Web-based Annotation of Anaphoric Relations and Lexical Chains

Maik Stührenberg and **Daniela Goecke** and **Nils Diewald** and **Alexander Mehler**
Bielefeld University
Germany
{maik.stuehrenberg|daniela.goecke|nils.diewald|alexander.mehler}@uni-bielefeld.de

Irene Cramer
Dortmund University
Germany
irene.cramer@uni-dortmund.de

Abstract

Annotating large text corpora is a time-consuming effort. Although single-user annotation tools are available, web-based annotation applications allow for distributed annotation and file access from different locations. In this paper we present the web-based annotation application *Serengeti* for annotating anaphoric relations which will be extended for the annotation of lexical chains.

1 Introduction

The relevance of corpus work for different tasks in the fields of linguistics is widely accepted. This holds especially for the area of (semi-)automatic text and discourse analysis which demands reference corpora in which instances of various levels of discourse structure have been annotated. Such annotation tasks are typically carried out by a combination of automatic and manual techniques. Manual annotation of large text corpora is a time consuming effort. Therefore, annotation tools are an indispensable means to overcome the limits of manual annotations. In spite of their limited level of automatization, such tools nevertheless help to semi-automatically support the annotation process and to secure consistency of manual annotations. This paper describes such an annotation tool which focuses on a certain type of discourse structures. More specifically, we deal with anaphoric relations and lexical cohesion. Our starting point is the observation that these two resources of textual cohesion (Halliday and Hasan, 1976) homogeneously induce *chain-like* discourse structures: one the one hand we have reference chains started by some antecedence and continued by some anaphora linked to the same antecedence. On the other hand, lexical cohesion generates so called *lexical chains* of semantically related tokens. Based on this observation we describe the annotation tool *Serengeti* which reflects this structural homogeneity on the level of its structural representation model as well as by its procedural annotation model. *Serengeti* includes an annotation scheme which is extended in order to support the annotation of reference chains and lexical chains. The paper is organized as follows: Section 2.1 describes the application scenario of anaphoric relations and the scheme we use to annotate them. Section 2.2 deals with the second application scenario: lexical chains. As our starting point was the former scenario, its extension to the latter one will be motivated by a separate case study of lexical chaining. Section 3 refers to related work, while Section 4 describes our annotation tool in detail. Finally, the application of *Serengeti* to annotating lexical chains is described in Section 5.

2 Annotating Large Text Corpora

The main focus of the joint work presented in this paper[1] is text technological information modelling and analysis of various types of discourse. Within our research group we deal with the integration of

[1]The work presented in this paper is a joint effort of the projects A2, A4 and B1 of the Research Group *Text-technological modelling of information* funded by the German Research Foundation. See `http://www.text-technology.de` for further details.

Proceedings of the Linguistic Annotation Workshop, pages 140–147,
Prague, June 2007.

heterogeneous linguistic resources. This applies especially to the *Sekimo* project (A2) which focusses on the application domain of anaphora resolution. We use the term 'heterogeneity' to refer to resources that differ either in terms of form (text, audio, video) or in terms of function (e. g. lexicons, annotated texts). Connection between these resources can be established with the means of XML, cf. Simons (2004). Integrating resources via an abstract interface is necessary due to different reasons: The resources used have often been developed independently from each other and a cascaded application of one resource to the output of another resource is not always possible. Furthermore, the output of different resources often cannot be encoded in a single structure without driving into incompatibilites (i. e. XML overlap). Therefore an architecture was developed which allows for the combination of the output structures of several linguistic resources into a single XML annotated document and which is described in detail in Witt et al. (2005) and Stührenberg et al. (2006) .

2.1 Anaphoric Relations

Motivation and Background Resolving anaphoric relations needs a variety of different information (e. g. POS, distance information, grammatical function, semantic knowledge, see, for example, Mitkov (2002) for an overview). Several resources are applied to a corpus of 47 texts and the output structures are combined into a single XML document using the architecture mentioned above. In order not only to integrate but also evaluate resources for a given linguistic task formally in terms of precision and recall, it should be possible to either switch on or switch off a given resource. In the application domain of anaphora resolution evaluation is done as follows. Each *discourse entity* or *referent* (cf. Karttunen (1976)) is annotated as an XML element which holds a variety of attribute information. Each XML element is reinterpreted as a feature vector; pairs of discourse entities between which an anaphoric relation holds form a single feature vector with additional information relevant for anaphora resolution (e. g. distance information, identity of grammatical form, semantic relatedness of underlying lemmata and the like). In order to evaluate different resource settings, decision trees with varying sets of feature vectors are used for the process of anaphora resolution. Xiaofeng et al. (2004) or Strube and Müller (2003) have shown the feasibility of decision trees for the domain of anaphora resolution; we have chosen this approach as it makes it possible to easily switch the information set for training and evaluation as opposed to e. g. rewriting rule sets. Both, training and evaluation as well as empirically based analysis of anaphora need an annotated reference corpus (Poesio et al., 2002). Scheme and annotation process are described in the following section.

The Annotation Scheme for Anaphoric Relations Several annotation schemes for annotating anaphoric relations have been developed in the last years, e. g. the UCREL anaphora annotation scheme (Fligelstone, 1992; Garside et al., 1997), the SGML-based MUC annotation scheme (Hirschmann, 1997), and the MATE/GNOME Scheme (Poesio, 2004), amongst others. In order to annotate discourse relations – either anaphoric relations or lexical chains (cf. Section 2.2) – two types of information have to be specified. First, the *markables*, i. e. the elements that can be part of a relation, have to be specified (cf. Müller and Strube (2003)). Second, the *relation(s)* between markables and their respective types and subtypes have to be defined. The markables form a basis for the annotation process and therefore have to be annotated in advance. Normally, for a domain under investigation, elements are denoted as being markables either via a specific element or via the use of a universal attribute. In our system, discourse entities are detected automatically on the basis of POS and parsing information. The annotation scheme for annotating anaphoric relations is an extension of the scheme presented by Holler et al. (2004) that has been developed for annotations in the context of text-to-hypertext conversion in the project B1 *HyTex*. We adopt the distinction between coreference and cospecification but we extend the annotation scheme for an explicit distinction between cospecification (direct anaphora) and bridging (associative or indirect anaphora). Thus, we add the primary relation type *bridgingLink* (denoting bridging) to the already existing one (*cospecLink*). Each primary relation type includes different secondary relation

Listing 1: The annotation format for anaphoric relations. Shortened and manually revised output

```
1  <chs:chs>
2   <chs:text>
3    <cnx:de deID="de8" deType="namedEntity" headRef="w36">
4     <cnx:token ref="w36">Maik</cnx:token></cnx:de>
5    <cnx:token ref="w37">hat</cnx:token> <cnx:token ref="w38">kein</cnx:token>
6    <cnx:token ref="w39">eigenes</cnx:token> <cnx:token ref="w40">Fahrrad</cnx:token>,
7    <cnx:token ref="w42">und</cnx:token>
8    <cnx:de deID="de10" deType="namedEntity" headRef="w43">
9     <cnx:token ref="w43">Marie</cnx:token></cnx:de>
10   <cnx:token ref="w45">fährt</cnx:token> <cnx:token ref="w46">nicht</cnx:token>
11   <cnx:token ref="w47">in</cnx:token>
12   <cnx:de deID="de11" deType="nom" headRef="w49">
13    <cnx:token ref="w48">den</cnx:token>
14    <cnx:token ref="w49">Urlaub</cnx:token></cnx:de>.
15   <cnx:de deID="de12" deType="nom" headRef="w53">
16    <cnx:token ref="w52">Zwei</cnx:token>
17    <cnx:token ref="w53">Kinder</cnx:token></cnx:de>,
18   <cnx:de deID="de13" deType="nom" headRef="w56">
19    <cnx:token ref="w55">eine</cnx:token>
20    <cnx:token ref="w56">Gemeinsamkeit</cnx:token></cnx:de>:
21  </chs:text>
22  <cnx:token_ref id="w36" head="w37" pos="N" syn="@NH" depV="subj" morph="MSC␣SG␣NOM" />
23  <chs:semRel>
24   <chs:bridgingLink relType="hasMember" antecedentIDRefs="de8␣de10" phorIDRef="de12"/>
25  </chs:semRel>
26 </chs:chs>
```

types that specify the subtype of the relation, e. g. *ident* or *hypernym* as secondary types of cospecLink or *meronym* or *setMember* as secondary types of bridgingLink. An example annotation of an indirect anaphoric relation (element `bridgingLink`, line 30) between the discourse entities `de12` (lines 18 to 21) and `de8` (lines 3 to 5) and `de10` (lines 9 to 11) can be seen in Listing 1.

2.2 Lexical Chaining

Motivation and Background Based on the concept of lexical cohesion (Halliday and Hasan, 1976), computational linguists (inter alia Morris and Hirst (1991)) developed a method to compute a partial text representation: *lexical chains*. These span over passages or even the complete text linking lexical items. The exemplary annotation in Figure 1 illustrates that lexical chaining is achieved by the selection of vocabulary and significantly accounts for the cohesive structure of a text passage. Items in a lexical chain are connected via semantic relations. Accordingly, lexical chains are computed on the basis of a lexical semantic resource such as WordNet (Fellbaum, 1998). Figure 1 also depicts

> Jan sat down to rest at the foot of a huge beech-tree. Now he was so tired that he soon fell asleep; and a leaf fell on him, and then another, and then another, and before long he was covered all over with leaves, yellow, golden and brown.
>
> **Chain 1:** sat down, rest, tired, fell asleep
> **Chain 2:** beech-tree, leaf, leaves
>
> Unsystematic relations not yet considered in lexical chaining: foot / huge – beech-tree; yellow / golden / brown – leaves

Figure 1: Chaining Example (adapted from Halliday et al. (1976))

several unsystematic relations, which should in principle be considered. Unfortunately, common lexical resources do not incorporate them sufficiently. Most systems consist of the fundamental modules shown in Table 1.

However, in order to formally evaluate the performance of a given chainer in terms of precision and recall, a (preferably standardized and freely available) test set would be required. To our knowledge such a resource does not exist – neither for English

Module	Subtasks
chaining candidate selection	preprocessing of corpora: determine chaining window, sentence boundaries, tokens, POS-tagging chunks etc.
calculation of chains / meta-chains	look-up: lexical semantic resource (e.g. WordNet), scoring of relations, sense disambiguation
output creation	rate chain strength (e.g. select strong chains), build application specific representation

Table 1: Overview of Chainer Modules

nor for German. We therefore plan to develop an evaluation corpus (gold standard), which on the one hand includes the annotation of lexical chains and on the other hand reveals the rich interaction between various principles to achieve a cohesive text structure. In order to systematically construct sound guidelines for the annotation of this gold standard, we conducted a case study.

Case Study Six subjects were asked to annotate lexical chains in three short texts and in doing so record all challenges and uncertainties they experienced. The subjects were asked to read three texts – a `wikipedia` entry (137 words), a newspaper article (233 words), and an interview (306 words). They were then given a list of all nouns occurring in the articles (almost all chainers exclusively consider nouns as chaining candidates), which they had to rate with respect to their 'importance' in understanding the text. On this basis they were asked to determine the semantic relations of every possible chaining candidate pair, thus chain the nouns and annotate the three texts. Just like previously reported case studies (Beigman Klebanov, 2005; Morris and Hirst, 2004; Morris and Hirst, 2005) aiming at the annotation of lexical chains, we found that the inter-annotator agreement was in general relatively low. Only the annotation of very prominent items in the three texts, which accounted for approximately one fifth of the chaining candidates, resulted in a satisfying agreement (that is: the majority of the subjects produced an identical or very similar annotation). However, all subjects complained about the task. They found it rather difficult to construct linearized or quasi-linearized structures, in short, chains. Instead, most of the subjects built clusters and drew very complex graphs to illustrate the cohesive relations they found. They also pointed out that only a small fraction of the candidate list contributed to their text understanding. This clearly supports our observation that most of the subjects *first* skimmed through the text to find the most prominent items, established chains for this selection and *then* worked the text over to distribute the remaining items to these chains. We therefore assume that lexical chains do not directly reflect reading and understanding processes. Nevertheless, they do in some way contribute to them. Many subjects additionally noted that a reasonable candidate list should also include multi-word units (e.g. technical terms) or even phrases. Furthermore, as already reported in previous work (Morris and Hirst, 2004), the semantic relations usually considered seem not to suffice. Accordingly, some subjects proposed new relations to characterize the links connecting candidate pairs. Given our own findings and the results reported in previous work, it is obviously demanding to find a clear-cut border between the concepts of lexical chaining, semantic fields, and co-reference/anaphora resolution. Definitely, the annotation of co-reference/anaphora and lexical chains is inherently analogous. In both cases an annotation layer consisting of labelled edges between pairs of annotation candidates is constructed. However, we assume that the lexical chaining layer might contain more edges between annotation candidates. As a consequence, its structure presumably is more complex and its connectivity higher. We thus plan to conduct an extended follow-up study in order to explore these differences between the annotation of lexical chains and co-reference/anaphora. We also intend to take advantage of – amongst other aspects – the inter-annotator comparison functionality provided by *Serengeti* (see Section 4 for a detailed description) in order to implement a formally correct inter-annotator agreement test.

3 Available Tools for Annotating Linguistic Corpora

Both the anaphora resolution and the lexical chaining scenario have shown the importance of an easyto-use annotation tool. Although a wide range of annotation tools is available, one has to separate tools for annotating multimodal corpora from tools for annotating unimodal (i. e. text) corpora. Dipper et al. (2004) evaluated some of the most commonly used tools of both categories (TASX Annotator, EXMARaLDA, MMAX, PALinkA and Systematic Coder). Besides, other tools such as ELAN[2] or Anvil[3] are available as well, as are tool kits such as the Annotation Graph Toolkit (AGTK)[4] or the NITE XML Toolkit.[5] While multimodal annotation demands a framework supporting the time-aligned handling of video and audio streams and, therefore, much effort has been spent on the design and development of tools, unimodal annotation has often been fulfilled by using ordinary XML editors which can be error-prone. Nevertheless, specialized annotation frameworks are available as well, e. g. MMAX can be used for multi-level annotation projects (cf. Müller and Strube (2001; 2003)). However, as annotation projects grow in size and complexity (often multiple annotation layers are generated), collaborative annotation and the use of annotation tools is vital.

- Ma et al. (2002), for example, describe collaborative annotation in the context of the AGTK. But since most of the aforementioned applications have to be installed locally on a PC, working on a corpus and managing annotations externally can be difficult.

- Another problem worth to be mentioned is data management. Having several annotators working on one text, unification and comparison of the markup produced is quite difficult.

- Furthermore, annotation tools help to increase both the quality and quantity of the annotation process.

Recent web technologies allow the design of web-based applications that resemble locally installed desktop programs on the one hand and provide central data management on the other hand. Therefore distributed annotation is possible regardless of location, provided that an internet connection is available. In this paper we propose the web-based annotation application *Serengeti*.

4 A new Approach: Serengeti

As the *Sekimo* project is part of a research group with interrelated application domains, annotation layers from different projects have been evaluated for their interrelationship (e. g. Bayerl et al. (2003; 2006)). This led directly to the open design of *Serengeti* – an annotation tool with the fundamental idea in mind: making possible the annotation of a single layer (or resource) and the use of the best annotation possible and the best available resources. *Serengeti* allows for several experts to annotate a single text at the same time as well as to compare the different annotations (inter-annotator-agreement) and merge them afterwards. Access to the documents is available from everywhere (an internet connection and a browser is required).

4.1 Technical Overview

Serengeti is a web application developed for Mozilla Firefox,[6] thus its architecture is separated into a client and a server side, following the principles and tools of AJAX (Asynchronous JavaScript and XML, cf. Garrett (2005)). While groups, documents and annotations are managed centrally on the server side, all user interactions are rendered locally on the client side.[7]

4.2 Graphical User Interface

The Graphical User Interface (GUI) of *Serengeti* is subdivided into several areas (cf. Figure 2). The main area renders the text to be annotated, roughly laid out in terms of paragraphs, lists, tables and non-text sections according to the input XML data. Additionally, predefined markables are underlined and followed by boxes containing the markables' unique identifiers. These boxes serve as clickable buttons to choose markables during the annotation. At this

[2] `http://www.lat-mpi.eu/tools/elan/`
[3] `http://www.dfki.de/~kipp/anvil/`
[4] `http://agtk.sourceforge.net/`
[5] `http://www.ltg.ed.ac.uk/NITE/`

[6] *Serengeti* is targeted at platform independence, so we've chosen Firefox, which is freely available for several operating systems. Future versions will support other browsers as well.
[7] Each *Serengeti* installation supports more than one workgroup. Server sided data management allows the use of versioning systems like *CVS* or, in our case, *Subversion*.

time, adding markables, i. e. changing the input data, is not allowed.[8] This ensures that all annotators use the same base layer. A section at the bottom of the interface represents the annotation panel with a list of all annotated relations on the left and all editing tools on the right side. An application bar at the top of the GUI provides functions for choosing and managing groups, documents and annotations.

4.3 Annotation Process

After logging in and choosing a document to annotate, new relations between markables can be created. The markables that take part in the relation are chosen by left-clicking the boxes attached to the underlined markables in the text and, if necessary, unchecked by clicking them once again. To encode the type of a relation between chosen markables, an input form at the bottom right of the page provides various options for specifying the relation according to the annotation scheme. The OKAY command adds created relations to the list, which can subsequently be edited or deleted. In regard to their state, relation bars in the list can be highlighted differently to simplify the post-editing (i. e. new relations, old/saved relations, commented relations or incomplete relations).[9] The user can save his work to the server at any time. After the annotation process is completed, the COMMIT command (located in the document menu) declares the annotation as finished.

4.4 Comparing Annotations and Reaching a Consensus

In order to achieve the best annotation results it is necessary to provide an opportunity for the evaluation of single annotations or comparing of multiple annotations on one single document (either by different annotators or identical annotators at different points in time). This allows for verification of the quality of the annotation scheme and for valid training data for automated natural language processing tools. For this purpose, a special user access, the *Consensus User (CU)*, has been developed as part of *Serengeti*'s concept. Loading a document as a CU, it is possible to choose a single annotation done by any other annotator (either work in progress or committed) as the basis for the final annotation. This is done with the same tools as those for the annotation process. If satisfied, the CU can declare the annotation as ultimately closed via the COMMIT command.

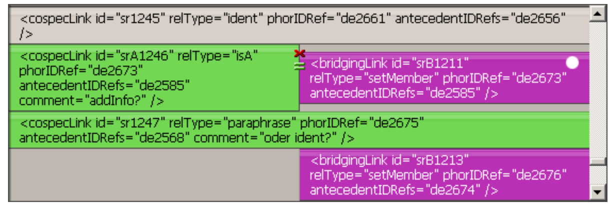

Figure 3: *Serengeti*'s comparison window in the lower left part of the GUI.

Furthermore, the CU can compare two annotations with each other. The relations annotated by both users are then displayed in the relation list and juxtaposed in case they differ in at least one aspect (e. g. different relation types as in Figure 3).[10] On this basis the CU can decide which relation to accept and which one to reject. Again, all editing options are at the user's disposal.
While editing single or multiple user annotations, the CU can save the current state of his work at any time. Afterwards these annotations will appear in the ANNOTATIONS MENU as well and can be selected for further evaluation and comparison.[11]

5 Extending Serengeti

Although one might doubt that *Serengeti* is directly applicable to annotating lexical chains, this can nevertheless be done straightforwardly using the annotation described in Section 2.1. Our starting point is as follows: As markables we refer to entities of the parser output (i. e. tokens) where a user can mark a token as the initial vertex of a chain. In order to reflect the findings of our case study on lexical chaining we distinguish two cases: Either the annotator decides that a newly entered token enlarges

[8] The definition of XML elements as markables and the layout and relation type specification is driven via an external configuration script, adjustable for each group.

[9] It is possible to hide relations according to their state as well.

[10] At this point the assignment of relations is important. Anaphoric relations, for example, are assigned to each other if their anaphoric element is the same. If there is more than one relation with identical anaphoric elements, the relations are sorted by their relation types and their antecedent(s).

[11] Comparisons require conflictless annotations, i. e. saved comparisons have to be free from juxtaposed relations.

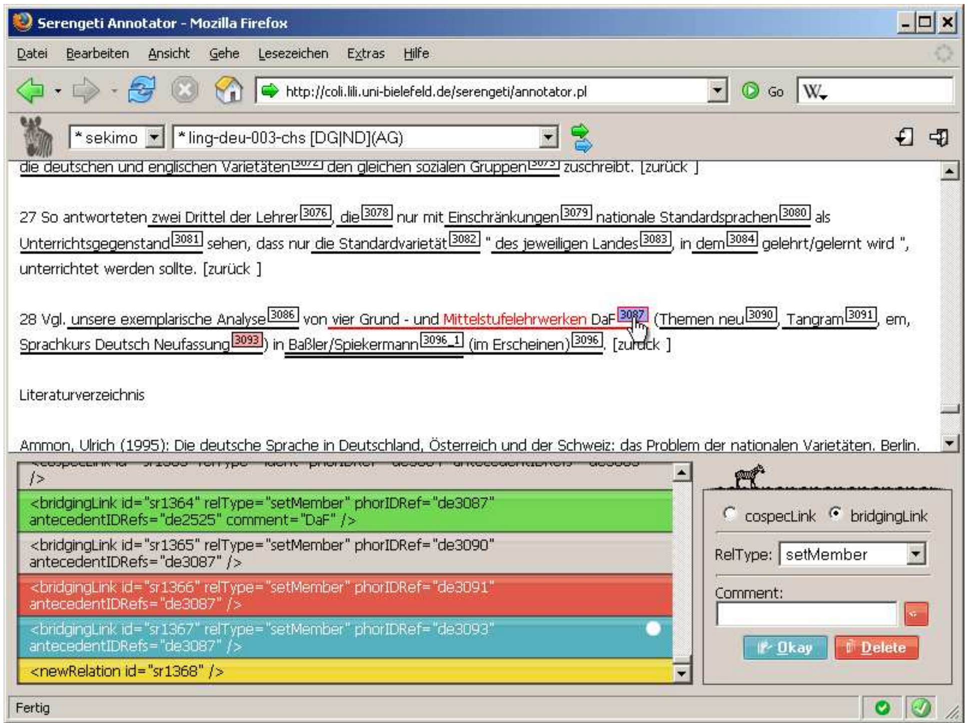

Figure 2: *Serengeti*'s User Interface. Screenshots of *Serengeti* Version 0.7.1

an already marked-up chain by explicitly relating it to one of its links or he implicitly assigns the token to that chain as a whole which is visually represented as part of *Serengeti*'s interface. In the first case we just face another use case of our annotation scheme, that is, a link between two tokens or spans of a text where this link may be typed according to some linguistic relation that holds between the spans, e. g. hyponymy. In the second case of an implicit chain assignment we proceed as follows: We link the newly processed token to the last vertex of the lexical chain to which the token is attached and type this relation non-specifically as *association*. As a result, we reduce this use case to the one already mapped by our general annotation scheme. In order to make this a workable solution, we will integrate a representation of lexical chains by means of *tag clouds* where each chain is represented by a subset of those lexical units which because of their frequency are most important in representing that chain. Following this line of extending *Serengeti*, we manage to use it as an annotation tool which handles anaphoric relations *as well as* lexical chains.

6 Discussion and Outlook

Serengeti can be used to create corpus data for training and evaluation purposes. An installation of *Serengeti* is available online.[12] Currently, the tool is being generalized to allow the annotation of lexical chains and several other annotation tasks. More specifically, we plan to incorporate any kind of chain-like structuring of text segments and to make the chains an object of annotation so that they can be interrelated. This will allow to incorporate constituency relations into the annotation process. Beyond that we will incorporate metadata handling to document all steps of the annotation process.

References

P. S. Bayerl, H. Lüngen, D. Goecke, A. Witt, and D. Naber. 2003. Methods for the Semantic Analysis of Document Markup. In C. Roisin, E. Muson, and C. Vanoirbeek, editors, *Proceedings of the 2003 ACM symposium on Document engineering (DocEng)*, pages 161–170, Grenoble. ACM Press.

[12] http://coli.lili.uni-bielefeld.de/serengeti/

B. Beigman Klebanov. 2005. Using readers to identify lexical cohesive structures in texts. In *Proceedings of ACL Student Research Workshop*.

S. Dipper, M. Götze, and M. Stede. 2004. Simple Annotation Tools for Complex Annotation Tasks: an Evaluation. In *Proceedings of the LREC Workshop on XML-based Richly Annotated Corpora*, pages 54–62, Lisbon, Portugal.

C. Fellbaum, editor. 1998. *WordNet. An Electronic Lexical Database*. The MIT Press.

S. Fligelstone. 1992. Developing a Scheme for Annotating Text to Show Anaphoric Relations. In G. Leitner, editor, *New Directions in English Language Corpora: Methodology, Results, Software Developments*, pages 153–170. Mouton de Gruyter, Berlin.

J. J. Garrett, 2005. *AJAX: A New Approach to Web Applications*. Adaptive Path LLC, February, 18. Online: `http://www.adaptivepath.com/publications/essays/archives/000385.php`.

R. Garside, S. Fligelstone, and S. Botley. 1997. Discourse Annotation: Anaphoric Relations in Corpora. In R. Garside, G. Leech, and A. McEnery, editors, *Corpus Annotation: Linguistic Information from Computer Text Corpora*, pages 66–84. Addison-Wesley Longman, London.

D. Goecke and A. Witt. 2006. Exploiting Logical Document Structure for Anaphora Resolution. In *Proceedings of the 5th International Conference.*, Genoa, Italy.

Michael A. K. Halliday and Ruqaiya Hasan. 1976. *Cohesion in English*. Longman, London.

L. Hirschmann. 1997. MUC-7 Coreference Task Definition (version 3.0). In L. Hirschman and N. Chinchor, editors, *Proceedings of Message Understanding Conference (MUC-7)*.

A. Holler, J.-F. Maas, and A. Storrer. 2004. Exploiting Coreference Annotations for Text-to-Hypertext Conversion. In *Proceeding of LREC*, volume II, pages 651–654, Lisbon, Portugal.

L. Karttunen. 1976. Discourse Referents. *Syntax and Semantics: Notes from the Linguistic Underground*, 7:363–385.

X. Ma, L. Haejoong, S. Bird, and K. Maeda. 2002. Models and Tools for Collaborative Annotation. In *Proceedings of the Third International Conference on Language Resources and Evaluation*, Paris. European Language Resources Association.

R. Mitkov. 2002. *Anaphora Resolution.* Longman, London.

J. Morris and G. Hirst. 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational linguistics*, 17(1):21–48, March.

J. Morris and G. Hirst. 2004. Non-classical lexical semantic relations. In *Proceedings of HLT-NAACL Workshop on Computational Lexical Semantics*.

J. Morris and G. Hirst. 2005. The subjectivity of lexical cohesion in text. In J. C. Chanahan, C. Qu, and J. Wiebe, editors, *Computing attitude and affect in text*. Springer.

C. Müller and M.l Strube. 2001. Annotating Anaphoric and Bridging Relations with MMAX. In *Proceedings of the 2nd SIGdial Workshop on Discourse and Dialogue*, pages 90–95, Aalborg, Denmark.

C. Müller and M. Strube. 2003. Multi-Level Annotation in MMAX. In *Proceedings of the 4th SIGdial Workshop on Discourse and Dialogue*, pages 198–207, Sapporo, Japan.

M. Poesio, T. Ishikawa, S. Schulte im Walde, and R. Viera. 2002. Acquiring lexical knowledge for anaphora resolution. In *Proc. of the 3rd Conference on Language Resources and Evaluation (LREC)*.

M. Poesio. 2004. The MATE/GNOME Scheme for Anaphoric Annotation, Revisited. In *Proceedings of SIGDIAL*, Boston, April.

G. Simons, W. Lewis, S. Farrar, T. Langendoen, B. Fitzsimons, and H. Gonzalez. 2004. The semantics of markup. In *Proceedings of the ACL 2004 Workshop on RDF/RDFS and OWL in Language Technology (NLPXML-2004)*, Barcelona.

M. Strube and C. Müller. 2003. A Machine Learning Approach to Pronoun Resolution in Spoken Dialogue. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, volume 1, pages 168–175. ACL 03.

M. Stührenberg, A. Witt, D. Goecke, D. Metzing, and O. Schonefeld. 2006. Multidimensional Markup and Heterogeneous Linguistic Resources. In D. Ahn, E. T. K. Sang, and G. Wilcock, editors, *Proceedings of the 5th Workshop on NLP and XML (NLPXML-2006): Multi-Dimensional Markup in Natural Language Processing*, pages 85–88.

A. Witt, D. Goecke, F. Sasaki, and H. Lüngen. 2005. Unification of XML Documents with Concurrent Markup. *Literary and Lingustic Computing*, 20(1):103–116.

Y. Xiaofeng, J. Su, G. Zhou, and C. L. Tan. 2004. Improving Pronoun Resolution by Incorporating Coreferential Information of Candidates. In *Proceedings of ACL*.