RUPRECHT-KARLS-UNIVERSITÄT HEIDELBERG
INSTITUT FÜR COMPUTERLINGUISTIK

The Twin-Candidate and Mention-Ranking Coreference Models

SEMINAR PAPER

Course
Coreference Resolution

Semester
WS 2017/2018

Lecturer
Dr. Sebastian Martschat

Author
Lyubomira Dimitrova

Email
dimitrova@cl.uni-heidelberg.de

Matr. number
3443834

April 12, 2018
Heidelberg

Table of Contents

1 Introduction

Coreference, in linguistics, or more particularly in discourse semantics, occurs when two or more words or phrases in a text refer to the same entity. **Coreference resolution** aims to group all such words and phrases (often called *mentions* or *markables*) according to their reference entities, effectively building a partition over an entire text, or even set of texts. In the field of natural language processing, coreference resolution represents an important step in many different problems and tasks, including, but not limited to, Information Retrieval, Machine Translation, and Automatic Summarization.

The first attempts to automatize coreference resolution include knowledge based e.g. (e.g. Hobbs, 1978), and knowledge-poor algorithms that rely on heuristics (e.g. Baldwin, 1997). These systems, however, sought to solve the pronoun resolution problem, above all, i.e. finding the correct antecedent of an anaphoric pronoun. The coreference resolution over all noun phrases (NPs) is much more complex, requiring a different set of features. Nevertheless, it is a common practice in computational linguistics to recast the coreference resolution problem as a sequence of **anaphora resolutions** (where an anaphor is defined as any referring expression that does not start a coreference chain). Then, a clustering algorithm is used on the produced antecedent-anaphor pairs, which through transitivity defines a final partition of the set of mentions.

The transition from hand-crafted to learning systems was gradual. It was the work of Soon et al. (2001) that showed machine learning approaches can achieve results "*competitive with that of state-of-the-art systems using non-learning approaches*" (Soon et al., 2001, Section 6). Their system, **the single candidate classifier**, reduces the coreference resolution problem to a binary classification problem, making binary decisions about whether two mentions (or markables) are coreferential or not. Since this method operates locally, over pairs of mentions, it has also been called the *mention-pair* coreference model. While Soon et al.'s model is in fact the first machine learning approach to rival non-learning systems on the MUC data sets (MUC-6, 1995; MUC-7, 1998), its reliance on local coreference decisions fails to capture the discourse dependencies present in every text. Yang et al. (2003) seek to remedy this potential drawback by having candidate antecedents 'compete' not only during test time, but also as part of the training process. Another approach which addresses the mention-pair model's shortcomings is the mention-ranking model of (Denis, 2007). Pascal Denis argues that classification

in general is less suitable for coreference resolution, since one aims to find the best possible antecedent of a mention, and not *all* antecedents. Instead, a ranking function is used to impose an ordering of the candidate antecedents.

We will see the twin-candidate and mention-ranking models in detail in the next sections. My goal is to make a direct comparison between them in order to examine their theoretical soundness, as well as their suitability to the coreference resolution task and to further refinement.

2 The Twin-Candidate Classifier

Like many other machine learning models, Yang et al. (2003) view the coreference resolution problem as a sequence of anaphora resolutions. Instead of pairs of mentions like in the single-candidate model, however, **triples** of mentions are taken into account. While a mention-pair model is trained on instances in the form $(m_i,\ m_j)$, where m_i (possible antecedent) and m_j (anaphor) are mentions and m_i precedes m_j, the twin-candidate classifier uses instances $(m_i,\ m_j,\ m_k)$, where m_k is the anaphor, and m_i, m_j two possible antecedents. The binary decision is still present. However, instead of coreferential/not coreferential, the decision is first/second, i.e., which antecedent candidate (m_i or m_j) is more likely a true antecedent for m_k. Intuitively, once this comparison is made for all pairs of candidates, the winner of the largest number of such round-robin contests would be chosen as the *true* antecedent for m_k.

The twin-candidate model is trained with a decision tree learning algorithm - C5.0 (Quinlan, 1993) - on training instances in the form:

$$\begin{aligned}\mathcal{T} = &\left\{(c_i,\ c_j,\ ana) \mid i > j,\ c_i \in positive\_set,\ c_j \in negative\_set\right\} \cup \\ &\left\{(c_i,\ c_j,\ ana) \mid i > j,\ c_i \in negative\_set,\ c_j \in positive\_set\right\}\end{aligned} \tag{1}$$

As the definition states, each training instance contains a positive candidate (one from the same coreference chain as the anaphor) and a negative candidate[1].

No details of the training process or the final decision tree were given in the paper. However, the authors report their feature set, built on "*features that can be obtained with low annotation cost and high reliability*" (Yang et al., 2003, Section 4.2). The feature set therefore includes features describing the antecedent candidates (linguistic form (i.e. whether a pronoun, a definite NP, a proper name etc.;

[1]See Yang et al. (2003, Section 3.1) for a specific example for the creation of training instances.

whether the candidate is a part of an appositive structure, or is the nearest NP to the anaphor). A further subset of features concerns the anaphor itself (linguistic form) and the relationship between the anaphor and any one of its candidates (presence or lack of morphosyntactic agreement, whether they match in string, or if one is an alias/acronym of the other). Finally, a last couple of features measure the distance between the two candidate antecedents in sentences and paragraphs.

After training, the twin-candidate classifier can be evaluated on unseen test data. Before resolution can begin, the NPs to resolve are collected with a **mention extraction pipeline** consisting of tokenization, sentence segmentation, named entity recognition, part-of-speech tagging, and NP chunking.

It is clear at this point that a disadvantage of this approach is its computational cost. For the n-th mention in a text, n^2 pairs of candidate antecedents need to be considered, allowing for a final complexity of $\mathcal{O}(n^3)$. In order to reduce these costs, and any data noise, a **candidate filter** is applied. Yang et al. (2003) use a filter during training as well as testing, based on a different strategy for pronouns and non-pronominal anaphors.

- **Training:**
  - *pronouns:* Include all non-pronominal candidate antecedents from the current sentence and the two previous sentences. If this set is empty, look in a previous sentence.
  - *non-pronouns:* Include all non-pronominal candidate antecedents.
- **Testing/Resolution:**
  - *pronouns:* Same as training.
  - *non-pronouns:* Include all non-pronominal candidates with a confidence value over 0.5, as calculated by a single-candidate model.

The treatment of non-pronouns during test time serves as a kind of implicit anaphoricity filter, i.e., aims to show whether the referring expression is an anaphor and has an antecedent, or is more likely non-anaphoric. Without such a filter, the twin-candidate classifier would always choose an antecedent, even for the first mention of an entity. This is due to the antecedent selection algorithm[2] - the candidate with the largest number of won round-robin contests is chosen for the *true* antecedent. If multiple candidates have the same score, the one closest

[2](Yang et al., 2003), Figure 1

to the anaphor is chosen. Using the single-candidate model as a kind of back-off during resolution is an attempt to judge whether a coreference link is at all sensible. See (Soon et al., 2001) for a more detailed explanation of the single-candidate threshold.

From Yang et al.'s candidate filtering strategy, we can also infer some rather common assumptions for the coreference resolution task - (1) a pronoun cannot start a coreference chain, and is unacceptable as an antecedent; (2) most pronouns find their antecedent in a small sentence window, while some non-pronouns could find theirs at a longer distance, or even be the first mention of an entity (Ariel, 1988).

In conclusion, the idea to use a pair of antecedents aims to capture the preference relationship between candidate antecedents, instead of regarding only the relationship between the anaphor and one candidate. Yang et al.'s end-to-end coreference system, relying on a twin-candidate model, with the aid of a competition learning approach, achieves the highest MUC F-score on the MUC-6 data set - 71.3% (Yang et al. (2003); Stoyanov et al. (2009)).

3 The Mention-Ranking Model

Denis and Baldridge (2008) view the coreference resolution similarly, as a sequence of anaphora resolutions. However, they argue that classification is unsuitable for, and oversimplifies the problem, proposing instead a **ranking approach** to coreference resolution. The main difference between this new model and the twin-candidate classifier consists in regarding all antecedent candidates at once, not pairwise.

Denis and Baldridge (2008) train a log-linear (maximum entropy) probabilistic model. The conditional probability of one antecedent candidate α_i being the true antecedent of an anaphoric expression π is calculated by:

$$P_{rk}(\alpha_i \mid \pi) = \frac{\exp\ w \cdot f(\pi, \alpha_i)}{\sum_k \exp\ w \cdot f(\pi, \alpha_k)} \quad , \tag{2}$$

where k is the number of antecedent candidates for the anaphor π, and $f(\pi, \alpha)$ the feature vector. This conditional probability with respect to the entire candidate set imposes a ranking among the antecedent candidates.

During training, a weights vector w needs to be learned that correctly sets apart the one *true* antecedent from all non-antecedents. A training instance ulti-

mately consists of the anaphor to resolve, one true antecedent, and a set of non-antecedents: $(\pi, \alpha^*, \mathcal{C}_{non})$. Different strategies are employed for choosing the *one* true antecedent for an anaphor:

- *pronouns:* Choose the closest preceding mention in the coreference chain as the *true* antecedent.
- *non-pronouns:* Choose the closest preceding non-pronominal mention in the coreference chain as the *true* antecedent.
- *all:* Collect as non-antecedents all mentions in a window of two sentences around the *true* antecedent, which do not corefer with π.

Again, some assumptions of the authors can be deduced from these strategies. For example, pronominal antecedents are allowed, but only for pronominal anaphora.

At test time, **gold mentions** are used, and the candidate set includes all preceding mentions. For each anaphoric expression π, the highest scoring antecedent from the candidate set $\mathcal{C}_\pi$ is chosen as the *true* antecedent α^*:

$$\alpha^* = \underset{\alpha \in \mathcal{C}_\pi}{\text{argmax}} \; w \cdot f(\pi, \alpha) \tag{3}$$

The ranker, like the twin-candidate classifier, also faces the problem of recognizing non-anaphoric expressions, because it always chooses an antecedent from the candidate set. Denis and Baldridge (2008) use a separate, explicit discourse status filter on each potential anaphor, resolving only the ones for which the decision is 'discourse-old', i.e., referring to an existing entity. They report an 80.8% accuracy of the discourse status filter on the ACE data set.

A second idea presented in the paper is the use of **separate, specialized models** for different types of anaphoric expressions. The authors train ranking models for third person pronouns, speech pronouns (e.g. I, you), proper names, definite noun phrases, and for the *other* category, which includes indefinite NPs and demonstratives. This division they motivate with the distribution of anaphors in the ACE corpus[3], as well as with the connection between the form of an NP and its anaphoric behaviour, proposed with the Accessibility Hierarchy of (Ariel, 1988). This division aims to provide the resolution task with a feature set tailored to each particular kind of referring expression. While all five models

[3](Denis and Baldridge, 2008), Table 1; more detailed in (Denis, 2007), Table 4.3

benefit from features concerning the candidate antecedent (linguistic form; surrounding part-of-speech tags), and features describing the relationship between the anaphor and the antecedent candidate (agreement in number, gender and person; distance between them in mentions and sentences; WordNet[4] sense similarity), **some features are model-specific**. For instance, the proper names model retains features concerning string similarity, appositive structures, or an acronym relationship between anaphor and antecedent, which are excluded in the two pronominal models. Features describing the anaphor itself are not present, covered instead by the specialized models.

After training, the authors proceed to evaluate their system on the ACE data sets, using **gold mentions**. The reported results beat their classifier system modeled on (Ng and Cardie, 2002b), proving that the ranker offers better antecedent selection capabilities than the single-candidate coreference model employing a best-first link selection strategy[5]. Specialized rankers relying on a discourse status classifier achieve 71.6%, 72.7%, and 67.0% MUC, B^3, and CEAF F-measures on the ACE corpus, respectively.

4 Comparison

4.1 Model

The main objective of the twin-candidate classifier of (Yang et al., 2003) is to learn a **preference criterion** between any two antecedent candidates for an anaphor. In contrast, the ranker aims to learn how to set apart the **one true antecedent** from all other candidates. As already discussed, the twin-candidate model examines pairs of antecedent candidates, while the mention-ranking model regards the entire candidate set at once.

Objectively, the latter should obtain more accurate results due to the more informed, less local context available for the antecedent selection. Denis (2007) argues that the *only* significant difference between the two coreference models is the objective function - one uses a ranking function, while the other - a classification funtion. The general approach, however, remains the same: both systems model coreference resolution as a sequence of anaphora resolutions, and use a

[4] `https://wordnet.princeton.edu`

[5] Instead of the closest-first strategy of (Soon et al., 2001), where the closest antecedent candidate with a score over 0.5 is chosen, the best-first approach simply picks the highest scoring candidate.

clustering algorithm to form the coreference chains. Resolutions are thus completely independent from one another, and no global coherence can be ensured.

From a general machine learning perspective, maximum entropy models are perhaps the most widely-used learners in natural language processing, attractive due to their robustness and ability to lessen the influence of incorrect model assumptions. Furthermore, Denis (2007) reports that the ranker has a complexity equal to that of the single-candidate model - $\mathcal{O}(n^2)$ (compared to $\mathcal{O}(n^3)$ for the twin-candidate model), and therefore no restrictions regarding the size of the candidate set. As discussed in (Denis, 2007, Section 3.5.3), increasing the context window and subsequently the size of the candidate set improves performance.

4.2 Features

We observe the feature sets of the two models[6], following the feature categorization of (Yang et al., 2003).

- **Features describing the anaphor:** While Yang et al. (2003) include features describing the anaphor, mostly concerned with its linguistic form, Denis and Baldridge (2008) have no use for them, since the specialized models assume that function.

- **Features describing the/an antecedent candidate:** The features describing the antecedent candidate alone are much more detailed in the twin-candidate classifier (10, versus 4 in the mention-ranking model). It is important to note that the 10 different features have to be present twice in the final feature vector, once for each of the two antecedent candidates. This brings us to the conclusion that the twin-candidate classifier needs a larger number of features, the extraction of which might complicate the training and testing process. Denis and Baldridge (2008) use additional three features to describe the context of the antecedent candidate, namely the part-of-speech tags of the previous and following word.

- **Features describing the candidate and the anaphor:** Other than the features describing the antecedent candidate, this is the only other feature type in the mention-ranking model. All five features present in (Yang et al., 2003) are common to both models - (1) full string match between anaphor and antecedent, agreement in (2) number and (3) gender, whether the anaphor and

[6] Yang et al. (2003), Table 1, and Denis and Baldridge (2008), Table 2

antecedent are in an (4) appositive structure, or one is an (5) acronym/alias of the other. However, Denis and Baldridge (2008) add further, more refined string similarity features (substring and head word matches), as well as a semantic compatibility feature, in the form of paired WordNet senses. Perhaps the most surprising difference lies in the use of distance features between anaphor and antecedent in the mention-ranking, but not in the twin-candidate model. Denis and Baldridge (2008) motivate their usefulness by stating that most pronouns find their antecedent at a shorter distance (1-2 sentences), a preference which distance features can encode.

- **Features describing the two candidates:** Yang et al. (2003) measure the distance between the two candidate antecedents, in sentences and paragraphs. However, no motivation is given for this decision. Of course, this category of features is present only in the twin-candidate classifier.

4.3 Recognizing anaphors

As mentioned in Sections 2 and 3, both models face the problem of determining whether a NP is anaphoric or not, whether it refers to a previously mentioned entity, or introduces a new entity. Yang et al. (2003), like many other classification-based approaches to coreference resolution (e.g. Soon et al., 2001) address this issue only **implicitly** - a NP is considered non-anaphoric only if the system fails to find an antecedent. Through the use of score thresholds and candidate filters, the classifier itself is coerced into also functioning as an anaphoricity classifier (no separate model is necessary). The twin-candidate classifier assumes all mentions to be potential anaphors, and examines all possible antecedent combinations. Such an approach, however, may be error-prone (since an overwhelming number of the entities in a text are mentioned only once) and computationally expensive.

The alternative, namely a separate anaphoricity filter, applied on each potential anaphor prior to its resolution, has been explored in the work of (Ng and Cardie, 2002a). A similar, **explicit** discourse status classifier, is employed in (Denis and Baldridge, 2008), described in detail in (Denis, 2007, Section 4.3). It is clear, however, that errors made by the discourse status filter propagate to the coreference resolution - some non-anaphors are resolved, and vice versa. Essentially, the discourse classifier's decisions are always taken on faith by the rankers, and, since the two models are separate, there is no guarantee of global coherence.

4.4 Results

The mention-ranking (Denis and Baldridge, 2008) and twin-candidate (Yang et al., 2003) coreference models are trained and tested on different data sets, so a direct comparison of the achieved results is impossible. Denis (2007) compares the ranker to the twin-candidate model of (Yang et al., 2005) on the ACE data set in the limited context of pronoun resolution. The results suggest that the ranker performs better than the classifier. However, there are a few differences between in the implementations of the twin-candidate model in (Denis, 2007) and (Yang et al., 2005). For example, while the original model uses a decision tree learning algorithm, the model in (Denis, 2007) takes a log-linear form. This discrepancy, combined with the pronoun-only resolution task, further complicates the comparison of the models described in Sections 2 and 3.

Another point to examine concerns the test time setup of the two systems. While Yang et al. (2003) use an NLP pipeline to extract the mentions in the test data, Denis and Baldridge use gold mention boundaries during test time. This is a fairly common practice in coreference resolution systems, both on the MUC and ACE data sets (e.g. Luo et al., 2004). However, it is argued by Stoyanov et al. (2009) that providing a coreference resolver with annotated mentions gives unrealistic evaluations of the system. In an external information extraction application, for instance, no annotated data is provided for the trained model, and its ability to correctly recognize (at the very least) NPs is essential.

4.5 Refinements & further work

Approaches building on the twin-candidate classifier are mostly confined to the work of Xiaofeng Yang.

The problem of recognizing non-anaphoric expressions is addressed in (Yang et al., 2005), where a joint learning framework for coreference and anaphoricity is proposed. This modified system avoids setting a specific threshold, as well as the use of a separate anaphoricity filter, and yields better results than the original classifier on both MUC data sets. (Yang et al., 2008) contains perhaps the most detailed examination of the twin-candidate classifier. The authors compare different antecedent selection strategies (round robin and tournament elimination), and learning algorithms (the C5 decision tree learner, a Ranking-SVM preference learner, and a maximum entropy model). The results suggest that a maximum en-

tropy twin-candidate model employing a round-robin antecedent selection strategy performs best on the ACE data set.

Although the oracle systems presented in (Denis and Baldridge, 2008, Section 5.3) suggest room for improvement in the performance of the specialized ranker models, the authors do not expand on the idea. However, many other authors employ the log-linear model for coreference resolution in their state-of-the-art performing systems. Durrett and Klein (2013), for instance, use a log-linear model in combination with a complex loss function and a set of shallow features. The structured prediction framework of (Martschat and Strube, 2015) recasts the mention-ranking approach as a latent structure in the form of an unlabeled graph. Even more recently, Clark and Manning (2016) use reinforcement learning to train a neural mention-ranking coreference model.

5 Conclusion

I presented and compared two well-known supervised machine learning approaches to coreference resolution. As the earlier work, (Yang et al., 2003) is predictably more influenced by the single-candidate classifier, and still relies on a decision tree learning algorithm and binary classification. However, it represents an important step towards reducing the locality of coreference decisions. (Denis and Baldridge, 2008) is, in fact, a further step in the same direction. As already mentioned, the two models are actually quite similar, especially in the context of entity-based coreference resolvers (e.g. Lee et al., 2013). Nevertheless, I believe the ranking approach of (Denis and Baldridge, 2008) is more suitable to the coreference resolution task, due to its ability to regard the entire set of antecedent candidates at once. This models the human intuition regarding anaphora resolution better than the twin-candidate classifier. Furthermore, the smaller feature set and the use of a deterministic model allow the ranker to differentiate between many, potentially similar, candidates, to find the "best" antecedent. Finally, its soundness is attested by the numerous works in the field of coreference resolution which employ the mention-ranking model, albeit with modifications.

While this ranking approach no longer holds the title of state-of-the-art, there is no doubt that it helped others attain it.

References

Ariel, M.

1988. Referring and accessibility. *Journal of linguistics*, 24(1):65–87.

Baldwin, B.

1997. Cogniac: high precision coreference with limited knowledge and linguistic resources. In *Proceedings of a Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*, Pp. 38–45. Association for Computational Linguistics.

Clark, K. and C. D. Manning

2016. Deep reinforcement learning for mention-ranking coreference models. *arXiv preprint arXiv:1609.08667*.

Denis, P.

2007. *New learning models for robust reference resolution*. The University of Texas at Austin.

Denis, P. and J. Baldridge

2008. Specialized models and ranking for coreference resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Pp. 660–669. Association for Computational Linguistics.

Durrett, G. and D. Klein

2013. Easy victories and uphill battles in coreference resolution. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Pp. 1971–1982.

Hobbs, J. R.

1978. Resolving pronoun references. *Lingua*, 44(4):311–338.

Lee, H., A. Chang, Y. Peirsman, N. Chambers, M. Surdeanu, and D. Jurafsky

2013. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39(4):885–916.

Luo, X., A. Ittycheriah, H. Jing, N. Kambhatla, and S. Roukos

2004. A mention-synchronous coreference resolution algorithm based on the bell tree. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*.

Martschat, S. and M. Strube

2015. Latent structures for coreference resolution. *Transactions of the Association of Computational Linguistics*, 3(1):405–418.

Ng, V. and C. Cardie

2002a. Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, Pp. 1–7. Association for Computational Linguistics.

Ng, V. and C. Cardie

2002b. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th annual meeting on association for computational linguistics*, Pp. 104–111. Association for Computational Linguistics.

Quinlan, J. R.

1993. *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Soon, W. M., H. T. Ng, and D. C. Y. Lim

2001. A machine learning approach to coreference resolution of noun phrases. *Computational linguistics*, 27(4):521–544.

Stoyanov, V., N. Gilbert, C. Cardie, and E. Riloff

2009. Conundrums in noun phrase coreference resolution: Making sense of the state-of-the-art. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, Pp. 656–664. Association for Computational Linguistics.

Yang, X., J. Su, and C. L. Tan

2005. A twin-candidate model of coreference resolution with non-anaphor identification capability. In *International Conference on Natural Language Processing*, Pp. 719–730. Springer.

Yang, X., J. Su, and C. L. Tan

2008. A twin-candidate model for learning-based anaphora resolution. *Computational Linguistics*, 34(3):327–356.

Yang, X., G. Zhou, J. Su, and C. L. Tan

2003. Coreference resolution using competition learning approach. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, Pp. 176–183. Association for Computational Linguistics.