c) \cong g(\{f'(x) : x \in \mathcal{P}(c)\})$$

em que $g()$ é a função de agregação dos resultados de $f'()$. Esta função $f'()$ pode ser $f()$, ou com pequenas alterações para que o seu resultado possa ser agregado posteriormente. Note-se que aplicação desta estratégia de decomposição pode levar aos mesmos resultados da tarefa inicial, ou pode resultar em pequenas perdas.

[1] Considerando 1% de células ocupadas, temos um total de 4 010 359 795 células. Se em cada célula armazenarmos quatro bytes, a matriz ocupa mais de 15 GB. De notar que cada célula acaba por usar mais do que quatro bytes e que a representação de uma matriz esparsa em memória não é muito económica.

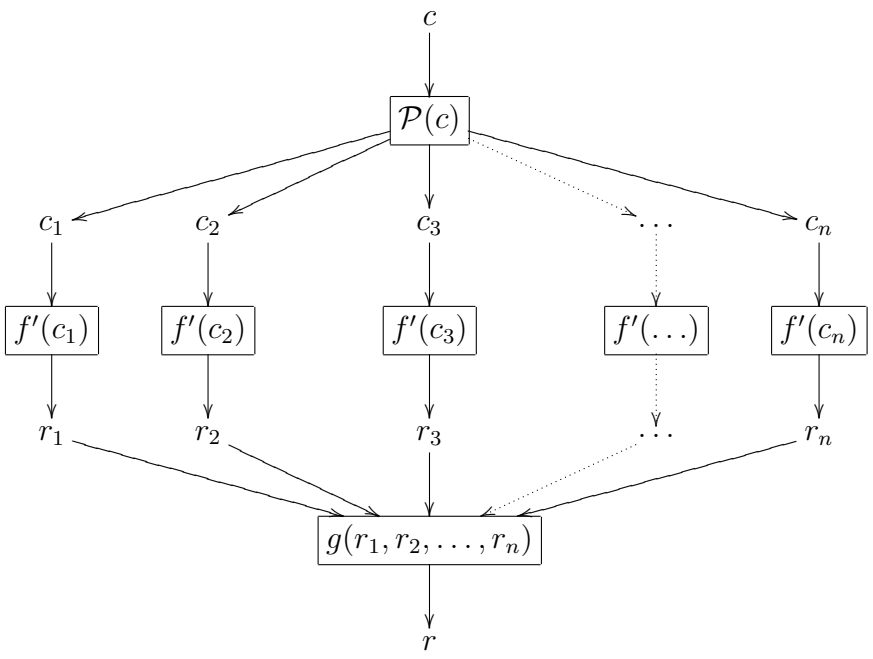


Figura 7.1: Estratégia de decomposição por partição, replicação e aglutinação.

Esta abordagem foi aplicada em várias etapas da dissertação. Seguem-se dois exemplos de funções $f()$ e $g()$ usadas para permitir o desenvolvimento escalável das ferramentas inclusas no NATools:

- na extracção de dicionários probabilísticos, discutida na secção 4.1, é criada uma matriz de co-ocorrências que é incomportável na memória de um computador comum. A solução passou pela divisão do corpus em pequenos sub-corpora e a criação da suas matrizes de co-ocorrências. Destas matrizes são extraídos dicionários probabilísticos de tradução que são posteriormente somados de acordo com uma fórmula discutida nessa mesma secção. Este é um exemplo de uma situação que permitiu a escalabilidade de um algoritmo que é dado como irremediavelmente limitado por vários investigadores (Nieto and O'Donnell, 2007);
- nas várias abordagens para extracção de exemplos discutidas no

capítulo 5 também foi aplicada esta mesma abordagem. Nesta situação o problema não era a incapacidade de processar todo o corpus sequencialmente, mas permitir a paralelização da extracção de exemplos. Para isso, a ferramenta de extracção de exemplos aceita um corpus e a especificação da partição a processar. Os exemplos de cada uma destas zonas são extraídos para ficheiros separados que são posteriormente aglutinados por simples concatenação.

Para além de permitir a escalabilidade de processamento de corpora, permite ainda a paralelização destes mesmos processos. Dado que as funções $f()$ processam as fatias independentemente podem ser paralelizadas em diferentes computadores (ou nodos de um cluster), sendo que apenas a função $g()$ não será paralelizada.

> A partição de um problema em tarefas mais pequenas permite a escalabilidade de tarefas. Este processo obriga à definição de uma função de partição que prepare partes a processar de forma independente, e uma função de agregação que junte os resultados obtidos.

Esta facilidade na paralelização de processos levou a que se desenvolvesse uma linguagem de domínio específico para a especificação de interdependências entre processos para permitir a sua paralelização sempre que possível. Esta linguagem e o respectivo escalonador são discutidos na secção 7.4.

7.3 Decomposição Cliente/Servidor

Durante a criação dos recursos discutidos nesta dissertação tornou-se necessário definir uma metodologia eficiente para a disponibilização de recursos. Um dos principais problemas na disponibilização de recursos corresponde à eficiência na consulta de corpora de grandes dimensões. Com isso em mente, foram definidos alguns pontos prioritários no desenvolvimento de uma ferramenta para a disponibilização de recursos.

Genericamente, a ferramenta deve:

- ser *open-source* e integrada (apenas um servidor para vários tipos de recursos);
- ser capaz de disponibilizar mais do que um corpus ao mesmo tempo, para diferentes pares de línguas;
- ser capaz de lidar com corpora de grandes dimensões (por exemplo, o EuroParl tem mais de um milhão de unidades de tradução, e o EurLex mais de 10 milhões);
- suportar uma API simples para a implementação de experiências e protótipos em poucas linhas de código (de forma a que o programador se possa centrar na tarefa que está a implementar e não nos detalhes de acesso eficiente a recursos);
- permitir a expansão futura com suporte multi-camada, para a adição de informação a cada palavra, como sejam o lema ou a sua categoria gramatical.

Existem vários tipos de recursos que foram extraídos ao longo desta dissertação. A sua consulta eficiente é imprescindível para que se possam desenvolver aplicações que os usem. Deste modo, a ferramenta deve permitir a consulta de:

- **concordâncias** quer **monolingues** (com base na língua origem ou na língua destino) e **multilingues**. Estas concordâncias devem suportar pesquisas por palavras, sequências e padrões;
- **dicionários probabilísticos de tradução**, obtendo para cada palavra o seu número de ocorrências e as suas prováveis traduções;
- **n-gramas** por língua, permitindo a pesquisa por n-gramas completos ou por padrões;
- **meta-informação** sobre cada um dos corpus disponibilizados, como sejam as línguas envolvidas e o número de unidades de tradução;
- **recursos parciais**, não obrigando a que para cada corpus seja necessário ter calculado todo o tipo de recursos (n-gramas, PTD, etc).

Os recursos podem ser consultados por diferentes tipos de aplicações. Algumas destas aplicações precisam de eficiência no carregamento dos recursos, enquanto que outras precisam de eficiência na resposta a consultas. Esta necessidade dupla levou ao desenvolvimento de uma arquitectura híbrida:

- Reduzir o tempo de carregamento de índices e dicionários é importante em abordagem interactivas, como o acesso com aplicações web. Para obter este tipo de performance foi implementada uma **arquitectura cliente/servidor**, em que os índices são carregados apenas uma vez, e as consultas são realizadas interactivamente comunicando via *sockets* com o servidor NATools.
- Reduzir o tempo gasto pela comunicação entre o cliente e o servidor é importante para tarefas em bloco, em que o tempo de carregamento é desprezável comparado com o tempo total da tarefa. Para esta abordagem, foi implementada uma **biblioteca dinâmica** para o acesso a recursos NATools, de forma a que este seja um acesso directo a determinado endereço de memória.

A abordagem cliente/servidor permite ainda que se possa paralelizar o processamento a dois níveis:

- no caso de se pretender disponibilizar corpora muito grandes, ou muitos corpora diferentes, contemplar a possibilidade de os distribuir por diferentes servidores (paralelização ao nível do servidor);
- distribuir o processamento também ao nível do cliente, configurando diferentes acções em diferentes máquinas, reduzindo a necessidade de replicação dos corpora. Um exemplo prático é a implementação de memórias de tradução distribuídas (Simões, Guinovart, and Almeida, 2004).

7.3.1 Arquitectura do Servidor

Os recursos disponibilizados pelo NatServer são resultado do processo de codificação e extracção de dicionários probabilísticos de tradução.

Durante este processo é criado um objecto NATools:

$$\begin{aligned}\text{NatObject} &= \text{TU}^{\star}_{\mathcal{A},\mathcal{B}} \\ &\times PTD_{\mathcal{A},\mathcal{B}} \times PTD_{\mathcal{B},\mathcal{A}} \\ &\times \left(\text{W}^2_{\mathcal{A}} \rightharpoonup \mathbb{N}\right) \times \left(\text{W}^3_{\mathcal{A}} \rightharpoonup \mathbb{N}\right) \times \left(\text{W}^4_{\mathcal{A}} \rightharpoonup \mathbb{N}\right) \\ &\times \left(\text{W}^2_{\mathcal{B}} \rightharpoonup \mathbb{N}\right) \times \left(\text{W}^3_{\mathcal{B}} \rightharpoonup \mathbb{N}\right) \times \left(\text{W}^4_{\mathcal{B}} \rightharpoonup \mathbb{N}\right) \\ &\times \text{Key} \rightharpoonup \text{Metadata}\end{aligned}$$

Este objecto contém o corpus alinhado ao nível da frase, os dicionários probabilísticos de tradução respectivos, n-gramas por língua, e meta-informação. É importante salientar que cada um destes objectos pode conter apenas alguns destes recursos. Do mesmo modo, o NatServer está preparado para que possa ser expandido com novos tipos de recursos.

O NatServer é configurado com uma lista de objectos NATools, correspondentes a diferentes corpora e, possivelmente, diferentes línguas.

Na sua versão cliente/servidor, o NatServer funciona como um servidor clássico de *sockets*, respondendo a uma API através de conexões em determinada porta. Na sua versão de biblioteca dinâmica, os objectos NATools ficam disponíveis por uma API standard.

As várias aplicações desenvolvidas tiram partido do NatServer, usando uma mesma API configurável, sendo apenas necessário indicar qual o modo em que deve funcionar (servidor ou biblioteca dinâmica).

A API disponibilizada pelo NatServer[2] corresponde às seguintes funções:

- *list:* listagem dos corpora disponíveis no servidor e das línguas envolvidas;
- *querryattr:* consulta das propriedades de meta-informação associadas a cada corpus;
- *queryptd:* consulta uma entrada num dicionário probabilístico de tradução para determinada língua e corpus;
- *conc:* pesquisa de concordâncias em determinado corpus de acordo com um padrão por língua;

[2]Note-se que a biblioteca Nat::Client implementa um conjunto de funções de ordem-superior que usam internamente a API disponibilizada pelo NatServer.

- *ngrams:* consulta de contextos (n-gramas) de acordo com o padrão e a língua especificados.

Estas funções são detalhadas de seguida.

Meta-Informação

Uma vez que o servidor suporta mais do que um corpus, e não obriga a que as línguas envolvidas sejam as mesmas, é importante que a API fornecida permita consultar este tipo de informação.

Em primeiro lugar, é preciso saber que corpus estão disponíveis. Para isso, a API inclui uma função que retorna a lista de identificadores dos corpora disponíveis, os seus nomes e línguas envolvidas:

$$list :\longrightarrow set\left(\mathbb{N} \times \text{Name} \times \text{Lang}^2\right)$$

O identificador de cada um dos corpora é necessário para o uso das restantes funções disponibilizadas, identificando o corpus a consultar. No geral, pretendeu-se que o servidor fosse *state-free*, para não ser necessário guardar informação sobre cada cliente entre invocações.

Os atributos de meta-informação associadas a um corpus são um conjunto de pares: nome do atributo e valor. Esta meta-informação inclui, por exemplo, o nome do corpus, descrição, línguas envolvidas, número de unidades de tradução e número de palavras em cada uma das línguas. A função *queryattr* permite obter os valores de cada um destes atributos.

$$queryattr : \mathbb{N} \times \text{Attribute} \longrightarrow \text{Value}$$

Os atributos não estão confinados ao conjunto definido pelas ferramentas NATools. O utilizador (ou um programa) pode adicionar meta-informação no ficheiro de configuração de um corpus. Por questões de segurança esta edição de propriedades não está disponível na API.

Dicionários Probabilísticos de Tradução

Como discutido no capítulo 4, ao processar um corpus é criado um dicionário probabilístico constituído por um par de dicionários, $d_{\mathcal{A},\mathcal{B}}$ e $d_{\mathcal{B},\mathcal{A}}$. Isto leva a que nas consultas de entradas em PTD seja necessário especificar, para além do corpus, a língua (ou direcção) a consultar. Para facilitar a interacção com o servidor, o cliente não especifica exactamente em que língua a palavra se encontra, mas se a consulta deve ser realizada na língua de origem ou na língua de destino (de modo a que cliente não precise de saber à partida que línguas estão disponíveis no dicionário).

A função de consulta pode ser formalizada como a invocação de

$$\mathit{queryptd} : \mathbb{N} \times \mathrm{W}_{\mathcal{A}} \times \mathrm{Lang} \longrightarrow \mathrm{Occs} \times \mathrm{Trans}$$

onde

$$\begin{aligned} \mathrm{Occs} &= \mathbb{N} \\ \mathrm{Trans} &= \mathrm{W}_{\mathcal{B}} \rightharpoonup [0..1] \end{aligned}$$

Esta função recebe o identificador do corpus, a palavra a procurar e a língua (origem ou destino) em que a palavra se encontra. O resultado da invocação é constituído pelo número de ocorrências da palavra pesquisada e as suas traduções com a respectiva medida de certeza.

Concordâncias

Existem dois tipos de concordâncias:

- a *pesquisa de palavras* numa ou nas duas línguas, sem que se defina qualquer tipo de ordem relativa entre as palavras procuradas;
- a *pesquisa de padrões* numa ou nas duas línguas, em que as palavras dos padrões devem ocorrer pela ordem especificada. Estes padrões permitem a especificação de buracos (ou *place-holders*), que correspondem a uma qualquer palavra (representados por um asterisco). Assim, a pesquisa de "`vinte e * mil`" encontra ocorrências de "`vinte e cinco mil`" e de "`vinte e três mil`" mas não de "`vinte e três milhões e cinco mil`".

A função de cálculo de concordâncias recebe o identificador do corpus a consultar e a expressão de pesquisa. Esta expressão de pesquisa pode ser um par de sequências de palavras (para a pesquisa simples), ou um par de padrões (para a pesquisa por padrões), de acordo com a assinatura apresentada:

$$conc : \mathbb{N} \times (\mathrm{W}_{\mathcal{A}} + \mathrm{Patt}_{\mathcal{A}})^{\star} \times (\mathrm{W}_{\mathcal{B}} + \mathrm{Patt}_{\mathcal{B}})^{\star} \longrightarrow set\,(\mathrm{S}_{\mathcal{A}} \times \mathrm{S}_{\mathcal{B}})$$

A função devolve um conjunto de unidades de tradução.

n-Gramas

Além da consulta de concordâncias, a possibilidade de obter contagens estatísticas sobre n-gramas é importante. Tarefas como a criação de modelos de língua (como discutido na secção 2.3.2) ou a aprendizagem para previsão de palavras tiram partido de n-gramas extraídos de corpora.

O NatServer suporta a consulta de bigramas, trigramas e tetragramas por língua, quer directamente (consultando quantas vezes determinado n-grama ocorre), quer usando padrões (n-gramas com *placeholders*).

$$ngrams : \mathbb{N} \times \mathrm{Lang} \times \mathrm{Patt} \longrightarrow (\mathrm{W}_{\mathcal{A}}^{\star} \rightharpoonup \mathbb{N})$$

Esta função retorna os n-gramas mais ocorrentes que estejam de acordo com o padrão procurado, juntamente com o seu número de ocorrências.

> A API disponibilizada directamente pelo NatServer é bastante simples. O módulo Perl NAT::Client implementa um conjunto de funções de ordem superior que tornam o desenvolvimento de clientes bastante rápido.

7.3.2 Desenvolvimento de Clientes

É crucial a existência de uma API que permita o desenvolvimento rápido e simples de clientes. Para isso, o pacote NATools inclui um módulo Perl (`NAT::Client`) com funções de alto nível para a interacção com o NatServer.

A secção 6.1 apresentou um conjunto de aplicações Web, implementado utilizando esta API. Esta secção mostra pequenos exemplos de clientes como motivação para a importância da existência desta API no desenvolvimento de protótipo.

Exemplo 1: Sistema de Concordâncias

Este exemplo implementa um sistema de concordâncias básico. Recebe na linha de comandos a sequência de palavras a procurar e realiza a pesquisa na língua de origem.

O programa completo não usa mais do que oito linhas de código:

```
1   use NAT::Client;
2   $server = NAT::Client->new( PeerAddr => 'localhost' );
3   $pattern = join(" ",@ARGV);
4   $concs = $server->conc({crp=>1}, $pattern);
5   for my $tu (@$concs) {
6       print "$tu->[0]\n";
7       print "$tu->[1]\n";
8       print "\n"
9   }
```

linha 1: carregar a API para a realização de consultas no servidor;
linha 2: criar um objecto de acesso ao servidor, especificando o endereço onde se encontra o NatServer;
linha 3: construir o padrão de pesquisa usando os argumentos indicados na linha de comandos;
linha 4: calcular a lista de concordâncias invocando o método *conc* no servidor. Neste exemplo é consultado o corpus com identificador 1, e a pesquisa é realizada na língua de origem. Se assim não fosse, seria necessário indicar a língua em causa;
linha 5–8: iterar sobre todas as concordâncias e imprimi-las.

Ao criar o objecto para ligação ao servidor é possível especificar que se pretende usar o NatServer como biblioteca dinâmica. Para isso basta alterar a invocação do construtor:

```
$server = NAT::Client->new(Local=>'/corpora/EurLex-PT-EN');
```

Segue-se um extracto do resultado da execução deste programa:

```
1  $ example parlamento europeu

2  Declaro reaberta a sessão do Parlamento Europeu , que tinha sido interrompida ...
3  Declaro reanudado el período de sesiones del Parlamento Europeo , interrumpido...

4  Senhora Presidente , coincidindo com a primeira sessão deste ano do Parlamento...
5  Señora Presidenta , coincidiendo con el primer período parcial de sesiones de ...
```

Exemplo 2: Palavras Relacionadas

Este exemplo já foi apresentado na secção 4.4.2. No entanto, nessa secção ainda não tinha sido apresentado o funcionamento do NatServer, pelo que o retomamos e explicamos detalhadamente. Segue-se o programa completo para o cálculo de palavras relacionadas.

```
1    use NAT::Client;

2    my $client = NAT::Client->new( Local => "EuroParl-PT-EN" );
3    my %r = ();

4    my $a1 = $client->ptd( "povo" );
5    for my $b1 (keys %{$a1->[1]}) {
6      my $c = $client->ptd( { from => 'target' }, $b1);
7      for my $d (keys %{$c->[1]}){
8        $r{$d} +=  $a1->[1]{$b1} * $c->[1]{$d};
9      }
10   }
11   for((sort {$r{$b} <=> $r{$a}} keys %r)[0..9]) {
12       printf "  %15s %.3f \n", $_, $r{$_}*100
13   }
```

linha 1: carregar a API para consulta ao servidor;
linha 2: criar um objecto de acesso ao NatServer em modo local;
linha 3: declarar o *array associativo* de resultados;
linha 4: consultar o dicionário probabilístico de tradução para determinada palavra (neste exemplo, a palavra "*povo*");
linha 5: iterar sobre as traduções da palavra em causa;
linha 6: para cada tradução, obter a sua entrada no dicionário probabilístico de tradução inverso;

linha 7: adicionar cada tradução da tradução à lista de resultados, associando-lhe uma medida de probabilidade;
linha 10: iterar sobre os resultados, imprimindo-os.

Para além do uso de um servidor NatServer, ou de uma biblioteca dinâmica, a API do módulo NAT::Client permite ainda a consulta de um dicionário probabilístico de tradução em formato textual (estrutura de dados Perl serializada com Data::Dumper).

```
my $c = NAT::Client->new(LocalDumper=>"EuroParl-PT-EN/PT.dmp");
```

Deste modo, qualquer programa que precise apenas de PTD pode funcionar exactamente com o mesmo código usando o servidor, biblioteca dinâmica ou um PTD em formato textual.

> A reutilização do mesmo código para a consulta de recursos em diferentes arquitecturas (cliente/servidor, biblioteca ou formato textual) permite uma maior facilidade no desenvolvimento e teste de aplicações.

7.3.3 Métricas de Eficiência

Esta secção apresenta algumas métricas para caracterizar a eficiência do uso do NatServer em ambiente cliente/servidor ou de biblioteca dinâmica. Os testes apresentados correspondem a um servidor com três corpora carregados: EuroParl PT:ES, EuroParl PT:EN e EuroParl PT:FR (cerca de um milhão de unidades de tradução em cada).

Os testes correspondem a 100 000 pedidos ao servidor das vinte primeiras concordâncias. Foram executados testes com concordâncias de palavras e com concordâncias de padrões, de modo a calcular o tempo médio de resposta a um pedido (e o número de pedidos respondido por segundo). A tabela 7.1[3] resume os valores obtidos.

[3] O servidor usava cerca de 600 megabytes de memória. O computador usado é um Intel Pentium IV, 3 GHz com 2 GB de RAM.

		seg/pedido	pedido/seg	ocor
1	cão	0.038	26.027	40
2	europa	0.010	98.090	36532
3	parlamento europeu	0.036	27.131	23841
4	"parlamento europeu"	0.036	27.485	23841
5	"europeu parlamento"	1.474	0.68	23841
6	PTD(parlamento)	0.001	1676.45	–

Os testes 1, 2 e 3 são referentes a concordâncias de palavras. Os testes 4 e 5 são referentes a concordâncias de padrões. O teste 6 é referente à consulta de um dicionário probabilístico de tradução.

Tabela 7.1: Análise de eficiência do NatServer.

Os testes 1 e 2 são muito semelhantes, mudando apenas a palavra procurada. Esta comparação é importante já que o servidor armazena o corpus por fatias, e carrega uma fatia de cada vez (por questões de gestão de memória). Assim, se uma das palavras aparece muitas vezes no corpus (como a palavra "*europa*"), a primeira fatia carregada do disco contém, em princípio, as 20 concordâncias pedidas. Por sua vez, se a palavra ocorre poucas vezes (como a palavra "*cão*"), é provável que seja necessário carregar mais do que uma fatia para encontrar as 20 ocorrências, pelo que o tempo de resposta será maior.

Os testes 3 e 4 comparam o uso de concordâncias de palavras ou de padrões, e mostram que o algoritmo de pesquisa está a ser praticamente o mesmo (uma vez que a grande maioria das ocorrências das palavras "*parlamento*" e "*europeu*" na mesma unidade de tradução, corresponde ao termo multi-palavra "*parlamento europeu*").

O teste 5 obriga à consulta de todo o corpus, já que não existe qualquer ocorrência do padrão "*europeu parlamento.*" Este par de palavras ocorre 23 841 vezes, mas nenhuma pela ordem pedida. Logo, o sistema terá de realizar 23 841 comparações de palavras, e de carregar todas as fatias do corpus para memória (uma de cada vez).

O teste 6 é um teste de cariz diferente uma vez que mede o tempo demorado a consultar uma entrada num dicionário probabilístico de tradução. A palavra procurada é indiferente já que todas as entradas têm o mesmo tamanho (dado o número de traduções limitado) e o sistema

de indexação é bastante eficiente, baseado em pesquisa binária.

A tabela 7.2 sumariza alguns testes de comparação entre a arquitectura cliente/servidor e o uso de uma biblioteca dinâmica, para a consulta de dicionários probabilísticos de tradução.

	pedido/seg
via Servidor	1 737.92
via Biblioteca — corpus carregado uma vez	45 454.55
via Biblioteca — corpus carregado por consulta	0.70

Tabela 7.2: Número de pedidos respondidos por segundo usando uma arquitectura cliente/servidor ou uma biblioteca dinâmica (na consulta de entradas de um PTD).

O servidor é capaz de responder a mais de 1700 pedidos por segundo, de consulta a um dicionário probabilístico de tradução. No caso de se usar um corpus local via biblioteca dinâmica, já é possível consultar 45 454 entradas por segundo. Note-se que este tempo considera que o corpus e dicionários foram carregados para memória apenas uma vez. Se o corpus e dicionário forem carregados por cada consulta, só será possível responder a 0.7 pedidos por segundo.

A abordagem correcta (cliente/servidor vs biblioteca dinâmica) depende em grande parte dos objectivos da aplicação em desenvolvimento.

Numa aplicação Web o tempo de carregamento de índices é incomportável (especialmente se considerarmos o caso em que existe mais do que um utilizador a realizar consultas, já que levaria a *time-out* nos acessos HTTP). No entanto, não há necessidade de grande eficiência no tempo de resposta para cada pedido. O importante é a obtenção de uma resposta em tempo finito. Neste tipo de aplicações a abordagem Cliente/Servidor é mais adequada.

Por sua vez, numa aplicação que realize muitas consultas (p.ex. para a extracção de exemplos de um corpus), o tempo de carregamento dos índices é desprezável, e o importante é que cada resposta seja obtida no menor tempo possível. Para este tipo de aplicações o uso de uma

biblioteca dinâmica traz grandes vantagens.

> A possibilidade de aceder aos recursos criados usando duas arquitecturas diferentes, mas com uma mesma API, permite que o programador possa tirar partido de toda a eficiência desejada sem necessidade de usar duas formas distintas de acesso aos recursos.

7.4 Escalonamento e Paralelização de Tarefas

Como discutido nas secções 7.1 e 7.2, existem estratégias de decomposição estrutural e de partição de tarefas que permitem dividir uma tarefa grande em várias sub-tarefas pequenas.

Depois da decomposição de uma tarefa, é necessário executar cada uma das sub-tarefas. No caso da decomposição estrutural, as sub-tarefas têm de ser executadas por ordem, uma vez que têm uma dependência directa (fazem parte de uma *pipeline* de tarefas). Por sua vez, a decomposição por partição permite que cada uma das partes seja processada de forma independente (uma vez que cada tarefa estará a processar uma parte diferente), e portanto, possam ser paralelizadas.

Para a possível paralelização de tarefas é necessário definir uma topologia de processamento: quais as inter-dependências entre cada uma das pequenas tarefas.

A ferramenta `Makefile::Parallel`[4] (Fonseca, 2007; Simões, Fonseca, and Almeida, 2007) foi desenhada como uma linguagem de domínio específico para a especificação de inter-dependências entre tarefa, e um escalonador de tarefas baseado no grafo de dependências descrito.

O escalonador do `Makefile::Parallel` (`pmake`) interpreta a especificação de dependências entre tarefas e executa-as em paralelo sempre

[4]Este trabalho foi desenvolvido em parceria com o Rúben Fonseca, na altura aluno do último ano da licenciatura em Engenharia de Sistemas e Informática, a quem mais uma vez agradeço o ânimo e a ajuda.

que possível. O nível de paralelismo depende do número de processadores disponíveis (em máquinas multi-processador ou em *clusters* de computadores).

A sintaxe escolhida para a linguagem de dependências é inspirada no formato dos ficheiros *Makefile*, com a diferença de que esta linguagem não especifica dependências entre ficheiros a construir, mas dependências entre tarefas (e informação de como as executar). Além disso, inclui um conjunto de elementos específicos para tirar partido do escalonador de um *cluster*, como sejam o tempo previsto para a completação da tarefa.

O desenvolvimento do `Makefile::Parallel` seguiu os seguintes requisitos:

- usar uma **linguagem compacta e formal** para especificar dependências entre processos;
- reutilizar **sintaxes conhecidas**, usadas em tarefas semelhantes;
- **embeber outras linguagens** para tirar partido da sua expressividade. Na `pmakefiles` podemos especificar acções nas linguagens Bash e Perl, que são linguagens reflexivas, e portanto permitem a alteração do seu código em tempo de execução;
- suportar **regras dinâmicas**: em algumas situações só podemos definir uma regra depois da anterior ter terminado (por exemplo, por faltar um valor calculado na tarefa anterior);
- suportar **regras paramétricas**, que possam ser instanciadas com diferentes valores, de forma a gerar automaticamente um grande número de regras a partir de uma mesma definição (o que permite a utilização dinâmica da decomposição por partição);
- **disponibilizar informação** como relatórios, tabelas de duração de processos e grafos de dependência para facilitar a análise da eficiência das várias ferramentas e da topologia definida.

O algoritmo de escalonamento do `pmake` é bastante simples. A especificação é analisada e o grafo calculado. A cada passo, o escalonador verifica que processos podem ser executados e executa-os. Sempre que um processo termina, é calculada a lista de processos que dele dependiam e, caso não tenham mais dependências, são iniciados.

Segue-se a descrição formal da linguagem de domínio específico, e alguns detalhes relativos à implementação do escalonador.

7.4.1 A Linguagem

Como foi referido, a linguagem especifica dependências entre tarefas usando uma sintaxe semelhante à usada pelas *Makefiles*, e pode ser vista como a formalização de uma rede de Pert.

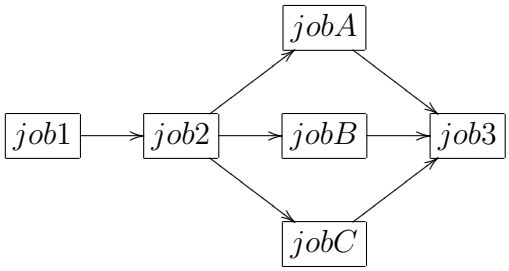
$$
\begin{array}{rcl}
jobs & \rightarrow & job^{+} \\
job & \rightarrow & jobName \text{ `:'} \; deps \; wallTime \; nrCpus \; actions \\
walltime & \rightarrow & \text{`(' } \mathbf{TIME} \text{ `)'} \\
nrCpus & \rightarrow & \epsilon \mid \text{`[' } \mathbf{INT} \text{ `]'} \\
jobName & \rightarrow & \mathbf{ID} \mid \mathbf{ID} \; \mathbf{VAR} \\
deps & \rightarrow & jobName^{\star} \\
actions & \rightarrow & action^{+} \\
action & \rightarrow & shellCmd \mid perlCmd \mid setDefinition \\
shellCmd & \rightarrow & \mathbf{TAB} \; \mathbf{SHELL} \\
perlCmd & \rightarrow & \mathbf{TAB} \text{ `sub\{' } \mathbf{PERL} \text{ `\}'} \\
setDefinition & \rightarrow & \mathbf{TAB} \; \mathbf{VAR} \text{ `<-' } \mathbf{SHELL} \\
 & \mid & \mathbf{TAB} \; \mathbf{VAR} \text{ `<-' `sub\{' } \mathbf{PERL} \text{ `\}'}
\end{array}
$$

Figura 7.2: Gramática simplificada da linguagem `Makefile::Parallel`.

A figura 7.2 mostra a gramática simplificada da linguagem reconhecida pela ferramenta `Makefile::Parallel`. Cada regra nesta linguagem corresponde a um processo e pode definir um ou mais arcos através das suas dependências. A regra é composta por um nome, a descrição de como esse processo se executa (um conjunto de acções), a lista de dependências (processos que têm de ser executados previamente), o tempo previsto para a completação da tarefa (elemento importante para o escalonador do *cluster* saber em que fila de trabalhos deve submeter a tarefa) e o número de processadores necessários (uma tarefa por si só

pode ser paralela).

Considere-se que o seguinte exemplo artificial de um grafo de dependências entre tarefas:



A especificação (simplificada: para maior legibilidade omitiram-se as acções semânticas e as linhas em branco entre regras) pode ser descrita por:

```
1          job1:
2          job2: job1
3          jobA: job2
4          jobB: job2
5          jobC: job2
6          job3: jobA jobB jobC
```

Suporte para acções em Bash e Perl

Embora a maior parte das aplicações que se deseja paralelizar sejam programas binários, ou ferramentas independentes, é importante existir uma linguagem expedita para realizar a cola entre as várias ferramentas, e os resultados obtidos, bem como para preparar o ambiente de execução. Para este conjunto de tarefas as linguagens ditas *de scripting* são as mais indicadas por permitirem de forma concisa especificar este tipo de tarefas.

Com o objectivo de permitir acções semânticas definidas integralmente ou parcialmente em Perl e em Bash, foi adicionado algum açúcar sintáctico à linguagem para as diferenciar.

Suporte para regras paramétricas

As regras paramétricas estão fortemente ligadas à decomposição de tarefas por partição, replicação da função de processamento, e posterior junção de resultados. Sem a possibilidade de definir regras paramétricas seria impossível a partição de tarefas num número de sub-tarefas dependente do tamanho dos dados a processar.

Por exemplo, as tarefas de codificação de corpora, extracção de dicionários probabilísticos de tradução e de extracção de exemplos, podem ser divididas em sub-tarefas independentes que processem partes distintas do corpus. No entanto, o número de sub-tarefas é dependente do tamanho do corpus, e deve ser calculado dinamicamente.

Enquanto que para um corpus pequeno são necessárias apenas uma ou duas fatias para a extracção de dicionários, para um corpus como o EuroParl são precisas cerca de 25. Como este valor é variável (depende do tamanho do corpus), seria necessário escrever uma *makefile* diferente para cada corpus a processar. Mesmo que assim fosse, enquanto que escrever uma *makefile* com uma ou duas regras é trivial, escrever uma com mais de 25 regras leva a que seja fácil cometer erros.

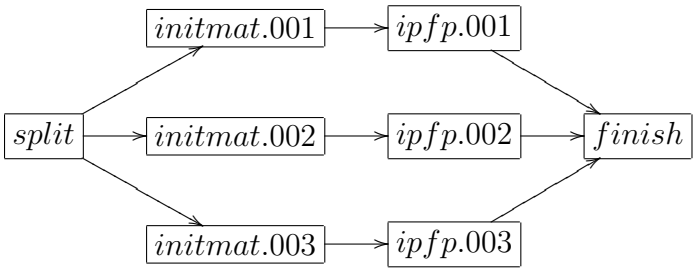
As regras paramétricas usam variáveis que são instanciadas com valores de um conjunto definido em tempo de execução por uma regra anterior. Consideremos uma variável i que seja definida pelo conjunto $i = \{001, 002, 003\}$. Então, as regras:

```
1       initmat.$i: split (5:00)
2             initmat crp.$i mat.$i

3       ipfp.$i: initmat.$i (10:00)
4               run-ipfp mat.$i ipfp.$i

5       finish: ipfp.$i (5:00)
6               join-results @i
```

seriam expandidas para:

```
1       initmat.001: split (5:00)
2               initmat crp.001 mat.001

3       initmat.002: split (5:00)
4               initmat crp.002 mat.002

5       initmat.003: split (5:00)
6               initmat crp.003 mat.003

7       ipfp.001: initmat.001 (10:00)
8               run-ipfp mat.001 ipfp.001

9       ipfp.002: initmat.002 (10:00)
10              run-ipfp mat.002 ipfp.002

11      ipfp.003: initmat.003 (10:00)
12              run-ipfp mat.003 ipfp.003

13      finish: ipfp.001 ipfp.002 ipfp.003 (5.00)
14              join-results 001 002 003
```

Esta expansão corresponde ao seguinte grafo de dependências:



As variáveis correspondem a conjuntos de valores e não apenas ao número de fatias a criar, já que por vezes é necessário definir regras com valores específicos (por exemplo, o *offset* correspondente à porção do corpus que deve ser processado) que deste modo são facilmente descritas.

Ainda em relação aos conjuntos, foi definida uma notação especial `@i` que pode ser usada nas acções semânticas (Perl ou Bash), e que são

expandidas com todos os valores do conjunto definido por essa variável (ver exemplo da regra `finish`).

7.4.2 O Escalonador

O escalonador (e interpretador da linguagem) foi escrito em Perl, e o reconhecedor da linguagem foi escrito em YAPP (Desarmenien, 2001), uma versão Perl do bem conhecido `yacc`. O facto de termos usado Perl levou a que o desenvolvimento fosse mais rápido.

Foi considerado crucial que o escalonador pudesse tirar partido de diferentes arquitecturas, de acordo com a plataforma onde fosse executado. Inicialmente implementaram-se dois escalonadores, um para ser usado numa computador normal, e um outro para ser usado num *cluster* com suporte para o escalonador *Portable Batch System* (PBS). No entanto, existem planos para implementações futuras de outros escalonadores, como um escalonador entre várias máquinas inter-ligadas com SSH.

Para facilitar a definição de novos escalonadores, foi criada uma classe abstracta que cada plataforma tem de implementar, e em que deve definir os seguintes métodos:

Launch usado para despoletar um novo processo na plataforma em causa;

Poll para obter o estado actual de determinado processo (parado ou a ser executado);

Interrupt para interromper um processo que esteja a ser executado;

GetID para obter um identificador único para cada um dos processos;

CanRun para confirmar com o escalonador da plataforma se pode ser despoletado um novo processo (ou se todos os processadores estão a ser usados).

Seguidamente, descrevem-se os dois subsistemas implementados: o escalonador local e o escalonador PBS.

Escalonador Local

Em situações de processamento de corpora pequenos, ou para debug, o uso de um *cluster* é desnecessário. É importante que possamos usar as mesmas *makefiles* em computadores pessoais, sem precisar de alterações. Esta foi a principal motivação para se criar um escalonador para esta arquitectura.

Este escalonador executa processos de acordo com a quantidade de processadores disponíveis. Numa máquina com um único processador as várias tarefas vão ser executadas sequencialmente, sem tirar qualquer partido do paralelismo. Se a máquina tiver mais do que um processador, é possível tirar algum partido de paralelismo, indicando ao escalonador o número de processadores disponíveis (com um parâmetro `-local=n`).

Escalonador PBS

Cada vez mais se pode considerar que qualquer *cluster* inclui uma ferramenta do tipo *Portable Batch System* (PBS), que suporta um conjunto bem definido de operações sobre tarefas. O PBS é um escalonador que permite ao *cluster* funcionar como um sistema de tarefas, em que são alocados recursos, como tempo de CPU e memória, numa base orientada ao processo. Cada um destes processos é colocado numa fila de espera, e executado assim que os recursos requisitados estejam disponíveis, e de acordo com um conjunto de prioridades estabelecidas pelos administradores (Sloan, 2004).

Embora existam várias implementações de sistemas PBS (OpenPBS, PBS-Pro, TORQUE), todas elas obedecem ao mesmo interface original pelo que se torna simples de usar numa grande variedade de *clusters*.

O `Makefile::Parallel` foi testado no SeARCH, um *cluster* formado por cerca de 180 CPUs em 50 nodos, disponível do Departamento de Informática da Universidade do Minho. O SeARCH usa o sistema operativo Linux, com o PBS TORQUE. Este escalonador é responsável por consultar uma lista de espera (criada com comandos PBS), e verificar quando é que estes terminam. Permite também que se possam matar

processos quando necessário. O escalonador do `Makefile::Parallel` interage com o escalonador PBS para executar tarefas.

Escalonador Genérico

O Escalonador Genérico (e o único que realmente faz escalonamento) incorpora todo o algoritmo de análise da especificação, construção do grafo de dependências, e é o responsável por comunicar com os subsistemas para despoletar tarefas, e verificar o estado das mesmas.

A interface com o escalonador é feita pela aplicação `pmake` que, quando invocada sem opções, interpreta uma especificação e executa-a no CPU local, despoletando um processo de cada vez. Durante a execução vai indicando que processos estão a ser executados, quando terminam, que tempo demoraram, e quais as regras paramétricas que foram criadas.

No final do processamento de toda a especificação, é construído um relatório que, para cada processo, indica a data de início e de término do processo, e tempo decorrido. Também gera um grafo de dependências que pode ser usado para analisar a topologia de processos.

Segue-se uma descrição sucinta das opções reconhecidas pela aplicação `pmake`:

`-debug` adiciona verbosidade às informações impressas durante a execução, e não remove os ficheiros usados para submeter processos (no caso do escalonador PBS);

`-continue` permite retomar a execução de uma `pmakefile` a partir de uma tarefa que tenha falhado (por uma falha na aplicação ou simplesmente no ambiente, como a falta de espaço em disco);

`-local=[`n`]` força o uso do escalonador local (embora seja o usado por omissão), permitindo a especificação do número de processadores ou de *cores* disponíveis;

`-pbs` indica que deve ser usado o escalonador PBS;

`-clean` permite a remoção de ficheiros auxiliares gerados com os *outputs* da execução (*standard error* e *standard output*).

7.4.3 Caso de estudo: Extracção de PTD

O nosso caso de estudo (e motivação) foi, como referido, o processamento de corpora de grandes dimensões com o NATools. Os processos de codificação de corpora e extracção de dicionários probabilísticos de tradução eram, já por si, processos independentes, aplicados a diferentes fatias de um corpus. A abordagem para a extracção de exemplos foi semelhante, não tanto por não ser possível executar o processo sobre todo o corpus, mas para tirar partido do paralelismo, reduzindo o tempo de execução.

Execução

A figura 7.3 mostra uma especificação do processo de extracção de dicionários probabilísticos. O processo completo também realiza o cálculo de n-gramas e a extracção de exemplos, mas essas regras foram omitidas.

Esta especificação começa por executar o processo `codify` que calcula o número de fatias em que o corpus deve ser cortado de acordo com o número de unidades de tradução. Segue-se todo o processo de extracção de dicionários probabilísticos de tradução com regras paramétricas (`initmat`, `ipfp`, `postipfp` e `postbin`). Posteriormente, dois processos somam os dicionários das várias fatias (`dicA` e `dicB`). O processo final gera os dicionários em formato textual (`dump`).

A figura 7.4 mostra um extracto das mensagens que o escalonador vai enviando para o utilizador, de forma a saber quando e que processos foram despoletados, quando terminam e quanto tempo demoraram.

A figura 7.5 mostra um grafo (gerado automaticamente com auxílio da ferramenta GraphViz (Gansner and North, 2000)) de uma execução do `Makefile::Parallel` sobre o corpus JRC-Acquis (usando uma especificação um pouco diferente da apresentada na figura 7.3).

```
codify: (20:00:00)
        nat-codify -id=EurLex EurLex-PT EurLex-EN
        i <- sub{ $nr = `cat EurLex/nat.cnf |grep nr-chunks|cut -f 2 -d "="`;
                  printf("%03d\n",$_) for (1..$nr); }

initmat$i: codify  (20:00:00)
        nat-initmat EurLex/source.$i.crp EurLex/target.$i.crp EurLex/mat.$i.in

ipfp$i: initmat$i (20:00:00)
        nat-ipfp 5 EurLex/source.$i.crp EurLex/target.$i.crp \
                   EurLex/mat.$i.in EurLex/mat.$i.out
        rm -f EurLex/mat.$i.in

postipfp$i: ipfp$i (20:00:00)
        nat-mat2dic EurLex/mat.$i.out EurLex/dict.$i
        rm -f EurLex/mat.$i.out

postbin$i: postipfp$i (20:00:00)
        nat-postbin EurLex/dict.$i \
                    EurLex/source.$i.crp.partials EurLex/target.$i.crp.partials \
                    EurLex/source.lex EurLex/target.lex \
                    EurLex/source-target.$i.bin EurLex/target-source.$i.bin
        rm -f EurLex/dict.$i

dicA: postbin$i (20:00:00)
        for a in @i; do \
           nat-dict add EurLex/source-target.bin EurLex/source-target.${a}.bin; \
        done
        for a in @i; do rm -f EurLex/source-target.${a}.bin; done

dicB: postbin$i (20:00:00)
        for a in @i; do \
           nat-dict add EurLex/target-source.bin EurLex/target-source.${a}.bin; \
        done
        for a in @i; do rm -f EurLex/target-source.${a}.bin; done

dump: dicA dicB (20:00:00)
        nat-dumpDicts -self EurLex
```

Figura 7.3: Especificação `Makefile::Parallel` para a extracção de dicionários probabilísticos de tradução.

```
2006/12/12 10:49:22 The job "ipfp005" is ready to run. Launching
2006/12/12 10:49:22 Launched "ipfp005" (23996)
2006/12/12 10:49:52 Process 23996 (ipfp005) has terminated [30s]
2006/12/12 10:49:52 The job "postipfp005" is ready to run. Launching
2006/12/12 10:49:52 Launched "postipfp005" (23997)
2006/12/12 10:50:02 Process 23997 (postipfp005) has terminated [10s]
2006/12/12 10:50:02 The job "postbin005" is ready to run. Launching
2006/12/12 10:50:02 Launched "postbin005" (23998)
2006/12/12 10:50:12 Process 23991 (initmat001) has terminated [1m]
2006/12/12 10:50:12 Process 23998 (postbin005) has terminated [10s]
2006/12/12 10:50:12 The job "ipfp001" is ready to run. Launching
2006/12/12 10:50:12 Launched "ipfp001" (23999)
```

Figura 7.4: Mensagens do `Makefile::Parallel` durante a execução.

Estes grafos também são cruciais em situações em que ocorrem erros, em que a tarefas que falham são marcadas a outra cor.

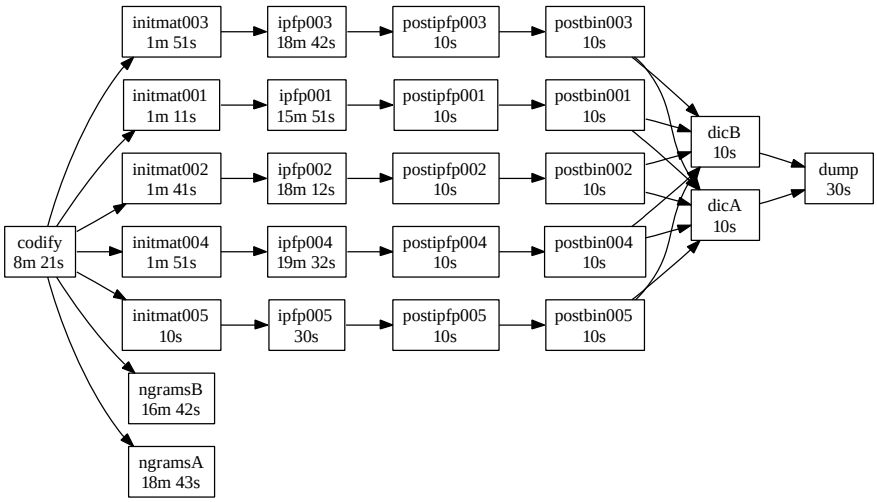


Figura 7.5: Grafo de dependências entre processos paralelos.

Juntamente com o grafo, é também criado um ficheiro com uma descrição temporal de todos os processos executados, tal como apresentado na figura 7.6.

```
ID          Start Time          End Time            Elapsed
codify      2006-12-12T10:41:10 2006-12-12T10:49:11 8m 1s
ngramsA     2006-12-12T10:49:11 2006-12-12T11:07:46 18m 34s
ngramsB     2006-12-12T10:49:11 2006-12-12T11:05:44 16m 33s
initmat001  2006-12-12T10:49:11 2006-12-12T10:50:12 1m
initmat002  2006-12-12T10:49:11 2006-12-12T10:50:43 1m 31s
initmat003  2006-12-12T10:49:11 2006-12-12T10:51:03 1m 51s
initmat004  2006-12-12T10:49:11 2006-12-12T10:50:53 1m 41s
initmat005  2006-12-12T10:49:11 2006-12-12T10:49:21 10s
ipfp005     2006-12-12T10:49:22 2006-12-12T10:49:52 30s
postipfp005 2006-12-12T10:49:52 2006-12-12T10:50:02 10s
postbin005  2006-12-12T10:50:02 2006-12-12T10:50:12 10s
[...]
```

Figura 7.6: Relatório de execução do `Makefile::Parallel`.

Métricas

A especificação de processamento de corpora actualmente em produção, inclui mais de 20 regras, em que cerca de 14 são paramétricas. Para o maior corpus processado (EurLex), foram submetidos ao cluster mais de um milhar de processos, e usados mais de 50 gigabytes de espaço em disco durante o pico de execução. O tempo de execução é de cerca de 12 horas, comparado com o tempo de execução em sequência de quase duas semanas[5].

A `Makefile::Parallel` foi apresentada num encontro da área (Simões, Fonseca, and Almeida, 2007) onde foi elogiada pela sua simplicidade e flexibilidade, bem como pelo facto de estar publicamente disponível quer para uso, quer para alteração, a partir do arquivo de módulos Perl CPAN (Comprehensive Perl Archive Network).

[5]O respectivo grafo é demasiado grande para ser aqui incluído. Os interessados podem visitar o gabinete 3.06 do Departamento de Informática, onde o grafo de mais de metro e meio de altura está actualmente a servir de papel de parede.

A Título de Conclusão

A decomposição estrutural e decomposição por partição são essenciais para o desenvolvimento de aplicações composicionais e escaláveis, dividindo uma aplicação complexa em pequenas tarefas.

Depois de aplicar os métodos de decomposição, obtém-se um conjunto de tarefas que têm de ser executadas por determinada ordem. Algumas destas tarefas podem ser executadas em paralelo de forma completamente independente.

Para a ordenação destas tarefas num grafo de dependências foi criada uma linguagem de domínio específico (`Makefile::Parallel`), e um processador (`pmake`) que escalona as várias tarefas.

Esta abordagem, que consiste na divisão de um problema, o seu processamento por partes, e reunião de resultados, foi escolhida bem antes de se apostar no uso de um *cluster*, e demonstrou ser eficaz quer no processamento sequencial quer no processamento paralelo.

A decomposição de numa arquitectura Cliente/Servidor permite maior facilidade na paralelização de processos, com o uso de vários servidores ou vários clientes. Para isso, foi desenvolvido o NatServer, um servidor de diferentes tipos de recursos. Permite a consulta de concordâncias, dicionários probabilísticos de tradução e n-gramas. Foi desenvolvido de forma modular o que lhe permite uma fácil expansão.

O NatServer foi desenvolvido tendo em vista vários ambientes, permitindo a abordagem Cliente/Servidor mas também a possibilidade de uso da mesma API com uma biblioteca dinâmica. Como foi discutido, cada uma destas arquitecturas tem as suas vantagens, e devem ser aplicadas em diferentes situações.

Foi desenvolvida uma API de ordem superior (`NAT::Client`) que tira partido da API suportada pelo NatServer e permite a construção rápida de protótipos.

Capítulo 8

Conclusões e Trabalho Futuro

> *The Babel fish is small, yellow and leech-like, and probably the oddest thing in the Universe. It feeds on brainwave energy received not from its own carrier but from those around it. It absorbs all unconscious mental frequencies from this brainwave energy to nourish itself with. It then excretes into the mind of its carrier a telepathic matrix formed by combining the conscious thought frequencies with nerve signals picked up from the speech centres of the brain which has supplied them. The practical upshot of all this is that if you stick a Babel fish in your ear you can instantly understand anything said to you in any form of language.*
>
> *Douglas Adams*
> *"The Hitch-Hikers Guide To The Galaxy"*

Ao longo deste trabalho foram apresentados métodos diversos para a extracção de recursos de tradução usando corpora paralelos. Os recursos obtidos explicitam relacionamentos bilingues entre palavras, termos ou segmentos de palavras, que podem ser usados para as mais diversas finalidades.

É importante referir que se deu especial ênfase na extracção de re-

cursos para a língua portuguesa, e que se constatou que existe muito trabalho a realizar nesta área.

Seguidamente, apresentaremos uma reflexão sumária sobre o trabalho realizado, dividindo-a em conclusões gerais, contribuições (de variados tipos) e trabalho futuro.

8.1 Conclusões

Foi possível retirar um conjunto de conclusões sobre as abordagens usadas e os recursos obtidos. Esta secção enumera as conclusões que nos parecem mais relevantes:

- O trabalho realizado permite concluir que o **tamanho dos corpora não são um factor limitativo** no seu processamento.
- Foram propostas metodologias para adaptar **algoritmos** de forma a que sejam **escaláveis**, permitindo assim o processamento de corpora paralelos de grandes dimensões.
- Foi demonstrada a exequibilidade da **extracção de dicionários probabilísticos de tradução** referente a todas as palavras de determinado corpus, **independentemente do seu tamanho**.
- Mostraram-se diferentes abordagens para o pré-processamento de dicionários probabilísticos de tradução que permitem a **extracção de dicionários específicos** de qualidade, como sejam dicionários de verbos ou de entidades mencionadas.
- É possível a **extracção de uma grande variedade de recursos bilingues de qualidade** usando corpora paralelos e dicionários probabilísticos de tradução. Por exemplo, os dicionários probabilísticos de tradução mostraram ser uma fonte eficaz para a detecção de âncoras entre línguas, permitindo uma maior robustez na análise de unidades de tradução.
- Concluímos que os **recursos bilingues** extraídos **permitem a extracção de novos recursos**. Assim como os dicionários probabilísticos de tradução foram usados para a extracção da maior parte dos recursos apresentados, outros recursos, como a termino-

logia bilingue extraída usando padrões de alinhamento, mostraram ser versáteis para a extracção de dicionários a usar em ferramentas de tradução automática (como foi visto na sua aplicação ao `Text::Translate`), e para a generalização de exemplos.

- Embora alguns dos métodos necessitem de informação específica para as línguas envolvidas (como listas de palavras-marca ou padrões de tradução), a **generalidade dos métodos** são **independentes de língua**.
- Os **recursos** obtidos são **úteis para uma grande diversidade de problemas** e áreas de investigação. No entanto, precisam quase sempre de pequenas adaptações locais para se integrarem na ferramenta ou finalidade em causa. Deste modo, foi disponibilizada uma *API de Ordem Superior* para o processamento eficiente de recursos de tradução.

8.2 Contribuições

Esta dissertação teve como principal objectivo a extracção de recursos de tradução, tendo um especial cuidado na extracção dos recursos que envolvem a língua portuguesa. Neste sentido, as principais contribuições deste trabalho correspondem a:

- um **conjunto de recursos** criados e extraídos pelos vários métodos apresentados (secção 8.2.1);
- **algoritmos e métodos** para a análise da extracção de dicionários probabilísticos de tradução, extracção de exemplos com base na Hipótese das Palavras-Marca, extracção de exemplos por cálculo da matriz de tradução, e extracção de terminologia base em padrões de alinhamento (secção 8.2.2);
- **ferramentas desenvolvidas** e incluídas no pacote NATools e `Makefile::Parallel` (secção 8.2.3).

8.2.1 Criação e Disponibilização de Recursos

Ao longo deste trabalho sentiu-se necessidade da criação de recursos, e em particular, de corpora paralelos. Neste sentido, investiu-se na detecção e extracção automática de corpora paralelos a partir da Web, de que o corpus EurLex é exemplo.

Igualmente importante é a disponibilização dos recursos. Esta disponibilização foi realizada de três formas:

- através de uma interface Web integrada que permite a consulta dos vários tipos de recursos calculados;
- através do *download* dos corpora paralelos, dicionários probabilísticos de tradução, listas de exemplos de tradução e de entradas terminológicas;
- através da criação de recursos prontos a utilizar por ferramentas específicas como sejam os dicionários *StarDict* para consulta *off-line* e integrada de recursos bilingues.

8.2.2 Contribuições Científicas

Em relação às contribuições científicas relativas a métodos e algoritmos, devem-se salientar as seguintes:

- a sistematização dos **métodos de decomposição** estrutural ou por partição, replicação e junção, que permitem o desenvolvimento de aplicações escaláveis sobre grandes corpora, facilitando a sua paralelização e distribuição;
- a demonstração de que é possível a aplicação do **algoritmo de extracção de dicionários** probabilísticos de tradução a corpora de qualquer tamanho sem qualquer limitação em termos de cardinalidade do domínio do dicionário final;
- a realização de várias experiências no **pré-processamento de corpora** para a extracção de dicionários probabilísticos de tradução com diferentes finalidades, e avaliação dos respectivos resultados;

- aplicação do algoritmo de *chunking* usando a **hipótese das palavras-marca para a língua portuguesa**;
- a abordagem na **extracção de exemplos** usando a **hipótese das palavras-marca** para segmentação e os dicionários probabilísticos de tradução para o alinhamento destes segmentos;
- a **extracção de exemplos** usando como base apenas as âncoras obtidas de **dicionários probabilísticos de tradução**;
- a definição de uma **linguagem de padrões com restrições** para a extracção de terminologia bilingue;
- a definição de uma **linguagem** para a **especificação de dependências** entre processos, para o seu posterior escalonamento tirando partido de paralelismo;

8.2.3 Contribuições Tecnológicas

Foram desenvolvidas várias aplicações que estão disponíveis livremente, para serem usadas e alteradas por toda a comunidade[1].

Neste campo, deve-se salientar o pacote NATools e as suas ferramentas constituintes:

- o **extractor de dicionários probabilísticos** de tradução que foi re-implementado com grandes melhorias a nível de eficiência, escalabilidade e resultados;
- um **servidor/biblioteca** para a disponibilização eficiente **de recursos de tradução**: concordâncias sobre corpora paralelos, dicionários probabilísticos de tradução e n-gramas;
- uma **linguagem de padrões** para a especificação de padrões de tradução, que permite a extracção de terminologia de grande qualidade;
- dois **extractores de exemplos**, usando dois algoritmos diferentes, um baseado em segmentação a um nível próximo do sintagma,

[1] Na verdade o NATools foi instalado e utilizado por vários grupos de investigação para o processamento de corpora nas mais diversas línguas, como o Galego, Alemão, Grego e Hebraico.

e outro baseado em âncoras definidas por dicionários probabilísticos de tradução, e extracção combinatória de exemplos;

- uma **aplicação Web integrada** para a consulta de recursos bilingues;

A `Makefile::Parallel` , constituída por uma linguagem de domínio específico para a especificação de dependências entre processos, e um escalonador eficiente para arquitecturas multi-processador e *clusters* computacionais, também demonstrou ser de grande utilidade.

Para além destas ferramentas interviu-se noutras, como sejam:

- o módulo `XML::TMX` que permite o processamento de memórias de tradução e onde foi implementada a abordagem híbrida DOM e SAX por questões de escalabilidade;
- o analisador morfológico jSpell, desenvolvido no projecto Natura e com dicionários morfológicos para as línguas portuguesa e inglesa;
- o módulo `Lingua::PT::PLNbase` com funcionalidades básicas de processamento de linguagem natural (p.ex. atomização e segmentação de texto);
- um detector de nomes próprios, `Lingua::PT::ProperNames`.
- um detector de língua, `Lingua::Identify`.

8.3 Trabalho Futuro

Temos consciência de que cada desafio resolvido levantou muitos novos e interessantes desafios. Infelizmente não foi possível encará-los todos, pelo que alguns foram *adiados* com grande pena nossa, e portanto não constituem o centro desta dissertação. Esta secção resume alguns desafios, que constituem um caminho natural na continuação deste trabalho[2]. Segue-se uma lista de áreas de investigação que nos parecem relevantes na sequência deste trabalho:

[2]É importante realçar que uma dissertação de doutoramento tem um intervalo temporal associado no qual não é possível incluir toda a investigação relevante.

- embora se tenha realizado várias avaliações de dicionários probabilísticos de tradução, existem muitas outras formas de avaliar estes recursos, e que trariam resultados interessantes;
- as várias experiências apresentadas na secção 4.3 para melhoria de dicionários necessitam de uma análise mais cuidada, e é crucial a definição de funções de aglutinação para junção dos resultados obtidos pelas diferentes abordagens;
- a experimentação do algoritmo de extracção de exemplos de tradução baseado na hipótese das palavras-marca para novos pares de língua, como sejam a língua portuguesa e espanhola;
- a definição de padrões para extracção de terminologia foi usada para o par de línguas português-inglês. Mais uma vez, seria interessante analisar o uso de padrões para a extracção de terminologia noutras línguas, mesmo nas em que a ordem das palavras não mude. Nestes casos, o uso de padrões não é imprescindível para a extracção genérica de exemplos, mas poderia ser usado para a extracção específica de terminologia bilingue usando restrições morfológicas;
- a expansão do servidor de recursos NatServer para o suporte de exemplos de tradução e de terminologia bilingue como se de corpora paralelos se tratassem;
- a incorporação de primitivas de alto nível no `Makefile::Parallel` que permitam a especificação de decomposição estrutural e decomposição por partição de forma mais natural e elegante;
- a experiência apresentada para a integração dos recursos obtidos em sistemas de tradução foi superficial. É necessário aprofundar este estudo com mais experiências de tradução, e com a extracção de recursos de tradução a partir de diferentes géneros de corpora paralelos. Estão já em curso experiências com o sistema de tradução Apertium.

O NATools, a interface Web para consulta de recursos, e os recursos extraídos ao longo deste trabalho, estão disponíveis em `http://natools.sf.net/`

Bibliografia

Almeida, J. João and Alberto Simões. 2006. Publishing multilingual ontologies: a quick way of obtaining feedback. In *ElPub 2006 — Digital Spectrum: Integrating Technology and Culture*, Bansko, Bulgaria, June.

Almeida, José João and Ulisses Pinto. 1994. Jspell – um módulo para análise léxica genérica de linguagem natural. In *Actas do X Encontro da Associação Portuguesa de Linguística*, pages 1–15, Évora.

Almeida, José João and José Carlos Ramalho. 1999. XML::DT a Perl down-translation module. In *XML-Europe'99*, Granada, Spain, May.

Almeida, José João and Alberto Simões. 2006. T_2O — recycling thesauri into a multilingual ontology. In *Fifth international conference on Language Resources and Evaluation, LREC 2006*, Genova, Italy, May.

Almeida, José João and Alberto Simões. 2007. XML::TMX — processamento de memórias de tradução de grandes dimensões. In José Carlos Ramalho, João Correia Lopes, and Luís Carríço, editors, *XATA 2007 — 5ª Conferência Nacional em XML, Aplicações e Tecnologias Aplicadas*, pages 83–93, February.

Almeida, José João, Alberto Manuel Simões, and José Alves Castro. 2002. Grabbing parallel corpora from the web. *Procesamiento del Lenguaje Natural*, 29:13–20, September.

Almeida, José João Dias. 2003. *Dicionários dinâmicos multi-fonte*. Tese de doutoramento, Escola de Engenharia – Universidade do Minho, December.

ALPAC, Automatic Language Processing Advisory Comittee. 1966. Languages and machines: computers in translation and linguistics. Technical report, Division of Behavioral Sciences, National Academy of Sciences, National Research Council, Washington, D.C.

Armentano-Oller, Carme, Rafael C. Carrasco, Antonio M. Corbí-Bellot, Mikel L. Forcada, Mireia Ginestí-Rosell, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Gema Ramírez-Sánchez, Felipe Sánchez-Martínez, and Miriam A. Scalco. 2006. Open-source portuguese-spanish machine translation. In *7th International Workshop on Computational Processing of Written and Spoken Portuguese, PROPOR 2006*, pages 50–59, Itatiaia, Rio de Janeiro, Brazil, May.

Armentano-Oller, Carme, Antonio M. Corbí-Bellot, Mikel L. Forcada, Mireia Ginestí-Rosell, Boyan Bonev, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Gema Ramírez-Sánchez, and Felipe Sánchez-Martínez. 2005. An open-source shallow-transfer machine translation toolbox: consequences of its release and availability. In *OSMaTran: Open-Source Machine Translation, A workshop at Machine Translation Summit X*, Phuket, Thailand.

Armstrong, Stephen, Marian Flanagan, Yvette Graham, Declan Groves, Bart Mellebeek, Sara Morrissey, Nicolas Stroppa, and Andy Way. 2006. MaTrEx: machine translation using examples. In *TC-STAR OpenLab Workshop on Speech Translation*, Trento, Italy.

ATRIL Language Engineering, 2003. *Déjà Vu X Professional Users' Guide*.

Bar-Hillel, Yehoshua. 1951. The present state of reseach on mechanical translation. *American Documentation 2*, pages 229–237.

Bar-Hillel, Yehoshua. 1952a. Mechanical translation: needs and possibilities. Technical report, MIT Library.

Bar-Hillel, Yehosua. 1952b. Operational syntax. Technical report, MIT Library.

Bar-Hillel, Yehosua. 1952c. The treatment of "idioms" by a translating machine. Technical report, MIT Library.

Bar-Hillel, Yehosua. 1960. The present status of automatic translation of languages. *Advances in Computers 1*, pages 91–163.

Berger, A., P. Brown, S. Della Pietra, V Della Pietra, J. Lafferty, H. Printz, and L. Ures. 1994. The Candide system for machine translation. In *ARPA Conference on Human Language Technology*.

Bernardini, Silvia, Marco Baroni, and Stefan Evert. 2006. A wacky introduction. In Marco Baroni and Silvia Bernardini, editors, *WaCky! Working Papers on the Web as Corpus*. Gedit Edizioni, September, pages 9–40.

Bey, Youcef, Christian Boitet, and Kyo Kageura. 2006. The TRANSBey prototype: an online collaborative wiki-based cat environment for volunteer translators. In *LREC-2006: Fifth International Conference on Language Resources and Evaluation. Third International Workshop on Language Resources for Translation Work, Research & Training (LR4Trans-III)*, pages 49–54, Genoa, Italy, 28 May.

Bowker, Lynne and Michael Barlow. 2004. Bilingual concordancers and translation memories: a comparative evaluation. In *Language Resources and Evaluation Conference*, Geneva, August.

Brown, Peter F., John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2).

Brown, Ralf. 2002. Example-based machine translation — a tutorial. Carnegie Mellon University, October, 9.

Brown, Ralf D. 2001. Transfer-rule induction for example-based translation. In Michael Carl and Andy Way, editors, *Workshop on Example-Based Machine Translation*, pages 1–11, September.

Brown, Ralf D., Rebecca Hutchinson, Paul N. Bennett, Jaime G. Carbonell, and Peter Jansen. 2003. Reducing boundary friction using translation-fragment overlap. In *MT Summit IX*, New Orleans.

Bull, W. E. 1952. Frequency problems in MT. [not traceable].

Bédard, Claude. 2000. Mémoire de traduction cherche traducteur de phrases (translation memory is looking for sentences translator). *Traduire ISSN 0395-773X*, 186:41–49.

Callison-Burch, Chris, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the role of bleu in machine translation research. In *11th Conference of the European Chapter of the Association for Computational Linguistics*.

Canals-Marote, Raul, A. Esteve-Guillén, A. Garrido-Alenda, M.I. Guardiola-Savall, A. Iturraspe-Bellver, S. Montserrat-Buendia, S. Ortiz-Rojas, H. Pastor-Pina, P.M. Pérez-Antón, and M.L. Forcada. 2001. El sistema de traducción automática castellano-catalán internostrum. *Procesamiento del Lenguaje Natural*, 27:151–156.

Cardoso, Nuno. 2006. Avaliação de sistemas de reconhecimento de entidades mencionadas. Master's thesis, Faculdade de Engenharia da Universidade do Porto.

Cardoso, Nuno, Leonardo Andrade, Alberto Simões, and Mário J. Silva. 2005. The XLDB Group at the CLEF 2005 Ad-Hoc Task. In C. Peters, F. Gey, J. Gonzalo, H. Mueller, G. Jones, M. Kluck, B. Magnini, and M. Rijke, editors, *Accessing Multilingual Information Repositories, 6th Workshop of the Cross-Language Evaluation Forum, CLEF 2005*, volume 4022 of *LNCS*, pages 54–60, Vienna, Austria, September. Springer-Verlag.

Carl, Michael. 1999. Inducing translation templates for example-based machine translation. In *MTSummit VII.*

Carl, Michael. 2001. Inducing probablilistic invertible translation grammars from aligned texts. In Michael Carl and Andy Way, editors, *Workshop on Example-Based Machine Translation*, pages 12–22, September.

Caseli, Helena de Medeiros and Maria Graça Volpe Nunes. 2003. Evaluation of Sentence Alignment Methods on Portuguese-English Parallel Texts. *SCIENTIA*, 14(2):1–14.

Caseli, Helena M., Maria G. V. Nunes, and Mikel L. Forcada. 2005. Evaluating the LIHLA lexical aligner on Spanish, Brazilian Portuguese and Basque parallel texts. *Procesamiento del Lenguaje Natural*, September.

Chandioux, John. 1976. METEO: un système operationnel pour la traduction automatique des bulletins metéorologiques destinés au grand public. *META*, 21:33–37.

Christ, Oliver, Bruno M. Schulze, Anja Hofmann, and Esther König, 1999. *The IMS Corpus Workbench: Corpus Query Processor (CQP): User's Manual*. Institute for Natural Language Processing, University of Stuttgart, March.

Collins, Bróna, Padraig Cunningham, and Tony Veale. 1996a. Adaptation-guided retrieval for example-based machine translation. In *AMTA'06, The 2nd Conference of the Association for Machine Translation in the Americas*.

Collins, Bróna, Pádraig Cunningham, and Tony Veale. 1996b. An example-based approach to machine translation. In *Expanding MT horizons: Proceedings of the Second Conference of the Association for Machine Translation in the Americas*, pages 1–13, Montreal, Quebec, Canada (Washington, DC: AMTA), 2–5 October.

Corbí-Bellot, Antonio M., Mikel L. Forcada, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Gema Ramírez-Sánchez, Felipe Sánchez-Martínez, Iñaki Alegria, Aingeru Mayor, and Kepa Sarasola. 2005. An open-source shallow-transfer machine translation engine for the romance languages of spain. In *European Association for Machine Translation, 10th Annual Conference*, pages 79–86, Budapest.

Correia, Ana Teresa Varajão Moutinho Pereira. 2006. Colaboração na constituição do corpus paralelo Le Monde Diplomatique (FR-PT). Relatório de estágio, Universidade do Minho, Braga, Dezembro.

Danielsson, Pernilla and Daniel Ridings. 1997. Practical presentation of a "vanilla" aligner. In *TELRI Workshop in alignment and exploitation of texts*, February.

Dempster, Arthur, Nan Laird, and Donald Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Royal Statistical Society, Series B*, 39(1):1–38.

Desarmenien, Francois. 2001. Parse::Yapp — perl extension for generating and using lalr parsers. Perl module, Comprehensive Perl Archive Network. `http://search.cpan.org/dist/Parse-Yapp/`.

Dodd, Stuart C. 1952. Model english for mechanical translation: an example of a national language regularized for electronic translators. Technical report, MIT Library.

Dominus, Mark Jason. 2005. *Higher Order Perl*. Morgan Kaufman.

Elliston, John. 1979. Computer aided translation: a business viewpoint. In B. M. Snell, editor, *Translating and the computer: proceedings of a seminar, London, 14th November 1978*, pages 149–158, Amsterdam: North-Holland.

Fonseca, Rúben. 2007. Paralelização de processos PLN. Relatório, Conselho de Cursos de Engenharia — Universidade do Minho, Braga, Fevereiro.

Frankenberg-Garcia, Ana and Diana Santos, 2001. *Apresentando o COMPARA, um corpus português-inglês na Web*. Cadernos de Tradução, Universidade de São Paulo.

Frankenberg-Garcia, Ana and Diana Santos. 2003. Introducing COMPARA, the portuguese-english parallel translation corpus. In Silvia Bernardini Federico Zanettin and Dominic Stewart, editors, *Corpora in Translation Education*. Manchester: St. Jerome Publishing, pages 71–87.

Gale, William A. and Kenneth Ward Church. 1991. A program for aligning sentences in bilingual corpora. In *Meeting of the Association for Computational Linguistics*, pages 177–184.

Gansner, Emden R. and Stephen C. North. 2000. An open graph visualization system and its applications to software engineering. *Software — Practice and Experience*, 30(11):1203–1233.

Garrido, Alicia, Amaia Iturraspe, Sandra Montserrat, Hermínia Pastor, and Mikel L. Forcada. 1999. A compiler for morphological analysers and generators based on finite-state transducers. *Procesamiento del Lenguaje Natural*, 25:93–98.

Garrido-Alenda, Alicia and M.L. Forcada. 2001. MorphTrans: un lenguaje y un compilador para especificar y generar módulos de transferencia morfológica para sistemas de traducción automática. *Procesamiento del Lenguaje Natural*, 27:157–162.

Garrido-Alenda, Alicia, P. Gilabert-Zarco, J.A. Pérez-Ortiz, A. Pertusa-Ibáñez, G. Ramírez-Sánchez, F. Sánchez-Martínez, M.A. Scalco, and M.L. Forcada. 2003. Shallow parsing for portuguese-spanish machine translation. In *Workshop on Tagging and Shallow Processing of Portuguese, TASHA 2003*, University of Lisbon, Portugal.

Garvin, Paul. 1972. *On machine translation: selected papers.* The Hague, Mouton.

Gilabert-Zarco, Patrícia, Javier Herrero-Vicente, Sergio Ortiz-Rojas, Antonio Pertusa-Ibáñez, Gema Ramírez-Sánchez, Felipe Sánchez-Martínez, Marcial Samper-Asensio, Míriam A. Scalco, and Mikel L. Forcada. 2003. Construcción rápida de un sistema de traducción automática español-portugués partiendo de un sistema español-catalán. *Procesamiento del Lenguaje Natural*, 31:279–284.

Green, Thomas R. G. 1979. The necessity of syntax markers. two experiments with artificial languages. *Journal of Verbal Learning and Behaviour*, 18:481–496.

Guinovart, Xavier Gómez and Elena Sacau Fontenla. 2004. Métodos de optimización de la extracción de léxico bilingüe a partir de corpus paralelos. *Procesamiento del Lenguaje Natural*, 33:133–140.

Guinovart, Xavier Gómez and Elena Sacau Fontenla. 2005. Técnicas para o desenvolvemento de dicionarios de tradución a partir de córpora aplicadas na xeración do Dicionario CLUVI Inglés-Galego. *Viceversa: Revista Galega de Traducción*, 11:159–171.

Harris, Zellig. 1946. From morpheme to utterance. *Language 22*, pages 161–183.

Harris, Zellig. 1954. Transfer grammar. *International Journal of American Linguistics*, 20:259–270.

Hayes, P., S. Maxwell, and L. Schmandt. 1996. Controlled english advantages for translated and original english documents. In *CLAW-96: First International Workshop on Controlled Language Applications*, pages 84–92, Leuven, Belgium, March.

Hiemstra, Djoerd. 1998. Multilingual domain modeling in Twenty-One: automatic creation of a bi-directional lexicon from a parallel corpus. Technical report, University of Twente, Parlevink Group.

Hiemstra, Djoerd. August 1996. Using statistical methods to create a bilingual dictionary. Master's thesis, Department of Computer Science, University of Twente.

Hutchins, John. 1986. *Machine Translation: past, present, future*. Chichester: Ellis Horwood.

Hutchins, John. 1997. Looking back to 1952: the first MT conferece. In *TMI-97: Theoretical and Methodogical Issues in Machine Translation*, Santa Fe, New Mexico, USA, july.

Hutchins, John. 2005. The history of machine translation in a nutshell. Technical report, University of East Anglia.

Juola, Patrick. 1995. *Learning to Translate: A Psycholinguistic approach to the induction of grammars and transfer functions*. Ph.D. thesis, Department of Computer Science, University of Boulder, Colorado.

Kaplan, A. 1950. An experimental study of ambiguity and context. Technical report, The RAND Corporation, Santa Monica. Reproduced in Mechanical Translation 2 (1955), pages 39–46.

Kay, Martin and Martin Röscheisen. 1993. Text-translation alignment. *Computational Linguistics*, 19(1).

Kenny, Dorothy. 2004. Translation memories and parallel corpora: Challenges for the translation trainer. In *Inaugural Conference of the International Association for Translation and Intercultural Studies*, Sookmyung Women's University, Seoul, Korea, 12–14 August.

Knight, Kevin. 2004. A statistical MT tutorial workbook. Prepared in connection with the JHU summer workshop, April, 30.

Knight, Kevin and Philipp Koehn. 2004. What's new in statistical machine translation. Tutorial at HLT/NAACL.

Koehn, Philipp. 2002. EuroParl: a multilingual corpus for evaluation of machine translation. Draft.

Koehn, Philipp, 2004. *Pharaoh, a Beam Search Decoder for Phrase-Based Statistical Machine Translation Models*. USC Information Sciences Institute, August 18.

Koehn, Philipp. 2006. Statistical machine translation: the basic, the novel, and the speculative. University of Edinburgh, April, 4.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL), demonstration session*, Prague, Czech Republic, June.

Landsbergen, Jan. 1987. Isomorphic grammars and their use in the rosetta translation system. In M. Kind, editor, *Machine translation today: the state of the art*, pages 351–372, Edinburgh: University Press.

McCowan, I., D. Moore, J. Dines, D. Gatica-Perez, M. Flynn, P. Wellner, and H. Bourlard. 2004. On the use of information retrieval measures for speech recognition evaluation. IDIAP-RR 73, IDIAP, Martigny, Switzerland.

Melamed, I. Dan. 2000. Models of translational equivalence among words. *Computational Linguistics*, 26(2):221–249.

Melamed, I. Dan. 2001. *Empirical Methods for Exploiting Parallel Texts*. MIT Press.

Mota, Cristina, Diana Santos, and Elisabete Ranchhod. 2007. Avaliação de reconhecimento de entidades mencionadas: princípio de AREM. In *Avaliação conjunta: um novo paradigma no processamento computacional da língua portuguesa*. IST Press.

Nagao, Makoto. 1984. A framework of a mechanical translation between japanese and english by analogy principle. In *International NATO symposium on Artificial and human intelligence*, pages 173–180, New York, NY, USA. Elsevier North-Holland, Inc.

Nieto, Ismael Pascual and Mick O'Donnell. 2007. Flexible statistical construction of bilingual dictionaries. *Procesamiento del Lenguaje Natural*, 39:249–255, September.

Nirenburg, Sergei. 1995. The pangloss mark iii machine translation system. Technical report, by NMSU CRL, USC ISI and CMU CMT.

Och, Franz Josef and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Oswald, V. A. 1952. Word-by-word translation. [not traceable].

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318, Philadelphia, July.

Petersen, Ulrik. 2004. Emdros — a text database engine for analyzed or annotated text. In *20th International Conference on Computational Linguistics*, volume II, pages 1190–1193, Geneva, August.

Prior, Marc, 2002. *OmegaT User Manual*, December. `http://www.omegat.org/`.

Pym, P. J. 1990. Pre-editing and the use of simplified writing for MT. *Translating and the computer: Proceedings of a conference, 10-11 November 1988*, 10:80–96.

RALI Laboratory. 2006. TransSearch. `http://www.tsrali.com/`.

Reifler, Erwin. 1952a. General MT and universal grammar. Technical report, MIT Library.

Reifler, Erwin. 1952b. MT with a pre-editor and writing for MT. Technical report, MIT Library.

Resnik, Philip. 1998. Parallel strands: A preliminary investigation into mining the web for bilingual text. In L.Gerber D. Farwell and E. Hovy, editors, *Machine Translation and the Information Soup (AMTA-98)*. Lecture Notes in Artificial Intelligence 1529, Springer.

Santos, Diana Maria de Sousa Marques Pinto dos. 1996. *Tense and aspect in English and Portuguese: a contrastive semantical study*. Ph.D. thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa.

Sarmento, Luís. 2006. BACO — a large database of text and co-occurrences. In *5th International Conference on Language Resources and Evaluation (LREC'2006)*, Genova, May.

Scott, Bernard. 2003. The logos model: An historical perspective. *Machine Translation*, 18(1):1–72, March.

SDL Trados. 2006. October. `http://www.trados.com/`.

Simões, Alberto and J. João Almeida. 2006a. Combinatory examples extraction for machine translation. In Jan Tore Lønning and Stephan Oepen, editors, *11th Annual Conference of the European Association for Machine Translation*, pages 27–32, Oslo, Norway, 19–20, June.

Simões, Alberto and J. João Almeida. 2006b. NatServer: a client-server architecture for building parallel corpora applications. *Procesamiento del Lenguaje Natural*, 37:91–97, September.

Simões, Alberto and José João Almeida. 2007. Avaliação de alinhadores. In Diana Santos, editor, *Avaliação conjunta: um novo paradigma no processamento computacional da língua portuguesa*. IST Press, pages 219–230.

Simões, Alberto, José João Almeida, and Xavier Gomez Guinovart. 2004. Memórias de tradução distribuídas. In José Carlos Ramalho and Alberto Simões, editors, *XATA 2004 — XML, Aplicações e Tecnologias Associadas*, pages 59–68, February.

Simões, Alberto, Rúben Fonseca, and José João Almeida. 2007. Makefile::Parallel dependency specification language. In Anne-Marie Kermarrec, Luc Bougé, and Thierry Priol, editors, *Euro-Par 2007*, volume 4641 of *LNCS*, pages 33–41, Rennes, France, August. Springer-Verlag.

Simões, Alberto, Xavier Gómez Guinovart, and José João Almeida. 2004. Distributed translation memories implementation using web-services. *Procesamiento del Lenguaje Natural*, 33:89–94, July.

Simões, Alberto M. and J. João Almeida. 2003. NATools – a statistical word aligner workbench. *Procesamiento del Lenguaje Natural*, 31:217–224, September.

Simões, Alberto Manuel and José João Almeida. 2001. `jspell.pm` — um módulo de análise morfológica para uso em processamento de linguagem natural. In *Actas da Associação Portuguesa de Linguística*, pages 485–495.

Simões, Alberto Manuel Brandão. 2004. Parallel corpora word alignment and applications. Master's thesis, Escola de Engenharia - Universidade do Minho.

Sloan, Joseph D. 2004. *High Performance Linux Clusters with OSCAR, Rocks, OpenMosix, and MPI*. O'Reilly.

Somers, Harold. 1999. Review article: Example based machine translation. *Machine Translation*, 14(2):113–157.

Somers, Harold, Ian McLean, and Daniel Jones. 1994. Experiments in multilingual example-based generation. In *3rd International Conference on the Cognitive Science of Natural Language Processing*, Dublin, Ireland.

Specia, L., M.G.V. Nunes, and M. Stevenson. 2005. Exploiting Parallel Texts to Produce a Multilingual Sense Tagged Corpus for Word

Sense Disambiguation. In *RANLP – Recent Advances in Natural Language Processing)*, volume 5, pages 525–531.

STAR AG, 2006. *Transit XV – User's Guide.*

Steinberger, Ralf, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaž Erjavec, Dan Tufiş, and Dániel Varga. 2006. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *5th International Conference on Language Resources and Evaluation (LREC'2006)*, Genoa, Italy, 24–26 May.

Sánchez-Martínez, Felipe and Mikel L. Forcada. 2007. Automatic induction of shallow-transfer rules for open-source machine translation. In *TMI, The Eleventh Conference on Theoretical and Methodological Issues in Machine Translation (TMI 2007)*, pages 181–190, Skövde, Sweden.

Sánchez-Martínez, Felipe and Hermann Ney. 2006. Using Alignment Templates to Infer Shallow-Transfer Machine Translation Rules. *Advances in Natural Language Processing.*

Sánchez-Villamil, Enrique, Susana Santos-Antón, Sergio Ortiz-Rojas, and Mikel L. Forcada. 2006. Evaluation of alignment methods for HTML parallel text. *Lecture Notes in Computer Science - Advances in Natural Language Processing - 4139*, pages 280–290, August.

Toma, Peter. 1977a. SYSTRAN as a multilingual machine translation system. In *"Overcoming the language barrier" – Third European Congress on Information Systems and Networks*, pages 569–581, Luxembourg, May.

Toma, Peter. 1977b. SYSTRAN: ein maschinelles Übersetzungssystem der 3 generation. *Sprache und Datenverarbeitung 1*, pages 38–46.

TRADOS Incorporated, 2003. *MultiTerm Terminology Solutions – User Guide*, July.

TRADOS Incorporated, 2005. *Trados 7 Freelance – Getting Started Guide*, June.

Varga, Dániel, Péter Halácsy, András Kornai, Viktor Nagy, László Németh, and Viktor Trón. 2005. Parallel corpora for medium density languages. In *Proceedings of RANLP'2005*, pages 590–596, Borovets, Bulgaria.

Veale, Tony and Andy Way. 1997. Gaijin: A template driven bootstrapping approach to EBMT. In *NeMNLP'97*, Sofia, Bulgaria.

Wells, R. S. 1947. Immediate constituents. *Language 23*, pages 81–117.

Wood, Mary M. 1993. Categorial grammars. London: Routledge.

Zipf, George. 1949. *Human Behavior and the Principle of Least Effort.* Addison-Wesley (Reading MA).

Apêndice A

Breve Introdução ao NATools

> *Real programmers don't draw flowcharts.*
> *Flowcharts are, after all, the illiterate's form of documentation. Cavemen drew flowcharts; look how much good it did them.*
>
> *anonymous*

Este apêndice apresenta uma breve introdução à codificação de um corpus usando as ferramentas NATools. Não tem como objectivo apresentar todas as ferramentas do pacote, mas apenas proporcionar uma introdução rápida à codificação de corpora.

A.1 Instalação

A instalação do NATools é simples, mas requer alguma experiência com sistemas operativos Unix, já que não são distribuídos binários da ferramenta. Também é sugerido que se use um sistema recente, já que algumas das bibliotecas e módulos Perl usados são bastante recentes.

Depois de descarregar o pacote, (p.ex. `NATools-x.xx.tar.gz`):

1. Começar por descompactar o ficheiro:
   `tar zxvf NATools-xxx.tar.gz`.
   e mudar a directoria actual `cd NATools-xxx`.;
2. Configurar o pacote utilizando a ferramenta `configure`.
   A instalação num Linux standard é obtida com:
   `./configure --prefix=/usr/local`.
3. Esta ferramenta irá indicar as dependências que não estão presentes no sistema. Antes de continuar deverão ser instaladas todas as dependências. O passo anterior pode ir sendo repetido várias vezes, até que não sejam encontradas faltas de dependências.
4. Assim que todas as dependências estejam instaladas e o passo de configuração não detecte falhas, realiza-se o passo de compilação, executando a ferramenta `make`.
5. Depois da compilação pode ser executado um passo de teste à ferramenta, utilizando o `make test`.
6. A instalação da ferramenta é realizada usando `make install`.
7. Finalmente, a directoria actual pode ser removida:
   `cd ..; rm -fr NATools-x.xx`

A.2 Codificação de Corpora

As ferramentas NATools reconhece dois tipos de ficheiros para corpora paralelos:

- o formato TMX[1] (Translation Memory Exchange), um standard para o intercâmbio de memórias de tradução entre ferramentas de tradução assistida por computador;
- o formato específico do NATools: um par de ficheiros, um para cada língua, em que cada unidade de tradução está separada da seguinte por uma linha com apenas um símbolo de dólar (`$`).
  Como exemplo, considere-se o par de ficheiros na tabela A.1. Note que o número de unidades em cada um dos ficheiros deve ser o mesmo!

[1] `http://www.lisa.org/standards/tmx/specification.html`

```
1  I saw a cat .
2  $
3  The cat was
4  fat .
5  $
```

```
1  Eu vi um
2  gato .
3  $
4  O gato era gordo .
5  $
```

Tabela A.1: Par de ficheiros no formato NATools.

Para codificar o corpus paralelo usa-se o comando `nat-create`, como descrito nas próximas subsecções. Este processo irá demorar algum tempo, dependendo do tamanho do corpus. O resultado será uma directoria com o nome do corpus, e um conjunto de ficheiros, como descritos na tabela A.2.

A.2.1 Codificação de um Ficheiro TMX

Para codificar um corpus em formato TMX, bem como a extracção do respectivo dicionário probabilístico de tradução usa-se o comando `nat-create`. Este comando recebe obrigatoriamente uma opção denominada `-id` que especifica o nome do corpus (e da directoria que irá ser criada). No caso de um corpus em formato TMX também deve ser adicionada a opção `-tmx`. Opcionalmente, pode-se usar a opção `-tokenize` para forçar a que o corpus seja atomizado.

A sintaxe básica é:

```
[foo@bar]$  nat-create -id=Corpus -tmx Corpus.tmx
```

A.2.2 Codificação de um par de Ficheiros NATools

Para usar este método é necessário um par de ficheiros alinhados ao nível da frase, com a sintaxe descrita anteriormente. A sintaxe do comando é idêntica à usada com um ficheiro TMX com a única diferença de que não se usa a opção `-tmx`.

```
[foo@bar]$ nat-create -id=Corpus linguaA.txt linguaB.txt
```

Ficheiro	Descrição
`nat.cnf`	propriedades do corpus e variáveis de configuração
`source.\d{3}`	cada uma das fatias do corpus original (língua de origem)
`target.\d{3}`	cada uma das fatias do corpus original (língua de destino)
`source.lex`	léxico correspondente à língua de origem
`target.lex`	léxico correspondente à língua de destino
`source.\d{3}.crp`	cada uma das fatias codificadas (língua de origem)
`target.\d{3}.crp`	cada uma das fatias codificadas (língua de destino)
`source.\d{3}.crp.index`	índices com *offsets* de unidades de tradução (língua de origem)
`target.\d{3}.crp.index`	índices com *offsets* de unidades de tradução (língua de destino)
`source.\d{3}.crp.invidx`	índices inversos de ocorrências de palavras (língua de origem)
`target.\d{3}.crp.invidx`	índices inversos de ocorrências de palavras (língua de destino)
`source-target.\d{3}.bin`	dicionário probabilístico de tradução (origem → destino) extraído de cada fatia
`target-source.\d{3}.bin`	dicionário probabilístico de tradução (destino → origem) extraído de cada fatia
`source-target.bin` (e `.dmp`)	dicionário probabilístico de tradução (origem → destino) resultante da soma das fatias
`target-source.bin` (e `.dmp`)	dicionário probabilístico de tradução (destino → origem) resultante da soma das fatias

Tabela A.2: Conteúdo de um Objecto NATools.

Apêndice B

Notação Matemática

In fact what I would like to see is thousands of computer scientists let loose to do whatever they want. That's what really advances the field.

Donald Knuth

Este apêndice apresenta um sub-conjunto da notação matemática usada para a representação de tipos e estruturas de dados usada nesta dissertação.

Os tipos de dados são habitualmente representados em letras maiúsculas, como TU ou S. No caso concreto desta dissertação, e para representar os vários constituintes de um corpus, usaremos:

$\mathrm{C}_{\mathcal{A}}$	Corpus na língua $\mathcal{A}$. Em casos específicos poderá usar-se C para representar um corpus paralelo;
$\mathrm{S}_{\mathcal{A}}$	Frases do corpus $\mathrm{C}_{\mathcal{A}}$ (da língua $\mathcal{A}$);
$\mathrm{W}_{\mathcal{A}}$	Palavras do corpus $\mathrm{C}_{\mathcal{A}}$ (da língua $\mathcal{A}$);
TU	Unidade de tradução, habitualmente $\mathrm{TU} = \mathrm{S}_{\mathcal{A}} \times \mathrm{S}_{\mathcal{B}}$

As instâncias são habitualmente representadas em letras minúsculas itálicas: $w_{\mathcal{A}}$, $d_{\mathcal{A},\mathcal{B}}$, etc.

Construtores de tipos

Notação mais usada na construção de tipos:

$set(A)$	*conjuntos de A*
$A \rightharpoonup B$	mapeamentos, correspondências de A para B
$A^\star$	*sequências de A*
$A \longrightarrow B$	*funções de A para B*
$A \times B$	*produtos*
$A + B$	*alternativas* (co-produtos)
$\perp$	*tipo singular* (vazio)

Mapeamentos, correspondências — $A \rightharpoonup B$

As correspondências unívocas dispõem das seguintes funções predefinidas:

Descrição	**Notação**
Mapeamentos em enumeração	$\left(\binom{a_1}{b_1}\ \binom{a_2}{b_2}\right)$
Mapeamentos em compreensão	$\binom{f(a)}{g(a)}_{a \in setexp}$
Domínio	$\mathrm{dom}\,(f)$
Contra-domínio	$\mathrm{rng}\,(f)$
Aplicação	$f(x)$

Sequências — $A^\star$

As sequências de um tipo A dispõem das seguintes funções de base:

Descrição	**Notação**
Sequências em enumeração	$\langle a_1, a_2, \ldots, a_n \rangle$
Sequências em compreensão	$\langle f(a) \mid a \in setexp \rangle$

Conjuntos — $set(A)$

Os conjuntos dispõem das seguintes funções predefinidas:

Descrição	**Notação**
Conjuntos em enumeração	$\{a_1, a_2, \ldots, a_n\}$
Conjuntos em compreensão	$\{f(a) \mid a \in setexp\}$
Reunião	$c_1 \cup c_2$
Intersecção	$c_1 \cap c_2$
Pertencer ao conjunto	$e \in c$
Não pertencer ao conjunto	$e \notin c$