

# Learning to Model Multilingual Unrestricted Coreference in OntoNotes

**Baoli LI**

Department of Computer Science  
Henan University of Technology  
1 Lotus Street, High&New Technology  
Industrial Development Zone, Zhengzhou,  
Henan, China, 450001  
csblli@gmail.com

## Abstract

Coreference resolution, which aims at correctly linking meaningful expressions in text, is a much challenging problem in Natural Language Processing community. This paper describes the multilingual coreference modeling system of Web Information Processing Group, Henan University of Technology, China, for the CoNLL-2012 shared task (closed track). The system takes a supervised learning strategy, and consists of two cascaded components: one for detecting mentions, and the other for clustering mentions. To make the system applicable for multiple languages, generic syntactic and semantic features are used to model coreference in text. The system obtained combined official score 41.88 over three languages (Arabic, Chinese, and English) and ranked 7<sup>th</sup> among the 15 systems in the closed track.

## 1 Introduction

Coreference resolution, which aims at correctly linking meaningful expressions in text, has become a central research problem in natural language processing community with the advent of various supporting resources (e.g. corpora and different kinds of knowledge bases). OntoNotes (Pradhan et

al. 2007), compared to MUC (Chinchor, 2001; Chinchor and Sundheim, 2003) and ACE (Doddington et al. 2000) corpora, is a large-scale, multilingual corpus for general anaphoric coreference that covers entities and events not limited to noun phrases or a limited set of entity types. It greatly stimulates the research on this challenging problem – Coreference Resolution. Moreover, resources like WordNet (Miller, 1995) and the advancement of different kinds of syntactic and semantic analysis technologies, make it possible to do in-depth research on this topic, which is demanded in most of natural language processing applications, such as information extraction, machine translation, question answering, summarization, and so on.

Our group is exploring how to extract information from grain/cereal related Chinese text for business intelligence. This shared task provides a good platform for advancing our research on IE related topics. We experiment with a machine learning strategy to model multilingual coreference for the CoNLL-2012 shared task (Pradhan et al. 2012). Two steps are taken to detect coreference in text: mention detection and mention clustering. We consider mentions that correspond to a word or an internal node in a syntactic tree and ignore the rest mentions, as we think a mention should be a valid meaningful unit of a sentence. Maximal entropy algorithm is used to model what a mention is and how two mentions link to each other. Generic features are designed to facilitate these modeling.

Our official submission obtained combined official score 41.88 over three languages (Arabic, Chinese, and English), which ranked the system 7<sup>th</sup> among 15 systems participating the closed track. Our system performs poor on the Arabic data, and has relatively high precision but low recall.

The rest of this paper is organized as follows. Section 2 gives the overview of our system, while Section 3 discusses the first component of our system for mention detection. Section 4 explains how our system links mentions. We present our experiments and analyses in Section 5, and conclude in Section 6.

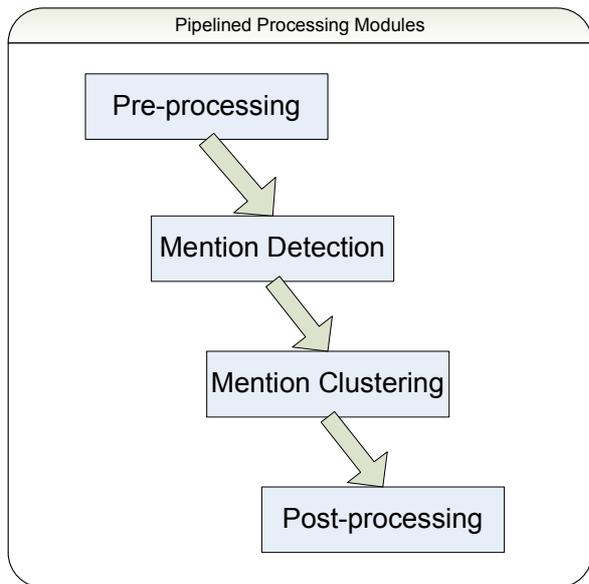


Figure 1. System Architecture.

## 2 System Description

Figure 1 gives the architecture of our CoNLL-2012 system, which consists of four pipelined processing modules: pre-processing, mention detection, mention clustering, and post-processing.

**Pre-processing:** this module reads in the data files in CoNLL format and re-builds the syntactic and semantic analysis trees in memory.

**Mention Detection:** this module chooses potential sub-structures on the syntactic parsing trees and determines whether they are real mentions.

**Mention Clustering:** this module compares pairs of mentions and links them together.

**Post-processing:** this module removes singleton mentions and produces the final results.

To facilitate the processing, the data files of the same languages are combined together to form big files for training, development, and test respectively.

Compared to the CoNLL-2011 shared task, the task of this year focuses on the multilingual capacity of a coreference resolution system. We plan to take a generic solution for different languages rather than customized approach to some languages with special resources. In other words, our official system didn't take any special processing for data of different languages but used the same strategy and feature sets for all three languages.

Stanford's Rule-based method succeeded in resolving the coreferences in English text last year (Pradhan et al. 2011; Lee et al. 2011). Therefore, we plan to incorporate the results of a rule-based system (simple or complex as the Stanford's system) if available and derive some relevant features for our machine learning engine. However, due to limited time and resources, we failed to implement in our official system such a solution integrating rules within the overall statistical model.

Intuitively, mentions are meaningful sub-structures of sentences. We thus assume that a mention should be a word or a phrasal sub-structure of a parsing tree. Mention detection modules focus on these mentions and ignore others that do not correspond to a valid phrasal sub-structure.

A widely used machine learning algorithm in solving different NLP problems, Maximal Entropy (Berger et al.1996), is used to model mentions and detect links between them. Compared with Naive Bayes algorithm, Maximum entropy does not assume statistical independence of the different features. In our system, Le Zhang's maximum entropy package (Zhang, 2006) is integrated.

In the following two sections, we will detail the two critical modules: mention detection and mention clustering.

## 3 Mention Detection

This module determines all mentions in text. We take the assumption that a mention should be a valid sub-structure of a sentence.

### 3.1 Methods

We first choose potential mentions in text and then use statistical machine learning method to make final decisions.

From the train and development datasets, we could obtain a list of POS and syntactic structure tags that a mention usually has. For example, below is given such a list for English data:

```

POS_TAG "NP" /*145765*/
POS_TAG "NML" /*910*/
POS_TAG "S" /*207*/
POS_TAG "VP" /*189*/
POS_TAG "ADVP" /*75*/
POS_TAG "FRAG" /*73*/
POS_TAG "WHNP" /*67*/
POS_TAG "ADJP" /*65*/
POS_TAG "QP" /*62*/
POS_TAG "INTJ" /*40*/
POS_TAG "PP" /*16*/
POS_TAG "SBAR" /*10*/
POS_TAG "WHADVP" /*7*/
POS_TAG "UCP" /*5*/
//POS_TAG "SINV" /*1*/
//POS_TAG "SBARQ" /*1*/
//POS_TAG "RRC" /*1*/
//POS_TAG "SQ" /*1*/
//POS_TAG "LST" /*1*/
SYN_TAG "PRP$" /*14734*/
SYN_TAG "NNP" /*3642*/
SYN_TAG "VB" /*733*/
SYN_TAG "VBD" /*669*/
SYN_TAG "VBN" /*384*/
SYN_TAG "VBG" /*371*/
SYN_TAG "NN" /*306*/
SYN_TAG "VBZ" /*254*/
SYN_TAG "VBP" /*235*/
SYN_TAG "PRP" /*137*/
SYN_TAG "CD" /*132*/
SYN_TAG "DT" /*77*/
SYN_TAG "IN" /*64*/
SYN_TAG "NNS" /*57*/
SYN_TAG "JJ" /*52*/
SYN_TAG "RB" /*19*/
SYN_TAG "NNPS" /*17*/
SYN_TAG "UH" /*7*/
SYN_TAG "CC" /*7*/
SYN_TAG "NFP" /*5*/
SYN_TAG "XX" /*4*/
SYN_TAG "MD" /*3*/
SYN_TAG "JJR" /*2*/
SYN_TAG "POS" /*2*/
//SYN_TAG "FW" /*1*/
//SYN_TAG "ADD" /*1*/

```

We remove tags rarely occurring in the datasets, such as FW and ADD for English and consider all words and syntactic structures of the rest categories as potential mentions.

To make a decision about whether a potential mention is a real one or not, we use a maximal entropy classifier with a set of generic features concerning the word or sub-structure itself and its syntactic and semantic contexts.

### 3.2 Features

The features we used in this step for each potential word or sub-structure include:

- a. Source and Genre of a document; Speaker of a sentence;
- b. Level of the Node in the syntactic parsing tree;
- c. Named entity tag of the word or sub-structure;
- d. Its head predicates and types;
- e. Syntactic tag path to the root;
- f. Whether it's part of a mention, named entity, or an argument;
- g. Features from its parent: syntactic tag, named entity tag, how many children it has, whether the potential word or sub-structure is the left most child of it, the right most child, or middle child; binary syntactic tag feature;
- h. Features from its direct left and right siblings: their syntactic tags, named entity tags, and binary syntactic tag features;
- i. Features from its children: its total token length, words, pos tags, lemma, frameset ID, and word sense, tag paths to the left and right most child;
- j. Features from its direct neighbor (before and after) tokens: words, pos tags, lemma, frameset ID, and word sense, and binary features of pos tags;

## 4 Mention Clustering

This component clusters the detected mentions into group.

### 4.1 Methods

For each pair of detected mentions, we determine whether they could be linked together with a maximal entropy classifier. The clustering takes a best-of-all strategy and works as the following algorithm:

```

INPUT: a list of mentions;
OUTPUT: a splitting of the mentions into groups;
ALGORIHTM:

```

1. For each detected mention *ANAP* from the last to the first:
  - 1.1 Find its most likely linked antecedent *ANTE* before *ANAP*
  - 1.2 if FOUND
    - 1.2.1 link all anaphors of *ANAP* to *ANTE*;
    - 1.2.2 link *ANAP* to *ANTE*

Figure 2. Algorithm for Clustering Detected Mentions

We used the probability value of the maximal entropy classifier’s output for weighting the links between mentions.

## 4.2 Features

The features we used in this step include:

- a. Source and Genre of a document; Speaker of a sentence;
- b. Sentence distance between the potential antecedent and anaphor;
- c. Syntactic tag of them, whether they are leaf node or not in the parsing tree;
- d. Syntactic tag bi-grams of them, and whether their syntactic tags are identical;
- e. Named entity tags of them, bi-gram of these tags, and whether they are identical;
- f. Syntactic tag path to root of them, bi-gram of these paths, and whether they are identical;
- g. Whether they are predicates;
- h. Features of anaphor: Its head predicates and types, words, pos tags, the words and pos tags of the left/right 3 neighbor tokens, and bi-grams;
- i. Features of antecedent: Its head predicates and types, words, pos tags, the words and pos tags of the left/right 3 neighbor tokens, and bi-grams;
- j. The number of identical words of the antecedent and the anaphor;
- k. The number of identical words in the neighbors (3 tokens before and after) of the antecedent and the anaphor.

The above features include not only those suggested by Soon et al. (2001), but also some context features, such as words within and out of the antecedent and the anaphor, and the overlapping number of the context words. Features about Gender and number agreements are not considered in our official system, as we failed to work out a generic solution to include them for all data of three different languages.

## 5 Experiments

### 5.1 Datasets

The datasets of the CoNLL-2012 shared task contain three languages: Arabic (ARB), Chinese (CHN), and English (ENG). No predicted names and propositions are provided in the Arabic data, while no predicted names are given in the Chinese data.

Tables 1 and 2 show statistical information of both training and development datasets for each language.

Language		# of Doc.	# of Sent.	# of Ment.	# of mentions that do not correspond to a valid phrasal sub-structure
ARB	Dev	44	950	3,317	262(7.9%)
	Train	359	7,422	27,590	2,176(7.9%)
CHN	Dev	252	6,083	14,183	677(4.8%)
	Train	1,810	36,487	102,854	6,345(6.2%)
ENG	Dev	343	9,603	19,156	661(3.5%)
	Train	2,802	75,187	155,560	4,639(3.0%)

Table 1. Statistical information of the three language datasets (train and development) (part 1).

Language		# of sentences per document		# of tokens per sentence	
		Avg.	Max	Avg.	Max
ARB	Dev	21.59	41	29.82	160
	Train	20.67	78	32.70	384
CHN	Dev	24.14	144	18.09	190
	Train	20.16	283	20.72	242
ENG	Dev	28.00	127	16.98	186
	Train	26.83	188	17.28	210

Table 2. Statistical information of the three language datasets (train and development) (part 2).

The total size of the uncompressed original data is about 384MB. The English dataset is the largest one containing 3,145 documents (343+2802), 84,790 sentences, and 174,716 mentions. The Arabic dataset is the smallest one containing 403 documents, 8,372 sentences, and 30,907 mentions. In the Arabic datasets, about 7.9% mentions do not

correspond to a valid phrasal sub-structure. This number of the Chinese dataset is 6%, while that of English 3%. These small percentages verify that our assumption that a mention is expected to be a valid phrasal sub-structure is reasonable.

The average numbers of sentences in a document in the three language datasets are roughly 21, 22, and 27 respectively, while the longest document that has 283 sentences is found in the Chinese train dataset. The average numbers of tokens in a sentence in the three language datasets are roughly 31, 19, and 17 respectively, while the longest sentence with 384 tokens is found in the Arabic train dataset.

## 5.2 Experimental Results

For producing the results on the test datasets, we combined both train and development datasets for training maximal entropy classifiers.

The official score adopted by CoNLL-2012 is the unweighted average of scores on three languages, while for each language, the score is derived by averaging the three metrics MUC (Vilain et al. 1995), B-CUBED (Bagga and Baldwin, 1998), and CEAF(E) (Constrained Entity Aligned F-measure)(Luo, 2005) as follows:

$$\text{OFFICIAL SCORE} = \frac{\text{MUC} + \text{B-CUBED} + \text{CEAF (E)}}{3}$$

Our system achieved the combined official score 42.32 over three languages (Arabic, Chinese, and English). On each of the three languages, the system obtained scores 33.53, 46.27, and 45.85 respectively. It performs poor on the Arabic dataset, but equally well on the Chinese and English datasets.

Tables 3, 4, and 5 give the detailed results on three languages respectively.

Metric	Recall	Precision	F1
<b>MUC</b>	<b>10.77</b>	<b>55.60</b>	<b>18.05</b>
<b>B-CUBED</b>	<b>36.17</b>	<b>93.34</b>	<b>52.14</b>
CEAF (M)	37.03	37.03	37.03
<b>CEAF (E)</b>	<b>55.45</b>	<b>20.95</b>	<b>30.41</b>
BLANC <sup>1</sup>	52.91	73.93	54.12
<b>OFFICIAL SCORE</b>	NA	NA	<b>33.53</b>

Table 3. Official results of our system on the Arabic test dataset.

<sup>1</sup> For this metric, please refer to (Recasens and Hovy, 2011).

Metric	Recall	Precision	F1
<b>MUC</b>	<b>32.48</b>	<b>71.44</b>	<b>44.65</b>
<b>B-CUBED</b>	<b>45.51</b>	<b>86.06</b>	<b>59.54</b>
CEAF (M)	45.70	45.70	45.70
<b>CEAF (E)</b>	<b>55.11</b>	<b>25.24</b>	<b>34.62</b>
BLANC	64.99	76.63	68.92
<b>OFFICIAL SCORE</b>	NA	NA	<b>46.27</b>

Table 4. Official results of our system on the Chinese test dataset.

Metric	Recall	Precision	F1
<b>MUC</b>	<b>39.12</b>	<b>72.57</b>	<b>50.84</b>
<b>B-CUBED</b>	<b>43.03</b>	<b>80.06</b>	<b>55.98</b>
CEAF (M)	41.97	41.97	41.97
<b>CEAF (E)</b>	<b>49.44</b>	<b>22.30</b>	<b>30.74</b>
BLANC	64.01	66.86	65.24
<b>OFFICIAL SCORE</b>	NA	NA	<b>45.85</b>

Table 5. Official results of our system on the English test dataset.

Comparing the detailed scores, we found that our submitted system performs much poor on the MUC metric on the Arabic data. It can only recover 10.77% valid mentions. As a whole, the system works well in precision perspective but poor in recall perspective.

Language	Recall	Precision	F1
Arabic	18.17	80.43	29.65
Chinese	36.60	87.01	51.53
English	45.78	86.72	59.93

Table 6. Mention Detection Scores on the test datasets.

Table 6 shows the official mention detection scores on the test datasets, which could be regarded as the performance upper bounds (MUC metric) of the mention clustering component. Taking the mention detection results as a basis, the mention clustering component could achieve roughly 60.88 (18.05/29.65), 86.65 (44.65/51.53), and 84.83 (50.84/59.93) for the Arabic, Chinese, and English data respectively. It seems that the performance of the whole system is highly bottlenecked by that of the mention detection component. However, it may not be true as the task requires removing singleton mentions that do not refer to any other mentions. To examine how

singleton mentions affect the final scores, we conducted additional experiments on the development datasets. Table 7 shows the mention detection scores on the dev datasets. When we include the singletons, the mention detection scores become 59, 63.75, and 71.27 from 31.46, 53.99, and 59.16 for the three language datasets respectively. They are reasonable and close to those that we can get at the mention clustering component. These analyses tell us that the requirement of removing singletons for scoring may deserve further study. At the same time, we realize that to get better performance we may need to re-design the feature sets (e.g. including more useful features like gender and number) and try some more powerful machine learning algorithms such as linear classification or Tree CRF (Bradley and Guestrin, 2010).

Language	Recall		Precision		F1	
	-Sing	+Sing	-Sing	+Sing	-Sing	+Sing
Arabic	19.42	47.58	82.88	77.61	31.46	59
Chinese	39.05	53.78	87.43	78.24	53.99	63.75
English	44.9	65.2	86.67	78.58	59.16	71.27

Table 7. Mention Detection Scores on the development (Dev) datasets. “-Sing” means without singletons, which is required by the task specification, while “+Sing” means including singletons.

	With gold mention boundaries (39.26)			With gold mentions (50.65)		
	ARB	CHN	ENG	ARB	CHN	ENG
<b>MUC</b>	<b>11.30</b>	<b>38.70</b>	<b>38.21</b>	<b>33.31</b>	<b>66.13</b>	<b>60.45</b>
<b>B-CUBED</b>	<b>54.25</b>	<b>59.27</b>	<b>59.51</b>	<b>53.74</b>	<b>66.84</b>	<b>57.18</b>
CEAF (M)	33.68	41.06	39.30	42.25	57.50	47.82
<b>CEAF (E)</b>	<b>28.84</b>	<b>31.86</b>	<b>31.39</b>	<b>34.81</b>	<b>46.83</b>	<b>36.58</b>
BLANC	51.46	61.47	61.33	57.96	73.47	67.12
MD Score	29.78	51.90	51.08	52.58	77.73	72.75
<b>Official Score</b>	<b>31.46</b>	<b>43.28</b>	<b>43.04</b>	<b>40.62</b>	<b>59.93</b>	<b>51.40</b>

Table 8. F1 scores of the two supplementary submissions with additional gold mention boundaries and gold mentions respectively.

Besides the official submission for the task with predicted data, we also provide two supplementary submissions with gold mention boundaries and gold mentions respectively. Table 8 summarizes the scores of these two submissions.

With gold mentions, our official system does achieve better performance with gain of 8.77 (50.65-41.88). On Chinese data, we get the highest score 61.61. However, the system performs worse when the gold mention boundaries are available. The F1 score drops 2.62 from 41.88 to 39.26. We guess that more candidate mentions bring more difficulties for the maximal entropy classifier to make decisions. The best-of-all strategy may not be a good choice when a large number of candidates are available. More efforts are required to explore the real reason behind the results.

## 6 Conclusions

In this paper, we describe our system for the CoNLL-2012 shared task – Modeling Multilingual Unrestricted Coreference in OntoNotes (closed track). Our system was built on machine learning strategy with a pipeline architecture, which integrated two cascaded components: mention detection and mention clustering. The system relies on successful syntactic analyses, which means that only valid sub-structures of sentences are considered as potential mentions.

Due to limited time and resources, we had not conducted thorough enough experiments to derive optimal solutions, but the system and the involvement in this challenge do provide a good foundation for further study. It’s a success for us to finish all the submissions on time. In the future, we plan to focus on those mentions that do not correspond to a syntactic structure and consider introducing virtual nodes for them. We may also explore different strategies when linking an anaphor and its antecedent. In addition, maximal entropy may not be good enough for this kind of task. Therefore, we also plan to explore other powerful algorithms like large linear classification and tree CRF (Bradley and Guestrin, 2010; Ram and Devi, 2012) in the future.

## Acknowledgments

This research was funded by HeNan Provincial Research Project on Fundamental and Cutting-Edge Technologies (No. 112300410007). We used the Amazon Elastic Compute Cloud (Amazon EC2) web service in our experiments. We thank Amazon.com for providing such great service for not only industrial applications, but also academic research.

## References

- Amit Bagga and Breck Baldwin. 1998. Algorithms for Scoring Coreference Chains. In Proceedings of the First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference.
- Adam L. Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra. 1996. A Maximal Entropy Approach to Natural Language Processing. *Computational Linguistics*, 22(1):39-42.
- Joseph K. Bradley and Carlos Guestrin. 2010. Learning Tree Conditional Random Fields. In Proceedings of the International Conference on Machine Learning (ICML-2010).
- Nancy Chinchor. 2001. Message Understanding Conference (MUC) 7. In LDC2001T02.
- Nancy Chinchor and Beth Sundheim. 2003. Message Understanding Conference (MUC) 6. In LDC2003T13.
- G.G. Doddington, A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassell, and R. Weischedel. 2000. The Automatic Content Extraction (ACE) Program: Tasks, Data, and Evaluation. In Proceedings of LREC-2000.
- Heeyoung Lee, Yves Peirsman, Angei Chang, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky. 2011. Stanford's Multi-Pass Sieve Coreference Resolution System at the CoNLL-2011 Shared Task. In Proceedings of the 15<sup>th</sup> Conference on Computational Natural Language Learning: Shared Task.
- Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing.
- George A. Miller. 1995. WordNet: a Lexical Database for English. *Communications of the ACM*. 38(11): 39-41.
- Sameer Pradhan, Eduard Hovy, Mitch Marcus, Martha Palmer, Lance Ramshaw, Ralph Weischedel. 2007. OntoNotes: A Unified Relational Semantic Representation. *International Journal of Semantic Computing*, 1(4): 405-419.
- 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): Shared Task.
- 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 15<sup>th</sup> Conference on Computational Natural Language Learning: Shared Task, pp. 1-27.
- Vijay Sundar Ram R. and Sobha Lalitha Devi. 2012. Coreference Resolution Using Tree CRFs. In Proceedings of the 13<sup>th</sup> Conference on Computational Linguistics and Intelligent Text Processing (CICLing-2012).
- Marta Recasens and Eduard Hovy. 2011. Blanc: Implementing the rand index for coreference evaluation. *Natural Language Engineering*, 17(4): 485-510.
- Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrase. *Computational Linguistics*, 27(4): 521-544.
- M. Vilain, J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman. 1995. A model theoretic coreference scoring scheme. In Proceedings of the Sixth Message Understanding Conference (MUC-6).
- Le Zhang. 2006. Maximum Entropy Modeling Toolkit for Python and C++. Software available at [http://homepages.inf.ed.ac.uk/s0450736/maxent\\_toolkit.html](http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html).