A Multi-classifier Approach to support Coreference Resolution in a Vector Space Model

Ana Zelaia
UPV/EHU
Manuel Lardizabal, 1
Donostia, 20018
Basque Country, Spain
ana.zelaia@ehu.eus

Olatz Arregi
UPV/EHU
Manuel Lardizabal, 1
Donostia, 20018
Basque Country, Spain
olatz.arregi@ehu.eus

Basilio Sierra
UPV/EHU
Manuel Lardizabal, 1
Donostia, 20018
Basque Country, Spain
b.sierra@ehu.eus

Abstract

In this paper a different machine learning approach is presented to deal with the coreference resolution task. This approach consists of a multi-classifier system that classifies mention-pairs in a reduced dimensional vector space. The vector representation for mention-pairs is generated using a rich set of linguistic features. The SVD technique is used to generate the reduced dimensional vector space.

The approach is applied to the OntoNotes v4.0 Release Corpus for the column-format files used in CONLL-2011 coreference resolution shared task. The results obtained show that the reduced dimensional representation obtained by SVD is very adequate to appropriately classify mention-pair vectors. Moreover, we can state that the multi-classifier plays an important role in improving the results.

1 Introduction

Coreference resolution deals with the problem of finding all expressions that refer to the same entity in a text (Mitkov, 2002). It is an important subtask in Natural Language Processing that require natural language understanding, and hence, it is considered to be difficult.

A coreference resolution system has to automatically identify the mentions of entities in text and link the corefering mentions (the ones that refer to the same entity) to form coreference chains. Systems are expected to perform both, mention detection and coreference resolution.

Preliminary researches proposed heuristic approaches to the task, but thanks to the annotated coreference corpora made available in the last years and the progress achieved in statistical NLP methods, machine learning approaches to the coreference resolution task are being proposed. (Ng, 2010) presents an interesting survey of the progress in coreference resolution.

In this paper we present a different machine learning approach to deal with the coreference resolution task. Given a corpus with annotated mentions, the multi-classifier system we present classifies mention-pairs in a reduced dimensional vector space. We use the typical mention-pair model, where each pair of mentions is represented by a rich set of linguistic features; positive instances correspond to mention-pairs that corefer. Coreference resolution is tackled as a binary classification problem (Soon et al., 2001) in this paper; the subsequent linking of mentions into coreference chains is not considered. In fact, the aim of our experiment is to measure to what extent working with feature vectors in a reduced dimensional vector space and applying a multi-classifier system helps to determine the coreference of mention-pairs. To the best of our knowledge, there are no approaches to the coreference resolution task which make use of multi-classifier systems to classify mention-pairs in a reduced dimensional vector space.

This paper gives a brief description of our approach to deal with the problem of identifying whether two mentions corefer and shows the results obtained. Section 2 presents related work. In Section 3 our approach is presented. Section 4 presents the case study, where details about the dataset used in the experiments and the preprocessing applied are

Proceedings of NAACL-HLT 2015, pages 17–24,
Denver, Colorado, May 31 – June 5, 2015. ©2015 Association for Computational Linguistics

given. In Section 5 the experimental setup is briefly introduced. The experimental results are presented and discussed in Section 6, and finally, Section 7 contains some conclusions and comments on future work.

2 Related Work

Much attention has been paid to the problem of coreference resolution in the past two decades. Conferences specifically focusing coreference resolution have been organized since 1995. The sixth and seventh Message Understanding Conferences (MUC-6, 1995; MUC-7, 1998) included a specific task on coreference resolution. The Automatic Context Extraction (ACE) Program focused on identifying certain types of relations between a predefined set of entities (Doddington et al., 2004) while the Anaphora Resolution Exercise (ARE) involved anaphora resolution and NP coreference resolution (Orăsan et al., 2008).

More recently, SemEval-2010 Task 1 was dedicated to coreference resolution in multiple languages. One year later, in the CoNLL-2011 shared task (Pradhan et al., 2011), participants had to model unrestricted coreference in the English-language OntoNotes corpora and CoNLL-2012 Shared Task (Pradhan et al., 2012) involved predicting coreference in three languages: English, Chinese and Arabic.

Recent work on coreference resolution has been largely dominated by machine learning approaches. In the SemEval-2010 task on Coreference Resolution in Multiple Languages (Recasens et al., 2010), most of the systems were based on these techniques (Broscheit et al., 2010; Uryupina, 2010; Kobdani et al., 2010). The same occurs at CoNLL-2011, where (Chang et al., 2011; Björkelund et al., 2011; dos Santos et al., 2011) were based on machine learning techniques. The advantage of these approaches is that there are many open-source platforms for machine learning and machine learning based coreference systems such as BART (Versley et al., 2008), the Illinois Coreference Package (Bengtson et al., 2008) or the Stanford CoreNLP (Manning et al., 2014), among others.

Nevertheless, rule-based systems have also been applied successfully (Lappin et al., 1994; Mitkov, 1998; Lee et al., 2013). The authors of this last system propose a coreference resolution system that is an incremental extension of the multi-pass sieve system proposed by (Raghunathan et al., 2010). This system is shifting from the supervised learning setting to an unsupervised setting, and obtained the best result in the CoNLL-2011 Shared Task.

Some very interesting uses of vector space models for the coreference resolution task can be found in the literature. (Nilsson et al., 2009) investigate the effect of using vector space models as an approximation of the kind of lexico-semantic and commonsense knowledge needed for coreference resolution for Swedish texts. They also work with reduced dimensional vector spaces and obtain encouraging results. In an attempt to increase the performance of a coreference resolution engine, (Bryl et al., 2010) make use of structured semantic knowledge available in the web. One of the strategies they adopt is to apply the SVD to Wikipedia articles and classify mentions in a reduced dimensional vector space.

3 Proposed Approach

The approach we present consists of a multi-classifier system which classifies mention-pairs in a reduced dimensional vector space. This multi-classifier is composed of several k-NN classifiers. A set of linguistic features is used to generate the vector representations for the mention-pairs. The training dataset is used to create a reduced dimensional vector space using the SVD technique. Mention-pairs in the training, development and test sets are represented using the same linguistic features and projected onto the reduced dimensional space.

The classification process is performed in the reduced dimensional space. To create the multi-classifier, we apply random subsampling and obtain training datasets $TD_1, \ldots, TD_i$ for the reduced dimensional space. Given a testing case q, the k-NN classifier makes a label prediction c^i based on each one of the training datasets TD_i, and predictions $c^1, \ldots c^i$ are combined to obtain the final prediction c_j using a Bayesian voting scheme. It is a binary classification system where the final prediction c_j may be positive (mentions tested corefer) or negative (mentions do not corefer). Figure 1 shows an illustration of the fundamental steps of the experi-

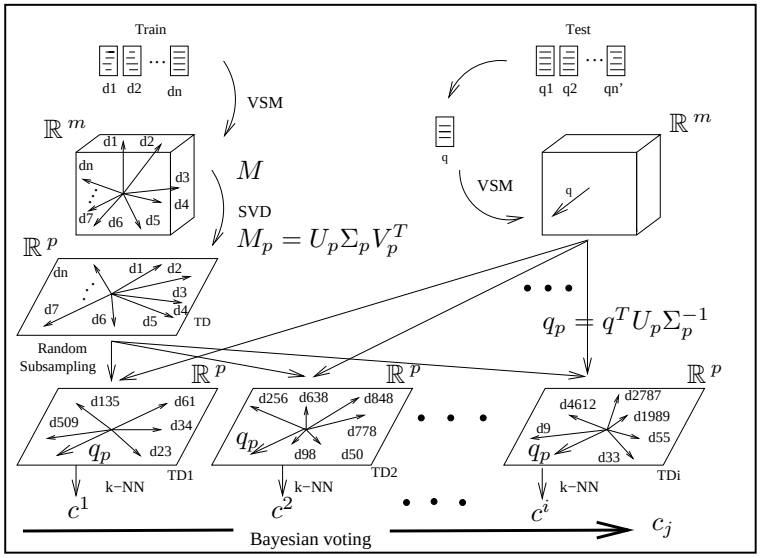

Figure 1: Fundamental steps of the proposed approach. $\mathbb{R}^m$ is the original vector space, $\mathbb{R}^p$ is the reduced dimensional space where vectors are projected. The multi-classifier is composed of several k-NN classifiers. c_j is the final classification label for testing case q.

ment.

In the rest of this section, details about the SVD dimensionality reduction technique, the k-NN classification algorithm, the combination of classifiers and the evaluation measures used are briefly reviewed.

3.1 The SVD Dimensionality Reduction

The classical Vector Space Model (VSM) has been successfully employed to represent documents in text categorization and Information Retrieval tasks. Latent Semantic Indexing (LSI) [1] (Deerwester et al., 1990) is a variant of the VSM in which documents are represented in a lower dimensional vector space created from a training dataset. To create such a lower dimensional vector space, LSI generates a term-document matrix M and computes its SVD matrix decomposition, $M = U\Sigma V^T$. As a result, r singular values are obtained, and terms and documents are mapped to the r-dimensional vector space. By reducing the r to p, a reduced dimensional space is created, the p-dimensional space onto which vectors are projected. This reduced dimensional space is used for classification purposes, and the cosine similarity is usually used to measure the similarity between vectors (Berry et al., 1995).

[1] http://lsi.research.telcordia.com,http://www.cs.utk.edu/∼lsi

It has been proved that computing the similarity of vectors in the reduced dimensional space gives better results than working in the original space. In fact, LSI is said to be able to capture the latent relationships among words in documents thanks to the word co-occurrence analysis performed by the SVD technique, and therefore, cluster semantically terms and documents. This powerful technique is being used to better capture the semantics of texts in applications such as Information Retrieval (Berry et al., 2005). LSI is referred to as Latent Semantic Analysis (LSA) when it is used as a model of the acquisition, induction and representation of language and the focus is on the analysis of texts (Dumais, 2004).

For the sake of the coreference resolution task, each document corresponds to a mention-pair, and words in each document are the linguistic feature values for the associated mention-pair. Section 4.2 gives details about the linguistic features used to represent each mention-pair. Matrix M is constructed for the selected feature values (terms) and all mention-pairs considered (documents). The SVD decomposition is computed and the p-dimensional reduced space is created. We use U as the reduced dimensional representation, and compute the coordinates to project mention-pair vectors onto the reduced space and compare them.

3.2 The k-NN classification algorithm

k-NN is a distance based classification approach. According to this approach, given an arbitrary testing case, the k-NN classifier ranks its nearest neighbors among the training cases, and uses the class of the k top-ranking neighbors to do the prediction for the testing case being analyzed (Dasarathy, 1991).

In our experiments, parameter k is set to 3. Given a testing mention-pair vector, the 3-NN classifier is used to find the three nearest neighbor mention-pair vectors in the reduced dimensional vector space. The cosine is used to measure vector similarity and find the nearests.

We also consider the k-NN classifier provided with the Weka package (Hall et al., 2009; Aha et al., 1991). We use it to obtain a honest comparison for the results.

3.3 Multi-classifier systems

The combination of multiple classifiers has been intensively studied with the aim of improving the accuracy of individual components (Ho et al., 1994). A widely used technique to implement this approach is *bagging* (Breiman, 1996), where a set of training datasets TD_i is generated by selecting n training cases drawn randomly with replacement from the original training dataset TD of n cases. When a set of $n_1 < n$ training cases is chosen from the original training collection, the bagging is said to be applied by random subsampling. In fact, this is the approach used in our work and the n_1 parameter is set to be 60% of the total number of training cases n. The proportion of positive and negative cases in the training dataset TD is preserved in the different TD_i datasets generated.

According to the random subsampling, given a testing case q, the classifier makes a label prediction c^i based on each one of the training datasets TD_i. Label predictions c^i may be either positive or negative. One way to combine the predictions is by Bayesian voting (Dietterich, 1998), where a confidence value $\text{cv}^i_{c_j}$ is calculated for each training dataset TD_i and label to be predicted. These confidence values are calculated based on the training collection. Confidence values are summed by label; the label c_j that gets the highest value is finally proposed as a prediction for the testing case q.

3.4 Evaluation measures

The approach presented in this paper is a binary classification system where the final prediction c_j may be positive (mentions tested corefer) or negative (mentions do not corefer). There are many metrics that can be used to measure the performance of a classifier. In binary classification problems precision and recall are very widely used. Precision (Prec) is the number of correct positive results divided by the number of all positive results, and recall (Rec) is the number of correct positive results divided by the number of positive results that should have been returned.

In general, there is a trade-off between precision and recall. Thus, a classifier is usually evaluated by means of a measure which combines them. The F_1-score can be interpreted as a weighted average of precision and recall; it reaches its best value at 1 and worst score at 0.

$$F_1 = \frac{2 \cdot \text{Prec} \cdot \text{Rec}}{\text{Prec} + \text{Rec}}$$

Accuracy is also used as a statistical measure of performance in binary classification tasks. Accuracy is the proportion of true results (both true positives and true negatives) among the total number of cases tested.

4 Case study

This section briefly reviews the dataset used in the experiments and the preprocessing applied.

4.1 Dataset

The OntoNotes v4.0 Release Corpus is used in the experiments[2]. It provides a large-scale multi-genre corpus with multiple layers of annotation (syntactic, semantic and discourse information) which also include coreference tags. A nice description of the coreference annotation in OntoNotes can be found in (Pradhan et al., 2007a) and (Pradhan et al., 2007b).

Although OntoNotes is a multi-lingual resource for English, Chinese and Arabic, for the scope of this paper, we just look at the English portion. We

[2]Downloaded from Linguistic Data Consortium (LDC) Catalog No.: LDC2011T03, https://catalog.ldc.upenn.edu/LDC2011T03. For more information, see OntoNotesRelease4.0.pdf and coreference/english-coref.pdf files in LDC directory

use English texts for five different genres or types of sources: broadcast conversations (BC), broadcast news (BN), magazine articles (MZ), newswires (NW) and web data (WB).

The English language portion of the OntoNotes v4.0 Release Corpus was used in the CONLL-2011 coreference resolution Shared task[3]. The task was to automatically identify mentions of entities and events in text and to link the corefering mentions together to form mention chains (Pradhan et al., 2011; Pradhan et al., 2012). Since OntoNotes coreference data spans multiple genre, the task organizers created a test set spanning all the genres. The training, development and test files were downloaded from the CONLL-2011 website, and the *_conll files were generated from each corresponding *_skel files using the scripts made available by the organizers.

The *_conll files contain information in a tabular structure where the last column contains coreference chain information. Two types of *_conll files may be generated, depending on how the annotation was generated; *gold_conll files were hand-annotated and adjudicated quality, whereas annotations in *auto_conll files were produced using a combination of automatic tools. *gold_conll files are used in the experiments presented in this paper.

4.2 Preprocessing

In order to obtain the vector representation for each pair of mentions, we used the features defined by (Sapena et al., 2011). The 127 binary features they define are related to distance, position, lexical information, morphological information, syntactic dependencies and semantic features. The authors developed a coreference resolution system called RelaxCor[4] and participated in the CoNLL-2011 shared task obtaining very good results. It is an open source software available for anyone who wishes to use it.

RelaxCor is a constraint-based hypergraph partitioning approach to coreference resolution, solved by relaxation labeling. It generates feature vectors for all mention-pairs in the *_conll files as part of the system and uses them to solve the task. We decided to use the perl scripts distributed by the authors and generate the positive and negative feature vectors for all *_conll files. These feature vectors consist of binary values for the 127 binary features and a label: a positive label (+) indicates that the feature vector corresponds to a corefering mention-pair, whereas a negative label (-) indicates that the two mentions do not corefer.

Note that each mention in a file is combined with all the rest of mentions in the same file to form mention-pairs and consequently, a very large amount of negative examples is generated, specially for large files. We decided to reduce the amount of negative examples, in a similar manner as (Sapena et al., 2011) and therefore, negative examples with more than five feature values different from any positive example in each file were eliminated. In order to obtain the training, development and test corpora for the 5 genres, we brought together the examples generated from files of the same split and genre. We removed contradictions (negative examples with identical feature values as a positive example) and examples that appeared more than once in the same corpus. We noticed that the size of the corpora was too large for some of the genres; the broadcast conversations (BC) genre training corpus for instance had more than 4 million examples. We decided to reduce all corpora to a reasonable size to compute the SVD.

	BC	BN	MZ	NW	WB
Train (+)	20206	44515	25103	31034	24501
Train (-)	26623	55921	23568	50687	26948
Dev (+)	4056	5920	3873	4776	3531
Dev (-)	5831	8609	4864	7615	5732
Test (+)	29363	10771	3918	15857	17146
Test (-)	16591	12480	3209	15759	5505

Table 1: Size of corpora used in the experiments.

Table 1. gives detailed information about the number of positive and negative mention-pairs in the training, development and test corpora used in the experiments. A matrix is constructed for each of the training corpus. Feature values that appear at least once in the corpus are selected as terms. Even though theoretically we could have a maximum number of 254 different terms in each training corpus (127×2, because the 127 features are binary), the real value is between 227 and 230. The

[3]http://conll.cemantix.org/2011/introduction.html

[4]http://nlp.lsi.upc.edu/relaxcor/

sizes of the matrices created are given by the number of terms and documents (sum of (+) and (-) examples in the training corpus) and can be seen in Table 2.

	BC	BN	MZ	NW	WB
Terms	227	230	227	229	230
Docs	46829	100436	48671	81721	51449

Table 2: Size of term-document matrices M.

5 Experimental Setup

To optimize the behaviour of the multi-classifier system, the number of TD_i training datasets is adjusted in a parameter tuning phase. This optimization process is performed in an independent way for each of the genres because the five genres correspond to texts coming from different sources and may have very different characteristics (Uryupina et al., 2012). Therefore, we treat them as five different classification problems.

The five development corpora are used to adjust parameter i (the amount of TD_i training datasets). We experimented with the following values for i: 5, 10, 20, 30, 40, 50, 60, 70, 80. Table 3 shows the optimal values obtained for each genre. This means that testing cases for the BC genre, for instance, are classified by a multi-classifier formed by 60 k-NN classifiers, after having generated 60 TD_i training datasets from the original TD.

	BC	BN	MZ	NW	WB
Optimal i	60	30	50	20	40
Singular Values	83	86	85	86	87

Table 3: Optimal values for the number of TD_i datasets. Number of singular values computed by SVD

Two different dimensional representations are experimented for mention-pair vectors. On the one hand, we consider mention-pair vectors represented in the original 127 dimensions. On the other hand, the SVD-computed dimensional vector representation is being experimented. Table 3 shows the number of singular values (dimensions) computed by SVD for each of the genres.

6 Experimental Results

Three experiments were carried out in the test phase using the optimal values for parameter i and the two different representations for mention-pair vectors. Table 4 shows the results obtained for each of the experiments: accuracy values in a first row (Acc.) and F_1-scores in a second (F_1).

In a first experiment (Exp.1), the Weka 3-NN classifier is applied to classify testing cases represented in the original 127 dimensional space. The same 3-NN classifier is applied in a second experiment (Exp.2), but training and testing cases are represented using the dimensions computed by SVD (see Singular Values in Table 3). In a last experiment (Exp.3), our approach is applied and a multi-classifier system classifies testing vectors in the same SVD-dimensional vector space as in the previous experiment. The multi-classifier is generated according to the optimal values for parameter i in each genre.

Exp.	BC	BN	MZ	NW	WB	Mean
1 Acc.	**0.719**	0.704	**0.706**	0.707	0.669	0.701
F_1	**0.762**	0.686	**0.731**	0.679	0.744	0.720
2 Acc.	0.672	0.725	0.662	0.725	**0.783**	0.713
F_1	0.742	0.71	0.717	0.715	**0.85**	**0.747**
3 Acc.	0.669	**0.755**	0.661	**0.742**	0.776	**0.721**
F_1	0.739	**0.728**	0.707	**0.716**	0.841	0.746

Table 4: Accuracy and F_1-score for the test corpora. Exp.1: 3-NN and 127 dimensions. Exp.2: 3-NN and SVD dimensions. Exp.3: multi-classifier and SVD dimensions. Last column: mean values

The results shown in bold in the first part of Table 4 are the best for each genre. Note that the two performance measures computed (accuracy and F_1-score) are very correlated in the five cases. Taking into account that the proportion of positive and negative examples varies from genre to genre, this correlation gives consistency to the interpretation of the results obtained.

The best results for BC and MZ genres are obtained in the first experiment, applying the 3-NN classifier to the 127 dimensional vectors (Exp.1, F_1-scores: 0.762 and 0.731, respectively). For the rest of the genres, the best results are obtained for the SVD-dimensional vectors. An F_1-score of 0.85 is

obtained for the WB genre in the second experiment (Exp.2). The approach proposed in this paper (Exp.3) achieves the best results for two out of the five genre, with an F_1-score of 0.728 for BN and 0.716 for NW.

The last column in Table 4 shows the mean accuracy and F_1-scores obtained in each experiment, taking into account the five genres as a whole (the best are shown in bold). The best mean F_1-score is obtained in Experiment 2, where vectors are classified in the SVD-dimensional vector space. In fact, this result is very closely followed by the one obtained in Experiment 3 with our approach, (mean F_1-scores: 0.747 and 0.746, respectively). The best mean accuracy is obtained when our approach is applied (mean accuracy: 0.721). This good results seem to suggest that the dimensions computed by the SVD technique are very appropriate to represent mention-pairs and classify them. Moreover, the use of the multi-classifier system gets to achieve even better results, outperforming the ones obtained by the other classification systems.

7 Conclusions and Future Work

In this paper a different machine learning approach to deal with the coreference resolution task is presented: a multi-classifier system that classifies mention-pairs in a reduced dimensional vector space created by applying the SVD technique. The results obtained for the OntoNotes corpus are very good, outperforming the ones obtained by other classification systems for some genres. Moreover, when mean results per experiment are considered, the SVD generated dimensional representation always achieves the best results, which seems to suggest that it is a very robust and suitable representation for coreference mention-pairs.

As future work, we plan to experiment with some other kind of multi-classifer systems and basic classifiers such as SVM. It is important to note that the approach may be applied to corpora in other languages as well.

Acknowledgments

We gratefully acknowledge Emili Sapena, who helped us solve some file format problems. This work was supported by the University of the Basque Country, UPV/EHU, ikerketaren arloko errektoreordetza / Vicerrectorado de Investigación.

References

David W. Aha, Dennis Kibler, and Marc K. Albert. 1991. *Machine Learning*, volume 6(1).

Eric Bengtson and Dan Roth. 2008. *Understanding the value of features for coreference resolution*. Proceedings of the EMNLP '08: 294–303.

Michael W. Berry, Susan T. Dumais, and Gavin W. O'Brien. 1995. *Using Linear Algebra for Intelligent Information Retrieval*, volume 37(4):573–595. SIAM.

Michael W. Berry and Murray Browne. 2005. *Understanding Search Engines: Mathematical Modeling and Text Retrieval*. SIAM.

Anders Bjrkelund and Pierre Nugues. 2011. *Exploring lexicalized features for coreference resolution*. Proceedings of the CONLL'11 Shared Task, 45–50.

Leo Breiman. 1996. *Bagging Predictors*. Machine Learning, volume 24(2):123–140.

Samuel Broscheit, Massimo Poesio, Simone Paolo Ponzetto, Kepa Joseba Rodriguez, Lorenza Romano, Olga Uryupina, Yannick Versley, and Roberto Zanoli. 2010. *BART: A multilingual anaphora resolution system*. Proceedings of the SemEval-2010, pages 104–107.

Volha Bryl, Claudio Giuliano, Luciano Serafini, and Kateryna Tymoshenko. 2010. *Using Background Knowledge to Support Coreference Resolution*. IOS Press, volume 215:759–764.

Kai-Wei Chang, Rajhans Samdani, Alla Rozovskaya, Nick Rizzolo, Mark Sammons, and Dan Roth. 2011. *Inference protocols for coreference resolution*. Proceedings of the CoNLL'11 Shared Task, 40–44.

Belur V. Dasarathy. 1991. *Nearest Neighbor (NN) Norms: NN Pattern Recognition Classification Techniques*. IEEE Computer Society Press.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. *Indexing by Latent Semantic Analysis*. Journal of the American Society for Information Science, 41(6):391–407.

Thomas G. Dietterich. 1998. *Machine Learning Research: Four Current Directions*. The AI Magazine, volume 18(4):97–136.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. *The Automatic Content Extraction (ACE) Program-Tasks, Data, and Evaluation*. Proceedings of the LREC-2004, 837–840.

Susan T. Dumais. 2004. *Latent Semantic Analysis*. ARIST (Annual Review of Information Science Technology), volume 38:189–230.

Mark Hall, Eibe Franke, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. *The WEKA Data Mining Software: An Update*. SIGKDD Explorations, volume 11(1):10–18.

Tin K. Ho, Jonathan J. Hull, and Sargur N. Srihari. 1994. *Decision Combination in Multiple Classifier Systems*. IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 16(1):66–75.

Hamidreza Kobdani and Hinrich Schütze. 2010. *Sucre: A modular system for coreference resolution*. Proceedings of the SemEval-2010, pp. 92–95.

Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2013. *Deterministic coreference resolution based on entity-centric, precision-ranked rules*. Computational Linguistics, 39(4):885–916.

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

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. *The Stanford CoreNLP Natural Language Processing Toolkit*. Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 55–60.

Ruslan Mitkov. 1998. *Robust pronoun resolution with limited knowledge*. Proceedings of the COLING'98, volume 2: 869–875.

Ruslan Mitkov. 2002. *Anaphora Resolution*. Pearson Education.

MUC-6. 1995. *Coreference task definition*. Proceedings of the MUC, 335–344.

MUC-7. 1998. *Coreference task definition*. Proceedings of the MUC.

Vincent Ng. 2010. *Supervised Noun Phrase Coreference Research: The First Fifteen Years*. Proceedings of the ACL'10, 1396–1411.

Kristina Nilsson and Hans Hjelm. 2009. *Using Semantic Features Derived from Word-Space Models for Swedish Coreference Resolution*. Proceedings of the NoDaLiDa'09, volume 4:134–141.

Constantin Orăsan, Dan Cristea, Ruslan Mitkov, and António Branco. 2008. *Anaphora Resolution Exercise: an Overview*. Proceedings of the LREC'08.

Sameer Pradhan, Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel 2007a. *Ontonotes: a Unified Relational Semantic Representation*. International Journal of Semantic Computing, volume 1(4):405–419.

Sameer Pradhan, Lance Ramshaw, Ralph Weischedel, Jessica MacBride, and Linnea Micciulla. 2007b. *Unrestricted Coreference: Identifying Entities and Events in OntoNotes*. Proceedings of the ICSC, pp. 446–453.

Sameer Pradhan, Martha Palmer, Lance Ramshaw, Ralph Weischedel, Mitchell Marcus, and Nianwen Xue. 2011. *CoNLL-2011 Shared Task: Modeling Unrestricted Coreference in OntoNotes*. Proceedings of the CONLL'11 Shared Task, 1–27.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. *CoNLL-2012 Shared Task: Modeling Multilingual Unrestricted Coreference in OntoNotes*. Proceedings of the CONLL'12 Shared Task, 1–40. Association for Computational Linguistics.

Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. 2010. *A multi-pass sieve for coreference resolution*. Proceedings of the EMNLP'10, pp. 492–501.

Marta Recasens, Lluís Márquez, Emili Sapena, M. Antónia Martí, Mariona Taulé, Véronique Hoste, Massimo Poesio, and Yannick Versley. 2010. *SemEval-2010 Task 1: Coreference Resolution in Multiple Language*. Proceedings of the SemEval-2010, pp. 1–8.

C. N. dos Santos and D. L. Carvalho. 2011. *Rule and tree ensembles for unrestricted coreference resolution*. Proceedings of the CONLL'11 Shared Task, pp. 51–55.

Emili Sapena, Lluís Padró, and Jordi Turmo. 2011. *RelaxCor Participation in CoNLL Shared Task on Coreference Resolution*. Proceedings of the CONLL'11 Shared Task, pp. 35–39.

Wee M. Soon, Hwee Ng, and Daniel C. Y. Lim. 2001. *A Machine Learning Approach to Coreference Resolution of Noun Phrases*. Association for Computational Linguistics, volume 27(4): 521–544.

Olga Uryupina. 2010. *Corry: A system for coreference resolution*. Proceedings of the SemEval-2010, 100–103.

Olga Uryupina, and Massimo Poesio. 2012. *Domain-specific vs. Uniform Modeling for Coreference Resolution*. Proceedings of the LREC-2012: 187–191.

Yannick Versley, Simone Paolo Ponzetto, Massimo Poesio, Vladimir Eidelman, Alan Jern, Jason Smith, Xiaofeng Yang, and Alessandro Moschitti. 2008. *Bart: a modular toolkit for coreference resolution*. Proceedings of the HLT-Demonstrations'08, pp. 9–12.