Requirements, Tools, and Architectures for Annotated Corpora

Nancy Ide*, Chris Brew†

*Department of Computer Science
Vassar College
Poughkeepsie, NY 12604-0520 USA
ide@cs.vassar.edu

†Department of Linguistics
The Ohio State University
Columbus OH 43210-1298, USA
cbrew@ling.ohio-state.edu

Abstract

This paper provides an overview of the needs for corpus annotation and exploitation, and some suggested strategies for development of a widely usable and reusable corpus-handling environment. The central plank of our argument is that cross-disciplinary acceptability is no longer an optional extra. The overall goal is to provide a framework which can be adapted to meet the needs of a research community which is intellectually, geographically, and linguistically diverse.

The ecology of corpora

Annotated text and speech corpora are a staple of language processing research, as well as other applications such as lexicography and corpus linguistics. The cost of creating an annotated corpus can be very high, both in direct financial terms and in terms of the opportunity cost of allocating skilled labor. So funders, whether public or commercial, have come to expect that the cost of corpus creation will be amortized over multiple research and development efforts. The more costly the corpus, the wider the market which it must reach. For example, developments in representation and handling of multi-lingual, multi-media, and multi-modal data has opened up possibilities for linking text and speech data as well as audio, video, and image. Such corpora will be very expensive, to the extent that their cost will have to be justified not just within a single research community but across disciplinary boundaries. In extreme cases corpora may have to achieve worldwide distribution in order to justify the cost of their creation. It is no accident that multinational umbrella bodies have sprung up to coordinate the dissemination of these peculiarly valuable research products.

Given these pressures, corpus designers have often been trailblazers for standardized data and annotation formats. Design ideas from corpus encoding and annotation have helped to shape the information architecture of the World Wide Web. As the flight from proprietary formats has spread from academia into the wider world, it is increasingly plausible to claim that we were right all along. But the original goals that motivated the involvement of corpus designers in the standardization process have not necessarily been achieved. XML will presumably be the encoding format of choice for the next few corpora; this tells us only that a measure of syntactic uniformity has been achieved. We still have to decide what information to encode, and how to ensure that it will meet future needs.

This paper provides an overview of the needs for corpus annotation and exploitation, and some suggested strategies for development of a widely usable and reusable corpus-handling environment. The central plank of our argument is that cross-disciplinary acceptability is no longer an optional extra. The overall goal is to provide a framework which can be adapted to meet the needs of a research community which is intellectually, geographically, and linguistically diverse.

Temporal diversity is a particular challenge. Corpora have a long active life, and as technology moves on, it becomes possible to exploit them in ways not envisaged by the original designers. This puts a high premium on flexibility. While we cannot predict future research needs, we can predict that there will be such needs. The corpus architecture must be such that it can be adapted to new situations as new research paradigms emerge. Also, in seeking to take account of the putative needs of potential future users we may (more or less by accident) build in the flexibility which is needed by current workers working in fields different from those envisaged by the designers of the corpus. These considerations will not give answers for every aspect of corpus design, but they will give some guidance, and help us to avoid egregious mistakes.

In summary, good corpora will be both *reusable,* i.e., potentially usable in more than one research project and by more than one research team, and *extensible*, i.e., capable of further enhancement. Such corpora have a better chance of surviving in the changing environment of research and development. It is an open question how to achieve the necessary flexibility.

Requirements

Consideration of the needs in several inter-related areas is necessary in order to achieve the development of tools and data that are reusable and extensible:

- *Annotation formats*--the format of the annotations themselves, e.g., morpho-syntactic tags;
- *Encoding formats*--the markup scheme used to identify and delineate elements in the data;
- *Data architecture*--the organization of data in terms of document structure, including linkage among related elements and documents;
- *Tools architecture*--a framework for tool interoperability.
- *Tool support components* – facilities that make it possible for the interoperable tools to work efficiently.

1.1 Annotation formats

The exact form of annotations need not be identical to achieve commonality. However, it is essential to work toward some set of specifications that enable mapping among annotations of the same type. For example, there exist numerous part of speech tagging schemes; problems for reusability arise when one set of tags cannot be automatically translated into another due to differences in theoretical approach. The EAGLES project[1] has developed a set of standards for various types of annotation, whose underlying design is informed by annotation categories identified by experts in the appropriate fields. The annotation categories are organized in layers with universally agreed-upon and applicable ones at the bottom, and modules for specific languages, applications, and/or theoretical approaches at higher levels. This model is being implemented for additional types of annotation by EAGLES/ISLE, involving researchers from both the U.S. and Europe.

1.2 Encoding formats and data architecture

A standardized encoding format is required for data interchange, and also for enabling easy human-readable display and access to data. This format may or may not serve as direct input to tools, but must be capable of capturing all information that is to be both input and output of tools.

As an international standard, the eXtensible Markup Language (XML) (Bray, Paoli, & Sperberg-McQueen, 1998) is the obvious basis for a standardized corpus and annotation encoding format, and is or will be used in several corpus encoding projects and corpus handling applications (e.g., LT XML, ATLAS, XCES, ANC). The XML framework provides numerous capabilities relevant to corpus-based work, including means for complex linkage within and between documents, easy data transformations using the XML Transformation Language (XSLT) (Clark, 1999), and display, manipulation, and search of data via the World Wide Web (see Ide, 2000 for a fuller discussion). However, it falls to the community to determine how to implement the facilities provided within the XML framework for corpus-handling purposes.

The cost of creating an annotated corpus can be very high. Often the first step--simply the rendering of the data, which may exist originally in the form of typesetter tapes, word processor output, etc., into a "clean" format---demands considerable time and effort. Also, although many types of annotation (e.g., part-of-speech identification, alignment of parallel texts, syntactic description, discourse segmentation, prosodic analysis, etc.) can be generated automatically, the results are never 100% error-free. Depending on the type of annotation, anything from a very small percentage to 40-50% of the data may be erroneous, thus requiring hand validation. Other types of annotation, such as co-reference annotation, semantic tagging, etc., must be performed almost entirely by hand. Therefore, apart from determining the encoding scheme, a requirement for corpus encoding is the identification of precise *levels* of markup and annotation, for example, as defined in the Corpus Encoding Standard (CES) (Ide & Priest-Dorman, 1996, Section 1.3). The CES provides means to identify whether markup and annotation have been automatically generated, whether it has been hand-validated, etc. It also distinguishes corpora on the basis of the kind and amount of annotation that has been performed, but at present, the specifications apply to text data only, and must be augmented to accommodate other kinds of data, such as speech.

1.3 Data architectures

Representation of corpora intended for use in language engineering must support the following:

- the range of annotation types
- alternative annotations and versions
- different languages
- different media and modalities (e.g., text, speech signal, audio, video, image)
- potentially complex linkage among documents, parts of documents, and different modalities

It fairly well established within the community that a "stand-off" data architecture fills these requirements. Using this scheme, the data to be annotated are contained in a base XML (or SGML) document; all annotations are in separate XML documents linked to the base, thus effectively forming a hypertextual representation of a corpus and any number of annotations. Links can be one-way or two-way (e.g., for parallel texts), and annotation documents can themselves be linked.[2] XML, which includes extensive linkage specifications for both inter- and intra-document linkage, also supports linkage between different media. So, for example, it is possible to link a speech signal, its orthographic transcription, two or more prosodic analyses using different annotation schemes, part of speech and syntactic annotation, and a video of the speaker.

1.4 Data models

Because XSLT provides a powerful retrieval mechanism that enables extraction and transformation of information from one or more XML documents, the use of a precise set of tags for encoding corpora and their annotations (e.g., those provided in the XCES--see Ide, Bonhomme, & Romary, 2000) has become less critical. However, if commonality is to be achieved, it is important to ensure a consistent underlying *data model* for corpora and annotations.

A data model is a formalized description of the data objects (in terms of composition, attributes, class membership, applicable procedures, etc.) and relations among them, independent of their instantiation in any particular form. A data model capable of capturing the structure and relations in diverse types of data and annotations is a pre-requisite for developing a common corpus-handling environment: it impacts the design of annotation schema, encoding formats and data architectures, and tool architectures.

Data models for annotated corpora have already been proposed, including the TIPSTER model (Grishman, 1998) and, more recently, a model based on an annotation graph formalism (Bird & Liberman, 2000) and implemented in the ATLAS system (Bird *et al.*, 2000).

[1] Expert Advisory Group for Language Engineering Standards, http//:www.ilc.pi.cnr.it/EAGLES/hone.html.

[2] For a fuller explanation of the stand-off data architecture, see Ide & Priest-Dorman, 1996. Part 5: Encoding Linguistic Corpora.

However, the TIPSTER model was designed primarily for use in information extraction tasks, and the annotation graph model is primarily designed for use with transcribed speech. These architectures are very general, so it cannot be argued that they lack expressive power: rather, the issue is that the tools and techniques that have been developed for operating with these formalisms have been motivated by examples drawn from particular domains. Our point is that a judgement about the convenience or appropriateness of a given style of corpus annotation or query processing typically rests on a set of implicit background assumptions about the kinds of annotation that will be needed.

Abstractly, an annotation is a one- or two-way link between an annotation object and a point (or a list/set of points) or span (or a list/set of spans) within a base data set. Links may or may not have a semantics--i.e., a type--associated with them. Points and spans in the base data may themselves be objects, or sets or lists of objects. This gives rise to several observations:

- the model assumes a fundamental linearity of objects in the base,[3] e.g., as a time line (speech); a sequence of characters, words, sentences, etc.; or pixel data representing images. It has been stated that "In some cases the time line is the only practical basis for cross reference" (Graff and Bird, 2000);
- the *granularity* of the data representation and encoding is critical: it must be possible to uniquely point to the smallest possible component (e.g., character, phonetic component, pitch signal, morpheme, word, etc.);
- an annotation scheme must be mappable to the structures defined for annotation objects in the model;
- an encoding scheme must be able to capture the object structure and relations expressed in the model, including class membership and inheritance, therefore requiring a sophisticated means to specify linkage within and between documents;
- it is necessary to consider the logistics of identifying spans by enclosing them in start and end tags (thus enabling hierarchical grouping of objects in the data itself), vs. explicit addressing of start and end points;
- it must be possible to represent objects and relations in some (fairly straightforward) form that is both usable by a variety of tools and prevents information loss;
- ideally, it should be possible to represent the objects and relations in a variety of formats suitable to different tools and applications.

1.5 Tools and tool architectures

It is well known that common language processing tools, e.g., segmenters, part of speech taggers, aligners, etc., have been "reinvented" numerous times over the past twenty years. Many tailor-made systems replicate much of the functionality of similar systems and in turn create programs that cannot be re-used by others, and so on in an endless software waste cycle (Ide & Véronis, 1993). To solve this problem, several projects have developed and implemented tools and tool architectures intended to facilitate reusability; for example:

- MULTEXT (Ide & Véronis, 1994), an EU project that developed fundamental data and tool architecture for corpora, based on the notions of tool modularity and a pipeline tool architecture, with an API interface for access to SGML-encoded documents; an SGML encoding standard for linguistic annotation (Ide, 1998a, b), and introduced the concept of "stand-off" annotation.
- LT XML (McKelvie, Brew, & Thompson, 1998), which adapted the MULTEXT architecture to view XML files as either a flat stream of markup and text, or tree-structured XML, and implements a powerful query language.
- GATE (Cunningham, Wilks, & Gaiauskas, 1996), which implements the Tipster object-oriented data model and tool architecture, and is also based on the notion of tool modularity for maximum extensibility.
- ATLAS (Bird, *et al.,* 2000), which implements a layered data and tool architecture similar to previous systems, based on an annotation graph formalism.

While each of these systems is slightly different, and newer systems have benefited from the design of earlier ones as well as advances in the technology and our understanding of the problem, they share several basic assumptions that can serve as the basis for development of a common corpus-handling environment.

To satisfy the need for flexibility, extensibility, and reusability--i.e., the need to adapt to different annotation schemes, different applications, different languages and different modalities--all agree on a modular, "plug-and-play" tool architecture based on a three-layered design: one for physical storage representation; one to translate to and from the physical storage representation to one or more internal formats, using the data model as the *lingua franca*; and an API to enable application development. All also agree on the need for powerful query capability, provision of an easy interface for annotation, and the use of "stand-off" annotation.

These emerging common practices serve as a general basis for future development, but there remain many details to work out, the most important of which is the data model that can serve as the common core for the full spectrum of corpus handling applications. And even within the tool architecture, we need to consider, for instance, to what degree we extend the notion of tool modularity (to the level of gross function, e.g., segmentation, or to an even finer grained level?), or how to best accommodate different languages and modalities (e.g., is an engine-based approach where, say, language-specific information is provided as data the best approach?). These and other issues remain ahead of us to resolve.

1.6 Tool support components

Once we have defined a tool architecture we are almost done, except that our corpora are large enough that a mere specification of the tool interfaces does not suffice to tell us how to instantiate those interfaces in a usable system. We pick on the issues of compression and indexing. For XML encoded text, excellent compression techniques exist, to the point that under some circumstances annotated corpora can take up less disk

[3] Note that this observation applies to the *fundamental* structure of stored data. Because the targets of a relation may be either individual objects, or sets or lists of objects, information with more than one dimension is accommodated.

space than their unannotated counterparts (Liefke and Suciu, 1999).

But indexing is a more complex issue. Some form of indexing of our corpora seems essential. Unless our system reaches some minimum level of responsiveness, it is unlikely to be used on a regular basis. Fortunately, good techniques exist for full-text search (Witten, Moffat and Bell, 1999) and these have been applied to linguistic corpora (Christ, 1993) producing search tools (CQP and Xkwic) which are practicable and robust.

Unfortunately, the CQP architecture relies on the assumption that the data being searched are not too different from the flat linear streams for which full text search was designed. It is feasible and extremely useful to generalize the idea of a flat stream of words to (for example) a flat stream of triples, each consisting of a word, a tag and a lemma. But as the data becomes more complex it becomes more difficult to maintain the perspicuity and efficiency of the full-text approach. It becomes increasingly attractive to adopt ideas from the database community, where the concerns of corpus designers are converging with those arising from work on semi-structured data (Bird, Buneman, & Tan, 2000; Goldman, McHugh, & Widom, 1999). If we can predict the type and frequency of the queries that will be posed over our corpora, it looks likely that we will be able to adopt existing or developing technologies to our purposes. In the next section we address the plausibility of the above premise.

1.7 What is the corpus search task?

Corpora are largely static

Linguistic corpora (with the exception of monitor corpora designed to track linguistic change) are largely static. New texts, if they are added at all, will typically be added a manageable rate. It is therefore reasonable to devote resources to the creation of indices, since we can be moderately sure that the corpora will not change radically. Many frequent search needs are fairly simple can already be supported by CQP-like technology, while more complex searches may be infrequent enough that we can get away with expressive but inefficient search technology.

Corpora are partly dynamic

However, over time new search needs will arise. Some like the historian's application mentioned by Welty and Ide[4] (Welty and Ide, 1999) will pose interesting challenges, but are unlikely to spark off a major change in patterns of corpus usage. In other cases types of search which used to seem rare or marginal may become alarmingly frequent. It could easily happen that a single successful paper using video data might lead to an avalanche of demand. So we should expect that new needs for corpus indexing will arise. Corpora may not change much, but they may come to be used in different ways.

[4] In this application we imagine that a historian wishes to find deocuments written by Government officials of the Civil War period. It is unlikely that this information is inferrable from the existing annotations of the corpus, so external knowledge must be recruited. An appropriate solution is to make use of the expressive power of a Description Logic.

Non-traditional data

Several types of non-traditional data may demand inclusion in our corpora. These include:

- Documents with diagrams, including engineering drawings.
- Illustrated books, including those in which body text and illustration are intermingled or overlaid one upon the other
- Manuscripts in which the physical details of the calligraphy and media matter.
- Interlinked texts, both the familiar ones found on the Internet and the less familiar ones that arise as the output of machine translation systems, speech transcription efforts and lexicographic endeavors.
- Databases of phonetic phenomena. Word-frequency lists, concordances.
- Personal and public information spaces. These include hard disk folder structures, mailing list archives, personal email archives, voice mailboxes and arbitrary combinations of the above.
- Dialogue: although we admit that it is odd to call this non-traditional, since it has been so heavily studied

All of these seem potentially of interest to researchers in language processing. But it is not immediately obvious which (if any) of many possible ways of coercing them to have a single privileged timeline will give satisfying search behavior.

While practical systems will almost certainly proceed on the assumption that data has a distinguished timeline, we think of this as a useful expedient, useful for the same reasons that CQP's assumption that linguistic structure can be reduced to a flat stream of tuples was useful. The working assumption makes it possible to create tools with predictable behavior, but also limits the delicacy with which we can pose queries that express our true research needs. Because CQP is a robust tool with a reasonably expressive language (essentially regular expressions over the stream of tuples), it can be pressed into service to provide heuristically valid answers to questions which might at first glance appear to demand full syntactic annotation. But it would really be better to have appropriately indexed syntactic structures. Similarly, it would be preferable to develop architectures, which offer the prospect of one day abandoning the temporary expedient of a single distinguished time line, should that prove appropriate.

Conclusions

We have argued that it falls to corpus designers to engage in the near impossible task of second-guessing the needs of future corpus users. Doing this will decrease the likelihood that we design corpora which are too narrowly focussed on the needs of particular research communities. It is conceivable that our guesses about future needs turn out to be correct, but it would be an error to tailor the design of our corpora for these needs. Rather, what is needed is an open architecture approach, which will allow future users to access corpora in the ways they find appropriate.

Based on the discussion above, we can make some general suggestions about how such an open architecture might be implemented. Clearly, where possible we should build on existing common practice, such as the emerging principles of tool architecture design outlined in section

1.5, and use existing standards where possible. For example, we can argue that it is reasonable to use the XML framework, which is an accepted international standard and compatible with developing web technology, as a base for corpus encoding and manipulation, by exploiting XSLT, XML schemas, XML QL, etc.

It is also necessary to consider the development environment for tools, data, and annotations, which must necessarily enable development by different researchers in different locations throughout the world--i.e., it is necessary to enable the *distributed* development of annotations and tools. Furthermore, in the future, data, annotations, and tools may themselves be stored on many servers in widely dispersed locations; we must find means to provide distributed services, accommodate locally adapted versions/copies, etc. Finally, there is the issue of access. Ideally, data and tools should be freely available for research, but questions of exploitation for commercial use, etc. must be resolved, and means found to control access, if necessary.

We have not discussed the single most important issue that arises in disseminating corpora, because it is not primarily a technical issue. This is the matter of intellectual property. Since corpora are valuable resources it may be difficult to persuade the owners of the copyright to grant clearance for research and/or commercial use. Part of the issue is that large corpora, especially those collected on an opportunistic basis, are complex and heterogeneous, with similarly complex patterns of authorship and ownership. But this is already the case, and as with the technical issues, substantial experience has been gained in avoiding the obvious mistakes. If, as seems likely (Whitaker, *et al.* 1999), future corpora come to include email, voice mailboxes or even video mailboxes, there will be important legal issues to solve. And, just as was the case with standardized data formats, corpus designers will have to be among the first to seriously address these issues. Does anybody have a good XML Schema for copyright release forms?

Acknowledgments

We are indebted to the participants in the “Large Corpora and Annotation Standards” workshop held in conjunction with ANLP/NAACL’00 for initial discussion of these ideas. We gained especially from a short conversation about non-traditional corpus data with John Henderson. The second author is also grateful to the Center for Cognitive Science at Ohio State University for providing travel funds. The authors blame each other for any errors.

References

Bird, S. & Liberman, M., 2000. A Formal Framework for Linguistic Annotation. *Speech Communication* (to appear).

Bird, S., Day, D., Garofolo, J., Henderson, J., Laprun, C. Liberman, M., 2000. ATLAS: A Flexible and Extensible Architecture for Linguistic Annotation. In *Proceedings of the Second International Language Resources and Evaluation Conference*. Paris: European Language Resources Association.

Bird, S., Buneman, P., & Tan, W-C., 2000. Towards a Query Language for Annotation Graphs. In *Proceedings of the Second International Language Resources and Evaluation Conference*. Paris: European Language Resources Association.

Bray, T., Paoli, J., Sperberg-McQueen, C.M. (eds.), 1998. Extensible Markup Language (XML) Version 1.0. W3C Recommendation. http://www.w3.org:TR/1998/REC-xml-19980210.

Christ, O., 1994. A modular and flexible architecture for an integrated corpus query system. In *Proceedings of COMPLEX'94*, Budapest.

Clark, J. (ed.), 1999. XSL Transformations (XSLT). Version 1.0. W3C Recommendation. http://www.w3.org/TR/xslt.

Cunningham, H., Wilks, Y., Gaizauskas, R., 1996. GATE -- a General Architecture for Text Engineering. In *Proceedings of the 16th International Conference on Computational Linguistics, COLING-96*, Copenhangen, Denmark, 1057-1060.

Goldman R., McHugh, J. & Widom, J., 1999. From Semistructured Data to XML: Migrating the Lore Data Model and Query Language. In *Proceedings of the Second International Workshop on the Web and Databases (WebDB '99),*

Graff, D. & Bird, S., 2000. Many uses, many annotations for large speech corpora: Switchboard and TDT as case studies *Proceedings of the Second International Language Resources and Evaluation Conference*. Paris: European Language Resources Association.

Grishman, R. (ed.) (1998). Tipster Text Architecture Design. http://www-nlpir.nist.gov/related_projects/tipster/.

Ide, N., 1998a. Encoding Linguistic Corpora. In *Proceedings of the Sixth Workshop on Very Large Corpora*, 9-17.

Ide, N., 1998b. Corpus Encoding Standard: SGML Guidelines for Encoding Linguistic Corpora. In *Proceedings of the First International Language Resources and Evaluation Conference,* 463-70.

Ide, N., 2000. The XML Framework and Its Implications for Corpus Access and Use. In *Proceedings of the EAGLES/ISLE Workshop on Meta-Descriptions and Annotation Schemas for Multimodal/Multimedia Language Resources and Data Architectures and Software Support for Large Corpora* (this volume). Paris: European Language Resources Association.

Ide, N. & Priest-Dorman, G., 1996. The Corpus Encoding Standard. http://www.cs.vassar.edu/CES.

Ide, N., & Véronis, J., 1994. MULTEXT: Multilingual Text Tools and Corpora. In *Proceedings of the 15th International Conference on Computational Linguistics, COLING'94*, Kyoto, Japan, 588-92.

Liefke, H. & Suciu, D., 1999 XMill: an Efficient Compressor for XML Data, University of Pennsylvania Technical Report MS-CIS-99-26. http://www.seas.upenn.edu/~liefke/xmill/xmill.html

McKelvie, D., Brew, C., & Thompson, H. 1998. Using SGML as a Basis for Data-Intensive Natural Language Processing. *Computers and the Humanities* 31:5, 367-388.

Welty, C. & Ide, N., 1999. Using the right tools: enhancing retrieval from marked-up documents. *Computers and the Humanities*. 33:10:59-84.

Whittaker, S. Hirschberg, J. Choi, J., Hindle, D., Pereira, F. Singhal, A. (1999) SCAN: designing and evaluating user interfaces to support retrieval from speech archives In *Proceedings of SIGIR’99*, 26-33

Witten, I.H., Moffat, A., Bell, T.C., 1999. *Managing gigabytes: compressing and indexing documents and images (*second edition). Morgan Kaufmann, San Francisco, CA.