Using Syntactic Dependencies to Solve Coreferences

Marcus Stamborg Dennis Medved Peter Exner Pierre Nugues
Lund University
Lund, Sweden
cid03mst@student.lu.se, dt07dm0@student.lth.se
Peter.Exner@cs.lth.se, Pierre.Nugues@cs.lth.se

Abstract

This paper describes the structure of the LTH coreference solver used in the closed track of the CoNLL 2012 shared task (Pradhan et al., 2012). The solver core is a mention classifier that uses Soon et al. (2001)'s algorithm and features extracted from the dependency graphs of the sentences.

This system builds on Björkelund and Nugues (2011)'s solver that we extended so that it can be applied to the three languages of the task: English, Chinese, and Arabic. We designed a new mention detection module that removes pleonastic pronouns, prunes constituents, and recovers mentions when they do not match exactly a noun phrase. We carefully redesigned the features so that they reflect more complex linguistic phenomena as well as discourse properties. Finally, we introduced a minimal cluster model grounded in the first mention of an entity.

We optimized the feature sets for the three languages: We carried out an extensive evaluation of pairs of features and we complemented the single features with associations that improved the CoNLL score. We obtained the respective scores of 59.57, 56.62, and 48.25 on English, Chinese, and Arabic on the development set, 59.36, 56.85, and 49.43 on the test set, and the combined official score of 55.21.

1 Introduction

In this paper, we present the LTH coreference solver used in the closed track of the CoNLL 2012 shared task (Pradhan et al., 2012). We started from an earlier version of the system by Björkelund and Nugues (2011), to which we added substantial improvements. As base learning and decoding algorithm, our solver extracts noun phrases and possessive pronouns and uses Soon et al. (2001)'s pairwise classifier to decide if a pair corefers or not. Similarly to the earlier LTH system, we constructed a primary feature set from properties extracted from the dependency graphs of the sentences.

2 System Architecture

The training and decoding modules consist of a mention detector, a pair generator, and a feature extractor. The training module extracts a set of positive and negative pairs of mentions and uses logistic regression and the LIBLINEAR package (Fan et al., 2008) to generate a binary classifier. The solver extracts pairs of mentions and uses the classifier and its probability output, $P_{coref}(\text{Antecedent}, \text{Anaphor})$, to determine if a pair corefers or not. The solver has also a post processing step to recover some mentions that do not match a noun phrase constituent.

3 Converting Constituents to Dependency Trees

Although the input to coreference solvers are pairs or sets of constituents, many systems use concepts from dependency grammars to decide if a pair is coreferent. The most frequent one is the constituent's head that solvers need then to extract using ad-hoc rules; see the CoNLL 2011 shared task (Pradhan et al., 2011), for instance. This can be tedious as we may have to write new rules for each new feature to incorporate in the classifier. That is

Proceedings of the Joint Conference on EMNLP and CoNLL: Shared Task, pages 64–70,
Jeju Island, Korea, July 13, 2012.

why, instead of writing sets of rules applicable to specific types of dependencies, we converted all the constituents in the three corpora to generic dependency graphs before starting the training and solving steps. We used the LTH converter (Johansson and Nugues, 2007) for English, the Penn2Malt converter (Nivre, 2006) with the Chinese rules for Chinese[1], and the CATiB converter (Habash and Roth, 2009) for Arabic.

The CATiB converter (Habash and Roth, 2009) uses the Penn Arabic part-of-speech tagset, while the automatically tagged version of the CoNLL Arabic corpus uses a simplified tagset inspired by the English version of the Penn treebank. We translated these simplified POS tags to run the CATiB converter. We created a lookup table to map the simplified POS tags in the automatically annotated corpus to the Penn Arabic POS tags in the gold annotation. We took the most frequent association in the lookup table to carry out the translation. We then used the result to convert the constituents into dependencies. We translated the POS tags in the development set using a dictionary extracted from the gold training file and we translated the tags in the training file by a 5-fold cross-validation. We used this dictionary during both training and classifying since our features had a better performance with the Arabic tagset.

4 Mention Extraction

4.1 Base Extraction

As first step of the mention selection stage, we extracted all the noun phrases (NP), pronouns (PRP), and possessive pronouns (PRP$) for English and Arabic, with the addition of PN pronouns for Chinese. This stage is aimed at reaching a high recall of the mentions involved in the coreference chains and results in an overinclusive set of candidates. Table 1 shows the precision and recall figures for the respective languages when extracting mentions from the training set. The precision is significantly lower for Arabic than for English and Chinese.

4.2 Removal of the Pleonastic *it*

In the English corpus, the pronoun *it* in the first step of the mention extraction stage creates a high number of false positive mentions. We built a classifier to discard as many of these pleonastic *it* as possible from the mention list.

[1] http://stp.lingfil.uu.se/ nivre/research/Penn2Malt.html

Language	Recall	Precision
English	92.17	32.82
English with named entities	94.47	31.61
Chinese	87.32	32.29
Arabic	87.22	17.64

Table 1: Precision and recall for the mention detection stage on the training set.

Feature name
HeadLex
HeadRightSiblingPOS
HeadPOS

Table 2: Features used by the pleonastic *it* classifier.

Table 2 shows the features we used to train the classifier and Table 3 shows the impact on the final system. We optimized the feature set using greedy forward and backward selections. We explored various ways of using the classifier: before, after, and during coreference resolving. We obtained the best results when we applied the pleonastic classifier during coreference solving and we multiplied the probability outputs from the two classifiers. We used the inequality:

$$P_{coref}(\text{Antecedent}, it) \times (1 - P_{pleo}(it)) > 0.4,$$

where we found the optimal threshold of 0.4 using a 5-fold cross-validation.

4.3 Named Entities

The simple rule to approximate entities to noun phrases and pronouns leaves out between $\sim$8% and $\sim$13 % of the entities in the corpora (Table 1). As the named entities sometimes do not match constituents, we tried to add them to increase the recall. We carried out extensive experiments for the three lan-

English	CoNLL score
Without removal	59.15
With removal	**59.57**

Table 3: Score on the English development set with and without removal of the pleonastic *it* pronouns.

English	Total score
Without named entities	58.85
With named entities	**59.57**

Table 4: Impact on the overall score on the English development set by addition of named entities extracted from the corpus.

Language	Without pruning	With pruning
English	56.42	**59.57**
Chinese	50.94	**56.62**
Arabic	**48.25**	47.10

Table 5: Results on running the system on the development set with and without pruning for all the languages.

guages. While the named entities increased the score for the English corpus, we found that it lowered the results for Chinese and Arabic. We added all single and multiword named entities of the English corpus except the CARDINAL, ORDINAL, PERCENT, and QUANTITY tags. Table 1 shows the recall and precision for English and Table 4 shows the named entity impact on the overall CoNLL score on the development set.

4.4 Pruning

When constituents shared the same head in the list of mentions, we pruned the smaller ones. This increased the scores for English and Chinese, but lowered that of Arabic (Table 5). The results for the latter language are somewhat paradoxical; they are possibly due to errors in the dependency conversion.

5 Decoding

Depending on the languages, we applied different decoding strategies: For Chinese and Arabic, we used a closest-first clustering method as described by Soon et al. (2001) for pronominal anaphors and a best-first clustering otherwise as in Ng and Cardie (2002). For English, we applied a closest-first clustering for pronominal anaphors. For nonpronominal anaphors, we used an averaged best-first clustering: We considered all the chains before the current anaphor and we computed the geometric mean of the pair probabilities using all the mentions in a chain. We linked the anaphor to the maximal scoring chain or we created a new chain if the score was less than 0.5. We discarded all the remaining singletons.

English	Total score
Without extensions	57.22
With extensions	**59.57**

Table 6: Total impact of the extensions to the mention extraction stage on the English development set.

As in Björkelund and Nugues (2011), we recovered some mentions using a post processing stage, where we clustered named entities to chains having strict matching heads.

6 Features

We started with the feature set described in Björkelund and Nugues (2011) for our baseline system for English and with the feature set in Soon et al. (2001) for Chinese and Arabic. Due to space limitations, we omit the description of these features and refer to the respective papers.

6.1 Naming Convention

We denoted HD, the head word of a mention in a dependency tree, HDLMC and HDRMC, the left-most child and the right-most child of the head, HDLS and HDRS, the left and right siblings of the head word, and HDGOV, the governor of the head word.

From these tokens, we can extract the surface form, FORM, the part-of-speech tag, POS, and the grammatical function of the token, FUN, i.e. the label of the dependency edge of the token to its parent.

We used a naming nomenclature consisting of the role in the anaphora, where J- stands for the anaphor, I-, for the antecedent, F-, for the mention in the chain preceding the antecedent (previous antecedent), and A- for the first mention of the entity in the chain; the token we selected from the dependency graph, e.g. HD or HDLMC; and the value extracted from the token e.g. POS or FUN. For instance, the part-of-speech tag of the governor of the head word of the anaphor is denoted J-HDGOVPOS.

6.2 Combination of Features

In addition to the single features, we combined them to create bigram, trigram, and four-gram features. Table 7 shows the features we used, either single or in combination, e.g. I-HDFORM+J-HDFORM.

We emulated a simple cluster model by utilizing the first mention in the chain and/or the previous antecedent, e.g. A-EDITDISTANCE+F-EDITDISTANCE+EDITDISTANCE, where the edit distance of the anaphor is calculated for the first mention in the chain, previous antecedent, and antecedent.

6.3 Notable New Features

Edit Distance Features. We created edit distance-based features between pairs of potentially coreferring mentions: EDITDISTANCE is the character-based edit distance between two strings; EDITDISTANCEWORD is a word-level edit distance, where the symbols are the complete words; and PROPERNAMESIMILARITY is a character-based edit distance between proper nouns only.

Discourse Features. We created features to reflect the speaker agreement, i.e. when the pair of mentions corresponds to the same speaker, often in combination with the fact that both mentions are pronouns. For example, references to the first person pronoun *I* from a same speaker refer probably to a same entity; in this case, the speaker himself.

Document Type Feature. We created the I-HD FORM+J-HDFORM+DOCUMENTTYPE feature to capture the genre of different document types, as texts from e.g. the New Testament are likely to differ from internet blogs.

6.4 Feature Selection

We carried out a greedy forward selection of the features starting from Björkelund and Nugues (2011)'s feature set for English, and Soon et al. (2001)'s for Chinese and Arabic. The feature selection used a 5-fold cross-validation over the training set, where we evaluated the features using the arithmetic mean of MUC, BCUB, and CEAFE.

After reaching a maximal score using forward selection, we reversed the process using a backward elimination, leaving out each feature and removing the one that had the worst impact on performance. This backwards procedure was carried out until the score no longer increased. We repeated this forward-backward procedure until there was no increase in performance.

7 Evaluation

Table 7 shows the final feature set for each language combined with the impact each feature has on the score on the development set when being left out. A dash (—) means that the feature is not part of the feature set used in the respective language. As we can see, some features increase the score. This is due to the fact that the feature selection was carried out in a cross-validated manner over the training set.

Table 8 shows the results on the development and test sets as well as on the test set with gold mentions. For each language, the figures are overall consistent between the development and test sets across all the metrics. The scores improve very significantly with the gold mentions: up to more than 10 points for Chinese.

8 Conclusions

The LTH coreference solver used in the CoNLL 2012 shared task uses Soon et al. (2001)'s algorithm and a set of lexical and nonlexical features. To a large extent, we extracted these features from the dependency graphs of the sentences. The results we obtained seem to hint that this approach is robust across the three languages of the task.

Our system builds on an earlier system that we evaluated in the CoNLL 2011 shared task (Pradhan et al., 2011), where we optimized significantly the solver code, most notably the mention detection step and the feature design. Although not exactly comparable, we could improve the CoNLL score by 4.83 from 54.53 to 59.36 on the English corpus. The mention extraction stage plays a significant role in the overall performance. By improving the quality of the mentions extracted, we obtained a performance increase of 2.35 (Table 6).

Using more complex feature structures also proved instrumental. Scores of additional feature variants could be tested in the future and possibly increase the system's performance. Due to limited computing resources and time, we had to confine the search to a handful of features that we deemed most promising.

All features	En (+/-)	Zh (+/-)	Ar (+/-)
StringMatch	-0.003	**-0.58**	**-1.79**
A-StringMatch+StringMatch	-0.11	—	—
Distance	-0.19	-0.57	-0.24
Distance+J-Pronoun	0.03	—	—
I-Pronoun	0.02	—	—
J-Pronoun	0.02	—	—
J-Demostrative	-0.02	0.01	—
BothProperName	—	0.03	—
NumberAgreement	**-0.23**	—	—
GenderAgreement	0.003	—	—
NumberBigram	—	0.06	—
GenderBigram	-0.03	0.01	—
I-HdForm	-0.16	—	-0.67
I-HdFun	0.05	—	—
I-HdPos	-0.02	—	-0.52
I-HdRmcFun	0.003	—	—
I-HdLmcForm	—	—	-0.05
I-HdLmcPos	0.01	—	—
I-HdLsForm	-0.08	—	-0.18
I-HdGovFun	0.06	—	—
I-HdGovPos	—	-0.003	-0.19
J-HdFun	0.003	—	—
J-HdGovFun	0.03	—	—
J-HdGovPos	-0.05	—	—
J-HdRsPos	—	—	-0.2
A-HdChildSetPos	—	0.06	—
I-HdForm+J-HdForm	0.08	—	-0.57
A-HdForm+J-HdForm	—	—	-0.46
I-HdGovForm+J-HdForm	—	-0.14	0.04
I-LmcForm+J-LmcForm	-0.07	-0.15	—
A-HdForm+I-HdForm+J-HdForm	0.11	—	—
F-HdForm+I-HdForm+J-HdForm	—	-0.1	—
I-HdPos+J-HdPos+I-HdFun+J-HdFun	—	-0.09	—
I-HdPos+J-HdPos+I-HdForm+J-HdForm	—	—	-0.05
I-HdForm+J-HdForm+SpeakAgree	—	-0.55	—
I-HdForm+J-HdForm+BothPrn+SpeakAgree	-0.11	—	—
I-HdGovForm+J-HdForm+BothPrn+SpeakAgree	**-0.23**	—	—
A-HdForm+J-HdForm+SpeakAgree	0.04	—	—
I-HdForm+J-HdForm+DocumentType	-0.4	-0.18	—
SsPathBergsmaLin	-0.07	—	—
SsPathForm	—	—	-0.19
SsPathFun	-0.08	—	-0.14
SsPathPos	-0.1	-0.11	-0.53
DsPathBergsmaLin	—	—	0
DsPathForm	0.07	—	—
DsPathForm+DocumentType	0.03	—	—
DsPathPos	0.07	-0.06	0.05
EditDistance	-0.05	-0.16	0
EditDistanceWord	—	—	-0.25
A-EditDistance+EditDistance	—	—	-0.02
A-EditDistance+F-EditDistance	—	-0.01	-0.01
A-EditDistance+F-EditDistance+EditDistance	—	—	-0.09
EditDistanceWord+BothProperName	0.02	—	—
ProperNameSimilarity	-0.03	—	—
SemRolePropJHd	0.01	—	—

Table 7: The feature sets for English, Chinese and Arabic, and for each feature, the degradation in performance when leaving out this feature from the set; the more negative, the better the feature contribution. We carried out all the evaluations on the development set. The table shows the difference with the official CoNLL score.

Metric/Corpus	Development set			Test set			Test set (Gold mentions)		
English	R	P	F1	R	P	F1	R	P	F1
Mention detection	74.21	72.81	73.5	75.51	72.39	73.92	78.17	100	87.74
MUC	65.27	64.25	64.76	66.26	63.98	65.10	71.22	88.12	78.77
BCUB	69.1	70.94	70.01	69.09	69.54	69.31	64.75	83.16	72.8
CEAFM	57.56	57.56	57.56	56.76	56.76	56.76	66.74	66.74	66.74
CEAFE	43.44	44.47	43.95	42.53	44.89	43.68	71.94	43.74	54.41
BLANC	75.36	77.41	76.34	74.03	77.28	75.52	78.68	81.47	79.99
CoNLL score			59.57			59.36			68.66
Chinese	R	P	F1	R	P	F1	R	P	F1
Mention detection	60.55	68.73	64.38	57.65	71.93	64.01	68.97	100	81.63
MUC	54.63	60.96	57.62	52.56	64.13	57.77	63.52	88.23	73.86
BCUB	66.91	74.4	70.46	64.43	77.55	70.38	63.54	88.12	73.84
CEAFM	55.09	55.09	55.09	55.57	55.57	55.57	65.60	65.60	65.60
CEAFE	44.65	39.25	41.78	47.90	38.04	42.41	72.56	42.01	53.21
BLANC	73.23	72.95	73.09	72.74	77.84	75.00	76.96	83.70	79.89
CoNLL score			56.62			56.85			66.97
Arabic	R	P	F1	R	P	F1	R	P	F1
Mention detection	55.54	61.7	58.46	56.1	63.28	59.47	56.13	100	71.9
MUC	39.18	43.76	41.34	39.11	43.49	41.18	41.99	69.78	52.43
BCUB	59.16	67.94	63.25	61.57	67.95	64.61	50.45	81.30	62.26
CEAFM	47.8	47.8	47.8	50.16	50.16	50.16	54.00	54.00	54.00
CEAFE	42.57	38.01	40.16	44.86	40.36	42.49	66.16	34.52	45.37
BLANC	62.44	67.18	64.36	66.80	66.94	66.87	67.37	73.46	69.87
CoNLL score			48.25			49.43			53.35

Table 8: Scores on the development set, test set, and test set with gold mentions for English, Chinese, and Arabic: recall R, precision P, and harmonic mean F1. The official CoNLL score is computed as the arithmetic mean of MUC, BCUB, and CEAFE.

Acknowledgments

This research was supported by Vetenskapsrådet, the Swedish research council, under grant 621-2010-4800, and the European Union's seventh framework program (FP7/2007-2013) under grant agreement no. 230902.

References

Anders Björkelund and Pierre Nugues. 2011. Exploring lexicalized features for coreference resolution. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 45–50, Portland, Oregon, USA, June. Association for Computational Linguistics.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Nizar Habash and Ryan Roth. 2009. CATiB: The Columbia Arabic treebank. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 221–224, Suntec, Singapore, August. Association for Computational Linguistics.

Richard Johansson and Pierre Nugues. 2007. Extended constituent-to-dependency conversion for English. In Joakim Nivre, Heiki-Jaan Kaalep, Kadri Muischnek, and Mare Koit, editors, *NODALIDA 2007 Conference Proceedings*, pages 105–112, Tartu, May 25-26.

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

Joakim Nivre. 2006. *Inductive Dependency Parsing*. Springer, Dordrecht, The Netherlands.

Sameer Pradhan, Lance Ramshaw, Mitchell Marcus, Martha Palmer, Ralph Weischedel, and Nianwen Xue. 2011. CoNLL-2011 shared task: Modeling unrestricted coreference in OntoNotes. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–27, Portland, Oregon, USA, June. Association for Computational Linguistics.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Proceedings of the Sixteenth Conference on Computational Natural Language Learning (CoNLL 2012)*, Jeju, Korea.

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