Sanaphor++: Combining Deep Neural Networks with Semantics for Coreference Resolution

Julien Plu[◊], **Roman Prokofyev**[*], **Alberto Tonon**[*], **Philippe Cudré-Mauroux**[*], **Djellel Eddine Difallah**[*], **Raphaël Troncy**[◊], **Giuseppe Rizzo**[∞]

[◊]EURECOM, France {julien.plu,raphael.troncy}@eurecom.fr

[*]eXascale Infolab, University of Fribourg {roman.prokofyev,alberto.tonon,philippe.cudre-mauroux,djelleleddine.difallah}@unifr.ch

[∞]ISMB, Italy giuseppe.rizzo@ismb.it

Abstract

Coreference resolution has always been a challenging task in Natural Language Processing. Machine learning and semantic techniques have improved the state of the art over the time, though since a few years, the biggest step forward has been made using deep neural networks. In this paper, we describe Sanaphor++, which is an improvement of a top-level deep neural network system for coreference resolution—namely Stanford deep-coref—through the addition of semantic features. The goal of Sanaphor++ is to improve the clustering part of the coreference resolution in order to know if two clusters have to be merged or not once the pairs of mentions have been identified. We evaluate our model over the CoNLL 2012 Shared Task dataset and compare it with the state-of-the-art system (Stanford deep-coref) where we demonstrated an average gain of 1.13% of the average F1 score.

1. Introduction

The task of coreference resolution aims to identify which mentions in a text refer to the same real-world entity. Although coreference resolution is mostly studied as a clustering problem, it has also been studied as a Semantic Web problem by using Named Entity Recognition (NER) and Named Entity Linking (NEL) approaches. We define a Semantic Web problem as a problem where we exploit the semantics represented in a knowledge base that is published on the Web. Coreference resolution is an important aspect of text understanding and has numerous applications such as Entity Linking (see, for instance, the first two editions of the Open Knowledge Extraction challenge (Nuzzolese et al., 2015; Nuzzolese et al., 2016)). As an example of coreference resolution, in the following piece of text *"Emmanuel Macron is the new French president. He has been elected with a large majority"*, the mentions *Emmanuel Macron* and *He* will be disambiguated to the same entity, e.g. `http://dbpedia.org/resource/Emmanuel_Macron`. The task of coreference resolution is considered as one of the most challenging in Natural Language Processing (NLP). As an example of its challenging nature, in the following sentence *"Curie shared the 1903 Nobel Prize in Physics with her husband, Pierre Curie"*, it is clear that the clusters {*Pierre Curie*} and {*Curie, her*} are disjoint and do not refer to the same entity, but it is ambiguous whether the pair of mentions *Pierre Curie* and *Curie* are coreferent or not. Actually, without the mention *her*, one does not know if the mention *Curie* refers to *Pierre Curie* or *Marie Curie*.

The contributions of this work are:

1. A new approach that leverages both deep learning and Semantic Web techniques to solve an NLP problem;
2. A model that is integrated into a widely used NLP toolkit, namely the Stanford CoreNLP;
3. A thorough evaluation showing that our technique improves the results over the standard CoNLL2012 Shared Task dataset compared to the state-of-the-art methods by 1.13% in terms of the average F1 score.

2. Related Work

Stanford deep-coref (Clark and Manning, 2016b) takes inspiration from multiple existing methods and implements them using a deep neural network. As a starting point, the framework pre-trains a cluster-ranking model that takes advantage of entity-level information, with a neural mention-ranking model inspired from (Wiseman et al., 2015). In (Wiseman et al., 2016), the authors extend their previous mention-ranking model (Wiseman et al., 2015) by integrating entity-level information taken from the output of a recurrent neural network running over the candidate antecedent-clusters. Nevertheless, this is a simple change with respect to their original mention-ranking model, but not a true clustering model as the deep-coref cluster ranker is. Coreference resolution systems, such as joint inference (McCallum and Wellner, 2003; Poon and Domingos, 2008; Haghighi and Klein, 2010) and those that construct coreference clusters incrementally (Luo et al., 2004; Yang et al., 2008; Raghunathan et al., 2010), integrate entity-level information. Stanford deep-coref takes inspiration from the second kind of system and, particularly, from a combination of cluster-ranking (Rahman and Ng, 2011; Ma et al., 2014) and easy-first clustering strategies (Stoyanov and Eisner, 2012; Clark and Manning, 2015). While most of the previous systems used handcrafted features to integrate linguistic constraints, Stanford deep-coref, in addition, uses a learning-to-search approach inspired from (Chang et al., 2015) in order to learn from data the entity-level distributed representation. Although Stanford deep-coref provides very good results, it often has issues when resolving a coreference that involves entities. For example, in the sentence *"Marie Curie shared the 1903 Nobel Prize in Physics with her husband, Pierre Curie"*, it outputs the following cluster: {*Marie Curie, her, Pierre Curie*} and not the two clusters {*Marie Curie, her*}, {*Pierre Curie*} as expected.

Finally, there also exists a category of coreference resolution approaches that uses structural knowledge from external data sources (Strube and Ponzetto, 2006; Ponzetto and Strube, 2006; Bryl et al., 2010; Uryupina et al., 2011) such as Wikipedia, YAGO or WordNet. SANAPHOR (Prokofyev et al., 2015) belongs to that category and uses DBpdia and YAGO to help disambiguate the different entities that are involved into a coreference cluster. SANAPHOR is plugged to the output of the Stanford decoref (Lee et al., 2011) coreference resolution system, and improves its results by linking the different entities by deciding if a cluster has to be merged with another one, or if it has to be split based on the type, and the link (from YAGO or DBpedia) of the disambiguated entities.

3. SANAPHOR and Stanford deep-coref

This section describes both methods implemented by SANAPHOR and Stanford deep-coref in order to detail their inner-workings and, then, have a clear understanding of the implications in our approach.

3.1. SANAPHOR

SANAPHOR receives as input the clusters of coreferences generated by the Stanford decoref coreference resolution system. Each cluster is a set of mentions extracted from the original text. Each mention comes in the form of a string and, potentially, an associated headword (the most salient word in the mention). The mentions can be either entities, pronouns, or determinants. The resolution then proceeds in two steps by *i)* representing entities with their semantic counterparts whenever possible, and *ii)* optimizing the clusters by merging or splitting them according to their semantic representation. For the first step, it uses an entity linking component in order to link the mentions that might be an entity against DBpedia. It focus on precision rather than recall by doing a strict match over an inverted index over DBpedia, in order to be, as sure as possible, that the mention corresponds to an entity. It uses Wikipedia redirect pages in order to handle the entities that have multiple possible aliases. In case a mention corresponds to an ambiguous entity (i.e. entities associated to a Wikipedia disambiguation page directly), it is discarded. Once a mention is linked, it uses a mapping between DBPedia and YAGO ontologies provided by the TRank Hierarchy (Tonon et al., 2013) to map DBPedia types onto YAGO types.

For the second step, SANAPHOR makes use of the semantic features previously computed in order to optimize the clusters provided by Stanford decoref. The first part of this optimization is the spliting of each cluster by comparing each mention pairwise. A cluster is split following three different cases: *i)* the two mentions being compared have been properly linked against DBpedia and these DBpedia links are different, *ii)* the two mentions being compared have not been properly linked against DBpedia but successfuly typed against the YAGO ontology and those YAGO types are different, or *iii)* over the two mentions being compared, one has been properly linked against DBpedia and the other one against the YAGO ontology, and their YAGO type are different. Since a coreference cluster might also contain non-annotated mentions, they identify the words that belong exclusively to one of the mentions, then assign all the other mentions to one of the new clusters based on the overlap of their words with the exclusive words of each new cluster. After applying these heuristics, new clusters are created and then possibly merged. In the end, clusters are merged *i)* if they share at least one mention that refers to the same entity, or *ii)* if the type of two mentions share the same hierarchy.

3.2. Stanford Deep-coref

Stanford deep-coref basically consists in one big neural network where each component can be seen as three different sub-networks, and where each sub-network has a specific task. These three sub-networks are used to train the cluster-ranking model. The first sub-network is the mention-pair encoder that produces distributed representations for pairs of mentions by passing relevant features through a feed-forward neural network. The input of this sub-network is composed of multiple features that can be grouped in five categories: embedding features, mention features, document genre, distance features and string matching features. More details about the features are given in the original paper (Clark and Manning, 2016b).

This mention-pair encoder is used as a feature for the two other sub-networks: the mention-ranking model, and the cluster-pair encoder. The former scores pairs of mentions by passing their representations (from the mention-pair encoder) through a single neural network layer. The latter produces distributed representations for pairs of clusters by applying a pooling operation over the representations of relevant mention pairs (i.e. pairs where one mention is in each cluster). More precisely, it concatenates the results of max-pooling and average-pooling. The mention-ranking model is pre-trained before the final cluster-ranking model is fed. The final neural network, the cluster-ranking model, is trained with the output of the pre-trained mention-ranking model, and with the cluster-pair encoder. It is important to notice that the mentions are sorted in descending order according to their score from the mention-ranking model before they are used as features. The cluster-ranking model scores pairs of clusters and the scores it produces are leveraged to determine if candidates must be merged or not.

4. Sanaphor++

In this section, we detail the different steps of Sanaphor++ and, in particular, how we extend Stanford deep-coref and SANAPHOR into one single method. The first step is to create a new logic that extends SANAPHOR to take into account ambiguous and novel or emergent entities. The second step is to extend Stanford deep-coref to handle the new logic described in the first step.

4.1. Ambiguous entities

Sanaphor++ is able to handle coreference for ambiguous entities. In SANAPHOR, an ambiguous entity is a mention for which the basic entity linking method finds multiple candidates, such as *Paris* that might refers to: *Paris in France*, *Paris in Texas*, *Paris Hilton*, *Paris the movie* or even *Paris the band*. To be able to handle those cases, we switch to a more robust entity linking system, ADEL (Plu,

2016), when we encounter an ambiguous entity. ADEL is a hybrid entity linking approach being agnostic to the kind of text (e.g. newswire, tweets, subtitles), the knowledge base used to disambiguate the entities (e.g. DBpedia, Musicbrainz), the type of entities to extract (e.g. Person, Date, Numbers, Location), and the language of the docucment.

4.2. Novel Entities

The second stage is to handle novel or emergent entities, that is, entities that do not exist (yet) in the knowledge base being used, in our experiment, DBpedia. The case of a novel entity occurs when both the SANAPHOR original entity linking method and ADEL give an empty result. In that case, we rely on the Stanford NER annotator that is integrated into the mention extraction process detailed in 4.3.. In that process, mentions come with their NER type attached. The model used to attach these types has been trained with the CoNLL2012 Shared Task dataset that contains the types of mentions defined as entities. Finally, like for the linking that maps DBpedia types to YAGO types, we have defined a mapping that links the NER types to YAGO types. Nevertheless, these NER types are high level types such as *Person* or *Organization* and they do not give a lot of details with respect to their semantics. Despite this lack of information on the deep semantic of these entities, it is still worth to handle because it helps the system to split clusters where cases that mix, for example, *Person* and *Organization*. This is often the case when a company is named according to the family name of the owner. Once these two new cases are handled, we need to find a way to modify Stanford deep-coref in order to make it able to take into account this extended logic: handling ambiguous and novel entities.

4.3. New Mention-pair Ranking Model

The final step is to make Stanford deep-coref able to handle the new logic seen before, to have the final Sanaphor++ pipeline. Thus, after a thorough study of Stanford deep-coref, we have found that it shares two common parts with SANAPHOR: *i)* the cluster merging part and the cluster-ranking model, and *ii)* the cluster splitting part and the mention-ranking model.

The cluster-ranking model has a very complex structure and modifying it to take into account the merging features was impractical, such that we decided to leave this as future work. Therefore, we decided to create a new three-dimensional vector, one dimension for each semantic feature: *i)* if the entities of the two mentions are the same, *ii)* if the type of the two mentions are the same, and *iii)* if the type of one mention is included into the hierarchy of the other one. They are then concatenated with the original vector in the input layer of the mention-pair encoder and then to the mention-pair ranking model.

The extraction of the new features must be implemented in Stanford deep-coref. To do so, we have to put the new process to extract the semantic features directly into the one that takes care of the Stanford deep-coref features. The CoNLL2012 Shared Task dataset is composed of three datasets: train, dev and test. The extraction of the mentions for the train and the set {dev, test} datasets is done differently. About the training dataset, the mentions are directly extracted from the annotated gold standard, whereas for the set {dev, test} datasets, the mentions are extracted using the Stanford mention annotator (see (Clark and Manning, 2015)), where its goal is basically to extract mentions from the text. It means that the annotations in the {dev, test} datasets are not used at all, because they are the evaluation datasets. Once the mentions have been extracted, either via the dataset or via the Stanford mention annotator, they are all put into different feature computation process, one for each kind of feature: embedding, mention, document, distance, string matching and now the semantic feature, i.e. the 3-dimensional vector. In order for the neural network to take into account these new features, we must modify its training objective and more precisely the mistake-specific cost function. The new mistake-specific cost function is represented in Equation 1.

$$\Delta(a, m_i) = \begin{cases} \alpha_{FN} & \text{if} \quad a = NA \wedge \Gamma(m_i) \neq \{NA\} \\ \alpha_{FA} & \text{if} \quad a \neq NA \wedge \Gamma(m_i) = \{NA\} \\ \alpha_{WL} & \text{if} \quad a \neq NA \wedge a \notin \Gamma(m_i) \\ 0 & \text{if} \quad a \in \Gamma(m_i) \vee e_a \in \Omega(m_i) \vee t_a \in \Phi(m_i) \vee T(e_a) \in \Phi(m_i) \end{cases} \tag{1}$$

Where NA indicates an empty antecedent; $\Gamma(m_i)$ denotes the set of true antecedents of m_i (i.e. mentions preceding m_i that are coreferent with it or $\{NA\}$ if m_i has no antecedent); a is a possible antecedent of m_i; e_a denotes the entity of a; $\Omega(m_i)$ denotes the set of the entities of the true antecedents of m_i; t_a denotes the type of a; $\Phi(m_i)$ denotes the set of the types of the true antecedents of m_i (i.e. this set includes also all the types that belongs to the hierarchy of a type); $T(e_a)$ denotes the type of the entity of a. The goal of this new mistake-specific cost function, is to make understand to the network if two mentions are likely to be compatible together and being a pair or not. We detail the clauses of this new function:

1. The first clause stands for the *false new* antecedents. It means that if the antecedent is an empty antecedent and if the set of true antecedents is not equal to empty, then this antecedent is likely to be a wrong pair;

2. The second clause stands for the *false anaphoric* antecedents. It means that if an antecedent is not an empty antecedent and if the set of true antecedents is equal to empty, then this antecedent is likely to be a wrong pair;

3. The third clause stands for the *wrong link* antecedents. It means that if an antecedent is not empty and this antecedent does not belong to the set of true antecedents, then this antecedent is likely to be a wrong pair;

4. The fourth clause stands for a correct coreferent decision. It means that if an antecedent is in the set of the true antecedents, or if the entity of the antecedent is in the set of the true entities, or if the type of the antecedent is in the set of the true types, or if one type belonging to the hierarchy of type of the antecedent is in the set of the true types, then this antecedent is likely to be a good pair.

	MUC			B^3			CEAF-E			
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**	**Avg F1**
Sanaphor++	**65.81**	**74.65**	**69.95**	**58.84**	**62.37**	**60.55**	**52.47**	**58.64**	**55.39**	**61.96**
Stanford deep-coref	64.3	72.93	68.34	57.46	60.91	59.14	52.11	58.24	55	60.83

Table 1: Sanaphor++ and Stanford deep-coref results

	BLANC		
	Precision	**Recall**	**F1**
Sanaphor++	**65.88**	**54.97**	**59.93**
Sanaphor	60.63	55.16	57.11
Stanford decoref	60.61	55.07	57.04
Stanford deep-coref	61.48	50.98	55.7

Table 2: Sanaphor++, Sanaphor, Stanford decoref and Stanford deep-coref BLANC results

The error penalties α_{FN}, α_{FA} and α_{WL} are hyperparameters that must be defined at the beginning of the training. We keep the values set by the original network respectively (α_{FN}, α_{FA}, α_{WL}) = (0.8, 0.4, 1.0).
Finally, we run the training of this new model with the same hyper parameters than the original Stanford deep-coref. A negative effect of adding these new features into the network is that it increases the training time from 3 days to 5 days. Once the training is done, it is possible to save the model, with the help of scripts provided by the author of Stanford deep-coref, into a format that can directly be used by Stanford CoreNLP toolkit (Manning et al., 2014).

5. Evaluation

5.1. Metrics

Many metrics have been proposed to evaluate the performance of coreference resolution systems, such as *MUC* (van Deemter and Kibble, 2000), B^3 (Bagga and Baldwin, 1998), or *CEAF* (Luo, 2005), including multiple variants of them. *MUC* counts the minimum number of links between mentions to be inserted or deleted when mapping a system response to a gold standard key set. B^3 overcomes the shortcomings of the *MUC* score, instead of looking at the links, it computes precision and recall for all mentions in the document, which are then combined to produce the final precision and recall numbers for the entire output. *CEAF* is calculated based on the best mapping between coreference expressions or entities, thus results in two types of *CEAF*: expression-based (*CEAF-M*) and entity-based (*CEAF-E*). Finally, *Avg F1* is an average of the F1 scores of the three previous ones. Afterward, to have a full comparison among the different systems we will use the most recent metric, BLANC (Recasens and Hovy, 2011).

5.2. Experimental Results and Settings

We evaluate our system on standard datasets from the CoNLL-2012 Shared Task on Coreference Resolution (Pradhan et al., 2012). We compare Sanaphor++ with the most recent version of Stanford deep-coref based on a deep reinforcement learning (Clark and Manning, 2016a) in Table 1. Finally, we compare Sanaphor++, Sanaphor, Stanford deep-coref and Stanford decoref with the BLANC score in Table 2.

The Sanaphor++ model has been exported into a Stanford CoreNLP Framework compliant format, in order to be interoperable and foster the usage through the Stanford CoreNLP Framework. We have run the Sanaphor++ model and the Stanford deep-coref model over the CoNLL2012 test dataset. The provided Stanford deep-coref model in the Stanford CoreNLP Framework is designed for *real-world* usage and gets lower scores than the ones provided in the corresponding paper (Clark and Manning, 2016a), because it does not take into account the CoNLL specific features such as speaker or document genre. For this reason, we have also designed our model to discard those features positioning our experimental setup in the worst experimental setup conditions.

As shown in Table 1, the new semantic logic brought by Sanaphor++ allows to compute a better mention-pair score, as all the scores are improved compared to Stanford deep-coref. While the results are promising, the new logic provides also wrong clusters, such as in the piece of text: *How does {the copyright thing} work on a Live365 stream? [...] About {the JW thing}, I work with a couple of them...*, the mentions *the copyright thing* and *the JW thing* have a high pair score since they share common surface forms and the same type, both are novel entities and have been typed to *THING*. The case also appears with ambiguous entities, such as in the piece of text: *It is she who asks if {Michael} and John could come too. [...] or that can't be tied to {Michael Jackson}*, the mentions *Michael* and *Michael Jackson* have a high pair score, because they share a common surface form, the same type and the same link. The first *Michael* refers to the character in Peter Pan which is completely different from the artist *Michael Jackson*. The problem is that *Michael* is highly ambiguous in this sentence.

Results from SANAPHOR have not been reported because it has been built over the Stanford decoref that uses the CoNLL specific features and then cannot be compared with the actual models of Sanaphor++ and Stanford deep-coref.

6. Conclusion and Future Work

We have presented a coreference resolution system that is able to capture the semantic of the entities into a single method based on two different systems that use divergent methods. The newly created model can be used through the Stanford CoreNLP toolkit as it uses the same structure as Stanford deep-coref. Finally, the results have shown improvements over the CoNLL 2012 Shared Task compared to Stanford deep-coref of 1.13% in terms of the average F1 score.

As future work, we want to integrate the merging logic from SANAPHOR into the cluster-ranking model in order to improve the clustering merge decision of this model. We will also investigate the impact on the training time and we will envisage possibilities to reduce it. We would like to evaluate how much the used entity linking system (here, ADEL) impacts the results by using and comparing with other entity linking systems. Finally, our last goal is to have models in other languages such as French.

7. Acknowledgments

We would like to thank Kevin Clark for his help in understanding and using Stanford deep-coref. This work has been partially supported by the French National Research Agency (ANR) within the ASRAEL project (ANR-15-CE23-0018), the French Fonds Unique Interministériel (FUI) within the NexGen-TV project and the innovation activities 3cixty (14523) and PasTime (17164) of EIT Digital (`https://www.eitdigital.eu`).

8. Bibliographical References

Bagga, A. and Baldwin, B. (1998). Algorithms for scoring coreference chains. In *The first international conference on language resources and evaluation workshop on linguistics coreference*.

Bryl, V., Giuliano, C., Serafini, L., and Tymoshenko, K. (2010). Using background knowledge to support coreference resolution. In *Proceedings of the 2010 Conference on ECAI 2010: 19th European Conference on Artificial Intelligence*.

Chang, K., He, H., III, H. D., and Langford, J. (2015). Learning to search for dependencies. *CoRR*.

Clark, K. and Manning, C. (2015). Entity-centric coreference resolution with model stacking. In *Association for Computational Linguistics (ACL)*.

Clark, K. and Manning, C. (2016a). Deep reinforcement learning for mention-ranking coreference models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*.

Clark, K. and Manning, C. (2016b). Improving coreference resolution by learning entity-level distributed representations. *arXiv preprint arXiv:1606.01323*.

Haghighi, A. and Klein, D. (2010). Coreference resolution in a modular, entity-centered model. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the 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 *Conference on Natural Language Learning (CoNLL) Shared Task*.

Luo, X., Ittycheriah, A., Jing, H., Kambhatla, N., and Roukos, S. (2004). A mention-synchronous coreference resolution algorithm based on the bell tree. In *ACL*.

Luo, X. (2005). On coreference resolution performance metrics. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*.

Ma, C., Doppa, J., Orr, J., Mannem, P., Fern, X., Dietterich, T., and Tadepalli, P. (2014). Prune-and-score: Learning for greedy coreference resolution. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*.

Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., and McClosky, D. (2014). The stanford corenlp natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*.

McCallum, A. and Wellner, B. (2003). Toward conditional models of identity uncertainty with application to proper noun coreference. In *Proceedings of the IJCAI-2003 Workshop on Information Integration on the Web*.

Nuzzolese, A., Gentile, A., Presutti, V., Gangemi, A., Garigliotti, D., and Navigli, R. (2015). The First Open Knowledge Extraction Challenge. In *12^{th} European Semantic Web Conference (ESWC)*.

Nuzzolese, A., Gentile, A., Presutti, V., Gangemi, A., Meusel, R., and Paulheim, H. (2016). The Second Open Knowledge Extraction Challenge. In *13^{th} European Semantic Web Conference (ESWC)*.

Plu, J. (2016). Knowledge Extraction in Web Media: At The Frontier of NLP, Machine Learning and Semantics. In *25^{th} World Wide Web Conference (WWW), PhD Symposium*.

Ponzetto, S. and Strube, M. (2006). Exploiting semantic role labeling, wordnet and wikipedia for coreference resolution. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*.

Poon, H. and Domingos, P. (2008). Joint unsupervised coreference resolution with markov logic. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Pradhan, S., Moschitti, A., Xue, N., Uryupina, O., and Zhang, Y. (2012). Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*.

Prokofyev, R., Tonon, A., Luggen, M., Vouilloz, L., Difallah, D., and Cudré-Mauroux, P. (2015). Sanaphor: Ontology-based coreference resolution. In *International Semantic Web Conference (1)*.

Raghunathan, K., Lee, H., Rangarajan, S., Chambers, N., Surdeanu, M., Jurafsky, D., and Manning, C. (2010). A

multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*.

Rahman, A. and Ng, V. (2011). Narrowing the modeling gap: A cluster-ranking approach to coreference resolution. *J. Artif. Intell. Res. (JAIR)*.

Recasens, M. and Hovy, E. (2011). Blanc: Implementing the rand index for coreference evaluation. *Natural Language Engineering*.

Stoyanov, V. and Eisner, J. (2012). Easy-first coreference resolution. In *COLING 2012, 24th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, 8-15 December 2012, Mumbai, India*.

Strube, M. and Ponzetto, S. (2006). Wikirelate! computing semantic relatedness using wikipedia. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2*.

Tonon, A., Catasta, M., Demartini, G., Cudré-Mauroux, P., and Aberer, K. (2013). Trank: Ranking entity types using the web of data. In *International Semantic Web Conference (1)*.

Uryupina, O., Poesio, M., Giuliano, C., and Tymoshenko, K. (2011). Disambiguation and filtering methods in using web knowledge for coreference resolution. In *FLAIRS Conference*.

van Deemter, K. and Kibble, R. (2000). On coreferring: Coreference in muc and related annotation schemes. *Computational Linguistics*.

Wiseman, S., Rush, A., Shieber, S., and Weston, J. (2015). Learning anaphoricity and antecedent ranking features for coreference resolution. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*.

Wiseman, S., Rush, A., and Shieber, S. (2016). Learning global features for coreference resolution. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*.

Yang, X., Su, J., Lang, J., Tan, C. L., Liu, T., and Li, S. (2008). An entity-mention model for coreference resolution with inductive logic programming. In *ACL*.