Computational Philology

Martin Weisser
English Language & Linguistics
Chemnitz University of Technology

Abstract

Computational Philology is a newly emerging course of studies at German universities. It combines aspects of 'traditional' teaching and research in *Language and Computing*, Computer-Aided Language Learning and multimedial presentation and publishing of linguistic data. The following chapter will present an overview as to which aspects of information technology (IT) and linguistics may be relevant in the design of a course in Computational Philology, and which problems may be expected in teaching linguists about many aspects of IT they may be somewhat unused to. The chapter is to a large extent based on experiences from a course that has already been taught at Chemnitz University of Technology (TUC) in the summer semester 2005.

What is Computational Philology?

Computational Philology is a newly emerging course of studies at German universities. It appears as if most institutions that have begun to introduce[1] it in recent years seem to see it as a modern form of the British tradition of *Literary and Lingustic Computing* or *Computing in the Humanities*. In contrast to *corpus* or *computational linguistics*, its main emphasis seems to be directed at providing students with a knowledge of some of the basic facts about how to handle electronic texts for various linguistic and literary purposes, but sometimes extending as far as incorporating elements of electronic and printed publishing of literary works, such as information about various data and annotation formats or typesetting and publishing mechanisms. Those universities/departments who offer a whole course of studies tend to teach all of these topics in great detail and sometimes also in a fairly technical manner, which will enable graduates of these courses to find jobs in the publishing sector, whereas shorter courses, such as the one taught at the TUC are meant to provide students of linguistics and literature with a greater understanding of electronic textual data and its uses in philological studies and electronically supported language teaching.

[1] such as e.g. the Universities of Hamburg and Munich

Applications of Computational Philology

In general, *computational philology* has very similar applications to *corpus linguistics*, the main difference perhaps being a stronger orientation towards the analysis of literary materials. Thus, stylometry or authorship identification (cf. McEnery/Oakes, 2000) are certainly amongst the predominant fields of application for it. However, general research into the differences between various – also non-literary – genres, registers or text-types, perhaps through e.g. Biber's multi-dimensional analysis (cf. Biber, 1988 or Biber et al, 1998: p. 147ff), certainly ought to be ranked amongst its fields of application, too, along with perhaps research in lexicology/-graphy (cf. Ooi, 1998), general grammar, computer-based pragmatics (cf. Leech/Weisser, 2000), L1 acquisition, etc.

Based on the assumptions expressed above, in the following I want to present an overview of what may be considered essential features of a short course in computational philology. Such a basic course should at the very least contain an introduction to the most important features of electronic data that students of language need to be aware of, as well as provide basic information and foster some necessary skills in working with such types of data. The notion of data handling skills is an especially important one in this context, as a purely theoretical knowledge of standards, techniques and resources in this field is actually of little practical use.

Introducing Corpora & Other Sources of Data

The first important step in teaching students of language and literature about using electronic resources for language research is certainly to make them aware of the different options for obtaining or creating electronic data. Once they are aware of these options, they can make an educated choice as to which types of existing data they might be able to work with for their particular purposes or whether it may be necessary for them to produce their own research materials.

Perhaps the best starting point for familiarising students with the idea of electronic data is to give them an introduction to corpora, their purposes and some of their most important design features. Although they ought to progressively become more and more aware of the nearly overwhelming abundance of corpora in existence, a brief overview of the historical development introducing the most important written corpora and concepts will provide the necessary grounding. Amongst the first general corpora to be discussed should certainly be the Brown and LOB corpora – together with their modern counter-parts Frown and FLOB – representing the earliest 'model' corpora for both major varieties of English. Particular aspects regarding

their composition need to be discussed and presented in the light of the historical background, including the emphasis on written language, particular genres and issues involving computational resources, prevalent at their inception. This should be followed by a brief discussion of corpora of other varieties following in the 'tradition' of this 1 Mio. word model and then lead up to providing information about the specific model adopted for the creation of the ICE corpora collection as a 'more coherent' and comprehensive attempt at capturing differences and commonalities between the individual national varieties of English. Having covered the earlier, 'size'-limited corpora, one can then turn towards a discussion of the modern *mega-corpora*, such as the BNC, ANC or Bank of English, and their relative merits.

The introduction of the mega-corpora, especially the BNC, provides a natural 'transition point' to start discussing general issues in *corpus design*, such as *balance* and *representativeness* – including the difficulties in achieving both – as well as the differences between spoken and written corpora. When discussing spoken corpora, such as the Lancaster SEC/MARSEC or the spoken part of the BNC, it is particularly important to stress the differences between spoken materials that are transcribed only orthographically vs. those that actually contain phonetic information, perhaps even proper phonetic transcriptions. Similarly, students need to be made aware of the reasons for 'true' phonetic corpora usually being comparatively small in size, due to the effort involved in creating a thorough phonetic transcription, and, on the other hand, what the implications are in terms of quality if spoken corpora are transcribed (phonetically or orthographically) either too hastily or by non-experts.

Once students realise the usefulness and basic concepts of/behind *general purpose corpora*, specialist corpora, such as learner corpora or domain-specific corpora, and the motivation(s) for creating them, can be discussed. When they have understood the reasons behind creating *specialist corpora*, it is also possible to raise their awareness for the potential of creating their own, needs-based, corpora from materials that are either available online or can be collected using standard linguistic data-collection methods.

Since Computational Philology, unlike Computational Linguistics and Corpus Linguistics, is not a methodology primarily aimed only at students of linguistics, but should also be of interest to students of literature, perhaps the first option for collecting electronic data online that is introduced should be the use of *text archives* or *repositories*. There are actually a number of these available online, including general archives such as the websites for the *Project Gutenberg* (http://www.promo.net/pg/) or the *Oxford Text Archive* (http://ota.ahds.ac.uk/), as well

as some more specialised on older forms of English[2]. Many of the texts that are available there a available free of *copyright restrictions* or can at least be used for academic research. Students should download some of these texts themselves to see how easily such material may be obtained online in order to encourage them towards the use of electronic texts. Furthermore, many of these texts also provide a good basis for recognising some of the *formats* and practising some of the *data analysis techniques* discussed during later stages of the course.

For the sake of completeness, two further means of data collection can be introduced at this point, 'getting data off the web' and techniques for collecting other types of linguistic data, such as *interviews*, *questionnaires*, etc. The former will usually be of interest only to more technologically advanced students, as it usually involves using command-line utilities, such as *wget* or *cURL* (cURL groks URLs), for retrieving data from web resources or even writing one's own programs in a scripting language like *Perl*, which contains modules for accessing and processing web pages, such as the LWP library. Most data collected from the internet, however, often require a certain amount of 'cleanup' before they can be processed, especially if they are in HTML or PDF format, which is why this can present an initial stumbling block for less technologically oriented students.

Data Formats

Due to the fact that many students still have little or no experience in handling the aforementioned types of data – let alone know what *plain text* is – , they need to be given at least a basic introduction that helps them to understand the most common linguistic and general data formats they may encounter, as well as the issues involved in processing such data. An introduction to *HTML*, which represents part of this, may in this case fulfil a dual purpose in helping students to understand textual structure and *mark-up*, as well as providing them with a head-start in producing their own web-based materials.

Since HTML is 'omni-present' on the internet and nearly everyone can be expected to have used web pages before, perhaps one of the best ways to start the introduction is to simply open any ordinary web page with the students and ask them to switch to the source view in their browser. As the general structure of most web pages tends to follow at least the basic conventions for writing HTML pages, one can use this as a starting point for explaining what (HTML) *tags* and *attributes* look like and what the difference between *textual* vs. *meta-*

[2] listed in the appendix

information is. The latter can be achieved quite nicely by pointing out the difference between the contents of the <head> and <body> tags on the HTML page. Once students have grasped the basic concept behind *mark-up* in this way, one can to continue with a general overview describing the development from *SGML* to *XML* and the implications for the annotation of linguistic data. Although students certainly still need to be aware of SGML as a former standard for linguistic annotation, which may still be encountered in such important corpora as the BNC[3], it is perhaps even more important to give them a basic grounding in XML as the 'technology of the future'.

Depending on the time available, at this point it may be useful to follow a 'dual track' approach to provide course participants with some practical experience in annotating their own data by getting them to produce a set of HTML and XML documents and to compare the similarities and options for presentation inherent in both document formats. A thorough introduction to XML, including advanced options such as XSLT, will in most cases probably not be possible, but an introduction to structural presentation and visualisation of linguistic data via *Cascading Style Sheets* (CSS) can demonstrate both the importance of proper document structuring and options for presenting data, such as colour coding, in a clear and efficient manner. Examples for the use of colour coding include such items as highlighting linguistically important terms, presenting sample data form different *word classes* (PoS) or *syntactic roles* in different colours or indicating features such as *subject-verb-agreement*.

In teaching students of linguistics how to edit HTML or XML data, there always seems to be one slight problem, which is that most of them will not be used to handling such data in a normal (*plain text*) *editor*. It is therefore advisable to plan a certain amount of time for illustrating the concept of plain text based linguistic materials and how to use a general editor, possibly also providing information about different encoding systems/issues. On the other hand, however, gaining some hands-on experience tends to improve the students' understanding of the issues involved both in the creation and use of such data, something that later facilitates the teaching of analysis methodologies.

Once a basic understanding of corpus *annotation/mark-up* is achieved, it is then possible to introduce certain de-facto standards in *corpus annotation* – such as *TEI* and *CES* – and also further, non-standardised annotation schemes, such as schemes for pragmatic or discourse annotation (cf. Leech et al, 1998, Leech/Weisser, 2003 & Weisser, 2002 & 2004), etc.

---

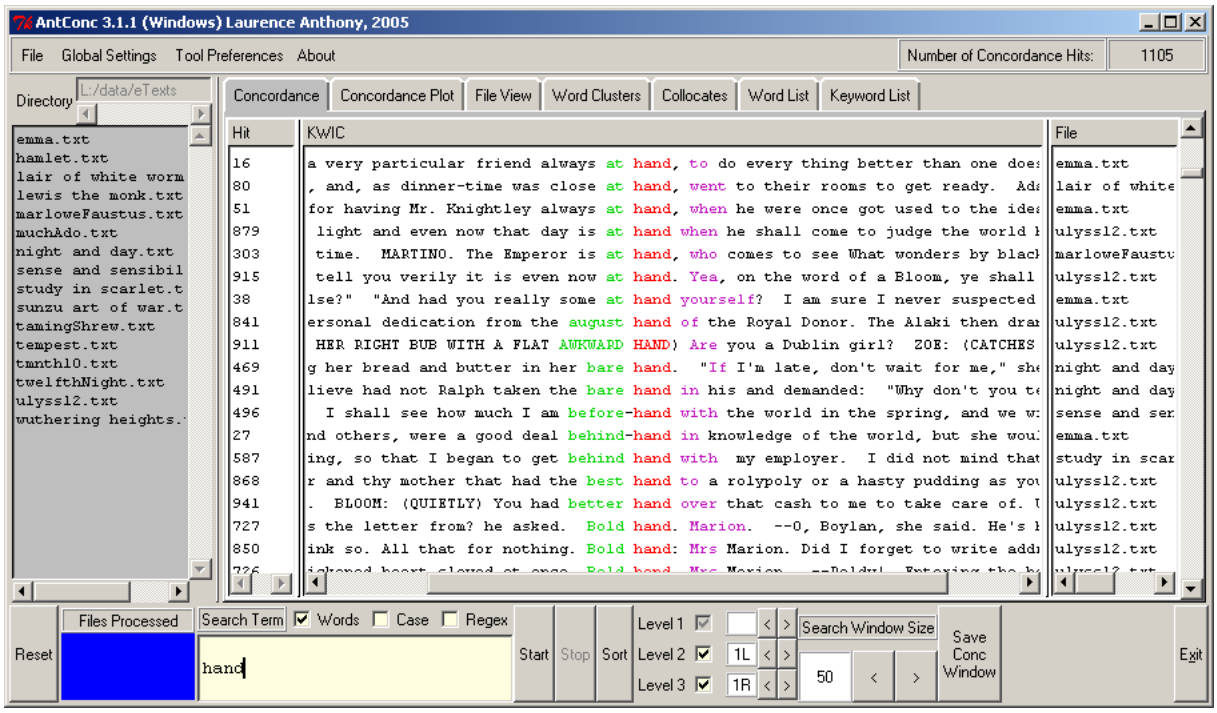
[3] although the BNC is now finally being converted to XML

Processing & Analysis

Processing or analysing linguistic data can essentialy be divided into two distinct sub-methods, one which is more *qualitative* than quantitative, and one which is almost purely *quantitative*. The first of these two involves searching for linguistic data and the latter producing 'statistically' interpretable data, such as *frequency lists*.

Searching

Searching linguistic data, although it may also be performed using command-line tools or even standard word-processing software, generally involves the use of *concordance* programs. A concordance program is a computer program that minimally allows the user to specify a specific *search term*, along with a list of files or directories to search through, and then lists all the occurrences of the search found in these files, usually in the so-called *keyword in context* (KWIC) format, illustrated in the graphic below.



The example above represents a concordance of the word *hand*, based on a number of rather haphazardly selected literary texts. It was produced using the excellent freeware concordancer *AntConc*, which provides many of the same advanced types of functionality that commercial concordancers, such as *Wordsmith* or *Monoconc*, offer. The word itself was chosen as an example due to its polysemous nature, but the concordance also illustrates other important

aspects usually associated with linguistic searching, e.g. the fact that a difference in case[4] may or may not fulfil textual functions and – perhaps more importantly – that *sorting* the results of a search in different ways may provide information about the syntactic or semantic behaviour of a search term.

One thing that is not illustrated in the graphic above is that often it is useful to be able to specify a more complex search term, which is why most concordancers provide facilities for using *regular expressions*, such as e.g. *hand(ed|ing|s)** to specify that the concordancer should look for occurrences not only of *hand*, but also for *hands*, *handed* or *handing* at the same time. A basic knowledge of regular expressions greatly enhances the efficiency in searching, which is why it is advisable to spend at least one session on this topic.

Apart from the stand-alone concordancers mentioned above, which are mainly useful with relatively small-sized corpora, there are also dedicated concordance interfaces to mega-corpora, such as the BNC, that students need to be aware of, since often corpus results from smaller corpora ought to be compared to and verified against results from larger general reference corpora (cf. Stubbs, 2001: p. 123/4). However, often such interfaces, such as *SARA* (cf. Aston/Burnard, 1998) or *BNC Web*, require institutions to have licences, which is why access to these resources may be limited. One notable exception to this is *VIEW*, a web-based interface to the BNC, which can be suggested as an alternative.

'Statistics' on Text

Doing 'Statistics' on text essentially involves frequency counts and establishing co-occurrence measures of lexical items in texts and interpreting them. However, even many students of linguistics still 'operate with' the most simple 'schoolbook' definitions of a word, i.e. assuming that a word is "something that is either delimited by spaces or punctuation". Therefore, one of the most important things to do when discussing quantitative methods is to make them aware of the problems one may encounter in dealing with real life data, where the distinctions may not be so clear-cut. For English, the best way to demonstrate this aspect is probably to give examples of compounds spelt in different ways. For instance, the word *ice cream* occurs in the BNC in the following three different ways:

[4] i.e. in the information technology sense, capital vs. small letters

word	matches	no of texts
icecream	28	17
ice-cream	368	174
ice cream	471	203

As the table above shows, the only compound form that would correspond to the simple definition of a word, i.e. the one without space or hyphen, is comparatively rare (although it occurs in 17 texts), and some native speakers may even intuitively judge this form to be incorrect. In contrast, the most frequent form would usually be interpreted as two words although it actually represents one *unit of sense*. The common drawback that most analysis programs have is that they cannot automatically determine such compounds, a fact that the analyst has to bear in mind when conducting any type of frequency analysis.

Similar problems may crop up because of *tokenisation* errors, since tokenisation, i.e. the act of splitting texts into lexical units, is usually performed on the same 'schoolbook' assumptions, so that often any sequence of characters (letters) that is followed by a punctuation mark is interpreted as a 'word'. Under certain circumstances, this may lead to numbers being split at decimal points (e.g. *2.5*) or thousand separators (e.g. *5,380*), times (e.g. *10:25 AM*), scores or ratios (e.g. *1:5*) being split at colons, etc. (cf. also Mikheev, 2003).

In most cases, students will not actually be able to overcome theses drawbacks unless they can write their own programs, but the main issue is actually to raise *awareness* of these problems so that students learn to interpret the results of any type of frequency analysis critically.

Another issue in this respect is the choice of proper units for analysis. Most software programs, such as concordancers, actually tend to create frequency lists simply by removing punctuation and then counting the resulting units. Within the limitations described above, this is not so much of a problem with regard to simple (single unit) frequency lists, but it may potentially distort the information presented by *n-gram*[5] lists, i.e. lists of two, three or more consecutive words counted as one unit, when such units are created across sentence boundaries.

[5] most commonly, bi- or tri-gram list are analysed since creating anything larger tends to get computationally very expensive in terms of memory usage and processing time.

Since n-grams already give an indication of co-occurrence, they represent the simplest from of *collocational* analysis. Other, more advanced, measures, such as *mutual information* (MI), *z-score* or *t-score* are discussed in Barnbrook (1996: pp. 89-101), which can be used as an introduction to these techniques, while at least some concordance programs, such as *AntConc*, provide facilities for generating collocation lists using them.

One further issue that needs to be discussed is that of using *stop word lists*, i.e. list of high-frequency *function words* that are generally assumed to provide little information about the lexical content of a text. The very least students should be able to understand about them is that, although they may quite usefully 'thin out' frequency lists in order to facilitate the identification of relevant *content* or *key words*, under certain circumstances, such as in the collocational analysis of idioms, they cannot simply be excluded because they often represent an integral and immutable part of such constructions.

Enriching One's Data

Previously having developed an understanding of the potential of annotations, it should now be relatively easy to demonstrate how adding further types of linguistic annotation to one's data may enhance the linguist's options for analysing his or her data. An introduction to this topic should minimally include information about the basic mechanisms involved in *Part-of-Speech (PoS) tagging*, since this form of annotation is perhaps the most useful and also most widely used one, apart from also providing a potential basis for further types of annotation, such as syntactic annotation. In order to demonstrate how students can obtain PoS-tagged data and to give them a basis for comparing different *tagsets*, tagging methods and formats, students can try out the *CLAWS* online trial service[6], as well as the freely available *Tree Tagger*. While the former is usually quite intuitive to use for most students, using the latter may require some practice for many students, though, as they need to learn how to operate and control the program from the Windows (or Solaris/Linux/Mac) *command line*. Once students know how to obtain this type of annotation, they should be made aware of how they can exploit it by specifying tags within their search options in whatever concordance programs they were introduced to earlier on the course.

Although most students will rarely ever have easy access to facilities that may provide them with further types of annotation, it is nevertheless important for them to understand what other

[6] at http://www.comp.lancs.ac.uk/computing/research/ucrel/claws/trial.html

options there may theoretically be available or could be produced manually in order to improve the value of their data. Garside et al. (1997) provides a good introduction to the different options for general corpus annotation and Leech et al. (1998) specifically discusses various options that may be relevant to the annotation of dialogue. Again, some types of annotation may be difficult to obtain, but a minimal discussion on the course should provide some information about syntactic annotation – including *skeleton* and *deep parsing* – and probably also about semantic[7] and pragmatic annotation (cf. especially Leech et al., 1998 and Weisser, 2004).

[7] see http://www.comp.lancs.ac.uk/computing/users/paul/publications/tokyo2002/ for an introduction

Abbreviations

Acronym	Expansion
ANC	American National Corpus
ICE	International Corpus of English
BNC	British National Corpus
BROWN	Brown Corpus of English
CES	Corpus Encoding Standard
CLAWS	Constituent-Likelihood Automatic WordTagging System
CSS	Cascading Style Sheets
FLOB	Freiburg Lancaster-Oslo-Bergen Corpus
FROWN	Freiburg Brown Corpus
HTML	Hypertext Markup Language
LOB	Lancaster-Oslo-Bergen Corpus
MARSEC	Machine-Readable Spoken English Corpus
SEC	Spoken English Corpus
TEI	Text Encoding Initiative
XHTML	Extensible HyperText Markup Language
XML	Extensible Markup Language
XSLT	XSL Transformations

Web Resources

Old English Texts

Labyrinth Library: Old English Literature (http://www.georgetown.edu/labyrinth/library/oe/oe.html)

The Complete Corpus of Anglo-Saxon Poetry (http://www.sacred-texts.com/neu/ascp/)

The Anglo-Saxon Chronicle (http://www.lonestar.texas.net/~jebbo/asc/asc.html)

Middle English Texts

The Middle English Collection (http://etext.lib.virginia.edu/mideng.browse.html)

Web Concordancing (BNC)

VIEW: http://view.byu.edu/

Programs & Programming Languages

AntConc (http://www.antlab.sci.waseda.ac.jp/antconc_index.html)

cURL (http://curl.haxx.se/)

Wget (http://www.gnu.org/software/wget/wget.html)

Perl (http://www.cpan.org/; http://www.activestate.com/Products/ActivePerl/)

(Stuttgart) Tree Tagger: http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html

References

Aston, G. & Burnard, L. (1998). *The BNC Handbook: Exploring the British National Corpus with SARA*. Edinburgh: EUP.

Barnbrook, Geoff. (1996). *Language and Computers*. Edinburgh: EUP.

Biber, D. 1988. *Variation Across Speech and Writing*. Cambridge: CUP.

Biber, D., Conrad, S. & Reppen, R. (1998). *Corpus Linguistics: Investigating Language Structure and Use*. Cambridge: CUP.

Dale, R., Moisl, H. & Somers, H. (Eds.). (2000). *The Handbook of Natural Language Processing*. New York: Marcel Dekker.

Garside, R., Leech, G. & McEnery, A. (1997). *Corpus Annotation: Linguistic Information from Computer Text Corpora*. London: Longman.

Granger, Sylviane. (Ed.) (1998). *Learner English on Computer*. London: Longman.

Kennedy, Graeme. (1998). *An Introduction to Corpus Linguistics*. London: Longman.

Lawler, J. & Aristar-Dry, H. (Eds.) (1998). *Using Computers in Linguistics: a Practical Guide*. London: Routledge.

Leech, G. & Eyes, L. (1997). Syntactic Annotation: Treebanks. in Garside, R., Leech, G. & McEnery, A. (Eds.). (1997). *Corpus Annotation: Linguistic Information from Computer Text Corpora*. London: Longman.

Leech, G., Myers, G. & Thomas, J. (Eds.) (1995). *Spoken English on Computer*. London: Longman.

Leech, G., Weisser, M., Wilson, A. & Grice, M. (1998). Survey and Guidelines for the Representation and Annotation of Dialogue. In Gibbon/Mertins/Moore. (Eds). (2000). *Handbook of Multimodal and Spoken Language Systems*. Dordrecht: Kluwer Academic Publishers.

Leech, Geoffrey & Weisser, Martin. (2000). Pragmatics and Dialogue. In Mitkov, R. (Ed.). (2003). *The Oxford Handbook of Computational Linguistics*. Oxford: OUP.

Leech, G. & Weisser, M. (2003). Generic Speech Act Annotation for Task-Oriented Dialogue. In Archer/Rayson/Wilson/McEnery (Eds.) (2003). *Proceedings of the Corpus Linguistics 2003 Conference*. Lancaster University: UCREL Technical Papers, vol. 16.

McEnery, A. & Oakes, M. (2000). Authorship Identification and Stylometry. In Dale/Moisl/Somers (Eds.). (2000). *The Handbook of Natural Language Processing*. New York: Marcel Dekker. pp. 545-562.

McEnery, Tony & Andrew Wilson. ([1]1996). *Corpus Linguistics*. Edinburgh: EUP.

Meyer, C. (2002). *English Corpus Linguistics*. Cambridge:CUP.

Mikheev, A. (2003). Text Segmentation. In Mitkov, R. (Ed.). (2003). *The Oxford Handbook of Computational Linguistics*. Oxford: OUP.

Ooi, V. (1998). *Computer Corpus Lexicography*. Edinburgh: EUP.

Stubbs, M. (2001). *Words and Phrases: Corpus Studies of Lexical Semantics*. Oxford: Blackwell.

Weisser, Martin. (2004). Tagging Dialogues in SPAACy. In Véronis, Jean (Ed.). *Traitement Automatique des Langues: Le traitement automatique des corpus oraux*. Vol. 45 – n° 2/2004. Cachan: Lavoisier. pp. 131-157.

Weisser, Martin. (2002). SPAACy - A Semi-Automated Tool for Annotating Dialogue Acts. *International Journal of Corpus Linguistics*, Vol. 8.1.