An Effective Hybrid Machine Learning Approach for Coreference Resolution

Feiliang Ren
Natural Language Processing Lab
College of Information Science and Engineering
Northeastern University, P.R.China
renfeiliang@ise.neu.edu.cn

Jingbo Zhu
Natural Language Processing Lab
College of Information Science and Engineering
Northeastern University, P.R.China
zhujingbo@ise.neu.edu.cn

Abstract

We present a hybrid machine learning approach for coreference resolution. In our method, we use CRFs as basic training model, use active learning method to generate combined features so as to make existed features used more effectively; at last, we proposed a novel clustering algorithm which used both the linguistics knowledge and the statistical knowledge. We built a coreference resolution system based on the proposed method and evaluate its performance from three aspects: the contributions of active learning; the effects of different clustering algorithms; and the resolution performance of different kinds of NPs. Experimental results show that additional performance gain can be obtained by using active learning method; clustering algorithm has a great effect on coreference resolution's performance and our clustering algorithm is very effective; and the key of coreference resolution is to improve the performance of the normal noun's resolution, especially the pronoun's resolution.

1 Introduction

Coreference resolution is the process of determining whether two noun phrases (NPs) refer to the same entity in a document. It is an important task in natural language processing and can be classified into pronoun phrase (denoted as ***PRO***) resolution, normal noun phrase (denoted as ***NOM***) resolution, and named noun phrase (denoted as ***NAM***) resolution. Machine learning approaches recast this problem as a classification task based on constraints that are learned from an annotated corpus. Then a separate clustering mechanism is used to construct a partition on the set of NPs.

Previous machine learning approaches for coreference resolution (Soon et al, 2001; Ng et al, 2002; Florian et al, 2004, etc) usually selected a machine learning approach to train a classification model, used as many as possible features for the training of this classification model, and finally used a clustering algorithm to construct a partition on the set of NPs based on the statistical data obtained from trained classification model. Their experimental results showed that different kinds of features had different contributions for system's performance, and usually the more features used, the better performance obtained. But they rarely focused on how to make existed features used more effectively; besides, they proposed their own clustering algorithm respectively mainly used the statistical data obtained from trained classification model, they rarely used the linguistics knowledge when clustering different kinds of NPs. Also, there were fewer experiments conducted to find out the effect of a clustering algorithm on final system's performance.

In this paper, we propose a new hybrid machine learning method for coreference resolution. We use NP pairs to create training examples; use CRFs as a basic classification model, and use active learning method to generate some combined features so as to make existed features used more effectively; at last, cluster NPs into entities by a novel cascade clustering algorithm.

The rest of the paper is organized as follows. Section 2 presents our coreference resolution system in detail. Section 3 is our experiments and discussions. And at last, we conclude our work in section 4.

2 Coreference Resolution

There are three basic components for a coreference resolution system that uses machine learning approach: the training set creation, the feature selection, and the coreference clustering algorithm. We will introduce our methods for these components respectively as follows.

2.1 Training Set Creation

Previous researchers (Soon et al., 2001, Vincent Ng et al., 2002, etc) took different creation strategies for positive examples and negative examples. Because there were no experimental results showed that these kinds of example creation methods were helpful for system's performance, we create both positive examples and negative examples in a unified NP pair wise manner.

Given an input NP chain of an annotated document, select a NP in this NP chain from left to right one by one, and take every of its right side's NP, we generate a positive example if they refer to the same entity or a negative example if they don't refer to the same entity. For example, there is a NP chain n1-n2-n3-n4 found in document, we will generate following training examples: (n1-n2, ± 1), (n1-n3, ± 1), (n1-n4, ± 1), (n2-n3, ± 1), (n2-n4, ± 1), and (n3-n4, ± 1). Where $+1$ denotes that this is a positive example, and -1 denotes that this is a negative example.

2.2 Feature Sets

In our system, two kinds of features are used. One is atomic feature, the other is combined feature. We define the features that have only one generation condition as atomic features, and define the union of some atomic features as combined features.

2.2.1 Atomic Features

All of the atomic features used in our system are listed as follows.

String Match Feature (denoted as *Sm*): Its possible values are *exact, left, right, included, part, alias,* and *other*. If two NPs are exactly string matched, return *exact*; if one NP is the left substring of the other, return *left*; if one NP is the right substring of the other, return *right*; if all the characters in one NP are appeared in the other but not belong to set {*left, right*}, return *included*; if some (not all) characters in one NP are appeared in the other, return *part*; if one NP is the alias of the other, return *alias*; if two NPs don't have any common characters, return *other*.

Lexical Similarity Features (denoted as *Ls*): compute two NP's similarity and their head words' similarity using following formula 1.

$$Sim(n_1, n_2) = \frac{2 \times SameChar(n_1, n_2)}{Len(n_1) + Len(n_2)} \quad (1)$$

Here $SameChar(n_1, n_2)$ means the common characters' number in n_1 and n_2 ; $Len(n_i)$ is the total characters' number in n_i .

Edit Distance Features (denoted as *Ed*): compute two NP's edit distance and their head words' edit distance (Wagner and Fischer, 1974), and the possible values are *true* and *false*. If the edit distance of two NPs (or the head words of these two NPs) are less than or equal to 1, return *true*, else return *false*.

Distance Features (denoted as *Dis*): distance between two NPs in words, NPs, sentences, paragraphs, and characters.

Length Ratio Features (denoted as *Lr*): the length ratio of two NPs, and their head words. Their possible values belong to the range $(0,1]$.

NP's Semantic Features (denoted as *Sem*): the POSs of two NPs' head words; the types of the two NPs (***NAM***, ***NOM*** or ***PRO***); besides, if one of the NP is ***PRO***, the semantic features will also include this NP's gender information and plurality information.

Other Features (denoted as *Oth*): whether two NPs are completely made up of capital English characters; whether two NPs are completely made up of lowercase English characters; whether two NPs are completely made up of digits.

2.2.2 Combined Features Generated by Active Learning

During the process of model training for coreference resolution, we found that we had very fewer available resources compared with previous researchers. In their works, they usually had some extra knowledge-based features such as alias table, abbreviation table, *wordnet* and so on; or they had some extra in-house analysis tools such as proper name parser, chunk parser, rule-based shallow coreference resolution parser, and so on (Hal Daume III, etc, 2005; R.Florian, etc, 2004; Vincent Ng, etc, 2002; etc). Although we also collected some aliases and abbreviations, the amounts are very small compared with previous researchers'. We hope we can make up for this by making existed features used more effectively by active learning method.

Formally, active learning studies the closed-loop phenomenon of a learner selecting actions or making queries that influence what data are added to its training set. When actions or queries are selected properly, the data requirements for some problems decrease drastically (Angluin, 1988; Baum & Lang, 1991). In our system, we used a pool-based active learning framework that is similar as Manabu Sassano (2002) used, this is shown in figure 1.

```
1. Build an initial classifier
2. While teacher can correct examples based on feature combinations
    a) Apply the current classifier to training examples
    b) Find m most informative training examples
    c) Have two teachers correct these examples based on feature combinations
    d) Add the feature combinations that are used by both of these two teachers to feature sets in CRFs and train a new classifier.
```

Figure 1: Our Active Learning Framework

In this active learning framework, an initial classifier is trained by CRFs [[1]] that uses only atomic features, and then two human teachers are asked to correct some selected wrong classified examples independently. During the process of correction, without any other available information, system only shows the examples that are made up of features to the human teachers; then these two human teachers have to use the information of some atomic features' combinations to decide whether two NPs refer to the same entity. We record all these atomic features' combinations that used by both of these human teachers, and take them as combined features.

For example, if both of these human teachers correct a wrong classified example based on the knowledge that "if two NPs are left substring matched, lexical similarity feature is greater than 0.5, I think they will refer to the same entity", the corresponding combined feature would be described as: "*Sm(NPs)-Ls(NPs)*", which denotes the human teachers made their decisions based on the combination information of "*String Match Features*" and "*Lexical Similarity Features*".

```
1. Select all the wrong classified examples whose
   CRFs' probability belongs to range [0.4, 0.6]
2. Sort these examples in decreasing order.
3. Select the top m examples
```

Figure 2: Selection Algorithm

In figure 1, "*information*" means the valuable data that can improve the system's performance after correcting their classification. The selection algorithm for "*informative*" is the most important component in an active learning framework. We designed it from the degree of correcting difficulty. We know 0.5 is a critical value for an example's classification. For a wrong classified example, the closer its probability value to 0.5, the easier for us to correct its classification. Following this, our selection algorithm for "informative" is designed as shown in figure 2.

When add new combined features won't lead to a performance improvement, we end active learning process. Totally we obtained 21 combined features from active learning. Some of them are listed in table 1.

Table 1: Some Combined Features

Sm(NPs)-Sm(HWs)-Ls(NPs)-Ls(HWs)
Sm(NPs)-Sm(HWs)-Ls(NPs)
Sm(NPs)-Sm(HWs)-Ls(HWs)
Sm(NPs)-Sm(HWs)-Lr(NPs)-Lr(HWs)
Sm(NPs)-Sm(HWs)-Lr(NPs)
Sm(NPs)-Sm(HWs)-Sem(HW1)-Sem(HW2)
Sm(NPs)-Sm(HWs)-Sem(NP1)-Sem(NP2)
Sm(NPs)-Sm(HWs)-Lr(HWs)
……

Here "*Sm(NPs)*" means the string match feature's value of two NPs, "*Sm(HWs)*" means the string match feature's value of two NPs' head words. "*HWs*" means the head words of two NPs. Combined feature "*Sm(NPs)-Sm(HWs)-Ls(NPs)*" means when correcting a wrong classified example, both these human teachers made their decisions based on the combination information of *Sm(NPs)*, *Sm(HWs)*, and *Ls(NPs)* . Other combined features have the similar explanation.

[1] http://www.chasen.org/~taku/software/CRF++/

And at last, we take all the atomic features and the combined features as final features to train the final CRFs classifier.

2.3 Clustering Algorithm

Formally, let $\{m_i : 1 \le i \le n\}$ be n NPs in a document. Let us define $S_a = \{N_{a1}, ..., N_{af}\}$ the set of NPs whose types are all ***NAM***s; define $S_o = \{N_{o1}, ..., N_{og}\}$ the set of NPs whose types are all ***NOM***s; define $S_p = \{N_{p1}, ..., N_{pk}\}$ the set of NPs whose types are all ***PRO***s. Let $g : i \mapsto j$ be the map from NP index i to entity index j. For a NP index $k (1 \le k \le n)$, let us define $J_k = \{g(1), ..., g(k-1)\}$ the set of indices of the partially-established entities before clustering m_k, and $E_k = \{e_t : t \in J_k\}$, the set of the partially-established entities. Let e_{ij} be the $j-th$ NP in $i-th$ entity. Let $prob(m_i, m_j)$ be the probability that m_i and m_j refer to the same entity, and $prob(m_i, m_j)$ can be trained from CRFs.

Given that E_k has been formed before clustering m_k , m_k can take two possible actions: if $g(k) \in J_k$, then the active NP m_k is said to *link* with the entity $e_{g(k)}$; otherwise it *starts* a new entity $e_{g(k)}$.

In this work, $P(L = 1 | E_k, m_k, A = t)$ is used to compute the link probability, where $t \in J_k$, L is 1 iff m_k links with e_t; the random variable A is the index of the partial entity to which m_k is linking.

Our clustering algorithm is shown in figure 3. The basic idea of our clustering algorithm is that ***NAM*s, *NOM***s and ***PRO***s have different abilities starting an entity. For ***NAM***s, they are inherent antecedents in entities, so we start entities based on them first.

For ***NOM***s, they have a higher ability of acting as antecedents in entities than ***PRO***s, but lower than ***NAM***s. We cluster them secondly, and add a ***NOM*** in an existed entity as long as their link probability is higher than a threshold. And during the process of the link probabilities computations, we select a NP in an existed entity carefully, and take these two NPs' link probability as the link probability between this ***NOM*** and current entity. The selection strategy is to try to make these link probabilities have the greatest distinction.

And for ***PRO*s**, they have the lowest ability of acting as antecedents in entities, most of the time, they won't be antecedents in entities; so we cluster them into an existed entity as long as there is a non-zero link probability.

3 Experiments and Discussions

Our experiments are conducted on Chinese EDR (Entity Detection and Recognize) &EMD (Entity Mention Detection) corpora from LDC. These corpora are the training data for ACE (Automatic Content Extraction) evaluation 2004 and ACE evaluation 2005. These corpora are annotated and can be used to train and test the coreference resolution task directly.

```
Input: $M = \{m_i : 1 \le i \le n\}$
Output: a partition $E$ of the set $M$
Initialize: $H^0 \leftarrow \{e_i = \{\{m_i : m_i \in S_a\}\}\}$
if $\exists m_x \in e_c \cap m_y \in e_d$, $c \ne d$, and $m_x$ is alias of $m_y$, then $H' \leftarrow H \setminus \{e_d\} \cup \{e_c \cup e_d\}$
foreach $m_k \in S_o$ that hasn't been clustered
    if $e_{k0}$ is NAM and $\exists d$ makes $\sigma(e_{td}, NOM) \ne 0$
        $P = \arg\max_{e_t} \underset{|d-k|=\min}{\{prob(m_k, e_{td}) \times \sigma(e_{td}, NOM)\}}$
    esleif $e_{k0}$ is NAM and $\forall d, \sigma(e_{td}, NOM) == 0$
        $P = \arg\max_{e_t} \underset{|d-k|=\min}{\{prob(m_k, e_{td}) \times \sigma(e_{td}, NAM)\}}$
    esleif $e_{k0}$ is NOM
        $P = \arg\max_{e_t} \; prob(m_k, e_{t0})$
    if $P \ge \theta$, $H' \leftarrow H \setminus \{e_t\} \cup \{e_t \cup \{m_k\}\}$
    else $H' \leftarrow H \cup \{m_k\}$
foreach $m_k \in S_p$ that hasn't been clustered
    $P = \arg\max_{m \in e_t} prob(m, m_k)$
    if $P > 0$, $H' \leftarrow H \setminus \{e_t\} \cup \{e_t \cup \{m_k\}\}$
    else $H' \leftarrow H \cup \{m_k\}$
return $H$
```

Figure 3: Our Clustering Algorithm

In ACE 2004 corpus, there are two types of documents: paper news (denoted as newswire) and broadcast news (denoted as broadca); for ACE 2005 corpus, a new type added: web log documents (denoted as weblogs). Totally there are 438 documents in ACE 2004 corpus and 636 documents in ACE 2005 corpus. We randomly divide these two corpora into two parts respectively, 75% of them for training CRFs model, and 25% of them for test. By this way, we get 354 documents for training and 84 documents for test in ACE 2004 corpus; and 513 documents for training and 123 documents for test in ACE 2005 corpus.

Some statistics of ACE2005 corpus and ACE2004 corpus are shown in table 2.

Our experiments were classified into three groups. Group 1 (denoted as ***ExperimentA***) is designed to evaluate the contributions of active learning for the system's performance. We developed two systems for ***ExperimentA***, one is a system that used only the atomic features for CRFs training and we took it as a baseline system, the other is a system that used both the atomic features and the combined features for CRFs training and we took it as our final system. The experimental results are shown in table 3 and table 4 for different corpus respectively. Bold font is the results of our final system, and normal font is the results of baseline system. Here we used the clustering algorithm as described in figure 3.

Group 2 (denoted as ***ExperimentB***) is designed to investigate the effects of different clustering algorithm for coreference resolution. We implemented another two clustering algorithms: *algorithm1* that is proposed by Ng et al. (2002) and *algorithm2* that is proposed by Florian et al. (2004). We compared the performance of them with our clustering algorithm and experimental results are shown in table 5.

Group 3 (denoted as ***ExperimentC***) is designed to evaluate the resolution performances of different kinds of NPs. We think this is very helpful for us to find out the difficulties and bottlenecks of coreference resolution; and also is helpful for our future work. Experimental results are shown in table 6.

In ***ExperimentB*** and ***ExperimentC***, we used both atomic features and combined features for CRFs classification model training. And in table5, table6 and table7, the data before "/" are experimental results for ACE2005 corpus and the data after "/" are experimental results for ACE2004 corpus.

In all of our experiments, we use recall, precision, and F-measure as evaluation metrics, and denoted as R, P, and F for short respectively.

Table 2: Statistics of ACE2005/2004 Corpora

	Training	Test
# of all documents	513/354	123/84
# of broadca	204/204	52/47
# of newswire	229/150	54/47
#of weblogs	80/0	17/0
# of characters	248972/164443	55263/35255
# of NPs	28173/18995	6257/3966
# of entities	12664/8723	2783/1828
# of ***neg*** examples	722919/488762	142949/89894
# of ***pos*** examples	72000/44682	15808/8935

Table3: ***ExperimentA*** for ACE2005 Corpora

	R	P	F
broadca	**79.0**/76.2	**75.4**/72.9	**77.2**/74.5
newswire	**73.2**/72.9	**68.7**/67.8	**70.9**/70.3
weblogs	**72.3**/68.5	**65.5**/63.3	**68.8**/65.8
total	**75.4**/73.7	**70.9**/69.3	**73.1**/71.4

Table4: ***ExperimentA*** for ACE2004 Corpora

	R	P	F
broadca	**74.7**/71.0	**72.4**/68.9	**73.5**/69.9
newswire	**77.7**/73.1	**73.0**/68.6	**75.2**/70.7
Total	**76.2**/72.0	**72.7**/68.7	**74.4**/70.4

Table5: ***ExperimentB*** for ACE2005/2004 Corpora

	R	P	F
algorithm1	61.0/63.5	59.5/62.8	60.2/63.2
algorithm2	61.0/62.4	60.7/62.8	60.9/62.6
Ours	**75.4/76.2**	**70.9/72.7**	**73.1/74.4**

Table6: ***ExperimentC*** for ACE2005/2004 Corpora

	R	P	F
NAM	80.5/81.4	77.9/79.2	79.2/80.1
NOM	62.6/62.5	54.4/56.8	58.2/59.5
PRO	28.4/29.8	22.7/24.0	25.2/26.6

From table 3 and table 4 we can see that the final system's performance made a notable improvement compared with the baseline system in both corpora. We know the only difference of these two systems is whether used active learning method. This indicates that by using active learning method, we make the existed features used more effectively and obtain additional performance gain accordingly. One may say that even without active learning method, he still can add some combined features during CRFs model training. But this can't guarantee it would make a performance improvement at anytime. Active learning method provides us a way that makes this combined features' selection process goes in a proper manner. Generally, a system can obtain an obvious performance improvement after several active learning iterations. We still noticed that the contributions of active learning for different kinds of documents are different. In ACE04 corpus, both kinds of documents' performance obtained almost equal improvements; in ACE05 corpus, there is almost no performance improvement for newswire documents, but broadcast documents' performance and web log documents' performance obtained greater improvements. We think this is because for different kinds of documents, they have different kinds of correcting rules (these rules refer to the combination methods of atomic features) for the wrong classified examples, some of these rules may be consistent, but some of them may be conflicting. Active learning mechanism will balance these conflicts and select a most appropriate global optimization for these rules. This can also explain why ACE04 corpus obtains more performance improvement than ACE05 corpus, because there are more kinds of documents in ACE05 corpus, and thus it is more likely to lead to rule conflicts during active learning process.

Experimental results in table 5 show that if other experimental conditions are the same, there are obvious differences among the performances with different clustering algorithms. This surprised us very much because both *algorithm1* and *algorithm2* worked very well in their own learning frameworks. We know R.Florian et al. (2004) first proposed *algorithm2* using maximum entropy model. Is this the reason for the poor performance of *algorithm2* and *algorithm1*? To make sure this, we conducted other experiments that changed the CRFs model to maximum entropy model [2] without changing any other conditions and the experimental results are shown in table 7.

The experimental results are the same: our clustering algorithm achieved better performance. We think this is mainly because the following reason, that in our clustering algorithm, we notice the fact that different kinds of NPs have different abilities of acting as antecedents in an entity, and take different clustering strategy for them respectively, this is obvious better than the methods that only use statistical data.

Table7: ***ExperimentB*** for ACE2005/2004 Corpora with ME Model

	R	P	F
algorithm1	48.9/48.3	44.2/50.3	46.4/49.3
algorithm2	57.4/59.5	52.3/61.4	54.7/60.4
Ours	**68.1/69.8**	**65.7/72.6**	**66.9/71.2**

We also noticed that the experimental results with maximum entropy model are poorer than with CRFs model. We think this maybe because that the combined features are obtained under CRFs model, thus they will be more suitable for CRFs model than for maximum entropy model, that is to say these obtained combined features don't play the same role in maximum entropy model as they do in CRFs model.

Experimental results in table 6 surprised us greatly. ***PRO*** resolution gets so poor a performance that it is only about 1/3 of the ***NAM*** resolution's performance. And ***NOM*** resolution's performance is also pessimistic, which reaches about 80% of the ***NAM*** resolution's performance. After analyses we found this is because there is too much confusing information for ***NOM***'s resolution and ***PRO***'s resolution and system can hardly distinguish them correctly with current features description for an example. For example, in a Chinese document, a ***NOM*** "*总统*" (means *president*) may refer to a person *A* at sometime, but refer to person *B* at another time, and there is no enough information for system to distinguish *A* and *B*. It is worse for ***PRO*** resolution because a ***PRO*** can refer to any ***NAM*** or ***NOM*** from a very long distance, there is little information for the system to distinguish which one it really refers to. For example, two ***PRO***s that both of whom are "*他*" (means *he*) , one refers to person *A*, the other refers to person *B*, even our human can hardly distinguish them, not to say the system.

Fortunately, generally there are more ***NAMs*** and ***NOMs*** in a document, but less ***PROs***. If they have similar amounts in a document, you can image how poor the performance of the coreference resolution system would be.

4 Conclusions

In this paper, we present a hybrid machine learning approach for coreference resolution task. It uses CRFs as a basic classification model and uses active learning method to generate some combined

[2] http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html

features to make existed features used more effectively; and we also proposed an effective clustering algorithm that used both the linguistics knowledge and the statistical knowledge. Experimental results show that additional performance gain can be obtained by using active learning method, clustering algorithm has a great effect on coreference resolution's performance and our clustering algorithm is very effective. Our experimental results also indicate the key of coreference resolution is to improve the performance of the ***NOM*** resolution, especially the ***PRO*** resolution; both of them remain challenges for a coreference resolution system.

Acknowledgments

We used the method proposed in this paper for Chinese EDR (Entity Detection and Recognition) task of ACE07 (Automatic Content Extraction 2007) and achieved very encouraging result.

And this work was supported in part by the National Natural Science Foundation of China under Grant No.60473140; the National 863 High-tech Project No.2006AA01Z154; the Program for New Century Excellent Talents in University No.NCET-05-0287; and the National 985 Project No.985-2-DB-C03.

Reference

Andrew Kachites McCallum and Kamal Nigam, 1998. Employing EM and pool-based active learning for text classification. In Proceedings of the Fifteenth International Conference on Machine Learning, pp 359-367

Cohn, D., Grahramani, Z., & Jordan, M.1996. Active learning with statistical models. Journal of Artificial Intelligence Research, 4. pp 129-145

Cynthia A.Thompson, Mary Leaine Califf, and Raymond J.Mooney. 1999. Active learning for natural language parsing and information extraction. In Proceedings of the Seventeenth International Conference on Machine Learning, pp 406-414

Hal Daume III and Daniel Marcu, 2005, A large-scale exploration of effective global features for a joint entity detection and tracking model. Proceedings of HLT/EMNLP, 2005

http://www.nist.gov/speech/tests/ace/ace07/doc, The ACE 2007 (ACE07) Evaluation Plan, Evaluation of the Detection and Recognition of ACE Entities, Values, Temporal Expressions, Relations and Events

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In International Conference on Machine Learingin(ICML01)

Lafferty, J., McCallum, A., & Pereira, F. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. Proc. ICML

Manabu Sassano. 2002. An Empirical Study of Active Learning with Support Vector Machines for Japanese Word Segmentation. Proceeding of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), 2002, pp 505-512

Min Tang, Xiaoqiang Luo, Salim Roukos.2002. Active Learning for Statistical Natural Language Parsing. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), 2002, pp.120-127.

Pinto, D., McCallum, A., Lee, X., & Croft, W.B. 2003. combining classifiers in text categorization. SIGIR' 03: Proceedings of the Twenty-sixth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.

Radu Florian, Hongyan Jing, Nanda Kambhatla and Imed Zitouni, “Factoring Complex Models: A Case Study in Mention Detection”, in Procedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, pages 473-480, Sydney, July 2006.

R Florian, H Hassan et al. 2004. A statistical model for multilingual entity detection and tracking. In Proc. Of HLT/NAACL-04, pp1-8

Sha, F., & Pereira, F. 2003. Shallow parsing with conditional random fields. Proceedings of Human Language Technology, NAACL.

Simon Tong, Daphne Koller. 2001. Support Vector Machine Active Learning with Applications to Text Classification. Journal of Machine Learning Research,(2001) pp45-66.

V.Ng and C.Cardie. 2002. Improving machine learning approaches to coreference resolution. In Proceedings of the ACL'02, pp.104-111.

W.M.Soon, H.T.Ng, et al.2001. A Machine Learning Approach to Coreference Resolution of Noun Phrases. Computational Linguistics, 27(4):521-544