Annotating Anaphoric and Bridging Relations with MMAX

Christoph Müller and Michael Strube
European Media Laboratory GmbH
Villa Bosch
Schloß-Wolfsbrunnenweg 33
69118 Heidelberg, Germany
{christoph.mueller, michael.strube}@eml.villa-bosch.de

Abstract

We present a tool for the annotation of anaphoric and bridging relations in a corpus of written texts. Based on differences as well as similarities between these phenomena, we define an annotation scheme. We then implement the scheme within an annotation tool and demonstrate its use.

1 Introduction

Anaphoric and bridging relations between discourse entities are of major importance for establishing and maintaining textual coherence. Consider the following examples, taken from the Heidelberg Text Corpus (HTC). The HTC is a collection of 577 short texts descriptive of the city of Heidelberg, which have been collected at our lab for a tourist information system in the course of the DeepMap project (Malaka & Zipf, 2000).

Im Gegensatz zu anderen Städten steht das Heidelberger Stadttheater nicht an herausgehobener Stelle, sondern es fügt sich in die Straßenfluch ohne Vorplatz ein. Der Haupteingang zeigt noch das alte Arkadenmotiv mit den flachen Segmentbögen. (HTC002)

In contrast to other cities, the Heidelberg theatre is not situated at a particularly exposed position, but it blends in with the street. The main entrance still shows the original motif of the arcades with the flat segments.

In the first sentence, das Heidelberger Stadttheater is introduced into the universe of discourse, and then referred to anaphorically by means of the pronoun es. In the next sentence, there is a bridging relation between the entities denoted by der Haupteingang and das Heidelberger Stadttheater.

Note that in each example it is the fact that the second discourse entity is either a definite NP or a pronoun that triggers the attempt to establish a relation to some previous entity: Normally, the use of a definite NP respectively a pronoun presupposes that the entity thus denoted has either already been introduced into the universe of discourse or is assumed to be familiar to the reader. This is the case in the anaphorically used pronoun in the first sentence. In the second sentence, the relation between the two discourse entities das Heidelberger Stadttheater and der Haupteingang is less obvious: The second one does not denote the same entity as the first but an entity that the second one is a part of.

The ability to automatically resolve these kinds of relations is an important feature of text understanding systems. For both the training as well as the evaluation of these systems, manually annotated corpora are required. The creation of these corpora in turn has a number of preliminaries. The phenomena anaphor, bridging and the closely related concept of antecedence need to be sufficiently defined (Section 2). On the basis of this definition an annotation scheme needs to be developed (Section 3). Finally, an annotation tool is required which implements the annotation scheme in a robust and efficient way. We review a selection of existing tools, then present MMAX (Multi-Modal Annotation in XML), our versatile Java tool (Section 4), and we demonstrate how the annotation scheme for anaphoric and bridging relations can be implemented in MMAX (Section 5).

2 Definition

In general, anaphoric as well as bridging relations hold between specifying expressions. These are those expressions that specify (i.e. are used to refer to) a particular extra-linguistic entity. In what follows, we briefly discuss the approach of Vieira and Poesio, 2000) and present our own definition. Since Vieira and Poesio, 2000) address the problem of bridging annotation, they try to find an operational and easily applicable definition. This is the main motivation for choosing Vieira and Poesio, 2000) (and not e.g. Clark, 1975), who introduced the term bridging as the background of our discussion. In the following discussion, two features of pairs of specifying expressions will be im-
important. The first one is cospecificatio (Sidner, 1983), also known as coreference, a relation holding between two or more specifying expressions which specify the same extra-linguistic entity. The second important feature is identity of the head noun. This feature is applicable to full NPs only and simply states that in a pair of NPs the head of each is realized by the same noun.

**Anaphor.** In (Vieira & Poesio, 2000), only those relations are classifiable as anaphoric in which both cospecificatio and identity of the head noun is given. Consequently, their rather strict definitio contains but one type, which the authors call direct anaphor. In contrast to this, we take only cospecificatio to be a necessary condition for anaphor. In case that identity of the head noun is also given, we call this direct anaphor as well. We believe, however, that additional sub-types of anaphor should be defined depending on the type of the anaphoric expression. Along these lines, we further distinguish pronominal anaphors, and those in which the object specified by the second expression is standing in a conceptual IS-A (or hyponym-hyperonym) relation to the object specified by the first one. Consider the NP [das Bauwerk] in the following example, which denotes a super-concept of [dem Gebäude].

Seit 1972 befinde sich das Romanische Seminar in [dem Gebäude Seminarstraße 3]. [Das Bauwerk] wurde 1847 [...] errichtet. (HTC010)

Since 1972, the Romance Seminar is situated in [the building Seminarstraße 3]. [The structure] was built [...] in 1847.

**Bridging.** Due to their strict definitio of anaphor, the term bridging is applied rather widely in (Vieira & Poesio, 2000). First, those expressions are classifiable as bridging which cospecific with their antecedent, but which do not by means of an identical, but a different head noun. Finally, also non-cospecificing expressions are classifiable as bridging as long as they stand in some lexical relation (i.e. hyponymy, meronymy or co-hyponymy) to their antecedent. The respective bridging sub-types are introduced by the authors accordingly. Our approach, in contrast, classifies as bridging only those expressions which are non-cospecificing and which stand in some conceptual relation to their antecedent. At this time, we consider the following relations to be relevant: cause-effect, part-whole, and entity-attribute, for which we give the following examples:


Therefore, [various examinations] [...] were conducted. [The result] (cause-effect) was the production of guidelines for the design of the [Hortus Palatinus]. After that, plans were made to restore [...] [the middle terrace] (part-whole).

**The Concept of Antecedence.** In pairs of anaphoric or bridging expressions, one member is regarded as the antecedent of the other. In fact, the task of resolving such a relation is often identifiable with finding the antecedent. While there are certainly cases where it is possible to find exactly one antecedent to a given expression, there are also cases where this decision is not obvious. This is true for anaphor as well as bridging antecedents: Consider the case of an NP that has been introduced into the universe of discourse, and that is referred back to anaphorically twice by means of a pronoun. We believe that it is not at all clear what is to be regarded as the antecedent of the second pronoun, since one could argue for the NP on the grounds of its semantic explicitness, or for the first pronoun on the grounds of its being the more recent cospecificating entity. The same is true with bridging expressions. In the example in Section 1, one could regard the pronoun [es] as the bridging antecedent of [der Haupteingang] as well.

**Preliminary Conclusion.** Our definitio of anaphor and bridging developed so far can be summed up as follows: Anaphoric relations can be observed between cospecificing expressions. A pair of antecedent and anaphor is always cospecificating, while on the other hand, given a set of (more than two) cospecificating expressions, determining the antecedent to a given expression is not necessarily trivial. Anaphors can be further differentiated according to the nature of the lexical items taking part in the relation. Bridging relations, in contrast, occur between non-cospecificating expressions only. Here, the criterion for division into sub-types is the conceptual relation between the entities denoted by the expressions taking part in the relation. Finally, it is possible for an expression to be antecedent to more than one anaphoric and / or bridging expressions at the same time, while we believe that the reverse, i.e. one expression having more than one antecedent, is not possible.

3 **Annotation Scheme**

The first step in the development of an annotation scheme is the definitio of the relevant markables, i.e. the class of entities in the text between which the relations to be annotated can possibly occur. It is in terms of these markables (with their attributes) and labelled relations between markables that the annotation itself is expressed.
In Section 2, we already roughly define what counts as a markable by stating that anaphoric and bridging relations hold between specifying expressions. To further distinguish, we introduce the attribute np form which allows to differentiate between the following subclasses: Proper noun, definite NP, indefinite NP, personal pronoun, possessive pronoun and demonstrative pronoun. In addition, other grammatical features of markables, like agreement or the grammatical role they play, might also be of interest. We capture these in two respective attributes, for which we specify a closed list of possible values to be assigned during annotation. These possible values are the combination person/number/gender for the first and subject, object and other for the second attribute.

In a given pair of expressions it is the way in which the second expression relates to the first one that determines whether an anaphoric or a bridging relation exists. It is natural, therefore, to represent this information on this second markable and only there. Moreover, this is the only way to allow for the representation of cases in which one markable is antecedent to several others. Since we rule out the possibility of one markable being anaphor or bridging expression to more than one antecedent, this information is easily represented by means of an attribute which identifies the markable as an anaphor or a bridging expression. We add a further attribute for the respective relation’s sub-type. For anaphoric expressions, the possible values for this attribute include direct, pronominal and IS-A, and for bridging expressions part-whole, cause-effect and entity-attribute, respectively.

Finally, the annotation of an anaphoric or bridging markable has to be complemented with information on which markable is its antecedent. This can be accomplished by supplying the markable with a further attribute. However, selecting the correct antecedent from several candidates can contain a considerable amount of interpretation on the part of the annotator. This is highly undesirable, because it is likely to force arbitrary decisions which in turn can introduce error and inconsistency into the annotation. It would be preferable, therefore, if the explicit identification of the antecedent would be optional. We do this by supplying in our annotation scheme a means to represent cospecification. With this additional representation, the annotation of anaphoric relations in our annotation scheme is a two-step process: Upon encountering an anaphoric markable and setting its general attributes, the markable is first annotated as being cospecifying with all other markables already in this set of cospecifying expressions. This is the only mandatory annotation, and together with the information that the markable is of the anaphoric type it perfectly well represents the anaphoric relation. The second, optional step consists in the specification of the markable’s exact antecedent. By separating the annotation of an anaphoric relations in this way, the concept antecedent becomes free to be used only in those cases where it is both relevant and unambiguously decidable. It is important to note that no relevant information appears to be lost here: Supplied that the linear order of markables within the text is preserved, it should be possible to establish definition antecedent to any anaphoric expression from a set of cospecifying expressions annotated within the scheme described above. Moreover, the important task of evaluating the annotation scheme is not affected either, because common evaluation algorithms for anaphor annotations (Vilain et al., 1995) do not depend on antecedence information, but treat anaphoric expressions as cospecifying equivalence classes.

What is even more important is that by the same means we can render optional the explicit specification of bridging antecedents as well. Two cases can be distinguished here: Whenever only a single candidate for antecedence exists, specifying it is trivial. Thus, the only cases where uncertainty as to the correct antecedent of a bridging expression can arise appear to be those in which multiple cospecifying candidates are available. Since bridging (as we define it) is a relation not between lexical items, but between extra-linguistic entities, and since cospecification is a transitive relation, a bridging relation can be sufficiently expressed by specifying any of the candidates. The major difference to the annotation of anaphoric relations is that in case of bridging, the selection of an antecedent is mandatory, but can be made at random, because what is really selected is not the markable but the extra-linguistic entity that it specifies.

4 Annotation Tool

This section deals with the question how the annotation scheme developed in the previous section can be implemented in a real annotation tool. Before presenting our own tool MMAX, we briefly review a selection of already existing tools.

4.1 Existing Tools

The Discourse Tagging Tool (DTTool) (Aone & Bennett, 1994) is a Tcl/Tk program for the annotation and display of antecedent-anaphor relations in SGML-encoded multilingual texts. While this field of application makes it a potential candidate for the implementation of our scheme as well, this is not the case, mainly because the tool lacks the possibility of assigning arbitrary combinations of attributes and possible values to markables, a feature that obviously is needed for the representation of different types of relations.

CLinkA3 (Orásan, 2000) is a more recent Java tool

3http://www.wlv.ac.uk/sles/compling/software/
for coreference annotation. In this case, it is a more structural constraint which prevents our annotation scheme from being implemented in this tool. This constraint results from the fact that CLinkA was built to implement the annotation scheme proposed in the MUC-7 Coreference Task Definition (Hirschman & Chinchor, 1997). In this scheme, co-specification is expressed in terms of antecedence only, a concept which we have shown to be problematic, and which our annotation scheme therefore avoids. Another problem with CLinkA is that it does not seem to support additional user-defined attributes either.

The Alembic Workbench is an annotation tool which, among other tasks, directly supports cospesification annotation. In contrast to DTTool and CLinkA, it also allows for the extension of the tag set, so that in principle the handling of different coreference phenomena is possible. The tool (like the other two mentioned before) processes SGML files into which annotation tags are inserted directly during annotation.

We regard this approach to annotation as a drawback, because it mixes the basic data (i.e. the texts to be annotated) with the annotation itself. This can give rise to problems, e.g. in cases where alternative annotations of the same data are to be compared.

Referee, a Tcl/Tk program for coreference annotation (DeCristofaro et al., 1999), is better in this respect in that it writes the annotations to a separate file leaving the annotated text itself unaltered. The format of this annotation file however, is highly idiosyncratic, rendering very difficult the subsequent analysis of the annotation. Moreover, this tool also represents cospesification in terms of antecedence only, making it impossible to annotate the former without specifying the latter. On the other hand, Referee directly supports the definitio of user-definable attributes.

Finally, the MATE Workbench is the most ambitious tool that we considered for the implementation of our annotation scheme. It has been developed in Java as a highly customizable tool for the XML-based annotation of arbitrary and possibly non-hierarchical levels of linguistic description. From a theoretical point of view, the MATE Workbench would thus be an ideal platform for the implementation of our annotation scheme. In practical terms, however, we found the performance of the program to be rather poor, rendering it practically unusable as soon as a certain corpus size was reached.

4.2 MMAX, an XML Annotation Tool

Since we found the existing tools that we considered to be insufficient for the task of implementing our annotation scheme, we decided to develop our own tool.

MMAX is written in Java for reasons of platform independence. It processes XML-encoded text corpora which make use of standoff annotation (Thompson & McKelvie, 1997). Using this technique allows us to keep apart the basic data and the annotation. XML support in Java is realized by means of the Apache implementations of an XML parser and XSL stylesheet processor.

4.2.1 The Data Model

In MMAX, written texts are represented in XML in terms of base-level and supra-base level elements. For each of these element types, Document Type Definitions (DTDs) exist which describe the structure of a well-formed element. In the following, we give DTD fragments and discuss their semantics.

A word is the most straightforward base level element for a written text. Apart from the representation of the word itself, each element of this type has an ID attribute which serves to uniquely identify the word within the text.

```xml
<!ATTLIST word id ID #REQUIRED>
```

Finally, the MATE Workbench is the most ambitious tool that we considered for the implementation of our annotation scheme. It has been developed in Java as a highly customizable tool for the XML-based annotation of arbitrary and possibly non-hierarchical levels of linguistic description. From a theoretical point of view, the MATE Workbench would thus be an ideal platform for the implementation of our annotation scheme. In practical terms, however, we found the performance of the program to be rather poor, rendering it practically unusable as soon as a certain corpus size was reached.

4.2 MMAX, an XML Annotation Tool

Since we found the existing tools that we considered to be insufficient for the task of implementing our annotation scheme, we decided to develop our own tool.

MMAX is written in Java for reasons of platform independence. It processes XML-encoded text corpora which make use of standoff annotation (Thompson & McKelvie, 1997). Using this technique allows us to keep apart the basic data and the annotation. XML support in Java is realized by means of the Apache implementations of an XML parser and XSL stylesheet processor.

4.2.1 The Data Model

In MMAX, written texts are represented in XML in terms of base-level and supra-base level elements. For each of these element types, Document Type Definitions (DTDs) exist which describe the structure of a well-formed element. In the following, we give DTD fragments and discuss their semantics.

A word is the most straightforward base level element for a written text. Apart from the representation of the word itself, each element of this type has an ID attribute which serves to uniquely identify the word within the text.

```xml
<!ATTLIST word id ID #REQUIRED>
```

The sequence of words which as a whole constitutes the complete text can be divided with respect to two criteria, a formal and a pragmatic one: Each word is part of a particular (formally defined) text, which consists of sentences, which in turn may be grouped into paragraphs. Each sentence has an ID and a span8 attribute which is a pointer to a sequence of word elements. In addition, every text can have an optional headline, which consists of any number of sentences. The formal structure of a text is described by the following DTD:

```xml
< ! [ D T D ] text ( ( headline? ), ( paragraph+ ) ) >
```

In pragmatic terms, on the other hand, a text can be regarded as a discourse, consisting of a series of discourse segments. Again, each discourse segment has an ID and a span attribute, as well as a function attribute indicative of its communicative function. This pragmatic structure can be translated into a DTD as follows:

```xml
< ! [ D T D ] discourse ( discourse_segment* ) >
```

8We use our own attribute here instead of the href attribute as define in XPointer, because our element differs from the latter both in semantics and implementation.
4.2.2 Markables

In MMAX, the XML elements representing markables possess a set of attributes which is only partly pre-defined. A closed set of fixed system attributes is complemented by an open set of user-definable attributes which depend on the annotation scheme that is to be implemented.

System Attributes. Each markable has an ID attribute which uniquely identifies it. In addition, a span attribute is needed as which maps the markable to one or more word elements. Finally, we introduce a type attribute the meaning of which will be described in the next subsection. Two additional system attributes serve to express the relations between markables. We argue that two basic relations are sufficient here.

The first is an unlabelled and unordered relation between arbitrarily many markables, which can be interpreted as set-membership, i.e. markables standing in this relation to each other are interpreted as constituting a set. Note that the interpretation of this relation is not pre-defined and needs to be specified within the annotation scheme. In order to express a markable’s membership in a certain set, a member attribute is introduced which has as its value some string specification. Set membership can thus be established/checked by unifying/comparing the member attribute values of two or more markables.

The second is a labelled and ordered relation between two markables, which is interpreted as one markable pointing to the other. Note that here, again, the nature of this pointing is not pre-defined. However, there is a structural constraint imposed on the pointing relation which demands that each markable can point to at most one other markable. Since there is no constraint as to how many different markables can point to another one, n:1 relations can be represented. A pointer attribute is required for the expression of the pointing relation. The range of possible values for this attribute is the range of existing markables’ IDs, with the exception of the current markable itself.

The DTD fragment for markables and their system attributes looks as follows:

```xml
<!ELEMENT markables (markable*)>
<!ATTLIST markable id ID #REQUIRED>
<!ATTLIST markable span CDATA #REQUIRED>
<!ATTLIST markable type CDATA #REQUIRED>
<!ATTLIST markable member CDATA #IMPLIED>
<!ATTLIST markable pointer IDREF #IMPLIED>
```

User-definable Attributes. It is by means of its user-definable attributes that a markable obtains its semantic interpretation within an annotation scheme. But even within a single scheme, it may be required to discriminate between different types of markables. In MMAX, the type attribute is introduced for this purpose. This attribute does not have any pre-defined possible values. Instead, a list of these has to be supplied by the annotation scheme. For each of these values, in turn, a list of relevant attributes and possible values has to be defined by the user. Depending on which of the mutually exclusive type attributes is assigned to a given markable during annotation, only the attributes relevant to this type will be offered in a separate attribute window for further specification.

5 Implementation

We utilize the system attribute type to discriminate between the three basic classes of markables, i.e. normal, anaphoric and bridging ones. The respective attributes and possible values for these mutually exclusive markable types can directly be adopted from the annotation scheme. Note that a subset of these is in fact identical for each type (np form, agreement and grammatical role), while other attributes’ possible values vary with the type of markable: For anaphoric markables, e.g., the sub-types direct, pronominal and IS-A are relevant, which make no sense for bridging expressions, and vice versa. This is directly supported by the adaptive attribute window. Figure 1 shows the attribute window in response to the selection of a value for the type attribute.

Cospecification between two or more markables is expressed by means of an identical member attribute. This value, though at this time realised as a string of the form setXX only, can be interpreted as what has been called universe entity, elsewhere, e.g. in the Mate Dialogue Annotation Guidelines. Adding

---

9Normal markables are those that are either not part of any relation or serve as the antecedent only.
10http://www.ims.uni-stuttgart.de/projekte/mdag
a markable to a set of cospecifying markables is accomplished in two steps: First, the set as a whole is selected by left-clicking any of its members. As a result, all members are displayed in a different color, the selected one in addition being highlighted. The markable to be added is then right-clicked, and the desired action chosen from a popup menu. Figure 2 shows this situation. Note that the attribute window has changed in response to the selection of the value anaphoric for the type attribute. Specifying the antecedent to an anaphoric expression is done as follows: First, the anaphoric markable is selected by left-clicking it. The desired antecedent is then right-clicked. Finally, selecting the appropriate menu item from a popup menu causes the anaphoric markable to point to its antecedent. The antecedent and the anaphoric respectively bridging expression are displayed in a different colour whenever the latter is selected. Note that by combination of the member and pointer attributes, cospecificatio and bridging can be represented simultaneously, which may be needed in cases of long-distance anaphor and short-distance bridging.

6 Outlook: Multi-modal Corpora

The definition of the phenomena anaphor and bridging presented in this paper as well as the annotation scheme developed were tailored mainly to the needs of written texts. This bias is apparent both in the way we define markables as well as in the attributes that we specify for them. In addition, our conception of antecedence was also influenced by it in that the notion of linear order (as can be observed between words in a written text) was at least implicit in it. All phenomena, however, are not limited to the domain of written text: They also occur in spoken language and dialogue. What is even more important: The means by which they are accomplished there include non-verbal elements like gazes and in particular pointing gestures. It is not at all clear yet if and how traditional definitions of phenomena like anaphor or bridging can be transferred to multi-modal corpora. In particular, phenomena like cross-modal anaphor and bridging need to be studied in more detail. As a step in this direction, we have already applied MMAX to the annotation of cospecificatio in multi-modal corpora (Müller & Strube, 2001).

Acknowledgements. We thank Lutz Wind for giving feedback on previous versions of MMAX. We also thank the two anonymous reviewers for their useful comments. The work presented here has been partially funded by the German Ministry of Research and Technology under grant 01 IL 904 D/2 (EMBASSI) and by the Klaus Tschira Foundation.

References


