Providing access to language resources through the World Wide Web: the Oslo Corpus of Bosnian Texts

Diana Santos
The Text Laboratory, University of Oslo
P.O.box 1102 Blindern, N-0317 Oslo, NORWAY
[diana.santos@ilf.uio.no]

Abstract

This paper describes a net-based service at the Text Laboratory of the University of Oslo, the access to the Oslo Corpus of Bosnian Texts, the first widely available corpus of the language spoken in Bosnia and Herzegovina.

The paper is devoted to the presentation of the several kinds of decisions involved in setting up such a service. By reporting our experience, and sharing some of the ideas implemented, we hope to help other providers with limited economic resources to set up similar services.

After a general description of the project, some criteria to evaluate a Web interface to a corpus are proposed and discussed in connection with our service. A threefold separation between interface, corpus encoding scheme and the corpus proper is suggested, and a view of the interface as adding capabilities is argued for.

Introduction

The Oslo Corpus of Bosnian Texts (OCBT) is the result of a joint project between the Department of East European and Oriental Studies and the Text Laboratory at the Faculty of Arts of the University of Oslo. The project's main goal is to make widely available, for purposes of linguistic research, recent Bosnian texts (1989-97). In order to accomplish this goal, a Web service, located at http://www.tekstlab.uio.no/Bosnian/Corpus.html, was implemented.

This service relies crucially on the use of the IMS Corpus Workbench[1] – and particularly of CQP, the Corpus Query Processor –, developed at the Institut für maschinelle Sprachverarbeitung at the University of Stuttgart.

We first provide a short introduction to the corpus proper, as well as information on the project itself. Then we present the functionalities offered by the Web service (put in perspective by comparing it with other sites which offer Web access to corpora), and discuss some options and problems. Finally, the conceptual separation between interface, corpus encoding and corpus itself is advocated.

Corpus contents

The texts, chosen by Professor Svein Mønnesland, were scanned or obtained in electronic form, and have the distribution and size displayed in Table 1. (The reader is nevertheless advised to consult the Web pages for updated information and for a detailed table of contents.)

Genre	Number of texts	Size in words
Fiction	32	655 044
Essays	37	451 395
Newspapers	11	257 808
Children's books	10	91 583
Islamic texts	7	43 039
Legal texts	7	22 824
Folklore	1	3 774
Total	**105**	**1 525 467**

Table 1: Corpus distribution and size.

Legal issues

The University of Oslo was granted permission, by all relevant publishers, to make the texts available world-wide through the Internet for research purposes. In most cases, not more than one third of the complete text was included in the corpus. In addition, precautions were taken to prevent users from downloading the whole corpus to their machines.

In order to comply with the publishers' wishes, and also in order to be able to provide good user support, users must require access to the corpus, giving information about themselves, as well as stating that they will not use the corpus for commercial purposes. In addition, corpus use is monitored.[2]

The language

Bosnian is the name of the language spoken in Bosnia and Herzegovina, which used to be called Serbo-Croatian before the split of former Yugoslavia (for a linguistic description, see Corbett (1990) or Browne (1993)). Due to the political changes, it has acquired in the last years features that individuate it in relation to Croatian and Serbian (Leko, 1998), although it shares a lot with both of these "new" languages.

Although Bosnian is theoretically written in the two alphabets, Cyrillic and Latin, in practice it is only used in the latter. It can thus be encoded using ISO-Latin2 (ISO-8859-2). In addition, there are some conventions to print it in ASCII, which for lack of a better name we call 'alongations'. Table 2 displays the Bosnian characters.

Bosnian character	Alongation(s)	ISO-Latin2 (octal) code
ć	ch	306

[1] See http://www.ims.uni-stuttgart.de/CorpusToolbox/

[2] So far, we have not had any illegal attempt to access more of the corpus than expected, though.

Ć	Ch,CH[3]	346
č	cc	310
Č	CC,Cc	350
đ	djj/dj[4]	320
Đ	Djj/Dj,DJJ/DJ	360
dž[5]	dz	
Dž	Dz	
DŽ	DZ	
ž	zz-zh[6]	256
Ž	ZZ-ZH,Zz-Zh	276

Table 2: Bosnian characters.

To our knowledge, this is the only corpus of Bosnian widely available. There is, however, a corresponding older corpus, Yugoslav, compiled by Henning Mørk at the Slavic Department of the University of Aarhus, in Danmark, which covers Serbo-Croatian in the four areas of former Yugoslavia and which is available as a set of text files for research purposes.[7]

Progress report

From the summer 1996 to the summer 1997, two Bosnian research assistants scanned and proofread the texts, saved them as Macintosh Word files with a text header, and stored them in a directory structure according to genre.

Since the author joined the project after this phase had been completed, only the subsequent steps will be reported and discussed.

One additional month of work was necessary to prepare the corpus to be encoded in the IMS Corpus Workbench format and keep track of the sources in a principled way. In particular, we created a (master) file containing a description of each corpus text, with an unique identifier, created a new version of all files as text only, and made sure the information on the header was in accordance with the master file. Then we wrote a program that automatically computed the correspondences, got rid of the headers and other annotation(s), converted the remaining text to ISO-Latin2 and created a corpus in a form ready for CQP. (In later versions, the program also computes a HTML version of the table of contents with updated token counts.)

Although, in principle, one should only need to build the corpus once, things never work this way in practice. We have built the corpus several times. Furthermore, this program has allowed us to try alternative encoding schemes, and was invaluable when we needed to recompute alterations to the corpus information (such as changes in the number of tokens after a revision of the corpus has taken place). According to our experience, it pays to automate the building process, even if you firmly believe “you only have to build a corpus once”. It goes without saying that such a program will keep evolving as well.

The first test version of the corpus, with a toy interface, was ready in mid-October 1997, and was made available at the University of Oslo for use by the members of the project and a few experts whose feedback was asked for. The first world-wide available version was announced in the appropriate lists only in the beginning of February this year.

Since work has shifted from corpus building to Web service design, the author's work (on a part-time basis – approximately 30% of full-time) has been spent in

1. programming (developing the CGI-interface to CQP, debugging and improving it)
2. designing and writing the Web pages
3. providing administrative/technical support to the users

Some time has also been devoted to finding usable fonts in various platforms, and documenting their use.

It is perhaps relevant to note that the two latter tasks consume the lion's share of the time devoted to the project, confirming that keeping a service alive (maintenance) is something that should be foreseen when launching this kind of projects. “Service” is fundamental in order to have successful results.

Point of the situation

At present, the Oslo Corpus of Bosnian Texts is available at the World Wide Web in version 2.1 of the interface, version 1.1 of the corpus, and has 35 registered users from 14 countries. A discussion list of issues related to the corpus has been created, and some mechanisms of “stimulating” active corpus use are in progress.

Serving a corpus through the Web

First we describe, from a user's point of view, what our service has to offer, then suggest some ways of evaluating Web interfaces to corpora and discuss the OCBT in these regards.

Kinds of functionalities

What can a linguist interested in Bosnian obtain by querying the OCBT through the Internet? At present, s/he can:

1. get concordances (KWIC) of words, phrases, suffixes prefixes, punctuation marks, or any combination of these, expressed in a powerful query language, the CQP query language (Schulze & Christ, 1996) – more on that below.
2. design to a limited extent the format of the KWIC (choosing character encoding, and size of the context)
3. get the distribution of the forms (i.e., if the query encompassed more than one form, which ones occurred, and how many times)
4. get the distribution of the sources
5. access the information about the source of each instance

[3] Two alongations separated by commas – in this and other entries – correspond to capitalized words, e.g. Ćao, or words all in capitals, e.g. ĆEVAPČIĆI.

[4] Two alongations separated by slashes – in this and other entries – correspond to input and ouput. Given that `dj` is a valid character sequence in Bosnian, the user is required to write `djj` in order to unambiguously select only the character đ in the query, but if s/he chose “all-ASCII” display s/he can appreciate the result following the standard conventions, i.e., see simply `dj` for đ.

[5] Although this is considered a character, it is typeset as two. That is why we do not provide the octal code (the ISO-Latin2 sequence consits of two codes: the one of `d` followed by the one of ž).

[6] Two alongations separated by dash correspond to (standard) alternatives.

[7] This corpus is available from the Consortium for Lexical Research, under the entry “Yugoslav Corpus” (http: //clr.nmsu.edu/cgi-bin/Tools/CLR/clrcat).

6. get a random sample of instances
7. restrict the query to elements included in the source identification, namely (any combination of) genre, date, and (when relevant) author, title, or newspaper name

What is it that a linguist would like to do and cannot? (This is obviously an open-ended question, which will be restricted by considering only what can already be done in other systems we know of.)

1. make queries based on part-of-speech (or other linguistic) annotation
2. have the system compute statistical measures (other than simple – absolute – frequency distributions)
3. make case-insensitive queries (one can create queries which are case insensitive, but must explicitly do so)[8]
4. see a larger part of the context (one can repeat the query with a larger context size, but not look at individual results in an extended window)
5. issue queries on corpora created by previous queries[9]
6. query a sample corpus instead of sampling the queries
7. sort the result of the query

In cases like 2, 4 or 7, only some programming needs to be done; in other cases, drastic changes / additions to the corpus or the encoding are required.

Evaluation parameters

In which ways can one evaluate interfaces to corpora? In addition to comparing the particular functionalities, some possible global parameters are discussed in this section.

Ease of use To a relatively large extent, it is subjective how easy a given interface appears to a new user, but it is generally acknowledged that menus and multiple choice forms are easier to manage than a complicated query language. Furthermore, the fewer choices one must do before being able to get any result, the better. It is often the case, however, that simplicity is in conflict with power (a complex statement, in order to be rendered "easy", can yield a bunch of menus), so one is either forced to restrict the query power in order to keep the system simple or give up ease of use above a given level of complexity.

Documentation and/or help available No matter which form of presentation or options were taken, documentation, examples, and other forms of help are always useful, if only for the reason that "crystal clear" keywords, commands or forms of displaying the results can actually turn out to be confusing or simply unknown to users. Particularly useful in this connection is a tutorial, which takes the user on a guided tour through a set of specific, well-chosen examples from the corpora available. The investment on documentation and help is often neglected in the planning phases of a software project, only to be found afterwards that a significant share of the project time has been spent in producing help materials so that users can know what they are doing. (Incidentally, Garside *et al* (1997), a recent book on corpus annotation, also emphasizes the need for thorough documentation.)

Power of queries Of course, an obvious criterion for the assessment of an interface to a corpus is its query power. However, this is not so easy to compare because – as will be discussed later in this paper – one must distinguish carefully between the interface, the encoding scheme and the corpus itself. Additionally, and even if one considers the corpus + interface as a single unit, the query power only makes sense relative to the kinds of questions a researcher is interested in. For example, which is more powerful: a system that allows a part of speech (POS) query like "noun" followed by "preposition", but which gives no information about the source of the examples, or a system that does not allow simple POS queries (only POS queries attached to specific lexical items) but allows one to see all information (and larger context) about the sources?[10] There is no objective answer, it seems to us.

Speed Regardless of the theoretical irrelevance of a factor like speed (and to some extent practical, due to the technological advances that multiply computing performance in few years), it is clear that there is a waiting threshold which it is unreasonable to exceed. Applications must be quick enough so that a user does not give up the interaction, or else they must have means for interrupting, resetting, or postponing computation.

Display of results The way the results are presented is an important issue. A user is very rarely content with simply looking at the results in his/her screen. Applications that do not provide good ways to "export" results will discourage serious users. Another issue connected with result displaying is truncation. Some people, including the author, consider it infuriating to have an arbitrary cut of the number of occurrences,[11] or only be able to make queries restricted to a letter of the alphabet, for example.

OCBT evaluation

We try here to provide a realistic evaluation of the OCBT according to the parameters above.

Ease of use The desire to provide an alternative interface to the OCBT which allows menu choices has been explicit since the project start. However, due to the power of the underlying query language, it was decided not to restrict it, in order not to penalize users willing to invest in its learning.

So, despite the good intentions, the only interface available is a slightly improved version of the CQP query

[8] A partial but efficient solution to this question has already been implemented, namely, the building of another version of the corpus all in lower case, which can be queried. The problem is that the results displayed do not preserve the case of the original. This has already proved useful, however, for lexical frequency studies.

[9] This is a main feature of CQP, which we disallow – at least for the moment – for security reasons.

[10] This is allegedly the case for the CQP version of the BNC (British National Corpus, http://info.ox.ac.uk/bnc/) as opposed to the SARA interface to the BNC (see e.g. Aston & Burnard (1996)). (We have unfortunately not yet been able to get a CQP-encoded version of the BNC from the BNC staff, although we have been waiting for one since June 1997.)

[11] As in the case of the "official" BNC service, http: //thetis.bl.uk/lookup.html

language:[12] In our interface, we removed some of the most "user-unfriendly" features of CQP, namely the need for semicolon, and the requirement for quotes even when queries have a single word as search domain.

Also, some relatively complicated commands were replaced by a menu choice.[13]

Finally, basic help messages were provided in order to help a first-time user. It should be noted that these messages exist thanks to our monitoring corpus use. So, despite its original security purpose, monitoring turned out to be an excellent tool for service improvement.

Documentation Since this corpus was created with the purpose of being served through the Web, all documentation was created from scratch for the Web site. We have invested some effort in providing a parallel service in Bosnian and English, in order to facilitate the use of the corpus for those who do not speak Bosnian (e.g. Slavists not specialized in Bosnian) or English (many native speakers of Bosnian).[14] (The English version was necessary, in addition, due to the fact that some of the members of the project, the author included, do not speak Bosnian.)

There are actually three kinds of documentation one can be concerned with, here described in computer jargon:

1. installation notes – what is needed for using the service, from permission to technical details
2. user's manual – what can be done, what does the corpus consist of, which encoding details are relevant to the user, what to do in case of problems
3. linguist's "cookbook" – how can one use this corpus for the purposes of linguistic investigation? Which data is already available, which studies have already been performed, which problems are the other users interested in? How can one go about exploring this particular corpus?

We have found this third item very important in our case – possibly applying as well to the majority of linguists whose languages do not benefit from a tradition of available computer corpora. Raising corpus-awareness seems to be an important part of providing a service which makes available corpora resources, if one wants the service to be useful.

In fact, and despite the considerable number of corpus linguistics books and courses on the Internet, in conferences and/or in book form, it is not evident that e.g. every Slavic linguist will read or attend them. The situation seems to be quite different, however, if there is a description – in terms of the language s/he is an expert in – of how to use a resource that already exists.

Therefore, we have started to compile an OCBT cookbook (Leko & Santos, in prep.), which should be available from our Web site.

It should be mentioned that there are excellent texts with precisely that purpose for other corpora, e.g. Aston & Burnard (1996) for the BNC.

A tutorial, like the one by James (1995), is also extremely useful. However, due to the fact that there are already comprehensive descriptions of the command language, both on the Web and as papers, written by their authors (Christ, 1994a, 1994b; Schulze & Christ, 1996), it did not seem necessary to provide another description in tutorial form.[15]

Query power This is the major strength of our service, namely the fact that it is based on the corpus workbench which, in our opinion, allows the most powerful queries.[16] The CQP query language offers

- full regular expression power over the character alphabet (ex: [17] `"[^A-Z].*a+[gh].+st"`)
- Boolean expressions over attribute-value expressions (ex:[18] `[(word = ".*ic" & pos != "noun" & ori="PU.*") | word = ".*jli.*"]`)
- regular expressions over attribute expressions (ex:[19] `"look|bring" [pos != "VB.*"]{0,10} "up" []* [word="foo.*" & pos = "N.*"]`)
- the possibility to specify the search space (ex:[20] `"man" []* "him" within 2 s`)
- the possibility to use label references
- the possibility to compute dynamic attributes.

Some of the limitations of the CQP command language, that linguists might react to, were circumvented with additional programming from our side. The most important was the identification, for each result, of the source it belonged to.

[12] Actually, it is a major task to produce a menu-based version of all potentialities provided by the CQP query language, as its presentation below will show.

[13] For example, in order to have the distribution of the results of a query in CQP, one would have to issue the following commands: `<query>; group Last match ori;` while it is enough to express the query and ask for "Distribution of sources" in our interface. (For CQP experts, note that we are not concerned here with the CQP query language, but with the CQP command language. In our opinion, Xkwic's way to execute the same task is not easier, either.)

[14] Each version of the OCBT is thus always released in two "language versions", and depending on the language of the HTML form the user is interacting with, the whole interaction happens in Bosnian or English. In fact, for the static HTML documents, we even have the whole service in three language versions, to cater for people with systems which do not support ISO-Latin2, and where the Bosnian characters are thus replaced by the alongations discussed above.

[15] Furthermore, there is already a CQP tutorial in Schulze & Christ (1996), using English and German corpora as targets.

[16] Other grounds for the choice of CQP/IMS Corpus Workbench were: excellent service, widely used, used in connection with many different languages, mature after some years and many versions, free for research purposes, design independent of corpus, use of regular expression syntax instead of introducing a specially devised vocabulary.

[17] This totally improbable query matches words (or annotations) beginning with a character different from uppercase `A` to `Z`, including one or more `a`'s followed by either a `g` or a `h` separated by one or more characters from the suffix `st`!

[18] This query succeeds for either words ending in `ic` whose 'pos' annotation is different from noun and whose 'ori' annotation begins by `PU` or words that have the sequence of characters `jli` somewhere in the word.

[19] This query succeeds for sequences of the word `look` (or `bring`) followed by `up` not farther than 10 words (which cannot be verbs – have a 'pos' attribute with a value beginning by `VB`) – followed by a word (at any distance) beginning with `foo`, and with 'pos' beginning with `N`.

[20] This query looks for occurrences of `man` followed by `him` which occur in a span of at most two sentences (more precisely, two instances of the structural attribute 's').

Other things (which are announced for the next release, and are already working in Xkwic[21]) had / will have also to be done to make the interface easier to use, as the computation of random samples or the sorting of the concordance.

Speed There are several factors influencing speed in a Web interface, and so far we have not really been concerned with this issue, except in the general option that all computing (corpus processing) is done on the server side, i.e., in our machines. We have, however, noted a considerable speed downgrading when upgrading from Version 1.0 to 2.0 of the interface. While CQP is itself very quick, some of the functions we added are not, in particular the computation, for each result, of which source it occurred in. We are therefore considering the alternative of providing an option of quick display, and only compute the sources of the examples if explicitly asked for.

Display of results In order to allow an easy re-use of the results, they are sent in plain HTML with a minimum of markup, which can thus be copied and manipulated by any text editor (or saved as text at once). We have also rejected the option of arbitrarily cutting the results, although users are asked to contact us directly if the result of their search exceeds a very large number. There is also a maximum context size for the display of the results of a query (to present a user from downloading the whole corpus with a few queries).

The reason for our display simplicity is twofold: Not only it is impossible to please all tastes, but if users are especially keen on viewing the results in a special format they are used to, nothing prevents them to import them to their preferred tool (which could then sort, highlight, change the font, etc., the way they are used to).

Technical details It is obviously impossible to speak generally of technical appropriateness. However, in the case of a language corpus whose potential interested users are expected to be in a wide range of computational environments, some of them with very limited resources, we believe it would be totally inappropriate to depend on the latest advances of Web programming, requiring users to download a series of applications and gadgets or having the latest versions of some browsers.

On the contrary, we devised a CGI application requiring simply a Web browser, whose only (advised, not obligatory) requirement was to be able to interpret ISO-Latin2[22] so that Bosnian texts were not distorted (we consider this as part of the minimal user's friendliness when serving a corpus of one particular language).

Even this “simple” requirement, however, in order to be seriously implemented, required several days of experimentation, test in various platforms and instruction writing.

We provide as well the option to have the results displayed in ASCII (with the output alongations discussed above). This may be useful, for example, to exchange results with other linguists who are not using the corpus themselves, and have therefore no ISO-Latin2 support in their machines.

Interface, Corpus Encoding Scheme and Corpus Content

In order to present a clear description of the OCBT Web service, we have to distinguish between

- the collection of texts included
- the corpus encoding scheme, with associated tools, in which it is encoded
- the Web service, which is a front-end to the corpus workbench

This is especially relevant in the OCBT case, since the second item, the IMS Corpus Workbench, is not in any way our work, although we benefited enormously from its existence and availability. Actually, we cannot but advise people to follow our example and use it as one of the three building blocks in a Web service of this kind.

Three building blocks

It is of – at least conceptual – interest to distinguish between the properties of an interface, of a corpus encoding scheme, and of the corpus itself. Things are not clear-cut in real life, though, and in general when one talks about a corpus one is also referring to its encoding scheme as an inseparable property of it.

Regarding the interface to the corpus, it has become relatively frequent that corpora come with a special interface associated to their encoding scheme (see e.g. the COSMAS query language,[23] the Cobuild direct service,[24] or even SARA – even though, as mentioned above, there is a version of the BNC encoded in CQP[25]).

It is also possible to have “pluralistic” corpora encoded in several encoding formats, as is the case of the LOB corpus (Johansson *et al*, 1978), which, in addition to be served in vertical or horizontal format as plain text files in a variety of character sets, can also be obtained in Wordcruncher (WC, 1989) format, with its special interface. In those cases, however, the corpus encoding schemes somehow determine the interface: it would be foolish to use another interface with a corpus indexed for WordCruncher, and people who prefer plain text files probably use either their editor of choice or general Unix tools to handle them. (Any PC or Mac program for concordances which used text files for input would also be appropriate, but, in the light of our threefold classification, this means simply that one would use such a program's encoding scheme and interface.)

It is, however, relatively infrequent to have a three-module model in which one can clearly distinguish the corpus, the corpus encoding scheme, and the interface as sufficiently distinct entities to deserve special attention, as is the case of the OCBT.[26]

[21] Xkwic is an X interface to CQP, and is an integral part of the IMS Corpus Workbench. See Christ (1995).

[22] In other words, that was able to make sense of HTML <META> tags with HTTP-EQUIV="Content-Type", and a definition of charset in the CONTENT attribute.

[23] http://corpora.ids-mannheim.de/~cosmas/ProtoDocs/Deutsch/HelpQuery.html

[24] http://titania.cobuild.collins.co.uk/javademo/index.html

[25] There is also a powerful Web interface to the BNC at the University of Zurich, BNCWeb, at http://escorp.unizh.ch/

[26] One might argue that this three-phase task division is already implicit in the IMS Corpus Workbench design, since Xkwic is also separate from the encoding of the corpus proper. To require everybody to use X and/or to have access to a Unix system would be contrary to our philosophy of imposing the least possible technical requirements from our users, though.

In fact, one can consider the phase of building the corpus (choosing (parts of) texts, scanning, revising them) into a set of plain text separate files a task completely separate from the subsequent encoding in a corpus workbench (the IMS Corpus Workbench), which was also separate from devising the Web interface required to access the corpus from anywhere in the world.

This three-phase process can be partially attributed to "historical" reasons, since the project began without a clear idea of the technical steps that would follow. Nevertheless, economical reasons played an important role: it was not realistic to devise an encoding scheme and interface especially for the corpus, as long as several corpus workbenches were available.

Since the project was meant to make the corpus worldwide available with a minimum of costs or technical requirements from the user's side, furthermore, the obvious solution was a WWW interface to the corpus. Given that there was no Web interface available off-the-shelf for CQP,[27] we had to devise our own.

One should mention, for completeness's sake, that there are corpus encoding schemes with a Web interface already associated, namely TactWeb (Rockwell, 1995) which is the Web 'version' of Tact (Lancashire, 1996). Accordingly, there are corpora services based on this solution, like Knut Hofland's beta server for Norwegian texts, at http://kh.hd.uib.no/tactweb/roman-bm.htm. For technical problems (irrelevant in the context of the present paper), related to the way the network connecting PC's with Windows was set up in the University of Oslo, our attempt to use TactWeb – and eventually compare this solution to the one based on CQP – had to be abandoned.

Interface vs. corpus encoding

One can look at an interface as simply a window to the corpus. From this point of view, the contents of the corpus, and the properties of the corpus encoding scheme, represent a natural limit to what can be provided through the interface.

But one may also look at an interface, however, as the means to access information that may lie buried in the corpus, or at least as a tool to access it. From this angle, an interface can provide functionalities, and so to say increase the information that is available from the corpus. This is something that happened in the OCBT service: When developing the interface, we changed, improved and added[28] in some ways to the corpus workbench and the corpus for which we were building the interface.

A note on encoding

To end the paper, it should be noted that we have been solely concerned with technical, not linguistic, encoding. Matters which are also extremely relevant to the setting up of a corpus service, and which may in fact determine to some extent technical encoding itself, have not been discussed here.

In particular, one such question is the use of SGML (or some other markup) as opposed to a set of 'parallel corpora', one for which annotation level, as is the approach followed by the IMS Workbench.

Since we have so far no linguistic annotation in the OCBT corpus, we have really no experience in such matters. We believe, nevertheless, that an encoding which allows incremental addition of annotations (after having been computed) will have advantages in relation to one where everything is (technically) at the same level.[29]

Acknowledgements

I thank Nedzad Leko and Jan Engh for reading and commenting on a previous version of the paper.

Thanks to Nedzad Leko for being an enthusiastic first user of the OCBT, and to the other members of the OCBT team for making the project possible.

External funding for (the first phase of) the project came from NFR (the Norwegian Research Council) through the programme for cultural exchange with Eastern Europe, and from Lambertseter and Helsfyr arbeidskontor.

References

Aston, Guy & Burnard, Lou (1996). *The BNC Handbook: Exploring the British National Corpus with SARA*. Edinburgh University Press, 1996

Browne, Wayles (1993). Serbo-Croat. In B. Comrie & G. Corbett (Eds.), *The Slavonic Languages* (pp. 306--387). London: Routledge.

Christ, Oliver (1994a). A (very) brief description of the query syntax, May 31, 1994, http://www.ims.uni-stuttgart.de/projekte/tc/CQPSyntax.html

Christ, Oliver (1994b). A modular and flexible architecture for an integrated corpus query system. *Proceedings of COMPLEX'94: 3rd Conference on Computational Lexicography and Text Research (Budapest, 7-10 July 1994)* (pp. 23 – 32). Budapest.

Christ, Oliver (1995). The Xkwic User Manual. Institut für maschinelle Sprachverarbeitung, Universität Stuttgart.

Corbett, Greville (1990). Serbo-Croat. In B. Comrie (Ed.), *The Major Languages of Eastern Europe* (pp. 125—143). London: Routledge.

Garside, Roger, Leech, Geoffrey & McEnery, Anthony (Eds.) (1997). *Corpus Annotation: Linguistic Information from Computer Text Corpora*. London & New York: Longman.

James, Zoe (1995). Cobuild*Direct* Tutorial, May 1995, http://titania.cobuild.collins.co.uk/direct_tutorial.html

Johansson, S., Leech, G. & Goodluck, H. (1978). *Manual of information to accompany the Lancaster-Oslo/ Bergen Corpus of British English, for use with digital computers*. Oslo.

Lancashire, Ian (1996). In collaboration with John Bradley, Willard McCarty, Michael Stairs and T.R. Wooldridge. *Using TACT with Electronic Texts: A Guide to Text Analysis Computing Tools*. New York: The Modern Language Association of America.

Leko, Nedzad (1998). Recent changes in the Bosnian language as reflected by and documented from the Oslo

[27] Though there were a few out on the Web when we started work: e.g., in addition to the demo at Stuttgart, we knew about the Czech National Corpus (http://ucnk.ff.cuni.cz/US/cnc/) and Verbmobil's interface. Others have subsequently appeared, such as Daniel Ridings's, serving Shona and Swedish at http://ldb20.svenska.gu.se/.

[28] Although in other ways we also restricted the CQP power, and downgraded its performance.

[29] What we know for sure, from other work at the Text Laboratory, is that it is not a trivial task to merge an SGML-encoded text with its POS annotations produced by an external tagger.

Corpus of Bosnian Texts. Ms, University of Sarajevo and University of Oslo.

Leko, Nedzad & Santos, Diana (in preparation). Exploring the OCBT: Raising corpus-awareness in Bosnian linguistics. The Text Laboratory, University of Oslo.

Rockwell, Geoffrey (1995). Computer Assisted Text Analysis: An Online Workbook, May 9, 1995, http://kh.hd.uib.no/tactweb/doc/TWIntro.htm

Schulze, Bruno Maximilian & Christ, Oliver (1996). The CQP User's Manual (Version 1.6). Institutt für maschinelle Sprachverarbeitung (IMS), Universität Stuttgart.

WC (1989). *WordCruncher: WCIndex Text Retrieval Software*. Version 4.30, Provo, Utah: Electronic Text Corporation, March 1989.