Relational Structures and Models for Coreference Resolution

Truc-Vien T. Nguyen[1] *Massimo Poesio*[1,2]
(1) CIMeC, University of Trento, Rovereto (TN), 38068, Italy
(2) University of Essex, Wivenhoe Park, Colchester CO4 3SQ, UK
{trucvien.nguyenthi,massimo.poesio}@unitn.it

Abstract

Coreference resolution is the task of identifying the sets of mentions referring to the same entity. Although modern machine learning approaches to coreference resolution exploit a variety of semantic information, the literature on the effect of relational information on coreference is still very limited. In this paper, we discuss and compare two methods for incorporating relational information into a coreference resolver. One approach is to use a filtering algorithm to rerank the output of coreference hypotheses. The filter is based on the relational structures between mentions and their corresponding relationships. The second approach is to use a joint model enriched with a set of relational features derived from semantic relations of each mention. Both methods have shown to improve the performance of a learning-based state-of-the-art coreference resolver.

Keywords: coreference resolution, relation extraction, machine learning.

1 Introduction

Much of the recent progress in statistical models of coreference resolution (Rahman and Ng, 2009, 2011; Ng, 2010) has come from the adoption of richer models of this interpretive task that overcome the limitations and simplifications of earlier models (Soon et al., 2001; Ng and Cardie, 2002), such as the assumption that resolving coreference involves linking mentions. There has also been some progress towards taking advantage of richer forms of information in general and of semantic knowledge in particular. Lexical knowledge has been shown to be clearly useful (Ponzetto and Strube, 2006) and is exploited by most state-of-the-art systems (Bengtson and Roth, 2008; Lee et al., 2011); it has been shown that encyclopedic knowledge as contained e.g., in Wikipedia can help as well (Ponzetto and Strube, 2006; Uryupina et al., 2011). But the ultimate goal is to develop a statistic-based integrated model of semantic interpretation in which coreference interacts with other aspects of interpretation such as predicate-argument structure recognition or discourse structure resolution, as argued in particular by (Hobbs, 1979) and implemented on a small-scale basis in the early, pre-statistical systems (Wilks, 1975; Hobbs et al., 1993; Alshawi, 1992).

Most work to this end has been concerned with the use of semantic role information to improve in particular the resolution of pronouns (Yang and Su, 2007; Ponzetto and Strube, 2006; Bean and Riloff, 2004). However, there has been much more limited investigation of the effect on coreference of the information provided by ACE-style relations. This is surprising given, first, that *prima facie*, such information should be very useful, and second, that annotated containing both coreference and relational information exist, most notably ACE-05. ACE-style relational information could be useful to increase precision, by ruling out coreference relations between entities already known to be related by other relations: if Jack is related by a ‘colleague’ relation with Mr. Smith, then most likely Jack and Mr. Smith are not coreferent. Such information could also be useful to increase recall: if Jack is related by a ‘works-for’ relation to an entity mentioned as ‘Foobar Inc.’ and by a ‘colleague’ relation with Mr. Smith, and Mr. Smith is related by a ‘works-for’ relation to an entity mentioned as ‘the international conglomerate’, then most likely ‘Foobar Inc.’ and ‘the international conglomerate’ are mentions of the same entity. Yet we are only aware of one study exploring the use of such information to improve coreference, namely (Ji et al., 2005), whose approach however was rule-based. In this paper we revisit the topic and compare rule-based methods with machine-learning approaches to integrating relational and coreference information.

The structure of the paper is as follows. In Section 2 we discuss previous work on using relational information for coreference. In Section 3 we describe relational information in the ACE corpora. In Section 4 we propose three methods for integrating relational information in a coreference resolver; the experimental setting used to evaluate these methods and the results we obtained are discussed in Section 5.

2 Related Work

The most closely related work to ours is the proposal by (Ji et al., 2005), who use heuristics to integrate constraints from relations between mentions with a coreference resolver. Their methodology involves a two-stage approach where the probabilities output from a MaxEnt classifier are rescored by adding information about the semantic relations between the two candidate mentions. These relations are automatically output by a relation tagger, which is trained on a corpus annotated with the semantic relations from the ACE 2004 relation ontology. Given a candidate pair 1.B and 2.B and the respective mentions 1.A and 2.A they are related to in the same document, Ji et al identify three lightweight rules to identify configurations informative of coreference:

1. If the relation between 1.A and 1.B is the same as the relation between 2.A and 2.B, and 1A and 2A don't corefer, then 1.B and 2.B are less likely to corefer.

2. If the relation between 1.A and 1.B is different from the relation between 2.A and 2.B, and 1.A is coreferent with 2.A, then 1.B and 2.B are less likely to corefer.

3. If the relation between 1.A and 1.B is the same as the relation between 2.A and 2.B and 1.A is coreferent with 2.A, then 1.B and 2.B are more likely to corefer.

While Ji et al. argue that the second rule usually has high accuracy independently of the particular relation, the accuracy of the other two rules depends on the particular relation. For example, the chairman of a company, which has a EMP- ORG/Employ-Executive relation, may be more likely to remain the same chairman across the text than a spokesperson of that company, which is in the EMP- ORG/Employ-Staff relation to it. Accordingly, the system retain only those rule instantiated with a specific ACE relation which have a precision of 70% or more, yielding 58 rule instances. For instances that still have lower precision, they try conjoining additional preconditions such as the absence of temporal modifiers such as "current" and "former," high confidence for the original coreference decisions, substring matching and/or head matching. In this way, they can recover 24 additional reliable rules that consist of one of the weaker rules plus combinations of at most 3 of the additional restrictions. They evaluate the system, trained on the ACE 2002 and ACE 2003 training corpora, on the ACE 2004 evaluation data and provide two types of evaluation: the first uses Vilain et al's scoring scheme, but uses perfect mentions, whereas the second uses system mentions, but ignore in the evaluation any mention that is not both in the system and key response. Using these two evaluation methods, they get an improvement in F-measure of about 2% in every case. In the main text of the paper, Ji et al. report an improvement in F-measure from 80.1% to 82.4%, largely due to a large gain in recall. These numbers are relatively high due to the fact that Ji et al. use a relaxed evaluation setting disregarding spurious links. A strict evaluation on exact mentions is able instead to yield an improvement in F-measure from 62.8% to 64.2% on the newswire section of the ACE corpus.

3 Relational Information in the ACE corpora

The ACE effort (Doddington et al., 2004) (Automatic Content Extraction) aims at developing technology for automatically carrying out inference in natural language text. The data includes the entities being mentioned, the relations among these entities that are directly expressed, and the events in which these entities participate. The program began with a pilot study in 1999. Moreover, data includes various source types (image, audio, text) and languages (English, Arabic).

We use the ACE 2005 Multilingual Training Corpus[1]. ACE defines 7 major entity types: FAC (Facility), GPE (Geo-Political Entity: countries, cities, etc.), LOC (Location), ORG (Organization), PER (Person), VEH (Vehicle) and WEA (Weapon). Relationship is defined in ACE as semantic relations between pairs of entities in texts. Note that relations in ACE are mostly directional (i.e., asymmetric), very few are symmetric, such as PHYS.Near that characterizes the two locations are nearby and PER-SOC.Family-Colleague that characterizes a family or colleague relationship.

Table 1 shows examples of ACE relations, the pair of arguments participating in the relation with their directionality, according to ACE guidelines and standards. In the models that integrate relational features, we mainly take the relation's direction into account to compute the features. In the following, we use the term *head* and *tail* to indicate the mentions where the relations are directed from and to, respectively.

[1] http://www.ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC2006T06

Relation type	Example	From	To
ART(artifact)	*My house is in West Philadelphia* ART.User-Owner("my", "my house")	"my"	"my house"
GEN-AFF	*U.S. businessman* GEN-AFF.Citizen("businessman", "U.S")	"businessman"	"U.S"
ORF-AFF	*the CEO of Yahoo* ORF-AFF.Employment("the CEO", "Yahoo")	"the CEO"	"Yahoo"
PART-WHOLE	*Northern Ireland in Belfast* PART-WHOLE.Geographical("Belfast", "Northern Ireland")	"Belfast"	"Northern Ireland"
PER-SOC*	*their colleagues* PER-SOC.Business("their", "their colleagues")	"their"	"their colleagues"
PHYS*	*a news conference in Paris* PHYS.Located("conference", "Paris")	"conference"	"Paris"

Table 1: Relation types in ACE 2005 and their directionality

4 Embedding Relational Information

In this section, we describe three methods for integrating relational information in a coreference resolver: the reranker, the enriched model and the joint model.

In traditional mention-pair coreference resolvers (Soon et al., 2001), the training and testing units are pairs $< x, y >$ of candidate antecedent and anaphor. The system extracts a vector v which contains syntactic and semantic features from these two mentions. A coreference resolver then learns a mapping function $v \rightarrow c$ where $c = (0, 1)$ indicates if x and y belong to the same coreference chain. In other words, a coreference resolver estimates $p(c|v)$, the probability that x is the antecedent of y given the feature vector v.

4.1 Reranking

The coreference reranker operates by first applying a baseline model trained using a maximum entropy classifier with the features proposed in (Soon et al., 2001) to determine whether two mentions $(antecedent, anaphor)$ are coreferent or not. We then use the resulting coreference chains c in combination with the relationships between mentions to construct a set of *relational structures*. We then extract from those structures a vector r of *relational features*. The coreference reranker then integrates v and r and improves the mapping function $(v, r) \rightarrow c$.

In other words, when integrated with relational information, the system extracts a vector r of relational features, which are derived from both the coreference chains c of the base model and relationships between pairs of mentions. The coreference reranker then integrates v and r and improves the mapping function $(v, r) \rightarrow c$.

Figure 1 shows the relational structure for the coreference chain on the left. The directionality specified in Table 1 is used to determine the relations belonging to the structure: only the relations whose first argument ('from' in Table 1) belongs to the coreference chain on the left are considered part of the relational structure for that coreference chain; which represents the coreference chains as group of mentions on the left, their relationships and other participants on the right. These structures are used to infer if it is likely that two mentions corefer, as described in the following.

From the relational structure we extract features that can supplement the information available to the base coreference resolver. Our set of features are inspired from those used by (Ji et al., 2005), but the method discussed in this subsection differs from theirs in three important respects, as discussed below.

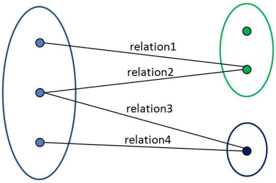

Figure 1: Relational structure

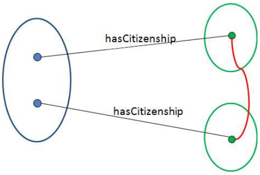

Figure 2: Coref_SameRelation

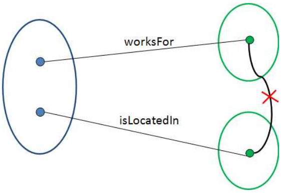

Figure 3: Coref_NotSameRelation

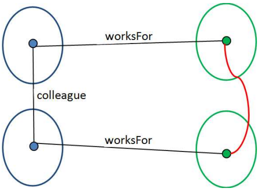

Figure 4: Coref_Transitivity

1. **Coref_SameRelation**: if two mentions in the same coreference chain have two relations directed from them with the same relation type and direction, then the two participants in those relations are likely to corefer, as illustrated in Figure 2.

2. **Coref_NotSameRelation**: if two mentions in the same coreference chain have two relations directed from them with different relation type and the same direction, or the same relation type but different direction, then the two participants in those relations are unlikely to corefer, as illustrated in Figure 3.

3. **Coref_Transitivity**: : if two mentions in different coreference chains have two relations directed from them with the same relation type and the same direction, and if these two mentions have the same semantic classes and participate in "maybe peer" relation (such as PHYS.Near or PER-SOC.Colleague), then the two participants in those relations are likely to corefer, as illustrated in Figure 4.

Our proposal differs from the work of (Ji et al., 2005) in three aspects. First, our approach is not rule-based but learning-based. Second, we do not compute the reliability weight for each rule; instead, we integrate each feature with relation type/direction directly to the learning model and let the model learn automatically. Finally, whereas their second and third rules are similar to our feature $FE\_Coref\_SameRelation$ and $FE\_Coref\_NotSameRelation$, we do not use the first rule (discussed in section 2) that refers to the two mentions in two different chains that have the same relation type/direction, since that rule is problematic. For example, the fact that *Bush* and *Obama* are mentions in different coreference chains with the same relation types/direction *leadership* with mentions of the entity *US*, doesn't mean that the two mentions *US* participating in the relationships cannot corefer.

4.2 Relational Features

An alternative approach is to use relational information to define features. Relational features are derived from relationships between mentions. As shown in Table 1, a relationship in ACE is defined between a pair of mentions, with a corresponding relation. In Table 1, each relationship is directed from one mention to another, the direction, as we notice, is *many-to-one* in most of cases and should be taken into account. Given a pair of *(antecedent, anaphor)*, we then extract relations for each mention and define the following features.

1. **FE_Related** characterizes relationships hold between a anaphor with its potential antecedent. Reasonably, relationships should not be hold between mentions of the same coreference chain.

2. **FE_SameRelation** determines if the pair *(anaphor, antecedent)* has two relations starting from them with the same relation type and direction. We argue that if the two mentions *(anaphor, antecedent)* have relationships of the same type/direction (e.g., *hasCitizenship* or *worksFor*), then it is more likely they are corefered.

3. **FE_SameRelationEntity** determines if the pair *(anaphor, antecedent)* has two relations starting from them with the same relation type/direction and directed to the same mention.

4. **FE_SameRelationWithPeer** determines if the pair *(anaphor, antecedent)* has two relations starting from them with the same relation type/direction and if the relations are directed to the two mentions of the same semantic type and connected by a "peer" relation, such as PHYS.Near or PER-SOC.

5. **FE_LeftRelation** describes the set of relation types in common between *antecedent* and *anaphor* where relations are those with these two mentions as *head*, as described in Table 1. We construct a vector from set of relations where *antecedent* and *anaphor* are the head, respectively, then compute the *dot-product* between the two vectors.

6. **FE_RightRelation** is the same as above, but applied for relations are those with these two mentions as *tail*.

7. **FE_SumRelation** computes the sum of FE_LeftRelation and FE_RightRelation.

8. **FE_SubtractRelation** computes the subtraction of FE_RightRelation and FE_LeftRelation. Given that the relation' direction is almost *many-to-one*, we argue that the *tail mention* promise to be more effective. Therefore, we compute the *dot-product* of *tail mention* and of *head mention* with respect to the pair *(antecedent, anaphor)* and take the subtraction of these two.

9. **FE_MultiplyRelation** computes the multiplication of FE_LeftRelation and FE_RightRelation.

4.3 Enriched Model and Joint Model

Given the baseline and set of additional relational features as described in the previous section, the enriched model works simply by adding those features into the baseline. Although the features *FE_Related, FE_SameRelationEntity, FE_SameRelationWithPeer* and *FE_SubtractRelation* are the best performers, the performance is almost consistent amongst the nine relational features.

However, we notice that, when integrated with each of nine relational features r_i (which we call 'individual model'), the increase in the performance is not always consistent amongst different documents. Therefore, we proceed with a joint model that learns jointly among separate individual models and picks the one with the highest score as the final answer. To train the basic models, we add each relational feature into the baseline and re-train. At testing time the model receiving the highest score is selected as the final answer.

5 Experiments and Results

5.1 Experimental Setup

Corpus. We use the ACE 2005 coreference corpus released by the LDC, which consists of the 599 training documents used in the official ACE evaluation. The corpus was created by selecting documents from six different sources: Broadcast News (bn), Broadcast Conversations (bc), Newswire (nw), Webblog (wb), Usenet (un), and conversational telephone speech (cts). For evaluation, we reuse the partition done by (Rahman and Ng, 2009) that splits the 599 documents into a training set and a test set following a 80/20 ratio, resulting in a partition of 482/117 documents.

In our experiments, we use the relation extraction model[2] proposed in (Nguyen and Moschitti, 2011). To extract mentions from both the training and test set, we used the model defined in (Nguyen et al., 2010, 2009) to train a mention extractor. When evaluated on the ACE 2005 data sets, since documents in the corpus are from six different sources with equivalent number of documents in each source, we perform 6-fold cross-validation where each fold consists of documents from one source. The performance of the relation and mention extractor is given in Table 2.

Task	Precision	Recall	F_1
Relation extractor	57.9%	59.4%	58.5%
Mention extractor	75.3%	67.7%	71.3%

Table 2: Performance of relation extraction and mention extraction

Baseline. As a baseline we train a maximum entropy classifier to generate the coreference chains. We use (Soon et al., 2001) set of features as implemented in the BART coreference toolkit[3]. The base model makes use of a maximum entropy classifier to train a *mention-pair* model, which determines whether two mentions are coreferent or not. Our baseline results are shown in Table 3 which also includes the results of another state-of-the-art coreference system of (Rahman and Ng, 2009). For this and the following experiments, all the results were computed using *MUC-score* with standard precision, recall, and *F-measure*.

	Gold mentions			System mentions		
System	Recall	Precision	F_1	Recall	Precision	F_1
Our Baseline	65.7	87.9	75.2	50.8	76.7	61.1
(Rahman and Ng, 2009)	71.7	69.2	70.4	70.0	56.4	62.5

Table 3: Performance comparison on the ACE 2005

5.2 Results

In this section, we report the results of different relationals model with the reranker, the enriched model and the joint model. Results with the reranking approach is shown in the second line of Table 4. Results of the enriched model with separate features and with the combination of all features, and results with the joint model are shown in Table 4.

[2] http://sourceforge.net/projects/reck/files/reck_v1.0.0.tar.gz/download

[3] http://www.bart-coref.org/

First, the *reranker* improves to 76.1 with *gold mentions* and 62.7 with *system mentions* when relational information is added. This suggests that the relational structures are effectively exploited with the three features as desribed in section 4.1 and that such information is somewhat complementary to the basic feature set as defined in (Soon et al., 2001).

	Gold mentions			**System mentions**		
Setting	**Recall**	**Precision**	F_1	**Recall**	**Precision**	F_1
Baseline	65.7	87.9	75.2	50.8	76.7	61.1
Reranking	**66.8**	**88.4**	**76.1**	**52.6**	**77.5**	**62.7**
FE_Related	65.7	88.2	75.3	51.0	77.0	61.4
FE_SameRelation	65.7	88.2	75.3	50.9	77.0	61.3
FE_SameRelationEntity	65.7	88.1	75.3	51.1	77.0	61.4
FE_SameRelationWithPeer	65.8	88.1	75.3	51.1	77.0	61.4
FE_LeftRelation	65.8	88.1	75.3	51.0	76.7	61.2
FE_RightRelation	65.8	88.0	75.3	50.9	77.0	61.3
FE_SumRelation	65.8	88.0	75.3	50.9	77.0	61.3
FE_SubtractRelation	65.8	88.0	75.3	51.0	77.0	61.4
FE_MultiplyRelation	65.7	88.0	75.3	51.2	77.0	61.4
Enriched Model	**66.3**	**88.7**	**76.0**	**52.1**	**76.7**	**62.0**
Joint Model	**67.0**	**88.9**	**76.4**	**54.5**	**75.7**	**63.3**

Table 4: Results with reranking, enriched and joint models

Second, the enriched model improves to 76.0 with *gold mentions* and 62.0 with *system mentions* when the base model is enriched with nine relational features. This suggests that the relation information between pairs of mentions can be encoded together with information merely from the mentions themselves.

Third, the joint model improves to 76.4 with *gold mentions* and 63.3 with *system mentions* when the enriched models are trained with separate relational features and the joint model chooses the best score for each testing instance. This suggests that the relational information, when possible to be encoded to yield better results as in the case of the *enriched model*, are not exploited as better as the *joint model* strategy. We also conducted *sign test* to measure the difference between the best model (i.e., joint model) and the baseline. The significance results are $\rho = 0.0047$ with *gold mentions* and $\rho = 0.0033$ with *system mentions*, which means that our results are statistically significant.

6 Conclusion

Previous results suggest that relational features are clearly helpful for coreference resolution in ACE. However, as we showed, there has been much more limited investigation of the effect on coreference of the information provided by ACE-style relations. Such information should be very useful, and that annotated containing both coreference and relational information exist, most notably ACE-05.

The *joint model* performs the best. That would suggest 1. relational features are helpful in linking one anaphor to its antecedent; 2. the integration of machine learning methods outperforms the merely addition of relational features, as in the enriched model.

We analyzed the impact of relational structures and features for coreference resolution. Our study demonstrates that both kinds of structures and features clearly give improvement to the coreference resolver. Most interestingly, as we shown, the integration of relational features, in combination of the ranking method, yields the best results. The joint model, that is taken by comparing the enriched models one with each other, turns out as very effective for both *gold mentions* and *system mentions*.

References

Alshawi, H., editor (1992). *The Core Language Engine*. The MIT Press.

Bean, D. and Riloff, E. (2004). Unsupervised learning of contextual role knowledge for coreference resolution. In Susan Dumais, D. M. and Roukos, S., editors, *HLT-NAACL 2004: Main Proceedings*, pages 297–304, Boston, Massachusetts, USA. Association for Computational Linguistics.

Bengtson, E. and Roth, D. (2008). Understanding the value of features for coreference resolution. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 294–303, Honolulu, Hawaii. Association for Computational Linguistics.

Doddington, G., Mitchell, A., Przybocki, M., Ramshaw, L., Strassel, S., and Weischedel, R. (2004). The automatic content extraction (ACE) program tasks, data, and evaluation. In *Proceedings of LREC*, pages 837–840, Barcelona, Spain.

Hobbs, J. R. (1979). Resolving pronoun references. *Coherence and Coreference*, 3:67–90.

Hobbs, J. R., Stickel, M., Appelt, D., and Martin, P. (1993). Interpretation as abduction. *Artificial Intelligence Journal*, 63:69–142.

Ji, H., Westbrook, D., and Grishman, R. (2005). Using semantic relations to refine coreference decisions. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 17–24, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Lee, H., Peirsman, Y., Chang, A., Chambers, N., Surdeanu, M., and Jurafsky, D. (2011). Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 28–34, Portland, Oregon, USA. Association for Computational Linguistics.

Ng, V. (2010). Supervised noun phrase coreference research: The first fifteen years. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1396–1411, Uppsala, Sweden. Association for Computational Linguistics.

Ng, V. and Cardie, C. (2002). Improving machine learning approaches to coreference resolution. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 104–111, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Nguyen, T. V. T. and Moschitti, A. (2011). Joint distant and direct supervision for relation extraction. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, Chiang Mai, Thailand.

Nguyen, T.-V. T., Moschitti, A., and Riccardi, G. (2009). Conditional random fields: Discriminative training over statistical features for named entity recognition. In *Proceedings of EVALITA 2009 workshop, the 11st International Conference of the Italian Association for Artificial Intelligence (AI*IA)*, Reggio Emilia, Italy.

Nguyen, T.-V. T., Moschitti, A., and Riccardi, G. (2010). Kernel-based reranking for named-entity extraction. In *Coling 2010: Posters*, pages 901–909, Beijing, China. Coling 2010 Organizing Committee.

Ponzetto, S. P. and Strube, M. (2006). Exploiting semantic role labeling, wordnet and wikipedia for coreference resolution. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 192–199, New York City, USA. Association for Computational Linguistics.

Rahman, A. and Ng, V. (2009). Supervised models for coreference resolution. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 968–977, Singapore. Association for Computational Linguistics.

Rahman, A. and Ng, V. (2011). Ensemble-based coreference resolution. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence - Volume Volume Three*, IJCAI'11, pages 1884–1889. AAAI Press.

Soon, W. M., Ng, H. T., and Lim, D. C. Y. (2001). A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544.

Uryupina, O., Poesio, M., Giuliano, C., and Tymoshenko, K. (2011). Using wikipedia for coreference resolution. In *Proceedings of the 24th Florida Artificial Intelligence Research Society Conference*.

Wilks, Y. (1975). A preferential pattern-matching semantics for natural language. *AIJ*, 6:53–74.

Yang, X. and Su, J. (2007). Coreference resolution using semantic relatedness information from automatically discovered patterns. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 528–535, Prague, Czech Republic. Association for Computational Linguistics.