

Towards Natural Interactive Question Answering

Gerhard Fliedner

Computational Linguistics
Saarland University
D-66123 Saarbrücken
fliedner@coli.uni-sb.de

Abstract

Interactive question answering systems should allow users to lead a coherent information seeking dialogue. Compared with systems that only locally evaluate a question, interactive systems facilitate the information seeking process and provide a more natural feel. We show that by extending a QA system to handle several types of anaphora and ellipsis, the naturalness of the interaction can be considerably improved. We describe an implementation in our prototype QA system for German and give a walk-through example of the enhanced interaction capabilities.

1. Introduction

Question Answering (QA) systems have received considerable interest from researchers in computational linguistics recently. The constant improvement of natural language processing tools and the increasing availability of broad-coverage linguistic resources have lead to a point where selectively applying comparatively deep linguistic methods can result in a noticeable improvement of system performance. The probably most prominent recent example is described in Moldovan et al. (2003).

There is general agreement that QA systems could profit greatly from providing means of user interaction up to actual dialogue capabilities (cf. Burger et al. (2001)). For people's information needs can rarely be formulated as one or more single questions, but rather should be accommodated by a coherent, information-seeking dialogue with starting questions and follow-up questions. However, only recently QA systems have become sophisticated enough to allow investigations into that direction.

In this paper, we will explore several phenomena concerning the use of anaphora and ellipsis in QA: From a linguistic point of view, an interactive approach to QA is characterised by the necessity to provide for discourse coherence, especially by use of anaphora (and coreference in general) and ellipses. For a user, it is, for example, very natural to use a pronoun to refer back to an entity under discussion in a follow-up question.

An interactive QA system should therefore be able to resolve coreference and ellipsis. This is not only a question of allowing the users to shorten their inputs to the system considerably, but more importantly it is essential for a more natural dialogue, as these are all devices that give cohesion to the discourse. Current systems have not yet proceeded very far into this direction: Systems in the TREC 2004 evaluation have mostly simply replaced pronouns by the subject under discussion (Voorhees (2005), but cf. also

Harabagiu et al. (2005)). In addition, a system will appear more natural in its interaction if it makes use of anaphora in its answers.

We aim to show in this paper that a coreference resolution module can be used to handle a number of interesting cases of anaphora resolution that occur in interactive QA systems. As many QA systems employ such a module to resolve coreference in candidate documents (the importance of coreference resolution for QA has been demonstrated, e. g. by Watson et al. (2003)), this extension will often be relatively straightforward.

We will not be concerned here with more advanced questions that arise in developing a full-scale dialogue system for QA. Research in cooperative human-machine dialogue has shown that 'real' dialogue has a large number of specific problems related to it, such as planning and plan recognition (Ferguson et al. (1996)) or spoken language phenomena in coreference (Eckert and Strube (2000)). We are not focusing here on the cognitive processes that underly such a free dialogue, either, but rather aim to show how by simple means improvements in naturalness and in ease-of-use may be achieved.

In the rest of this paper, we describe the implementation of several types of anaphora and ellipsis in a prototype QA system for German that makes use of comparatively deep linguistic methods. We first give a short overview of our system. Then we describe the types of anaphora and ellipsis that we have integrated into the system and how the integration was done. We then turn to an extended walk-through example to give an impression of the way in which interaction with the QA changes.

2. Our QA System

We are currently implementing and investigating a QA system for German that makes use of linguistic resources (Flidner, 2004a).

Our approach is based on the idea of obtaining a full linguistic parse in an offline 'indexing' phase for all documents in the text collection that is later to be searched. While this would not have been possible even a couple of years ago, the availability of fast parsers combined with dropping hardware costs have made it a real possibility: Setting up compute clusters with several dozen nodes no

Work reported in this paper was partly carried out by the author at DFKI Saarbrücken within the Collate project (*Computational Linguistics and Language Technology for Real Life Applications*), conducted by DFKI GmbH and Saarland University, funded by the German Ministry for Education and Research, Grant numbers 01 IN A01 B and 01 IN C02.

longer poses an unsurmountable challenge. Thus, this approach is feasible for large text collections, such as newspaper archives, at least. See also Bouma (2003).

Both the original text collection and the question put to the system by the user are parsed by the same parser. For parsing, we use a tool-chain of partial parsers, similar to the ideas described in Abney (1996). All tools in the chain exchange data by XML. As for German far fewer natural language processing tools are available than for English, most of the tools in the chain had to be built from scratch or adapted from existing tools. In the following, we list the most important steps used in processing.

Morphological analysis. Texts are first analysed morphologically using the Gertwol two level morphology developed by Lingsoft, Oy, Helsinki. Its base lexicon consists of some 350 000 stems; the morphology supports the full inflectional, derivational and compounding morphology of German. Compounding plays an especially important role in German, as compounds are generally written in one word and as ad-hoc compounding is extremely frequent.

Topological analysis. German sentences show a comparatively rigid overall structure. German sentence topology assumes a number of different sentence fields with rules for the positioning of constituents, especially inflected and non-inflected verb forms. These are captured by a rule-based topological parser that identifies the overall sentence structure with very good accuracy (Braun, 2003).

Named Entity Recognition. Named entities (NE) of several types (most importantly, personal names, company names and date expressions) are recognised by a combination of gazetteers and regular expressions.

NP/PP Chunking. NPs and PPs are recognised by a chunker based on extended regular expressions that recognises even complex German NPs and PPs with embedding and coordination (Flidner, 2002). Results from the NE recogniser are used directly by the chunker so that complex (e. g. modified or conjoined) NEs can be analysed as well.

Deriving a Dependency Representation. Results from all previous steps are integrated into a single dependency-style representation called Preds (*Partially resolved dependency structure*). This representation involves certain abstractions (e. g. over active and passive) and introduces grammatical functions such as deep subject or modifier as dependency relations between the words.

GermaNet Annotation. All words in the parse are mapped onto GermaNet (the German version of WordNet, Kunze and Lemnitzer (2002)). Currently, no word sense disambiguation is performed.

Coreference Resolution. This module resolves a number of coreferences between anaphora and antecedents, both intrasentential and intersentential. As it forms an important part of the work described in this paper, we will describe it in more detail below.

FrameNet Annotation. Based on these comparatively rich representation, a German FrameNet (cf. Baker et al. (1998), Erk et al. (2003)) level is then added. FrameNet provides a grouping of words denoting the same abstract type of event into one frame (e. g. BUYING) with fixed

names for the different ‘participant roles’ (e. g. BUYER), thus abstracting away from surface level variations such as using words from different parts of speech or certain idiomatic expressions.

For storing and efficiently retrieving these structures we have developed a database model that allows storing the dependency trees together with the additional lexical semantic information.¹ For retrieving linguistic structures from this database, we have defined a similarity measure that allows introducing a notion of similarity both between nodes and also between basic trees (of depth one). We mainly derive these similarity measures from the GermaNet and FrameNet hierarchies, i. e. based on hyponymy within a lexical semantic ontology. This notion of local tree similarity that also allows basic tree transformations can be directly integrated into the database search, making the efficient retrieval of similar (instead of identical) linguistic trees possible. It allows, for example, finding that ‘*The sale of Mannesmann to Vodafone...*’ contains an answer to the question ‘*Who bought Mannesmann?*’, as the former event is classed into the FrameNet frame SELLING and the latter into BUYING which are related in the FrameNet hierarchy (together with the proper mapping of the relevant roles). This is described in more detail in Flidner (2005).

We will now turn to the issue of adding interaction to the system described here. As full parses are available both for the users’ questions and for the potential answers, this system is especially suited for carrying out experiments with interactive QA. In fact, it is one of the goals of our work to investigate the usefulness of linguistic resources for interactive QA.

3. Adding Interaction to our QA System

As mentioned above, our system has a module that handles different kinds of nominal coreference, namely pronouns, coreferent definite NPs and Named Entities. We will now describe this module in some more detail, as it is used as a basis for anaphora resolution in the user interaction.

3.1. Basic Anaphora Resolution Algorithm

Coreference resolution is based on the algorithm described in Lappin and Leass (1994), using some of the additional features introduced by Kennedy and Boguraev (1996). This choice is motivated as follows (cf. Mitkov (2002)): Lappin and Leass’s algorithm requires syntactic parses of the text but then achieves high accuracy rates. As our system is based on obtaining full dependency parses, all information that is needed by the algorithm is present, obviating the employment of knowledge-poorer approaches. Moreover, it does not require the use of annotated corpora, as machine learning approaches would.

The algorithm works as follows: During the processing of the text, all NPs (including pronominal NPs and named entities) are put on a possible-antecedent stack. The candidates receive a salience score that is computed by checking several salience features and adding a constant weight

¹Actually, the ensuing structures may form directed acyclic graphs, as links between anaphora and their antecedents may destroy the ‘treeness’.

for every present feature. The used features include checks for same-sentence, for grammatical functions (where subject antecedents score highest), for non-embedded NPs and for syntactic parallelism between antecedent and anaphoric NP.

Whenever a possible anaphor is encountered (we consider all pronominal NPs², NEs and most definite NPs as possible anaphora, see also below), the current stack is searched for a suitable antecedent: First, morpho-syntactically incompatible antecedents are filtered (such as noun phrases that differ in number or gender from the pronoun under consideration). Then, the remaining candidate with the highest salience score is selected as the most likely antecedent. Whenever antecedent chains are found (i. e. one discourse entity is anaphorically referred to more than once), they form an equivalence class, and all elements of the chain receive an increase in salience.

While processing a text, all candidates remain on the stack and thus accessible. However, whenever a new sentence starts, the salience scores of all candidates on the stack are decreased by 50%. This ensures that more recent candidates are more likely to be chosen as antecedents.

We have extended this basic algorithm to handle two additional classes of anaphora: NEs and definite NPs. We will describe the necessary changes in turn.

3.2. Coreference of Named Entities

The first extension concerns handling coreference between named entities. Named entities are added to the possible-antecedent stack and can thus become antecedents for pronoun anaphora and definite descriptions (see below).

Named entities can also be coreferent with other named entities. Quite often, short forms will be used in these cases. For personal names, for example, only the last name may be used in anaphoric use (*'US president George W. Bush arrived at New Delhi today. Bush said...'*).

To handle these cases properly, two additional mechanisms need to be integrated. First, the NE recogniser mentioned above is equipped with a simple learning module: Once an NE is recognised, a number of possible short forms are derived and 'learned' as possible NEs. In the example above, after recognising the proper name *George W. Bush*, the word *Bush* would receive an additional NE reading within the rest of the text. Thus, when encountering *Bush* in the second sentence the NE recogniser would provide an NE reading where otherwise only the 'shrub' reading would have been available. A similar heuristic makes certain substrings of recognised company names available as NEs (thus *Lockheed Corp.* may be shortened to *Lockheed*). By restricting these learned NE forms to use within the same document, we minimise possible precision problems.

The coreference resolution algorithm then needs to establish the correct coreference links. To handle this, NEs are considered as candidates for anaphora. When one is encountered, the current possible-antecedent stack is searched for a suitable antecedent. First, a filter is applied to all candidates: Only other NEs are considered as antecedents of

NEs. A candidate antecedent must be compatible with the NE under consideration. As an example, we list the most important heuristics for personal names: All 'parts' that are present must match. If, for example, both NEs have a last name part and a first name part, then they must match (blocking, e. g. *George W. Bush* as a possible antecedent for *Laura Bush*). If the first name part is missing in either or both, they are considered a possible match (allowing *George W. Bush* as a possible antecedent for *Bush*).

After this filtering, the same salience criteria as for pronoun anaphora are used to find the optimal candidate.

3.3. Definite Descriptions

To handle definite NPs, we have added a comparatively simple semantic compatibility check based on GermaNet. This works as follows: Whenever a definite NP is encountered, it is considered as a possible anaphoric definite description. However, not all definite NPs actually are anaphora. They may also introduce a so-called discourse-new entity (with different subtypes). The problem of identifying discourse-new definites is known to be a difficult one (cf. e. g. Poesio et al. (2004)). After some experimentation, we finally settled for just two indicators for identifying discourse-new definites, namely modification of the NP by either an apposition or a genitive modifier, as all other suggested heuristics proved either too restrictive or too difficult to implement.

For all definite NPs that were not filtered out by the discourse-new heuristics, the most likely antecedent is then searched on the possible-antecedent stack. A filter based on semantic compatibility is used. For every pair of definite NP and antecedent candidate (only non-pronominal NPs are allowed as antecedent candidates), this filter checks whether a close hyponymy or hyperonymy relationship between the two can be found using GermaNet. As mentioned above, currently no word-sense disambiguation is done. That means that for ambiguous words all available readings (synsets) in GermaNet are compared. Checking for hyponymy and hyperonymy ensures that both generalisations and specialisations can be correctly identified (cf. Vieira and Poesio (2000)).³

The closeness between hyponyms is currently established using relative path-length in GermaNet as a heuristics: The path-length between the two synsets must not be more than half of the path length between the more specialised concept (the hyponym). We established this heuristic by experimentation. In principle, it should be replaced by a corpus-based similarity measure (cf. Patwardhan et al. (2003)). However, such measures are currently not available for GermaNet in a pre-computed form.

It should be noted that a number of bridging anaphora cannot be handled by the algorithm as it stands, namely those where a different type of semantic relationship between anaphor and antecedent exists, such as part-of (meronymy/holonymy) or similar. Thus, cases like the following currently remain unresolved: *'Great Britain will buy planes from Lockheed. A government spokesperson*

²Pleonastic *it* (as in *it is known*, German: *'es'*) is recognised as non-referential and filtered.

³We noted that using the morphology during parsing greatly improves recall for the semantic compatibility check as it allows to identify at least the head of compound words.

said...’ Here, one would like to establish that the *British* government is referred to.

To allow NEs as antecedents for definite descriptions, all recognised NEs receive a semantic type that directly maps into GermaNet. Company NEs, for example, receive synsets corresponding to *firm* and *factory* as semantic type. Thus, coreference can be correctly established (for example in the typical beginning of a news story: *Lockheed Corp. (Burbank, CA): The plane manufacturer...*).

Note that through the establishment of equivalence classes for coreference chains cases of pronominal antecedents for NEs and definite NPs can also properly be handled. As an example, let us consider the following text snippet: ‘*George W. Bush arrived in New Delhi today. He said... The president will meet...*’ Here, coreference between *he* and *George W. Bush* is established first. Then, this chain is found as the most likely antecedent group for *the president*, as this is semantically compatible with the type *person* for the personal name NE.

3.4. User Interaction

The same coreference resolution module is used to resolve coreference both in the system’s document collection and in the users’ questions to the system. We simply consider the interaction with the user a coherent text that user and system cooperatively produce.

This is, of course, a simplifying assumption that does not take a number of issues in human-machine dialogue into account (see above). We think that it is still a useful view as it allows us to extend our QA system towards a more natural interaction by comparatively simple methods.

Based on this assumption, we apply the coreference resolution algorithm described above to both the users’ inputs and the answers that the system produces. We can then resolve anaphora in the input by replacing them with their antecedent, as returned by the coreference resolution module. We use the modified query to search the whole document collection, not only the document in which the last answer was found, as the requested information may in fact reside in another document.

We have described above that the document collection in which the system searches for answers is completely parsed and that a parsed representation is actually searched in a database. This parsed representation is not only used to generate a suitable answer (see below) but also allows us to put possible antecedents from the system’s answers onto the coreference stack, providing full linguistic information such as gender etc. Thus, the user can anaphorically refer not only to earlier questions but also to the system’s answers, again adding to the natural feel of the dialogue.

Ellipses are notoriously difficult to handle in natural language processing systems. We have therefore currently restricted possible ellipses in the user’s questions to two cases: First, if a follow-up question consists only of one phrase containing a question word, this phrase is used to replace the question phrase of the last question. Second, if the ‘question’ consists of a single NP/PP, then this NP/PP is used to replace the first one in the last question with which it is semantically compatible (again using the semantic compatibility check described above).

This is achieved in the following way: A copy of the parsed representation of the last question is kept. When the user enters an incomplete question that fits one of the two patterns described above (viz. the question consists of a single phrase – NP, PP or adverbial *wh*-word –, either containing a *wh*-word or not), the parsed representation derived from this phrase replaces the representation of either the phrase containing the question word or the suitable NP/PP. Thus, a complete question representation is obtained for these elliptical cases. Our approach is thus similar to that described in Kehler (1993). Examples for both cases are shown below.

Another requirement for a natural interactive QA system is that the system’s answers are actually generated from a linguistic representation. Only this ensures that the answer fits the question in a natural way.

We have taken this idea a bit further and have implemented a parameterisable language generation module. It is possible to change the required ‘verbosity’ of the answer. Examples are shown below. We think that providing these different levels of verbosity is an important improvement for a QA system. In an experiment, Lin *et al.* found that users seem to prefer ‘answers in context’ over bare NP answers (Lin *et al.*, 2003). As a further means of justification, our system can also output the sentence or sentences from the original document on which the answer is based.

These different options currently have to be selected manually. The user may switch them by explicit commands at any time during the session. In a more advanced system, these settings might be part of a user model that could be selected either as a prototype model (e. g. ‘novice user’) or even automatically.

In ‘neutral’ and ‘brief’ verbosity mode, the system generally produces constituent answers. The constituent used as an answer is the one that corresponds with the *wh*-phrase of a *wh*-question (mostly NPs or PPs, adverbs or whole subordinate clauses are additional possibilities). Note that when question and answer sentence are matched, the similarity match described above that allows local changes in the linguistic representation is used. In ‘neutral’ verbosity mode, some additional material, especially appositions, may also be output, while in ‘brief’ mode, only a core answer is generated.

Having a special integrated generation module also allows us to make the system’s answer even more natural by generating pronouns for entities that either the user or the system has already referred to: The system keeps a list of already mentioned entities. A request to generate an NP with the same referent then produces the personal pronoun. Here again, both the user’s questions and the system’s answers are considered.

4. One example

It should be noted that while the example given here and similar examples can be handled by the current system, it is far from a broad, general coverage. We do not claim that the system itself is a useful interactive QA system as it stands, but rather that it shows what we believe is minimally required in such a system.

The following sentences are to be used as the basis of our

example ‘dialogue’. They are taken from an actual newspaper article.⁴

One of the most-wanted Mafia bosses, super-killer Lorenzo Tinnirello has been arrested near Palermo on Saturday. The 34-year-old has 119 murders on his conscience... The contract killer was arrested by five *carabinieri* in plain clothes... Together with him, another member of the Mafia, 54-year-old Giovanni D’Agati was arrested.

(*Frankfurter Rundschau*, 4 September 1994)

Now, let us turn to the example query session.

Q1: Whom have the carabinieri arrested?

A1: One of the most-wanted Mafia bosses, super-killer Lorenzo Tinnirello

To arrive at this answer, the system had to correctly resolve the coreference between *contract killer* in the original sentence and its antecedent. As described above, in ‘neutral verbosity mode’, the personal name and the appositions of the first occurrence in the text are output. While this may be considered as ‘over-answering’ the question, in general such additional information may be considered helpful.

A1a: Lorenzo Tinnirello

When switched back to verbosity mode ‘brief’, only the personal name is generated. This minimal answer contains only a core answer.

A1b: They have arrested one of the most-wanted Mafia bosses, super-killer Lorenzo Tinnirello.

In ‘clausal mode’, the whole sentence answering the question is output. Note that the reference to the *carabinieri* has been replaced by the correct personal pronoun. It should also be noted that though the original text had the sentence in passive voice, the underlying representation is neutral in that respect. The system generates the active voice equivalent, as the question was in active voice.

Q2: When was he arrested?

A2: On Saturday

Here, the pronoun is resolved to an antecedent (namely *Lorenzo Tinnirello*) that was only mentioned in a system answer, leading to a more natural dialogue flow.

Of course, the actual answer is far from perfect: One would prefer to get a ‘normalised’ answer, such as ‘*On Saturday*,

3 September 1994’. As normalising and resolving time expressions in general is far from easy (cf. the research done in the context of TimeML and automatic annotation such as Verhagen et al. (2005)), we have currently not implemented it.

Q3: Where?

A3: Near Palermo

This example shows how an elliptical question is handled: The question phrase replaces that of the last question. Thus, this question is ‘expanded’ to the representation of ‘*Where was Lorenzo Tinnirello arrested?*’.

Q4: Who was arrested together with Tinnirello?

A4: Giovanni D’Agati

‘Short forms’ (here, the last name) of personal names can be used to refer back to persons that have already been mentioned. This possibility is provided by the interaction of the ‘learner’ module of the NE recogniser and the coreference resolution for NEs, both again applied to questions and answers.

Q5: How old is he?

A5: 54 years

Here, *Giovanni D’Agati* is chosen as the most likely antecedent. This simple case shows that it might be useful to extend the system so that it can ask clarification questions in cases where two candidates have a very similar salience. However, it must be carefully considered when clarification is really necessary, as constant clarification would interrupt the dialogue flow. In many cases it is far easier to let the user ask a follow-up question such as the following.

Q6: And Tinnirello?

A6: 34 years

In this case, the single NP is treated as an elliptical question and replaces the only semantically similar phrase in the preceding question (namely *he*, that is *Giovanni D’Agati*).

5. Conclusion

We have shown in this paper a number of ways in which a QA system may be extended to handle a dialogue-style user interaction. We believe that this is only possible when the system actually does a linguistic processing of both questions and answers.

Our experiments with such dialogue-style interaction have only begun. So far, we have only got couple of positive comments from informal tests with two users but we have not yet conducted a systematic evaluation. For a useful evaluation of the approach, that is, of course, indispensable.

However, the evaluation of such a system will be even more challenging than for current systems. While in current evaluations such as TREC, a list of questions can be

⁴For convenience’s sake, we have translated the whole example from German to English. Our interest in this article was prompted by the fact that it contains an answer to a question in the QA@CLEF 2003 competition (Magnini et al., 2003), namely Q47: ‘*What is the nickname of the Mafia boss Lorenzo Tinnirello?*’

evaluated independently from each other, taking the interaction into account crucially changes this setting, as shown in our rather simple example dialogue. This means that an evaluation would need to centre more on individual users (cf. Fliedner (2004b)). This might, however, mean to give up the ‘shared task’ style of evaluation with its very important advantage of direct comparability of systems. Here, additional research will be needed to establish suitable new evaluation schemata.

6. Acknowledgements

We would like to thank our former colleague Christian Braun who has contributed largely to the design and implementation of the parser used in our QA system. We would further like to thank the anonymous reviewers for their valuable comments.

7. References

- Steven Abney. 1996. Partial parsing via finite-state cascades. In *ESSLLI Workshop on Robust Parsing*, pages 8–15.
- Colin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *COLING 98*.
- Gosse Bouma. 2003. Question answering for dutch using dependency relations. Project description, http://odur.let.rug.nl/~gosse/Imix/project_description.pdf; 2006/03/06.
- Christian Braun. 2003. Parsing German text for syntactosemantic structures. In *Workshop “Prospects and Advances in the Syntax/Semantics Interface”*, pages 99–102.
- John Burger, Claire Cardie, Vinay Chaudhri, Robert Gaizauskas, Sanda Harabagiu, David Israel, Christian Jacquemin, Chin-Yew Lin, Steve Maiorano, George Miller, Dan Moldovan, Bill Ogden, John Prager, Ellen Riloff, Amit Singhal, Rohini Shrihari, Tomek Strzalkowski, Ellen Voorhees, and Ralph Weischedel. 2001. Issues, tasks and program structures to roadmap research in question & answering (Q&A). Document Understanding Conferences Roadmapping Documents.
- Miriam Eckert and Michael Strube. 2000. Dialogue acts, synchronising units and anaphora resolution. *Journal of Semantics*, 17(1):51–89.
- Katrin Erk, Andrea Kowalski, and Manfred Pinkal. 2003. A corpus resource for lexical semantics. In *IWCS 2003*.
- George Ferguson, James Allen, and Brad Miller. 1996. TRAINS-95: Towards a mixed-initiative planning assistant. In *AIPS-96*, pages 70–77.
- Gerhard Fliedner. 2002. A system for checking NP agreement in German texts. In *ACL Student Workshop*.
- Gerhard Fliedner. 2004a. Deriving FrameNet representations: Towards meaning-oriented question answering. In *NLDB 2004*, volume 3136 of *LNCS*, pages 64–75.
- Gerhard Fliedner. 2004b. Issues in evaluating a question answering system. In *LREC Workshop “User-Oriented Evaluation”*, pages 8–12.
- Gerhard Fliedner. 2005. A generalised similarity measure for question answering. In *NLDB 2005*, volume 3513 of *LNCS*, pages 380–383.
- Sanda Harabagiu, Andrew Hickl, John Lehmann, and Dan Moldovan. 2005. Experiments with interactive question-answering. In *ACL’05*, pages 205–214.
- Andrew Kehler. 1993. A discourse copying algorithm for ellipsis and anaphora resolution. In *EACL’93*.
- Christopher Kennedy and Branimir Boguraev. 1996. Anaphora for everyone: Pronominal anaphora resolution without a parser. In *Coling’96*, pages 113–118.
- Claudia Kunze and Lothar Lemnitzer. 2002. Germanet – representation, visualization, application. In *LREC 2002*, pages 1485–1491.
- Shalom Lappin and Herbert J. Leass. 1994. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):535–561.
- Jimmy Lin, Dennis Quan, Vineet Sinha, Karun Bakshi, David Huynh, Boris Katz, and David R. Karger. 2003. What makes a good answer? The role of context in question answering. In *INTERACT 2003*.
- Bernardo Magnini, Simone Romagnoli, Alessandro Vallin, Jesús Herrera, Anselmo Peñas, Víctor Peinado, Felisa Verdejo, and Maarten de Rijke. 2003. The multiple language question answering track at CLEF 2003. In *CLEF 2003*.
- Ruslan Mitkov. 2002. *Anaphora Resolution*. Pearson.
- Dan Moldovan, Sanda Harabagiu, Roxana Girju, Paul Morescu, Finley Lacatusu, Adrian Novischi, Adriana Badulescu, and Orest Bolohan. 2003. LCC tools for question answering. In *TREC 2002*.
- Siddharth Patwardhan, Satanjeev Banerjee, and Ted Pedersen. 2003. Using measures of semantic relatedness for word sense disambiguation. In *CICLing 2003*, volume 2588 of *LNCS*.
- Massimo Poesio, Olga Uryupina, Renata Vieira, Mi-jail Alexandrov-Kabadjov, and Rodrigo Goulart. 2004. Discourse-new detectors for definite description resolution: A survey and a preliminary proposal. In *ACL Workshop on Reference Resolution*.
- Marc Verhagen, Inderjeet Mani, Roser Sauri, Jessica Littman, Robert Knippen, Seok B. Jang, Anna Rumshisky, John Phillips, and James Pustejovsky. 2005. Automating temporal annotation with TARSQI. In *ACL’05 Demo and Poster Session*, pages 81–84.
- Renata Vieira and Massimo Poesio. 2000. An empirically based system for processing definite descriptions. *Computational Linguistics*, 26(4):539–593.
- Ellen M. Voorhees. 2005. Overview of the TREC 2004 question answering track. In *TREC 2004*.
- Rebecca Watson, Judita Preiss, and Ted Briscoe. 2003. The contribution of domain-independent robust pronominal anaphora resolution to open-domain question-answering. In *Symposium on Reference Resolution and its Applications to Question Answering and Summarization*, pages 75–82.