Coreference Resolution with and without Linguistic Knowledge

Olga Uryupina

Computational Linguistics
Saarland University
Saarbrücken, 66041, Germany
ourioupi@coli.uni-sb.de

Abstract

State-of-the-art statistical approaches to the Coreference Resolution task rely on sophisticated modeling, but very few (10-20) simple features. In this paper we propose to extend the standard feature set substantially, incorporating more linguistic knowledge. To investigate the usability of linguistically motivated features, we evaluate our system for a variety of machine learners on the standard dataset (MUC-7) with the traditional learning set-up (Soon et al., 2001).

1. Introduction

Robust Coreference Resolution (CR) is essential for various NLP tasks, such as Information Extraction or Question Answering. Although there has been much attention to the problem, state-of-the-art Coreference Resolution algorithms still only have a moderate performance (around 60% F-measure for coreference chains on the MUC-7 data).

Cristea et al. (2002) claim that the main problem, at least for pronoun resolution, comes from complex anaphora cases that CR systems still cannot successfully handle. We see two possible solutions to the problem: one can either choose a more sophisticated statistical model or give better knowledge (more elaborated features) to the system.

Various recent studies have investigated the first possibility — extending or significantly changing CR modeling. These include, for example, sample selection (Harabagiu et al., 2001; Ng and Cardie, 2002a; Uryupina, 2004b), clustering (Cardie and Wagstaff, 1999), Bell trees (Luo et al., 2004), or sequence modeling with Conditional Random Fields (McCallum and Wellner, 2003).

The second possibility, giving the algorithm more knowledge, has not been much investigated so far — virtually all the systems rely on very few (10-20) features. A remarkable exception is the approach with 53 features advocated by Ng and Cardie (2002b): the authors report improvement, however, only with manual feature selection and after adjusting the modeling scheme.

Our study is motivated by the fact that modern linguistic theories identify a lot of factors potentially relevant for coreference resolution. Our goal is to encode this knowledge and use it for a full-scale computational CR project.

The system has 351 nominal features (1096 boolean/continuous), representing surface, syntactic, semantic and salience-based properties of markables and markables' pairs. All the values are computed fully automatically.

Although a learning-based system could benefit from a richer feature set, two following potential problems may arise:

1. Robustness and noise. More sophisticated features cannot be extracted reliably. Automatic extraction of sophisticated features inevitably leads to a noisy dataset.
2. Overfitting. With (much) more features we have a higher risk that the overfitting problem occurs.

We investigate the usability of our rich feature set in empirical evaluation experiments. To allow fair comparison, we run several machine learners on a standard corpus (MUC-7) with a traditional set-up (the setting used by Soon et al. (2001), see Section 4.1. below).

The rest of the paper is organized as follows. First, we briefly describe the data used. Section 3 introduces our extended feature set. The evaluation experiments are discussed in Section 4. Section 5 summarizes the conclusions and shows the directions for future work.

2. Data

For our experiments we use the MUC-7 corpus (Hirschman and Chinchor, 1997): this is a standard Coreference dataset for English, and various approaches have been evaluated on these data. We segment each text into sentences with the (Reynar and Ratnaparkhi, 1997) program, parse them with the Charniak's parser (Charniak, 2000), and, separately, extract named entities with the C&C NER module (Curran and Clark, 2003b). The parser's and the NE-tagger's outputs are merged to create a pool of markables. We consider the following entities (e.g., they and only they are checked for possible coreference):

- Noun Phrases (as identified by the parser): [$_{NP}$ a spin-off], [$_{NP}$ [$_{PRN}$ you]], [$_{NP}$ [$_{PRN}$ this]],...
- Pronouns (NP-pronouns and possessives, as identified by the parser): [$_{PRN}$ you], [$_{PRP}$ your],...
- Proper Names (as identified by the NE-tagger): [$_{NE}$ New York]

Note that these classes of entities overlap: for example, a pronoun may as well be a noun phrase. Complex NPs (containing an embedded non-possessive NP) are discarded.

Overall we have 5049 markables in the training sub-corpus (30 "dry-run" documents) and 3369 markables in the testing sub-corpus (20 "formal" documents).

To create training data we pair each markable (*candidate anaphor*) with some of the preceding ones (*candi-*

date antecedents)[1]. Thus, learning instances correspond to $(anaphor,\ candidate\_antecedent)$ pairs. The instance is *positive* if the anaphor and the candidate antecedent belong to the same coreference chain and *negative* otherwise. This procedure results in 34601 training instances (1703 positive, 32898 negative).

3. Features

Coreference is a complex phenomenon and therefore many factors can potentially be relevant for the resolution. We have roughly divided them into the following four groups: lexicographic, syntactic, semantic, and discourse/salience-related properties. Overall we have 351 feature[2], all of them are computed automatically using the parser's and the NE-tagger's output and the WordNet ontology (Miller, 1990).

3.1. Lexicographic Similarity

An anaphor and its antecedent often have similar, though not the same surface form: for example ("the new company", "company") or ("CHINA's Foreign Trade Minister Wu Yi", "Ms. Wu"). Various studies suggest different improvements to vanilla name-matching, for example, stripping off the determiners (Soon et al., 2001) or using approximate matching (Strube et al., 2002). For our system, we have carried out a systematic investigation of possible extensions to the naive name-matching algorithm.
We decompose our problem into three major sub-tasks:

- normalization: *low_casing*, *no_punctuation*, and *no_determiner*;
- substrings selection: *last_noun*, *last_word*, *first_word*, and *rarest_word*;
- matching: *exact_match*, *approximate_match* (Minimum Edit Distance), *matched_part* (overlap), *abbreviation*;

One can, for example, compute minimum edit distance (*matching*) between the down-cased (*normalization*) last nouns (*substring*) of an anaphor and an antecedent. The resulting value can be used as a surface similarity measure between the two.
By combining solutions to these subproblems and discarding the trivial ones, we have come up with 122 name-matching features. A detailed description can be found in Uryupina (2004a).

3.2. Syntactic Knowledge

Earlier papers on Coreference Resolution, in particular, on pronominal anaphor, have exploited various syntactic properties of markables and their contexts. Based on very simple syntactic information, the algorithms proposed by Hobbs (1978) and Lapin and Leass (1994) achieve very good performance even compared to modern approaches. Recent advances in the parsing technology (for example, (Charniak, 2000)) make syntactic information more and more valuable for any kind of natural language processing: with the performance level of around 90%, state-of-the-art parsers and taggers provide reliable and robust knowledge. Syntactic information can be viewed as explicit indicators for (appositions, copulas) or against (constraints on parse tree structure) coreference. In our system we use the following syntactic information:

- **Internal structure of a markables** is encoded in a set of features. We distinguish between definite NPs; NPs with determiners "this", "that", and "these"; pronouns (subdivided into personal, possessive, and reflexive classes); named entities; and other markables. Following Vieira and Poesio (2000), we account for pre- and post-modification (restrictive or not).
- **Tree-based constraints** are relevant for intrasentential coreference. Following traditional research on co-indexing (Barker and Pullum, 1990), we have implemented s- and c-commands. We also check for simpler properties of parse trees: whether the anaphor and the antecedent are sister nodes, or are a subject and an object of the same verb, or violate span conditions.
- We have developed elaborated high-precision heuristics for identifying **appositions** and **copulas**. For example, for appositions, we check if a candidate apposition-looking construction is not a part of coordination ("NP, NP, .. , CONJ NP"), age description ("PERSON, NUMBER"), or address ("LOCATION, LOCATION").
- We also have features to encode **number and person agreement**. The latter can be relaxed if one of the markables is a part of a quoted string ("$[\text{I}]_{ante}$'m killing two birds with one stone," said the $[$34-year-old construction-equipment salesman$]_{ana}$).
- Finally, we identify **grammatical roles** of the markables. This information cannot be directly obtained from the output of a shallow parser, so, we have developed a simple model, conditioning grammatical roles on the parent tag in the parse tree. In addition, we have more elaborated heuristics for the subject role.

Our system has 64 syntactic features.

3.3. Semantic Compatibility

Soon et al. (2001) have pointed out that 63.3% of the recall errors made by their system were due to "inadequacy of current surface features". Ng and Cardie (2002b) report that their algorithm has only moderate performance on common nouns.
This leads us to the conclusion that if we want to resolve difficult anaphors, we have to incorporate semantic knowledge into our algorithm.

- In our system we account for **gender** and **semantic class** of the markables. We combine different

[1]The choice of candidate antecedents follows the sampling strategy proposed by Soon et al. (2001) and is described in section 4.1. below.

[2]The SVM^{light} and Maxent programs do not support nominal values, so, for the corresponding experiments we have converted all the nominal features into binary ones, which resulted in 1096 features.

knowledge sources to compute the gender values: for common nouns, we climb up the WordNet ontology, whereas for proper names we search for a gender descriptor ("Mr", "Miss",...) and, if the search fails, consults our lists of female and male first names, downloaded from the U.S. Census website. For the present experiments, we use the same set of semantic classes as (Soon et al., 2001). They are obtained from the WordNet trees.

Having computed the gender and semantic class values for individual markables, we can also obtain **agreement** values.

- Following Harabagiu et al. (2001), we compute and encode **the WordNet path** from the anaphor to the antecedent.
- The semantic agreement values are only very rough estimators of semantic compatibility. The WordNet path parameters, on the contrary, are too fine-grained and lead to very sparse data. To have more realistic measures, we also use four WordNet **similarity measures** described in (Budanitsky and Hirst, 2001): the ones proposed by Jiang and Conrath, Leacock and Chodorow, Lin, and Resnik.

Our system has 29 semantic features.

3.4. Discourse and Salience

Modern pronoun resolution algorithms, for example, various instantiations of the Centering theory (Grosz et al., 1995), rely on the *salience* properties of discourse entities. Below we describe the discourse and salience-related features used by our system:

- First, we split the document into "text blocks" (for example, PREAMBLE, or BODY) and then further into paragraphs. We measure the **distance** between the anaphor and the antecedent in various ways: in markables, sentences, or paragraphs — the most recent entities are also the most salient.
- We identify **salient** markables according to various criteria proposed in the literature: linear order, hierarchy of grammatical roles, and centering parameters. We have developed a family of boolean features signaling, for example, that "*ante* is a CB of some sentence" or "*ante* is a first NP in some paragraph" .
- We combine the above-mentioned two types of features with syntactic constraints to create a set of boolean features suggesting **different salient antecedents** for a given anaphor, for example, "*ante* is the $\text{CB}_{<salience-CB>}$ of the $\text{previous}_{<distance>}$ sentence" or "*ante* is the $\text{closest}_{<distance>}$ $\text{subject}_{<salience-hierarchy>}$ with compatible $\text{agreement}_{<syntactic\ constraints>}$ features".
- Yang et al. (2004) propose to use the coreference information of the candidate antecedent: when we are processing an anaphor, the entities to the left are already resolved, so, we can, for example, compute the antecedent (**ante_ante**) proposed by our system for the candidate antecedent. Following Yang et al. (2004), we encode the salience properties of the ante_ante. In addition to the features proposed there, we also use the size of the antecedent's chain (the part of the chain from the beginning of the document to the anaphor) The biggest chains correspond to the main topics of the document, therefore, entities from these chains are more likely to be antecedents.

Our system has 136 discourse and salience-based features.

4. Evaluation

We evaluate our system on a standard dataset — the MUC-7 corpus (Hirschman and Chinchor, 1997). We train several classifiers with different machine learning algorithms to compare their performance on the traditional (Soon et al., 2001) and extended (our system) feature sets.

As a naive baseline, we take the "one chain" classification: all the markables in a document are merged to form a single coreference chain.

As a more intelligent baseline, we use a reimplemented version of the system proposed by Soon et al. (2001). It is a well-established algorithm, often cited as a reference point. Below we briefly describe the algorithm of Soon et al. (2001), introduce the machine learners we use, and discuss the evaluation results.

4.1. Intelligent baseline: Reimplementation of (Soon et al., 2001)

Soon et al. (2001) have presented the first full-scale learning-based CR system, achieving a performance level comparable to the best (knowledge-based) systems in the MUC-7 competition. It relies on just 12 very simple surface features, shown in Table 1.

The algorithm works as follows. First it pairs each anaphor in the training corpus with its closest antecedent to create a positive instance and with all the markables in between to create negative instances. The feature vectors for these instances are given to the C5.0 decision tree learner. The C5.0 internal parameters (pruning level and the minimum number of instances per leaf node) are optimized with 10-fold cross-validation.

The learner outputs a classifier that is applied to the test corpus. For each candidate anaphor in the test corpus, test instances are constructed by pairing this anaphor with the preceding markables (starting with the closest one and proceeding backward). These test instances are submitted to the classifier. Once an instance is classified as positive, it is annotated as the antecedent for the anaphor in question, and the algorithm goes on to the next candidate anaphor. Ng and Cardie (2002b) propose using not the closest positive antecedent, but the one with the highest confidence. As we want to stay as close as possible to the original (Soon et al., 2001) system, we do not follow this suggestion.

In our reimplementation, we use the same feature set and the same setting. However, we train not only a decision tree-based classifier, but also several others. As our main goal is to compare two feature sets and not to achieve the best performance level, we do not optimize learning parameters.

Feature	Values	Description
DIST	continuous	distance in sentences between *ana* and *ante*
I_PRONOUN	0,1	*ante* is a pronoun
J_PRONOUN	0,1	*ana* is a pronoun
STR_MATCH	0,1	*ana* and *ante* match after stripping off the determiners
DEF_NP	0,1	*ana*'s determiner is "the"
DEM_NP	0,1	*ana*'s determiner is "this," "that," "these," or "those"
NUMBER	0,1	*ana* and *ante* agree in number
SEMCLASS	0,1,?	*ana* and *ante* have compatible semantic classes
GENDER	0,1,?	*ana* and *ante* agree in gender
PROPER_NAME	0,1	*ana* and *ante* are both proper names
ALIAS	0,1	*ana* is an alias of *ante* or vice versa
APPOSITIVE	0,1	*ana* is in apposition to *ante*

Table 1: Features used by Soon et al. (2001)

Learner	(Soon et al., 2001) Features			Our features			Error (F)
	Recall	Precision	F-score	Recall	Precision	F-score	reduction
Ripper	44.6	74.8††	55.9	65.8††	51.1	57.5	3.8
Slipper	84.7	33.8	48.4	85.8	33.9	48.6	-0.0
C4.5	53.5	72.8††	61.7	65.1††	64.1	64.6	8.2
SVM^{light}	50.9	68.8	58.5	63.9††	67.0	65.4	19.9
Maxent	49.2	64.1	55.7	50.5	72.2††	59.4	9.1
Baseline	85.8	33.9	48.6	85.8	33.9	48.6	N/A
(Soon et al., 2001) system	56.1	65.5	60.4	N/A	N/A	N/A	N/A

Table 2: Performance on the test data (MUC-7) for different feature sets, training on all the training instances. Significantly better recall and precision figures are marked by †† (χ^2-test, $p < 0.01$) for each machine learner correspondingly.

4.2. Machine Learners

We use five publicly available machine learners in our experiments to be sure that the effect is not accidental. Each learner has advantages and disadvantages for our task.

RIPPER (Cohen, 1995) is an information gain-based decision rule induction system. The main advantage of Ripper for CR is that its rules can be composed of only very few features. It allows RIPPER to capture coreference links signaled by a single feature (for example, two parts of the copula construction are coreferent, even if they seem to have incompatible properties) The main disadvantage of Ripper is that it is very unstable: even a minor change in the feature set can potentially result in a major rearrangement of the learned classifier, and, thus, affect the system's performance in a rather unpredictable way.

SLIPPER is a newer, improved algorithm based on RIPPER and confidence-rate boosting.

C4.5 (Quinlan, 1993) is a decision tree learner. For our task it has essentially the same advantages and disadvantages as RIPPER. As an additional drawback, C4.5 is not very effective when some features (for example, grammatic roles) have a lot of not equally important nominal values. Most state-of-the-art Coreference Resolution algorithms (McCarthy and Lehnert, 1995; Vieira, 1999; Soon et al., 2001) rely on decision trees.

SVM^{light} (Joachims, 1999) is an implementation of Support Vector Machines (Vapnik, 1995), that are known for their good performance, especially for NLP tasks. In particular, SVMs have a built-in capacity control to deal with overfitting. This is especially important for our extended feature set.

Maxent (Le, 2004) is an implementation of GIS Maximum Entropy modeling (Curran and Clark, 2003a). As SVMs, ME-based classifiers, being the most non-committal models, are less prone to overfitting.

4.3. Performance and Learning Curves

Table 2 shows the system's performance for our two different feature sets: the one proposed in (Soon et al., 2001) and the one described in Section 3. above.

The SLIPPER learner could not resolve the problem: for both features sets, the SLIPPER classifier merges virtually all the markables into one chain, and, thus, performs at the baseline level.

All the other learners show better performance when a richer feature set is used. The most substantial improvement is achieved by SVM.[3] For all the learners, the Recall goes up reflecting the system's ability to resolve more difficult anaphors (significant for Ripper, C4.5, and SVM). The Precision, however, goes down indicating that this can be done only with substantial noise (significant for Ripper, C4.5).

A remarkable exception is the Maximum Entropy classifier: both its Recall and Precision go up when we add linguistically motivated features. However, the Recall goes up only

[3]To our knowledge, the system's performance with the extended feature set and the SVM classifier (F-score of 65.4%) is the best up-to-date result on the MUC-7 data.

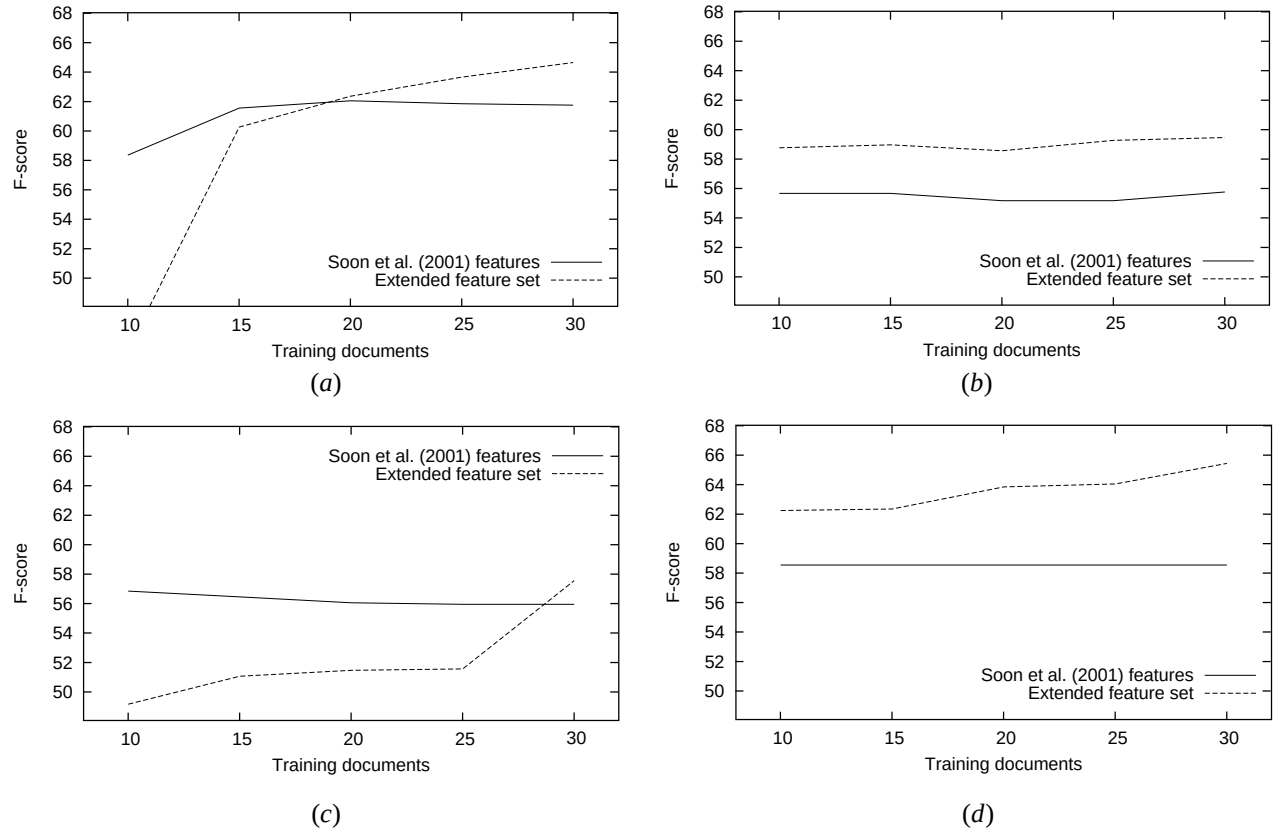

Figure 1: Learning curves (F-score) for different machine learning algorithms on the MUC-7 data: (a) C4.5, (b) Maxent, (c) Ripper, (d) SVM^{light}.

slightly, indicating that the system has not acquired new cases of anaphora, but, instead, has learned a more accurate classification for the old ones.

To investigate, why our extended feature set leads only to a moderate improvement, we have conducted second series of experiments. Our hypothesis is that the training corpus is too small to learn more sophisticated classifiers and the feature set extension leads to the overfitting problem that hinders the performance.

To see, how the system's performance depends on the amount of training data, we have learned classifiers from the first 10, 15, 20, 25, or all 30 "dryrun" documents. The resulting learning curves (F-score) are shown on Figure 1.

The curves clearly suggest that even very few documents are sufficient to learn a reliable classifier with the (Soon et al., 2001) features. However, when we increase the amount of training data, the performance remains on essentially the same level or sometimes even goes down (MaxEnt).

For the extended feature set, on the contrary, the performance is very low when only 10 training documents are available. With more training material available, the extended feature set leads to better and better classifiers, showing no sign of convergence. This suggests that one can get a much better Coreference Resolution algorithm using our linguistically motivated feature set by annotating more documents.

5. Conclusion

In this paper we have investigated the usability of linguistically-motivated features for statistical Coreference Resolution. We have encoded various relevant linguistic factors in 351 features and evaluated our system on a traditional dataset, comparing it to the knowledge-poor algorithm proposed in (Soon et al., 2001).

Our experiments show that the proposed extension of the feature set results in a moderate, though consistent, improvement in the system's performance. However, as the learning curves show no signs of convergence, we believe that more substantial improvement can be achieved by adding more training material.

This suggests the first direction of our future work: we plan to train our system on a bigger corpus, for example, on the ACE data.

We also plan to investigate more closely the impact of different feature groups on the overall performance, in particular, various possibilities for feature selection and for ensemble learning with different feature splits.

6. References

Chris Barker and Geoffrey K. Pullum. 1990. A theory of command relations. *Linguistics and Philosophy*, 13(1):1–34.

Alexander Budanitsky and Graeme Hirst. 2001. Semantic distance in wordnet: An experimental, application-

oriented evaluation of five measures. In *Proceedings of the NAACL Workshop on WordNet and Other Lexical Resources*.

Claire Cardie and Kiri Wagstaff. 1999. Noun phrase coreference as clustering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 82–89.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 132–139.

William W. Cohen. 1995. Fast effective rule induction. In *Proceedings of the 12th International Conference on Machine Learning*, pages 115–123.

Dan Cristea, Oana Postolache, and Ruslan Mitkov. 2002. Handling complex anaphora resolution cases. In *Proceedings of the 4th Discourse Anaphora and Anaphor Resolution Colloquium*.

James Curran and Stephen Clark. 2003a. Investigating gis and smoothing for maximum entropy taggers. In *Proceedings of the 11th Meeting of the European Chapter of the Association for Computational Linguistics*, pages 91–98.

James R. Curran and Stephen Clark. 2003b. Language independent NER using a maximum entropy tagger. In *Proceedings of the Seventh Conference on Natural Language Learning*, pages 164–167.

Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. 1995. Centering: A framework for modelling the local coherence of discourse. *Computational Linguistics*, 21(2):203–226.

Sanda Harabagiu, Răzvan Bunescu, and Steven Maiorano. 2001. Text and knowledge mining for coreference resolution. In *Proceedings of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 55–62.

Lynette Hirschman and Nancy Chinchor. 1997. MUC-7 coreference task definition. In *Message Understanding Conference Proceedings*.

Jerry Hobbs. 1978. Resolving pronoun references. *Lingua*, 44:311–338.

Thorsten Joachims. 1999. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT-Press.

Shalom Lapin and Herbert J. Leass. 1994. An algorithm for pronominal anaphora resolution. *Computational LInguistics*, 20(4):535–561.

Zhang Le, 2004. *Maximum Entropy Modelling Toolkit for Python and C++*.

Xiaoqiang Luo, Abe Ittycheriah, Hongyan Jing, Nanda Kambhatla, and Salim Roukos. 2004. A mention-synchronous coreference resolution algorithm based on the bell tree. In *Proceedings of the 42nd Annual Meeting of the Association for Computationa Linguistics*.

Andrew McCallum and Ben Wellner. 2003. Toward conditional models of identity uncertainty with application to proper noun coreference. In *Proceedings of the IJCAI Workshop on Information Integration on the Web*.

Joseph McCarthy and Wendy Lehnert. 1995. Using decision trees for coreference resolution. In *Proceedings of the 14th International Conference on Artificial Intelligence*, pages 1050–1055.

George Miller. 1990. Wordnet: An on-line lexical database. *International Journal of Lexicography*.

Vincent Ng and Claire Cardie. 2002a. Combining sample selection and error-driven pruning for machine learning of coreference rules. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Vincent Ng and Claire Cardie. 2002b. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computationa Linguistics*, pages 104–111.

J. Ross Quinlan. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufman.

Jeffrey C. Reynar and Adwait Ratnaparkhi. 1997. A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*.

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics (Special Issue on Computational Anaphora Resolution)*, 27(4):521–544.

Michael Strube, Stefan Rapp, and Christof Müller. 2002. The influence of minimum edit distance on reference resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 312–319.

Olga Uryupina. 2004a. Evaluating name-matching for coreference resolution. In *Proceedings of the Language Resources and Evaluation Conference*.

Olga Uryupina. 2004b. Linguistically motivated sample selection for coreference resolution. In *Proceedings of the 4th Discourse Anaphora and Anaphor Resolution Colloquium*.

Vladimir V. Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer.

Renata Vieira and Massimo Poesio. 2000. An empirically-based system for processing definite descriptions. *Computational Linguistics*, 26(4):539–593.

Renata Vieira. 1999. Applying inductive decision trees in resolution of definite NPs. In *Proceedings of the Argentine Symposium on Artificial Intelligence.*

Xiaofeng Yang, Jian Su, Guodong Zhou, and Chew-Lim Tan. 2004. Improving pronoun resolution by incorporating coreferential information of candidates. In *Proceedings of the 42nd Annual Meeting of the Association for Computationa Linguistics*.