An MT prototype from English to Portuguese

Diana Santos
IBM Portugal
Praça de Alvalade, 7
1799 Lisboa Codex
PODIANA at LISHQ

Abstract

The goal of this text is to describe the machine translation work at the IBM-INESC Scientific Group, Lisbon, in the framework of Mentor.2

MENTOR88/P is the resulting prototype of one-year work on machine translation from English to Portuguese. The first goals were to develop a convenient architecture, and fulfill it with knowledge from all final system's properties.

Its results seem to show that the architecture and the kind of knowledge for the task are suited to, in the near future, produce an effective automated translation from English technical documents into Portuguese.

Introduction

In short, what we shall present is our choice of an architecture for machine translation, emphasizing some points that seem to us more interesting, and referencing shortly the work already done.

In this paper we look at the system from three different viewpoints:

- the design of the system as a whole, describing its more striking or characteristic features
- the use made of it for machine translation proper, or which translation knowledge is stored in the prototype
- some practical performance measure of the system’s today form

At last, we conclude with some ideas about future work, where we claim that the architecture and the current implementation can be enhanced powerfully.

General description

As most general traits, the system can be considered as transfer-based (as opposed to an interlingua approach), unidirectional, and with an overall architecture that should be independent of the target language.

We proceed by emphasizing some of its relevant features.

---

1 This paper describes the work done in collaboration with Paulo Libano Monteiro, at the IBM-INESC Scientific Group.

2 MENTOR is not a product name and is only used as an internal project name in IBM.
PEG-based

- As a most important feature, MENTORP uses PEG [4] for the analysis of the English text. So, being the graph produced by PEG the departing point of our system, one of the general options was to follow its philosophy, namely both in the way and type of linguistic knowledge being used and in the programming language chosen, PLNL [5].

In fact, although there are several current translation projects [7] [10] using PEG as the analysis phase, we think that we are unique in trying to use all information inside PEG. As far as we know, other approaches have mainly used PEG's output trees, or a selected subset of PEG's internal information as the starting point to a new representation.

On the contrary, we use the whole PEG graph as the first intermediate structure for the transfer model. We believe that there are advantages in this choice, namely that most information that was necessary to perform analysis is also very useful to deeply characterize the English sentence, and consequently to achieve a better translation of it.  

Transfer model

- We imposed complete separation between the three phases of the system, namely analysis, transfer, and generation. Particularly, this means that transfer should create an internal representation for the Portuguese structure, analogue to the one created by PEG.

This is a direct consequence of a transfer architecture, in the sense that an intermediate representation in the target language should be defined, and moreover, we believe it should be at the same (representational) level of the one for the input sentence.

In MENTOR88/P, the transfer input is the PEG graph, only devoid of information concerning English only (morphology) or historical information (useful for debugging of the parsing process). Therefore, at the beginning of generation too, there must be a graph structure for Portuguese bearing the same kind of features and information. This means that uniquely information pertaining to Portuguese only (like morphology and surface forms) should be missing.

On the other hand, this second intermediate structure should have no trace of having come from English or whatever language it was transferred from.

Transfer architecture

One of the points we believe most interesting in the system being described, is

- Clear distinction between the two aspects of transfer: structural and lexical, since they access different information and produce different results. It is however important to emphasize that while they are conceptually separated, the design was conceived to allow for their interaction, interleaving the triggering of the two tasks in a certain order.

Their definition in our system is the following:

- **lexical** the choice among several possible translations of the correct Portuguese word or expression
- **structural** the changes of the sentence’s structure from the English to the Portuguese way

On one hand, modularity is achieved, since the two tasks are conceived and implemented separately.

On the other hand, they are allowed to influence each other in several ways: different Portuguese structures can arise because of the selection of different translations of a word, while, on the other hand, the structural environment can (and most frequently does) influence the lexical choice.

The algorithm

This process is better explained by the global algorithm of the transfer phase, which, in a top-down right-to-left manner, performs the following for each node:

---

3 Namely, we are referring to subject/object pointers, "hidden" nodes that convey important structural information, like the PPOBJ (object of a preposition), conjunction-related attributes, and several scalar data such as lexical or sentential subcategorization, that cannot obviously be displayed in tree form.

An MT prototype from English to Portuguese

IBM Internal Use Only

123
1. Each immediately dominated node
   a. is copied, selectively cleaned from spurious information, and guaranteed that it is the only copy.
   b. is fulfilled with relevant information for lexical transfer, coming from above.
   c. is used to access the bilingual dictionary (with its BASE) to choose the right Portuguese word.

2. The node type is reevaluated, being transformed in a more general class in several cases.
3. The structural transfer rules are triggered by the current node.

As can be seen from above, the graph is only traversed once (and in an ordered way), while we assume lexical transfer can at most influence structural transfer one level above, and not higher levels.

**Structural transfer implementation**

- To allow for this, the statement of structural rules obeys the following restriction: transformations can only concern the nodes immediately below.

This is not, however, a constraint on the linguistic rules that can be expressed in the system (which can account for changes of arbitrary depth), but only on the programming style of these rules.

- To make the writing of the structural rules easier and more high-level, a set of primitive transfer functions was implemented, that take care of uninteresting details and constitue the basis of the rules actually written [9].

Some examples of structural rules are presented in appendix.

**Lexical transfer implementation**

Lexical transfer is performed by a simple procedure connected to the bilingual dictionary access, that interprets the declaration stated in the dictionary entry. This procedure treats the several candidates assembled in the dictionary as follows:

1. Matches part of speech
2. Tests positive conditions, in the order they are stated
3. The same for negative conditions
4. Returns the list of the candidates that can be translations
5. In case the system is working in “NLP-INTERACTION” mode, and the size of the former list is different from one, it interacts with the user

We are convinced, however, that a much higher-level language for specifying lexical transfer is convenient, and therefore we only present this (and some bilingual entries in appendix) so that the complete prototype at this stage can be described and fully understood.

**Generation model**

Generation has to produce a correct Portuguese sentence after the graph received from the transfer phase. This graph defines the syntactical structure of the sentence. The specific tasks of generation may be divided, in the actual implementation, into two groups:

1. Process the existing information in order to get the necessary number, gender and person agreements, and the right cases for pronouns.
2. Inflect the words, contracting and hyphenating the final string.

**Generation architecture**

In our system, the generation architecture is very similar to that of transfer, as we are engaged in creating an homogeneous system, from every point of view. We use thus for generation the same
data structure, the same general philosophy, and the same programming tools (language and dictionary system).

We traverse the graph again only once from top to down, and access the dictionary one level down from the current node.

However, and while we distinguish also two separate tasks, they do not interact in both directions, as in transfer, since the second comes logically after the first, and is moreover only performed for the terminal nodes (the leaves of the graph).

Concerning the programming style of generation rules, it is the same as for transfer: changes are propagated one level at a time.

**Declarative**

1. Structural differences between the two languages are stated in rule form,

2. The information necessary to select the right word (lexical transfer) is stored in the bilingual dictionary as a set of positive and/or negative conditions,

3. Portuguese morphology is also stated in several sets of rules.

We believe that a declarative knowledge representation makes both the statement of the knowledge easier to human eyes and more independent of how to actually implement it. In this way it is also simpler to abstract the information stored in our system (and consequently use it for other purposes) than whether it were “hardwired” in control flow.

**Dictionaries**

- Dictionary information (except for lexical transfer) is reduced to the storing of exceptions: in the bilingual dictionary, exceptions to the general structural transfer rules; in the target language dictionary, morphological irregularities.

It should be acknowledged at this point that the transfer lexicon data that we are using in our system was derived from the ITS Portuguese transfer dictionary developed at Brigham Young University in Provo, Utah[8], using for the moment only its part-of-speech information. Other features and abilities of our own design were only incorporated in a very small scale.

However, we want to mention the methodology chosen (for both bilingual and generation dictionaries), which consists of coding the general rule in rule form, and only signal in the dictionary when specific words imply a different processing.

**Description of the knowledge stored in the system**

Instead of describing particular pieces of knowledge implemented, we prefer to focus on a more general way of describing the knowledge, referencing the interested reader to other more specific texts[9][6].

We shall go through several components of our system, namely structural transfer, lexical transfer, tense transfer, agreement generation, and surface generation.

Some justification on linguistic grounds for the options described in the previous section on the system’s design will also be tried.

**Structural transfer**

Since the transfer design gives structure a prominent role, as can be seen by its imposing the flow of the whole translation process and, more specifically, the order by which lexical transfer is performed, some linguistically motivated explanation should be attempted.

**Justification**

Apart from the computational adequacy there seem to be some reasons to support this approach to transfer. First of all, the fact that an organized and finite classification as syntactical structure may govern instead of the unpredictable and probably uncoverable universe of lexical transfer,
makes the system design more reliable. The pillars of the system are defined on solid and well-known bases: the set of different structures handled and represented by the parser.

But the theory that presided at the creation of the structure itself implies some assumptions that support, in our view, the path followed. In fact, the representation PEG creates is not isomorphic to the one stated in its rules [5]. Therefore, the different levels of its graph hold a much greater significance than if they reflected simply the order of application or form of the rules in the grammar.

We can thus postulate, that there is a meaning associated to the structure of the graph itself, and it is by taking it into account that the design is justified. We assume that, at any particular node in PEG's graph, there is enough information to characterize the phrase it represents. Therefore, rules centered at the nodes deal with language itself and not with the way analysis was led; one node and not a cluster of nodes represents a given structure. These facts motivate the way we state transfer.

In short, we hope to have made clear, in the above lines that the algorithm used is not only a computational device, but it is justified by the way PEG looks at language and therefore arrives at its representation.

Some of the (meta)rules implemented

These are general patterns followed in the system

- Phrase reclassification: we defined a supertype for the current PEG classification of nodes (expressed in their attribute SEGTYPEP2) to handle general translation rules, refining afterwards the classification (through restoring of the original values). This last action enables us to handle the specific translation problems connected to each type computed by PEG.

- Handling of "hidden pointer attributes" [9]: we handle the object of the prepositional phrase first, and then apply the rules to the PP as a whole. This allows for the treatment of the body of a PP in the same way as any other phrase.

- Treatment of constructs of arbitrary length: the way rules must be stated (either changing the current node or signalling down the change) allows for an elegant (recursive) treatment of language, with no constraints on relative closeness of structures.

Lexical transfer

Identification of problems

In general, the choice of a corresponding term in the target language can be based on syntax, on universal "semantic" markers, on semantics, pragmatics or even world knowledge. It can also be approached in a restricted domain context, or treat explicitly (and separately) idioms.

For the moment we have only concentrated on syntax and universals (that is, those characteristics, like animate, time or space, that are found in every human language).

The reason for choosing lexical transfer based on syntax is obvious, taking into account that PEG provides such a detailed description of the syntactical structure of the sentences it analyses, and that we are determined to do the best use of PEG.

However, there is a more general reason for choosing these two criteria as departing points and also as a research subject: the fact that we are aiming at broad coverage. In this situation, we try to develop a processing that may be as general and broadly applicable as possible.

Identifying of a few general patterns used in lexical transfer

From the small experimenting conducted with lexical transfer in the building of this prototype, we were able to identify some patterns:

- Transitive/intransitive use of a verb

---

For this matter we are strongly indebted to some very interesting discussions with Shalom Lappin of Haifa Scientific Center, and in general to the ideas presented by the MENTORH team concerning lexical transfer as a whole [2].

An MT prototype from English to Portuguese

IBM Internal Use Only
While in English the existence or not of the (deep) direct object conveys some meaning, in Portuguese that distinction is veiled by different lexical realizations. So, the determination of grammatical roles seems to play an important part for lexical transfer (for instance in "he returned / he returned the book"5, or "he quit / he quit the job"6).

- Passive/active context

On the other hand, the same verb used in passive or active form is sometimes translated into different verbs in Portuguese, which seems to indicate that the surface syntax must also be taken into account. (This is also related to the point above and/or to the existence of an animated subject or actor, but it definitely shows this way (for instance in "I intend / the program is intended"7, or "the problem shows / the program was shown"8, or even, "the program runs / the program is run"9).

- Human or animated subject (normally related to other conditions as well)

This feature was found to be one of the most used in the statement of the conditions for lexical transfer (for instance in "he works / the program works"10, "he pays / crime pays"11).

- Existence of a particular preposition connected to a verb

Even the actual (string) preposition that is associated with a verb must be considered for correct translation of the verb (see "look at/look for/look after/look like"12).

We must acknowledge that we are in what concerns lexical transfer in a very early stage, so that we can only sketch some of the problems and solutions we tried, without no conclusive data to back them.

**Several correct translations**

An important point concerning lexical transfer in our system is that one translation is not chosen, but the set of all possible ones in a given situation (see 4. on page 3).

This means that, although we currently use the first one, other more complex mechanisms can be incorporated later, for instance to avoid unnecessary word repetitions, or to test some stylistic criteria in word choice.

Also, it stands in the line of declarativeness and modularity that reigns in the whole system, since an entry associated with one particular Portuguese translation is independent of the others, having its own conditions for selection and its own structural features.

**Tense transfer**

This is a very important issue in translating from English into Portuguese, since though some syntactic devices are similar (like the use of the auxiliaries "be" for passive or "have" for past perfect), the Portuguese language presents a much broader tense choice, and specially there is not a direct mapping in many cases.

This is one of the most important issues for translation, and thus a tense transfer module (also written in PLNLP rules) is incorporated in MENTOR88/P.

Several tasks are performed, from the purely syntactical ones, as deleting auxiliaries for future or getting the correct auxiliaries for passive, progressive or perfect aspects, to others that we call conceptual, such as:

5 In Portuguese, "ele voltou / ele devolveu o livro".
6 "ele desistiu / ele deixou o trabalho"
7 "eu tenciono / o programa destina-se"
8 "o problema aparece / o problema foi mostrado"
9 "o programa funciona / o programa é executado"
10 "ele trabalha / o programa funciona"
11 "ele paga / o crime compensa"
12 "olhar para/procurar/cuidar de/parecer-se com"
• Creation of new tenses (in the sense there are no corresponding ones in English)
• Changing voice (e.g., passive into active voice)
• Changing mood (e.g., indicative to subjunctive)
• Changing the whole tense (e.g. present perfect to 'imperfeito', negated imperative to present subjunctive, etc.)

The task is far from complete, and only a very restricted set of cases is yet covered in our system. However, enough has been done to assess the adequacy of the overall system, given that at least one situation of each point of the list above is handled.

Generation

Justification

Also on this subject there seems to be language-based reasons for the processing we present. Particularly, the top-down approach seems to coincide with the "natural" flow of information [6]. (For example, the dominating noun imposes the gender and number of the dominated nodes.)

Motivation

Since generation is target-language specific, some information will be given here about the actual tasks performed, which we divide conceptually between agreement and surface generation.

To motivate the need for agreement processing, we mention that in Portuguese
• adjectives agree in number and gender with the noun they modify (and in case of a conjunction of nouns with different gender, obey a specific algorithm for gender determination)
• past participles in passive voice agree in gender and number with the subject
• pronouns agree in number, gender and person with the entities they refer to
• verbs agree in number and person with their subject

So there is the need to derive and appropriately store the necessary information for agreement, after which the following tasks are performed

1. Define gender for nouns
2. Compute feminine form for adjectives (if necessary)
3. Compute plural form for nouns and adjectives (if necessary)
4. Inflect a verb according to tense, person and number
5. Compute case of a pronoun and inflect it
6. Compute an adverb from an adjective
7. Compute surface forms like contraction or hyphenation of clitics

The algorithm followed for surface generation is, however, very simple:

1. Look in the generation dictionary to see if the information needed is stored there (as it should be in case of an exception)
2. If not, apply the appropriate (morphological) rules to get the result

Dictionary contents

Apart from the irregular forms of the cases described above, the generation dictionary still holds all pronoun cases (indexed by its base form) and other information concerning the contraction of determiners and pronouns with the preceding prepositions.
Description of some practical aspects

In this section we present the system as a working device.

Development methodology

A short description of how the work proceeds in our system follows:

1. The departing point is the English text and the way PEG analyses it.
2. A straightforward rule or set of rules is written to handle the general case for a given problem (for instance, infinitive clauses, pronoun position in the sentence, or noun compounds).
3. Incremental growth follows: by elaboration and by refinement, that is, by contemplating new cases and also by making clearer distinctions.
4. Studies on the corpus\(^{13}\) are done, creating a test set and running the system on it.
5. The appropriate corrections and improvements are done, until
6. the failures that represent the limits of our method are acknowledged (for instance, true ambiguities, or lack of information to distinguish between different cases).

An important guideline in the development of MENTOR88/P is the aim of universal coverage, which implies the following strategies:

- Try to solve the most general problems first.
- Try to solve everything we can before handling particular cases.
- Try to come to a definite and accurate statement of the limits of our methods before "patching" some problems that could even not need any "ad-hoc" treatment.
- Try to cover general problems, categories, or classes of words first, and only then handle exceptions.

Performance

Some results of MENTOR88/P will be presented, to allow for our conclusion that the overall design is fitted for translation from English. They concern the evaluation of two different kinds of problems, and were done on original corpus' sentences (in the first case, every sentence containing at least an occurrence of an infinitive clause, while in the second case, the first 145 sentences where appeared the verb "be").

The first test had as goal to evaluate a rather deep and extensive work on the translation of the infinitive clauses based on syntax. In 120 sentences containing a total of 141 occurrences, a percentage of 93% of correct translation was achieved.

The second test was intended to evaluate both the relative importance of the lexical transfer of the verb "be", related to its use as a syntactical auxiliary, and the performance of the set of conditions (stored in the bilingual dictionary) for selecting between the two alternatives "ser" and "estar". This problem is, by the way, also the deepest work on lexical transfer addressed nowadays in the system. The use of "be" as the main verb was present in 41% of the cases, and the overall translation of "be" was correct in 94,5% occurrences of the set.

To allow for some evaluation by the reader, some sample translations are presented in appendix, that illustrate some MENTOR88/P capabilities.

Future work

One of the most important issues a prototype must be concerned with, is, in our opinion, the way it can be expanded and made to incorporate bigger quantities of knowledge, but especially, qualitative improvements without becoming awkward or totally unsuitable.
Also in this aspect we believe MENTOR88/P is able to receive a lot of improvement with no significant changes of philosophy or implementation.

The first issue that should be described is the (almost) all-syntactic approach. There is nothing, either in the architecture or in the rules we have been programming, that enforces or stresses this (present) characteristic.

In fact, we are deeply convinced that several kinds of different rules could be described at a higher level, as soon as we get the new layers that are being incorporated in PEG (or post-PEG processing) that will allow for other description levels in the sentence (like deep grammatical roles) and/or discourse characterization (of the kind topic/focus).

However, the syntactic rules should still be there to work as a safety net if the other information is not available. Also, and while we would need more generation processing when transferring at a higher level, it would have to be added to the already existing generation module, which remains necessary.

What we are trying to say is that, although we intend later to formulate transfer rules at different conceptual levels than the syntactic one, we need to go through the intermediate representation for Portuguese as it is now (namely with all syntax specified). Therefore, improvement of the system is an addition of new layers (both in transfer and in generation) but not at all a replacement.

Another important question that at this moment is only planned but we hope to begin working on as soon as possible, is a style component, consisting of several rules for transfer and for generation, and also in the lexical transfer domain (processing the several candidates).

Finally, the addressing of idioms and the development of a consistent and adequate bilingual dictionary are a must, and will certainly have to be dealt with in the future development of this project.

**Conclusions**

In conclusion, we believe that the system presented is appropriate for the task of machine translation, and we attempted to demonstrate it for the English-to-Portuguese case. We believe it to be modular, easily expandable, and expressive enough to encompass the large range of phenomena covered by the translation task.

We are convinced that the relatively large amount of problems dealt with in the small time frame of the project can be presented as a sign of hope. But we are also aware that much work still remains to be done before a prototype can be turned into a real working system, for whose purpose the building of a vast bilingual dictionary is determinant.

In this paper, we tried to justify the architecture presented both from a pragmatic and a linguistic point of view. A presentation of some of the general problems handled and some performance measures were intended to illustrate the translation task that is our goal to perform. Finally, we stated what was in our opinion the logical continuation for this work.

**Acknowledgements**

I am indebted to all other participants in MENTOR for fruitful and interesting discussions, and particularly to Lauri Carlson for his remark on the importance of the linguistic knowledge involved in a system independently of organization issues, and to Shalom Lappin for long and very detailed discussions on lexical transfer.

I would like to thank both Karen Jensen and Steve Richardson for their constant support during the development of this prototype answering my questions and doubts.

And finally, I would like to make clear that the work presented here is mainly a team work, to which I am most grateful to Paulo Libano Monteiro.

**References**


An MT prototype from English to Portuguese IBM Internal Use Only
Some sample translations

i want the functions to have been correctly tested the next time

\[ \Rightarrow \text{eu quero que as funções tenham sido correctamente testadas a próxima vez} \].

he listened to no excuses and never returned, but the money was eventually returned after some time

\[ \Rightarrow \text{ele não escutou nenhuma desculpas e nunca voltou mas o dinheiro foi finalmente devolvido depois de algum tempo} \].

basically, to see if it is safe to execute a command, read the manual

\[ \Rightarrow \text{basicamente, para ver se é seguro executar um comando, le o manual} \].

to want the man to pay to enter the house to buy some cigarettes was to kill him

\[ \Rightarrow \text{querer que o homem pagasse para entrar na casa para comprar alguns cigarros foi matar o} \].

the meeting was in Paris, but she wasn't there

\[ \Rightarrow \text{a reunião foi em Paris mas ela não esteve lá} \].

this program set allows users to avoid many problems

\[ \Rightarrow \text{este conjunto de programa permite que os utilizadores evitem muitos problemas} \].

Appendix

This appendix is intended to illustrate the internals of the system, including some simple transfer rules and lexical transfer information, together with some more detailed examples of its output.

Some structural transfer rules

This first rule transforms a noun phrase into a prepositional one, in case there is an attribute called CHTOPP (holding a node representing the preposition). This rule can be triggered in several environments, like the translation of Noun Noun phrases, or of verb arguments asking for a preposition in Portuguese.
It should be noted that the rule only contains knowledge about changing the structure conveniently.

(5700) NP(CHITOPP) --> PP(%%NP, -DONE, plower<seg,'PPOBJ', 'NP', 'before', 'end'>, PRP=CHITOPP, padd<seg, PRP, 'PRMODS', 'before'>, SEGTYP="PP", SEGTYP2="PP", -CHITOPP)

The next rule handles infinitive clauses premodifying another clause. The only processing done is to mark it to be translated as an infinitive, in case it is also the subject of the main clause. Other rules at other levels will deal with that same infinitive clause (and if its translation was not determined here, it will be by other criteria).

(4910) VP('infcls3'.NOTIN.DONE, DONE='infcls3'...DONE, pfind<seg, 'SEGTYP2', 'INFCL', 'PRMODS', 'FOUND'>.NE.0) --> VP(%%VP, <SUBJECT.EQ.top<FOUND>, CHICNTO(top<FOUND>)=1, TENS(top<FOUND>)=TENS>, -FOUND)

Bilingual entries

Some bilingual entries, both with structural markers and with lexical transfer information, are presented. Very simple entries were chosen.

RETURN
voltar(VERB (NCOND self OBJECT self PASSIVE segtype PTFRTCL))
devolver(VERB)
retornar(VERB)

LOOK
procurar(VERB (COND PREP for) (PREPO for OBJECT))
olhar(VERB (PREPO at para))

ANY
nenhum(ADJ (COND NPIND NEG))
qualquer(ADJ)

FORGET
esquecer(VERB REFLEX (PREPO OBJECT de))

HAPPY
feliz(ADJ (INFPREP por))

Some more detailed examples

The two examples below can be useful to distinguish the tasks done in transfer and generation, being the intermediate structures built by analysis and transfer printed in a simplified way (PEG's output trees).

In both sentences, differences in structure and some generation processing can be seen.

I want the girl to go.

<table>
<thead>
<tr>
<th>DECL</th>
<th>NP1</th>
<th>PRON1</th>
<th>&quot;i&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>VERB1</td>
<td>&quot;want&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NP2</td>
<td>DET1</td>
<td>ADJ1</td>
<td>&quot;the&quot;</td>
</tr>
<tr>
<td>NOUN1</td>
<td>&quot;girl&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>?</td>
<td>INFCL1</td>
<td>INFTO1</td>
<td>&quot;to&quot;</td>
</tr>
</tbody>
</table>

An MT prototype from English to Portuguese
Árvore portuguesa

TRAD1   DECL2*  NP3 PRON2* "eu"
       VERB3* "querer"
       VP1   COMPL1 "que"
       NP4   DET2  ADJ2* "o"
       NOUN2* "rapariga"
       VERB4* "ir"
       PUNC2  ""

Geração

eu quero que a rapariga vá.

when i gave it to him i forgot her.

Árvore portuguesa

TRAD1   DECL2*  SUBCL1 CONJ1 "when"
       NP1   PRON1* "i"
       VERB1* "gave"
       NP2   PRON2* "it"
       ?PP1  PREP1 "to"
       PRON3* "him"
       NP3   PRON4* "i"
       VERB2* "forgot"
       NP4   PRON5* "her"
       PUNC1  ""

Geração

quando eu lhe o dei eu esqueci me dela.

Some Portuguese dictionary entries

Only exceptions are stored, as mentioned.

enigma: enigma (NOUN (GENER MASC))
longe: longe (ADV)
mão: mã (NOUN (PLURFORM mãos))
mal: mal (ADV (COMPFORM pior))
mau: mau (ADJ (FEMFORM má) (COMPFORM pior))
ter: ter (VERB (PRES1S tenho) (PRES2S tens) (PRES3S tem) (PRES3P têm))
tu: tu (PRON (ACUSFORM te) (DATIFORM te) (PREPFORM ti))