A Rule-based Reference Resolution Method for Dutch Discourse Analysis

Rieks op den Akker , Marjan Hospers,
Erna Kroezen and Anton Nijholt
University of Twente
Department of Computer Science
Enschede, the Netherlands
(infrieks|kroesper|anijholt)@cs.utwente.nl

Danny Lie
Carp Technologies
Hengelosestraat 705
Enschede, The Netherlands
lie@carp-technologies.nl

Abstract

This paper presents a knowledge-poor method for the solution of anaphoric and deictic expressions in Dutch texts. The method is developed for use in a text summarization system. Anaphora resolution plays an important role in the analysis of the original text as well as in the generation of the text summary.

Keywords: Anaphora resolution, Automatic Text Summarization, Dutch.

1 Introduction

An automatic text summarization system should provide a coherent summary of the contents of the original text. Although complete understanding of the text is often not necessary to come up with a good summary, it is rather important in order for selecting the important and coherent parts (sentences) that anaphorically and deictically used words or phrases are recognized and that the proper antecedents of anaphora are identified. In the generation of the summary some anaphorically used words need to be substituted by their referents especially if that part of the text that contains the referent is not included in the summary. In particular references to objects like figures, tables and footnotes have to be identified and resolved properly.

In this paper we present and discuss a method for resolving anaphorically used words and phrases for a commercial Dutch automatic text summarizer. The method is knowledge poor: no grammatical analysis is used; no conceptual knowledge base either. The choice for such a method is made because the system should be fast and easy to use and maintain, and in this way it can be ported easier to other languages. Further, we want to experiment with different algorithms, and we are going to integrate the algorithm with the grammatical and semantical analysis of the text summarizer. The current anaphora resolution module is for interactive use: in case the system is not sure it will ask the user for help.

The paper is organized as follows. In the following section we explain basic terminology and we present what Dutch words can be used anaphorically and deictically and what properties they have with respect to their possible referents. We discuss some peculiarities of Dutch compared with German and English. In section 3 we give a global description of the Dutch text summarizer to provide the context in which our method is used. Section 4 presents some well-known algorithms for anaphora resolution that motivated our own algorithm presented in section 5. Experimental results with some small corpora of different types are presented and discussed in section 6. Finally we come to conclusions and provide some insight in plans for improvements of the algorithm.

2 Anaphoric and deictic used words in Dutch

Anaphoric used words are words that are referring back to something that was earlier mentioned or that is known because of the discourse situation and/or the text as it is read or heard. The anaphorically used word is called ’the anaphor’, the text to which it refers ’the antecedent’. The extralingual entity they corefer to is called the referent. Deictic used words are words that refer to something directly or indirectly present in the situation. The word is then used instead of a gesture, or utterance of the word is accompanied by gesturing. Cataphorically reference is reference to something that follows in the text or that will be specified later by the text.

Anaphora resolution is the process of determining the antecedent of an anaphor. The antecedent can be in the same sentence as the anaphor, or in another sentence. The first case is called intra-sentential referencing, the second case inter-sentential referencing.

There are different types of words that can be used anaphorically. Examples of these types for Dutch are:

- Personal pronouns, which refer to different kinds of persons. Examples are: hij (he), zij (she), haar (her), hem (him).
- Possessive pronouns: zijn (his), haar (her), hun (their), et cetera.
- Reflexive pronouns: zich, zichzelf (himself, herself, themselves et cetera).
- Reciprocal pronouns: elkaar (each other, one another).
- Demonstrative pronouns: die, dat (that, those), deze, dit (this, these), het (it), et cetera.
- Relative pronouns: die, dat (that), wie (who), wat (what), et cetera.
- Numbers and words as *de eerste (the first)* or *de laatste (the last)* can be used anaphorically.
- Words as *het (it)* and *dat (that)* can refer to whole sentences or parts of sentences, or even to whole chapters.
- Noun phrases can be used to refer to something that is mentioned earlier.

Most of these words can have an antecedent in the text, but it is not necessary. For example, sometimes ”het” (”it”) doesn’t refer to something: *het regent (it is raining)*. Reflexive and reciprocal pronouns must have an antecedent: they can only be used anaphorically, and the antecedents have to be in the same sentence.

Examples of words that can be used deictically in Dutch are: ik (I), jij (you), wij (we), mij (me), ons (us), dat (that), et cetera. Noun phrases can also be used deictically: die man (that man).

Neuter nouns are nouns that are preceded by ”het” (or can be preceded by ”het”; e.g. *het meisje (the girl)*). Male nouns and female nouns are nouns that are preceded by ”de” (or can be preceded by ”de”; e.g. *de jongen (the boy)*). ”Deze” and ”die” don’t refer to neuter nouns. ”Dit”, ”dat”, ”het” and ”’t” don’t refer to male or female words. ”Deze”, ”dit”, ”dat”, ”het” and ”’t” don’t refer to proper nouns.

Anaphora resolution methods are dependent on the language. Different languages can have grammatical differences and different word orders. Some languages make use of cases (in German we have more cases that give hints/clues for resolving the anaphorical referent than in Dutch or English). In English, we have some different rules for use of plural and singular nouns as in Dutch (five dollars can be used also in case the intended referent is the unity of five dollars; in Dutch we use plural noun *vijf guldens/euros* if we intend the plurality and the singular form *vijf gulden/euro* if we intend to refer to the amount of money as a unity). And in Dutch, we make difference between ”deze”/ ”die” (which can’t refer to neuter words) and ”dit”/”dat” (which can’t refer to male or female words). In English, this difference is not made.

3 Anaphora Resolution for Automatic Text Summarization purposes

An automatic text summarization system based on semantic analysis is implemented in Lie (1998a). A detailed description of this system goes beyond the scope of this article, so for more information is referred to Lie (1998b). Instead, we will provide enough information to understand the application for anaphora resolution within this system.

First, the original text will be grammatically parsed and semantically analysed. This results in a semantic structure, of which an example is shown in Figure 1.

Next, a large number of heuristics and rules (more

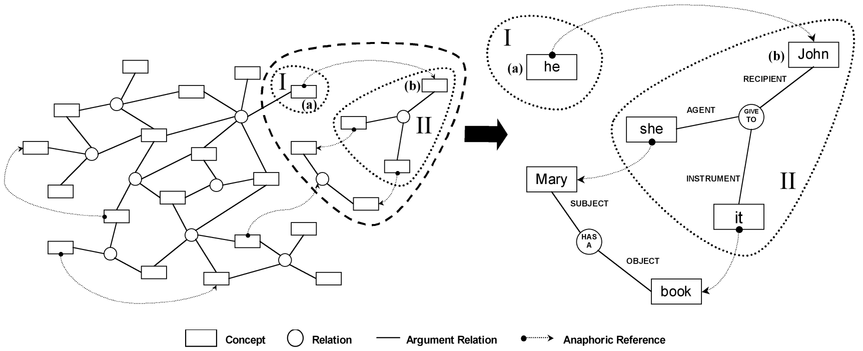

Figure 1: An example of a semantic structure

than one hundred thousand [1]) determines how this semantic structure will be pruned (see Figure 2). Finally, the resulting structure will be transformed back into a human readable text using some form of text generation.

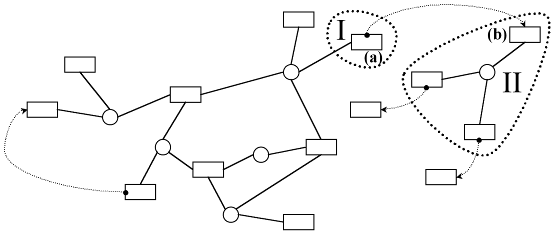

Figure 2: The pruned semantic structure

The rules determine the importance of every part of the semantic structure, by analysing the structure itself. One of the many aspects the rules may observe are anaphora and their antecedents (shown as dotted arrows in Figure 1 and 2). For example, one of the rules may decide that node **(a)** in part **I** is very important (see Figure 1). Subsequently, another rule decides that part **II** is also important, because node **(a)** is an anaphor that refers to node **(b)**, which is part of part **II**.

The last part of the system we will describe shortly is the text generation component. Using templates, and heuristics (amongst others), every part of the pruned semantic structure will be transformed into human readable sentences (see Figure 3). During this transformation, anaphors are automatically replaced by their antecedents if they have not already been mentioned before.

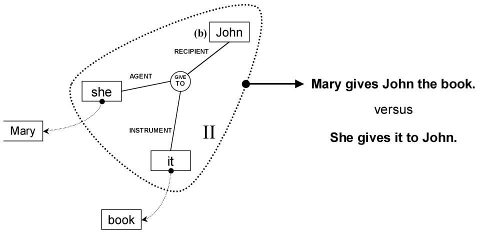

Figure 3: Text generation supported by anaphora resolution

We have shown that the rules partially depend on the existence of resolved anaphora. However, our observations indicate that because of the large number of rules, in most cases the absence of resolved anaphora will be dealt with correctly by other rules. In other words, the rules will be able to correctly determine the importance of a part of the semantic structure, even when resolved anaphora are absent.

There is however another component of the text summarization system that depends more on correctly resolved anaphora. The text generation part of the system replaces anaphora with their antecedents in case they have not already been mentioned in the text generated so far. In order to produce legible text, it is extremely important to minimize the number of errors (i.e. wrong antecedents). Tests indicate that for legibility, it is better to omit the antecedent entirely than produce a wrong one. These observations lead to the fact that we strongly focus on precision rather than recall. We would certainly prefer an anaphora resolving algorithm that only resolves about 10% of the anaphora without error, above an algo-

[1] The heuristics are hand-crafted, whereas the rules are automatically generated by a transformation based error-driven machine-learning algorithm (Brill, 1995). In the remainder of this article, we will refer to both the heuristics and the rules by using the term ”rules”.

rithm that resolves more than 80% with an error rate of 10%.

4 Other algorithms for anaphora resolution

Several articles about existing algorithms are considered. The three most important algorithms where our algorithm is based on are the algorithm for robust pronoun resolution with limited knowledge of Mitkov (1998), the RAP algorithm of Lappin and Leass (1994) and the algorithm of Kennedy and Boguraev (1996).

Mitkov's algorithm is intended for resolving pronouns in technical manuals. It only makes use of a part-of-speech tagger and simple noun phrase rules. Noun phrases that precede the anaphor within a distance of two sentences are identified. They are checked for gender and number agreement with the anaphor. Then the algorithm applies fourteen antecedent indicators to the remaining candidates. Some of the indicators are genre-specific. The candidates get a score for each indicator. The noun phrase with the highest aggregate score is proposed as antecedent. In case of a tie, priority is given to the candidate with the higher score for particular indicators.

Lappin and Leass describe RAP (Resolution of Anaphora Procedure), an algorithm for identifying noun phrase antecedents of third person pronouns and reflexives and reciprocals. RAP uses the syntactical information of McCord's Slot Grammar parser and a simple dynamic model of attentional state. Salience measures are derived from syntactic structure for all possible antecedents. The algorithm for example looks at grammatical role of the candidate antecedent, parallelism in grammatical role between the candidate antecedent and the anaphor, the frequency of appearance and the recentness of appearance of the candidate antecedent. Semantic conditions and real-world knowledge are not used.

The algorithm of Kennedy and Boguraev is an adaptated and extended version of RAP. It uses the output of a part-of-speech tagger enriched with annotations of grammatical function of lexical items in the input text stream. The basic logic of this algorithm parallels that of RAP. A significant point of divergence between the two algorithms is the determination of disjoint reference. RAP relies on syntactical configuration information, but in Kennedy and Boguraev's algorithm this information is absent. It relies on inferences from grammatical function and precedence to determine the disjoint reference.

5 Our algorithm

Our algorithm uses the output of a part-of-speech tagger as its input. The part-of-speech tags are used to fill in the features *number* (singular or plural) and *part of speech*. Further, a lexicon with names is used to identify words with certain tags as proper nouns. If a word is a proper noun, the feature *gender* is filled in.

Anaphors which are resolved by the algorithm are personal and possessive pronouns, reflexive and reciprocal pronouns and relative and demonstrative pronouns. Possible antecedents are nouns, proper nouns and anaphors (if two anaphors refer to the same thing or person, then ”the second anaphor refers to the first anaphor” is a good solution).

When resolving an anaphor, an antecedent is chosen from a list with potential antecedents, if that is possible. Before a candidate is chosen, all candidates which do not agree with the antecedent in number, gender or person are removed from the list.

All possible antecedents get a score, the salience. The salience of an antecedent is aggregated from different scores for a set of properties of the antecedent. Some properties always raise or lower the chance that a word is the antecedent of an anaphor. Other properties only raise or lower the chance that a word is the antecedent of an anaphor when the anaphor is of a special type. The salience is used to choose the right antecedent: the antecedent with the highest salience is chosen. If two or more candidates have the same salience, the most recent of them is chosen.

In some cases there is no antecedent to choose, or there is a great chance that the chosen antecedent is a wrong one (e.g. if the anaphor is ”ik” (”I”), ”jij” (”you”), ”wij” (”we”), ”jullie” (”you”)). Then comments are given as output (together with the eventually chosen antecedent), from which it becomes clear that there is a great chance that the chosen antecedent is wrong or why no antecedent can be chosen. When this is the case, the user can decide to make the connection between the anaphor and the chosen antecedent, to make no connection between the anaphor and an antecedent, or to look in the text for the right antecedent and make the right connection him/herself. This choice is left to the user, because the best choice depends on where the algorithm is used for.

5.1 Properties that always raise or lower the chance that a word is the antecedent

1. Definite descriptions, such as ”het meisje” (”the girl”), are more likely candidates than indefinite descriptions, such as ”een meisje” (”a girl”).

2. Noun phrases without a preposition are more likely candidates than noun phrases, which start with a preposition.

3. When a possible antecedent is an anaphor itself, then the salience is somewhat lowered. This is done because an anaphor only may be chosen as the antecedent if there is really no common antecedent or if there is a good chance that a noun or proper noun is the wrong antecedent, while it is likely that the two anaphors refer to the same thing or person. This is often the case in sentences with reflexive and reciprocal pronouns.

4. When a possible antecedent is in focus, its salience is raised. A heuristic is used to determine which words are in focus: the first noun (or proper noun) in a non-imperative sentence is in focus (Mitkov, 1999).

5. The salience of words, which occur more often in the text, is raised.

6. Dependent of how much sentences the candidate occurred before the anaphor, the salience is lowered. Candidates which are further away are less likely to be the antecedent.

7. Once a candidate is chosen as antecedent, the salience of this candidate is raised. There is a good chance that a following anaphor refers to the same antecedent.

8. If the salience of a word becomes lower than a certain value, this word is no longer a candidate.

5.2 Personal and possessive pronouns

The algorithm distinguishes between anaphors of the first/second person and third person pronouns. Third person pronouns nearly always have an antecedent in the text, while first and second person pronouns don’t. First and second person pronouns almost always refer to persons, so all candidates that are not persons or proper nouns are removed of the list with candidates. This is not the case when the anaphor is a third person pronoun.

If once is referred to a word with a third person pronoun, there will not be referred to the same word with a first or second person pronoun. Candidates which are chosen as the antecedent of a third person pronoun are therefore removed when a first or second person pronoun is resolved. This is not the case with singular first person pronouns: the word ”ik” (”I”) often occurs in an utterance of a person who is mentioned earlier in the text.

The salience of proper nouns is raised. When the anaphor is a first or second person pronoun the salience of other anaphors is raised, at the moment that it is sure that an antecedent will be chosen. If there is an other anaphor that agrees with the anaphor for all constraints, then there is a great chance that this anaphor refers to the same person as the anaphor which will be resolved. When the anaphor is a third person pronoun, the salience of nouns, which indicate persons, is raised.

When there are no possible antecedents to choose, comments are given as output. Dependent on the properties of the anaphor, it then refers to the speaker(s), the author(s), the reader(s) or the listener(s) or people in general.

When the language that is used in the text is correct, then it will not happen that there is referred to the same word sometimes with ”hij” (”he”) and sometimes with ”zij” (”she”). Therefore, if the gender of the anaphor is male, then the gender of the chosen antecedent becomes male too. When the gender is female, the gender of the chosen antecedent becomes female too. This can be problematical if a word is wrongly chosen as antecedent. If this is the case, then it can happen that the word is not chosen as antecedent when it has to be. But most of the time, adapting the gender of the antecedents causes the algorithm to work better.

5.3 Reflexive and reciprocal pronouns

The words ”zich”, ”zichzelf” and ”elkaar” require an antecedent in the same sentence. This is not the case for ”zelf”, but most of the time the antecedent of ”zelf” is also in the same sentence, so al candidates that are not in the same sentence are removed. The most recent candidate which agrees with the anaphor for all constraints is chosen as antecedent (thus not the candidate with the highest salience).

5.4 Relative and demonstrative pronouns

Relative and demonstrative pronouns are ”dit”, ”dat”, ”deze”, ”die”, ”het” and ”’t” (”this”, ”that”, ”these”, ”those”, ”it”). The algorithm distinguishes between pronouns preceding a noun and pronouns not preceding a noun.

”Dat”, ”het” and ”’t” are not resolved when they are preceded by a verb. When they are preceded by a verb, they most of the time refer to something that follows later in the text, or they don’t refer at all. The same holds if ”het” or ”’t” is the first word of a sentence.

The antecedents of relative and demonstrative pronouns are noun phrases or, if the anaphor is not preceding a noun, (parts of) sentences. Anaphors are not chosen as antecedents for relative and demonstrative pronouns.

Words where the pronoun cannot refer to (see section 2, anaphoric and deictic used words in Dutch) are removed from the list with candidates. Words which have a high chance to be the antecedent get a higher salience (e.g. if the pronoun is ”dit”, words that are preceded by ”het” get a higher salience).

For anaphors, which don’t precede a noun, the most recent antecedent that agrees with the anaphor for all constraints is chosen, if there is one after all antecedents that don’t agree are removed. Otherwise, if the anaphor is ”dat”, and there is a comma in the same sentence, the part of the sentence before the anaphor is chosen as antecedent. If there is no noun phrase and no comma in the same sentence, the algorithm looks at the structure of the preceding sentence, and a part of that sentence is chosen as antecedent, or the whole sentence. ”Deze” and ”die” don’t have a whole sentence or a part of a sentence as antecedent. If there is no suitable noun phrase, they will not be resolved.

If the anaphor is preceding a noun, the algorithm first looks if the noun and the anaphor agree in number. If this is not the case, then it is probably no anaphor, and thus it is not resolved. If they agree in number, al possible antecedents which do not agree in number are removed.

Indications of time (e.g. ”dit jaar” (”this year”), ”die middag” (”that afternoon”), ”dat moment” (”that moment”)) are not resolved, unless there is a very distinct antecedent, that is a word which is contained by the indication of time or which contains the indication of time.

For anaphors preceding a noun, the most recent antecedent with the highest salience is chosen. If the noun is meant for a person, the salience of proper nouns is raised. When the noun is not meant for a person, the salience of proper nouns is lowered. If the noun has occurred earlier in the text, or a word that contains the noun or is contained by the noun, than the salience of the noun phrase with that word is raised.

6 Experimental results

The algorithm is tested with a corpus consisting of a number of texts from different types (newspaper articles, articles from different types of magazines and journals and fragments from books). In the corpus are 440 personal and possessive pronouns, 241 relative and demonstrative pronouns and 40 reflexive and reciprocal pronouns.

Testing took place two times: once with the part-of-speech tags from the tagger, and once with improved part-of speech tags. The part-of-speech tags in the second test are improved in the sense that wrong tags from the tagger are changed by hand. The set of tags was not changed. The results of the two tests can be found in tables 1 and 2.

From the tables can be seen that the accuracy of the tagger is of great importance for the recall and precision of the method. The solutions of the algorithm are better when the part-of-speech tags are correct.

We don’t compare these results with the results of other algorithms. This would be the same as comparing apples with pears: other algorithms are in almost al cases made for other languages, use other tagsets and are implemented for other utilizations.

Type of pronouns	Recall	Precision
Personal and possessive pronouns	69,3% (305/440)	73,4% (224/305)
Relative and demonstrative pronouns	54,8% (132/241)	36,4% (48/132)
Reflexive and reciprocal pronouns	37,5% (15/40)	80,0% (12/15)

Table 1: Results when part-of-speech tags from tagger are used.

Type of pronouns	Recall	Precision
Personal and possessive pronouns	97,5% (429/440)	80,2% (344/429)
Relative and demonstrative pronouns	72,6% (175/241)	57,8% (101/175)
Reflexive and reciprocal pronouns	85,0% (34/40)	85,3% (29/34)

Table 2: Results when part-of-speech tags are improved.

The algorithm was implemented for utilization in automatic text summarization. As said in section 3, for our method of text summarization, tests indicate that it is better to omit the antecedent entirely than produce a wrong one. Therefore, anaphors are not resolved in a number of cases on purpose, this is especially the case for relative and demonstrative pronouns, because these are the most difficult pronouns to resolve. This is the case because relative and demonstrative pronouns can have more sorts of antecedents, because parts of sentences, whole sentences or even more sentences can be the antecedent of these pronouns.

Sometimes a part of a sentence is chosen, while it had to be the whole sentence, or the other way round, and sometimes the wrong noun phrase is chosen as antecedent for relative and demonstrative pronouns. Grammatical analysis can give better results in some of these cases, or we can choose to raise the number of cases in which pronouns are not resolved.

Mistakes are made with plural personal and possessive pronouns. Often, these pronouns don't refer to an unbroken noun phrase, but for example to two persons who are mentioned earlier in the text. For example: *Claudel wordt - hoewel 24 jaar jonger dan Rodin - zijn minnares. Meer dan 15 jaar werken ze intensief samen. (Claudel becomes - although 24 years younger than Rodin - his lover. They work intensively together for more than 15 years.)* ”Ze” refers to Claudel and Rodin, but the algorithm chooses only Rodin as antecedent.

Another error, which is made often is that, the wrong proper noun is chosen as antecedent, for example a place-name instead of a personal name, because it is nearer to the anaphor. This is a consequence of the choice to use a tagset with a small amount of tags. When different tags are used for different kinds of proper nouns, much less of these errors would be made.

Sometimes, a female person is chosen as antecedent for ”he” or a male person as antecedent for ”she”, when the text is about more persons. This can be the case when a person is once mentioned with first- and surname, and later only with his or her first name. When the first- and surname phrase is chosen as antecedent, only the gender of the surname is adapted to the gender of the anaphor. The first name can later be chosen wrongly as antecedent.

Especially the part of the algorithm for relative and demonstrative pronouns has to be improved. Some of the errors can be solved by using grammatical analysis or discourse structure information. However, we have to weight the pros and cons against each other: with each processing of the text new errors are introduced (in tables 1 and 2 can be seen that the results of part-of-speech tagging considerably influence the results of anaphora resolution). If there are more errors introduced than solved, it is better to choose to raise the number of cases in which pronouns are not resolved.

7 Conclusions and future work

We presented a knowledge-poor rule-based method for identifying anaphorically and deictically used words and phrases in Dutch texts to be used in a text summarizer. From experiments with different kinds of texts (see Tables 1 and 2 in the previous section) we see that the accuracy of the tagger and the tag system used by the tagger is of great importance for the quality - in terms of recall and precision - of the method. It will be clear that some of the erroneous solutions proposed by the system could be prevented by using a conceptual knowledge base or by using grammatical analysis or discourse analysis.

In the future, we want to use more heuristics, and in addition, we want to implement a machine-learning strategy, which learns automatically to choose the right candidate. This machine-learning strategy will deliver heuristics again, which can simply be combined with the heuristics made by hand. The heuristics determine a salience for each candidate on the base of the context, and therefore they all have the form: *f_salience(anaphor, anaphor_context, candidate, candidate_context)* → *salience*.

In the context are only rules with respect to focus and part-of-speech tags now. This can be extended with grammatical and possibly semantical information.

Finally the system should bother the human user when and only when it cannot be sure about the solution it has proposed.

As soon as the algorithm works satisfying, we want to port it to other languages. Doing this, we will possibly discover new and interesting problems.

Finally, we are working on improving the part-of-speech tagger, so that it is parameterised with respect to the set of tags. In this way, we can simply perform experiments with different sets of tags.

References

E. Brill. 1995. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–565.

C. Kennedy and B. Boguraev. 1996. Anaphora for everyone: Pronominal anaphora resolution without a parser. *Proceedings of the 16th International Conference on Computational Linguistics (COLING'96)*, 16:113–118.

S. Lappin and H. Leass. 1994. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):535–561.

D.H. Lie. 1998a. Automatic summary generation: a natural language processing approach. *M.Sc. Thesis*.

D.H. Lie. 1998b. Sumatra: A system for automatic summary generation. *Proceedings TWLT14*, pages 173–176.

R. Mitkov. 1998. Robust pronoun resolution with limited knowledge. *Proceedings of the 18th International Conference on Computational Linguistics (COLING'98/ACL'98)*, 18:869–875.

R. Mitkov. 1999. Multilingual anaphora resolution. *Machine Translation*, 14:281–299.