Computational Linguistics and Chinese Language Processing
Vol. 12, No. 4, December 2007, pp. 459-484

A Knowledge-Based Approach for Unsupervised Chinese Coreference Resolution

Grace Ngai[*] and Chi-Shing Wang[*]

Abstract

Coreference resolution is the process of determining the entity that noun phrases refer to. A great deal of research has been done on this task in English, using approaches ranging from those based on linguistics to those based on machine learning. In Chinese, however, much less work has been done in this area. One reason for this is the lack of resources for Chinese natural language processing. This paper presents a knowledge-based, unsupervised clustering algorithm for Chinese coreference resolution that maximizes performance using freely and easily available resources. Experiments to demonstrate the efficacy of such an approach are performed on two data sets: TDT3 and ACE05, and the ACE value coreference resolution results achieved through our approach are 52.5% and 55.2% respectively. An oracle experiment using gold standard noun phrases achieved even more impressive results of 77.0% and 76.4%. To analyze the causes of errors, this paper also looks into false alarms and misses in documents.

Keywords: Coreference Resolution, Modified K-means Clustering, Stacked Transformation-based Learning, Unsupervised Learning

1. Introduction

Noun phrase (NP) coreference resolution is an important subtask in natural language processing (NLP) applications such as text summarization, information extraction, data mining, and question answering. The subject has attracted much attention in recent years, although much more in regards to the English language than to the Chinese language, and has been included as a subtask in the MUC (Message Understanding Conferences) and ACE (Automatic Content Extraction) programs. NP coreference resolution is the process of detecting noun phrases in a document and determining whether these noun phrases refer to the same entity. As defined in ACE [2005], an entity is “an object or set of objects in the world.”

[*] Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong
Tel: +852-27667279 Fax: +852-22154652
E-mail: {csgngai; cscswang} @comp.polyu.edu.hk

[Received April 11, 2007; Revised August 31, 2007; Accepted September 10, 2007]

Phrases that refer to an entity are known as *mentions*, which may be either anaphors or antecedents. An anaphor is an expression that refers back to something mentioned previously in a discourse, and the something that the anaphor refers back to is its antecedent. Thus, in the passage in Figure 1, the term 克林頓總統 *(President Clinton)* in the second line of the passage is an anaphoric reference to its antecedent 克林頓 *(Clinton)*, which begins the passage. This anaphor 克林頓總統 *(President Clinton)* is in turn the antecedent of the second 他 *(he)*. All three of these terms, 克林頓 *(Clinton)*, 克林頓總統 *(President Clinton)*, and the second 他 *(he)*, are mentions of the same entity and refer, of course, to former U.S. president Bill Clinton. Generally speaking, it is a simple matter for human beings to quickly and accurately identify such coreferences. However, the cues that are used by humans for noun phrase coreference resolution are not easily transferred to the computer. Even in English, the most heavily studied language, the accuracy of automated NP coreference resolution is currently unsatisfactory. In Chinese, which has its own particular characteristics and difficulties, NP coreference resolution is a topic where even more work remains to be done.

[克林頓$_1$]說，華盛頓將逐步落實對[韓國$_2$]的經濟援助。[金大中$_3$]對[克林頓$_1$]的講話報以掌聲。[他$_3$]說：「[克林頓總統$_1$]在會談中重申，[他$_1$]堅定地支持[韓國$_2$]擺脫經濟危機。」

[Clinton$_1$] said that Washington would progressively follow through on economic aid to [Korea$_2$]. [Kim Dae-Jung$_3$] applauded.

Figure 1. An excerpt from the text, with coreferring noun phrases annotated. English translation in italics.

Central to the development of efficient and reliable approaches to automatic NP coreference resolution is the issue of what features should be used to identify the coreference. Ng and Cardie [2002b] listed 53 features, including gender agreement, number agreement, head noun matches, semantic class agreement, positional information, contextual information, apposition, abbreviation, and others. At one extreme, efficiency alone forbids the use of all of these features; at the other, no single linguistic feature is completely reliable. With the careful selection of combinations of suitable features, there may be a tradeoff to be made between the efficiency of using fewer features and the accuracy to be obtained from using more. Before such an approach can be tested, there are a number of difficulties that need to be addressed, not the least of which being the limitations of currently available NLP applications and ontologies used in coreference resolution. For example, applications, such as named entity recognition, and ontologies, such as WordNet and HowNet, are currently used to identify features such as semantic class. However, these identifications are not always accurate, especially where new terms, domains or languages are concerned. Domain adaptation then becomes an issue, or ontology coverage becomes less than ideal.

As already mentioned, Chinese NP coreference resolution involves certain difficulties which are not found in the English language. First, from the point of view of NLP, Chinese suffers from a lack of usable morphological and orthographic features. For example, in English, morphological features such as number agreement can indicate coreference, and this contributes to the accuracy of automatic part-of-speech (POS) tagging. Chinese, however, does not use morphological changes to indicate number agreement. As for orthography, Chinese does not, for example, use capitalization whereas English can make use of capitalization to mark elements such as proper names, place names, and abbreviations. Perhaps the greatest difficulty of written Chinese is that, unlike English, it does not mark word boundaries. Word segmentation is thus required, yet various segmentations of even a simple Chinese sentence may produce a variety of meanings, making a range of NLP tasks, for example, POS tagging, highly problematic.

A second important problem faced in Chinese NP coreference resolution is a lack of Chinese corpora (specifically coreference data sets) that are either free of charge, freely available, or sufficiently free of error for use as benchmarking data sets for training and for measuring performance. The principal reason for this is that building a reasonably large coreference corpus is a labor-intensive task, especially with regard to annotation, which cannot be undertaken by any but the largest institutions. For example, the ACE corpus from the ACE program is large and is annotated for a very comprehensive number of grammatical, semantic, and discourse features. It is available, at a cost, for use in problems involving coreference resolution.

In this paper, we propose an approach to Chinese NP coreference resolution that, with small amounts of training and time investment, can accurately identify chains of coreference in unannotated texts. The approach first uses an automatic, Penn Treebank trained parser [Zhang *et al*. 2003] to identify mentions and then filters out those that are not likely to refer to an entity using heuristic rules based on POS information. The resulting mentions are then linked into possible chains using a clustering algorithm and specific linguistic features. The advantages of this approach are, first, that the proposed algorithm is unsupervised and therefore requires no training set, relying instead on word lists, dictionaries, and gazetteers that are freely available and easily compliable; and second, that features may be easily added or deleted. This makes our method suitable for scenarios where such a system needs to be quickly compiled for a new genre or language, where pre-existing resources are not adequate. After describing the proposed system, we will demonstrate the efficacy of our algorithm by achieving satisfactory performance on two different corpora.

The rest of the paper is laid out as follows: Section 2 gives an overview of the previous work in this area. Section 3 describes our algorithm, Section 4 introduces the experimental setup, and Section 5 gives details of our evaluation. Section 6 contains the analysis of our results, and is followed by our conclusions.

2. Previous Work

In this section, we will start with a description of the most common approaches to coreference resolution and contrast them with the approach that we will be taking. Since we will be concentrating on the problem in Chinese, we will also include an introduction to the work conducted to date on Chinese NP coreference resolution.

2.1 Supervised Machine Learning Approaches

Much of the previous work in NP coreference resolution has used statistical, machine learning approaches, and one of the most frequently used approaches is that of binary classification. These algorithms link up mentions into coreference chains by first identifying an anaphoric noun phrase, and then using a predetermined number of features in an effort to identify the best antecedent for each mention. Soon *et al.* [2001] proposed a 12-feature classifier based on a decision tree, which returns a number between 0 and 1 to indicate the likelihood that two noun phrases corefer. Their training data came from and was applied to the MUC corpora. Positive examples were generated from each anaphoric NP_j and its immediately adjacent antecedent NP_i. Negative examples were generated by taking all noun phrases between each antecedent-anaphor pair, NP_{i+1}, NP_{i+2} … NP_{j-1}, and pairing them with the anaphor, NP_j. They found that the *alias*, *appositive*, and *string match* features contributed the most to performance. Ng and Cardie [2002b] extended this approach with three extra-linguistic changes: the clustering approach, the creation of training instances, and the definition of string match features. They also made use of additional features. Their system achieved good results on the MUC-6 and MUC-7 data sets, with F-Measure scores of 70.4 and 63.4, respectively. Ultimately, however, binary classification is flawed in that, at any given time, it takes into account only the relationships between two NPs rather than a longer chain. For example, given three NPs: NP_a, NP_b, and NP_c, it is possible that the model might think that NP_a and NP_b are coreferential, and also that NP_b and NP_c are coreferential, yet at the same time think that NP_a and NP_c are not. This creates a problem when the system tries to create coreference chains where all of the phrases in the chain refer to the same entity. Second, a phrase by itself usually lacks sufficient descriptive information to allow a completely confident decision to be made. Where the reference is to a human, it can be quite difficult to decide if two pronouns are anaphor-antecedent pairs simply by looking at the pronoun alone.

Several approaches have been proposed to compensate for these failings of the NP-NP approach. Yang *et al*. [2004b] adopted an NP-cluster framework, which considers the relationships between phrases and coreferential clusters. To describe the cluster properties, they introduced six additional features: cluster gender, number, semantic agreement, cluster length, cluster string similarity, and longest phrase similarity. Experiments have shown that this approach outperforms the NP-NP based approach. McCallum and Wellner [2004] introduced three conditional undirected graphical models of identity uncertainty based on conditional random fields. Their model avoids the problem of pair-wise coreference decisions being made independently of the relationships of each element of a pair. Rather than making a decision based on a single measurement to one other node, measurements are made to all nodes. This method improves upon the NP-NP based algorithm, but its supervised approach requires access to a large amount of data in order for meaningful statistics to be gathered.

2.2 Unsupervised Machine Learning Approaches

Supervised methods to coreference resolution have been successful at achieving good performance; however, they require annotated corpora as training data. This is not a problem with well-studied languages such as English, where language resources such as corpora and linguistics tools are plentiful, but it does create problems for other languages or even for less well-studied genres and domains.

Cardie and Wagstaff [1999] proposed an unsupervised approach that casts the problem of coreference resolution as a clustering task that applies a set of incompatibility functions and weights in the distance metric. Their algorithm starts by forming each entity into a singleton cluster, and then iteratively compares pairs of clusters. If the distance between two phrases in two clusters that are being compared is less than some threshold, the clusters are merged, provided that all their phrases are compatible. This mechanism can easily incorporate new constraints and preferences, but the merging algorithm is greedy in that it will take the first match rather than the best match.

2.3 Knowledge-Based Approaches

In addition to machine learning, knowledge-based approaches have also been widely used to provide rules for filtering features for NP resolution. Zhou and Su [2004] presented a constraint-based multi-agent strategy. This strategy first uses general heuristics such as morphological and semantic consistency to filter out invalid antecedent candidates, and then an antecedent for the anaphor is chosen based on the principle of proximity. This strategy offers two different types of agents: a set for filtering out less informative antecedent candidates and another set for matching coreference types. This strategy has been shown to be efficient and accurate. In addition, Bean and Riloff [2004] pioneered an approach to identify

NP coreferences by using information extraction patterns to identify contextual role knowledge. This approach first identifies definite, non-anaphoric noun phrases, and then uses case resolution to identify the most easily resolved phrases. Remaining non-resolved phrases are then evaluated against eleven sources of knowledge that include four contextual caseframes, that is, normalized extraction patterns. The final resolution is made using a Dempster-Shafer probabilistic model [Bean and Riloff 2004].

Knowledge-based approaches have the advantage in that, usually, little or no annotated corpora are required. However, they do rely heavily on hand-crafted heuristics or rules, which also require large investments of time and effort to create.

2.4 Feature Selection

The most desirable features for use in coreference resolution are robust and inexpensive, perform well over various domains, and can be obtained automatically. Features may be lexical, grammatical, semantic, syntactic, contextual, or heuristic. Given the broad range of features that may be chosen, there is currently no definitive classification of their relative merits or effects on system performance.

2.5 Coreference in Chinese Texts

To our knowledge, the previous work that has been conducted on the subject describes only two approaches to Chinese noun phrase coreference resolution, with both of them being supervised methods. Florian *et al.* [2004] used a language-independent framework to process Chinese data on the Entity Detection and Tracking (EDT) task, which is very similar to coreference resolution. EDT contains two subtasks, detection and tracking. The Entity Detection subtask finds all possibly coreferring phrases. The Entity Tracking subtask combines the detected phrases into groups referring to the same object. The authors formulate the detection subtask as a classification problem using a Robust Risk Minimization classifier combined with a Maximum Entropy classifier. Much like base noun phrase chunking, it labels each word token, indicating whether it starts a phrase, is inside a phrase, or is not within any phrase. They tackle the mention tracking subtask with a novel statistical approach that processes each phrase in turn, starting with the leftmost phrase in a document. For the current phrase, they make a decision to either link it with one of the existing clusters, or to make it start a new cluster. The authors reported achieving good results with English, Chinese, and Arabic. They obtained 58.8 on the ACE03 evaluation data on Chinese, but they noted that their algorithm was trained on only 90k characters for Chinese, in contrast to 340k words in English, which they believe to be insufficient for purposes of generalization.

Zhou *et al.* [2005] proposed a unified transformation-based learning (TBL) framework and tested it on Chinese EDT. They considered five types of entities: person, geographic or political entity, organization, location, and facility. They use the MSRSeg word segmentation algorithm and integrate it with an adapter to chunk Chinese characters into words. The mention detection model then tags each segmented word with a semantic type. The TBL tracking model then looks at every pair of words and classifies them as being coreferent or not, based on the values of six features (string match, edit distance, token distance, mention type, entity type, and lexical string). They report a performance of 63.3 on the ACE03 data set.

One of the biggest obstacles in Chinese noun phrase coreference resolution is that the amount of available data and resources lags far behind what is available in English. As a comparison, the ACE03 training corpus for Chinese was 90k characters, compared with over 340k words for English. In addition, there are many gazetteers and lexicons available in English but not many for other languages. These factors combine to make it difficult to get good performance in supervised efforts at noun phrase coreference in languages other than English.

2.6 Evaluation Metrics

Noun phrase coreference resolution is unlike other NLP tasks in that it does not decompose readily into either a task of bracketing or classification. As a result, it is not easy to extend current evaluation metrics to noun phrase coreference resolution. In this section, we will look at two of the most common evaluation metrics and explain how they work.

Traditionally, performance of noun phrase coreference resolution has been measured using precision and recall, as measured by Vilain *et al.*’s scoring algorithm [Vilain *et al*. 1995]. The algorithm defines recall as follows:

$$R = \frac{\sum(|C_i| - |p(C_i)|)}{\sum(|C_i| - 1)} . \tag{1}$$

Each C_i is a gold standard entity (*i.e*., a set of mentions that we know refer to the same entity), and $p(C_i)$ is the partitioning of C_i by the automatically identified entities. For example, suppose that the gold standard annotation identifies two entities, C_1 and C_2, where C_1 contains the mentions {1,2,3,4,5} and C_2 contains the mentions {6,7,8,9,A,B,C}. Now, assume that the automatically identified entities are partitioned as {1,2,3,4,5} {6,7} {7,8,A,B,C}. $|C_1|$ would therefore be 5, and $p(C_1)$ would be 1. Likewise, $|C_2|$ would be 7 and $p(C_2)$ would be 2. The recall for this scenario would then be calculated to be 90%. For precision, the roles of the automatically identified and gold standard entities are reversed.

Vilain *et al*’s evaluation metric was used for the MUC program, but as Baldwin *et al.* [1998] pointed out, it does have the weakness of yielding unintuitive results for some scenarios. For example, the baseline method of assuming that all identified mentions refer to the same entity actually yields a fairly good result by Vilain’s metric. There are several reasons for this counterintuitive result: first, the metric does not distinguish between different kinds of errors; second, it inherently favors outputs with fewer entities; and third, it ignores single-mention entities.

The ACE program introduced a different evaluation metric, the ACE value [ACE 2005], which has often been referred to as a cost-based metric. The idea is to evaluate system output by application value. A system with a completely correct output would get an ACE value of 100%, while a system producing no output would get an ACE value of 0%. Negative ACE values can also be given to systems with outputs that are drastically incorrect. The overall value is calculated by looking at each of the system-generated entities and calculating its value based on a product of two factors:

$$Value_{sys\_entity} = Entity\_Value(sys\_entity) \cdot Mentions\_Value(\{sys\_mentions\})$$

Entity_Value is a function calculated over each gold standard entity. It takes into account how well the gold standard and system outputs match each other on the entity level (*e.g.* whether the mentions in the entity were detected and resolved correctly by the system). *Mentions_Value* is a function measuring how well the mentions detected by the system match those of the gold standard (*e.g.* they may match, the system may identify extra mentions, or may miss some altogether). Errors that are penalized are misses (mentions that are in the gold standard but not in the system output), false alarms (mentions that appear in the system output but not in the gold standard), and mistakes (inexact overlaps between system output and gold standard). The heaviest penalties come from misses and false alarms, with misses penalized at a heavier rate than false alarms.

Even though the ACE value was developed partly to correct some of the drawbacks of the MUC metric, it does have a number of problems of its own. One of the biggest complaints is that ACE values are difficult to interpret. For example, if a system achieves an ACE score of 90%, this does not mean that the system correctly identified 90% of the entities and mentions in the corpus, but rather, that the cost of the system is 10% of one that does not give any output [Luo 2005]. Other criticisms are that it tends to be inconsistent in how it penalizes the systems for various mistakes [Zelenko 2005].

Despite all of the problems associated with its use, the ACE score remains the most widely used and accepted metric for evaluating noun phrase coreference system performance. Therefore, we will use this metric for our own evaluations.

3. Our Algorithm

Coreference resolution, although often referred to as a single task, can actually be divided into two subtasks. The first is entity or mention detection, which identifies anaphors and antecedents in a document, followed by noun phrase coreference resolution, or mention tracking, whereupon we decide upon the entities referred to by the identified phrases. Since trying to tackle both subtasks at once would necessitate the drawing up of an extremely complex model, almost all approaches in previous work have handled the two phases separately. Our algorithm will follow its predecessors and do the same.

3.1 Mention Detection

To start off the mention detection phase, we had our corpus parsed by a probabilistic Chinese parser [Zhang *et al*. 2003], which was trained on the Chinese Penn Treebank. As a precursor to doing a full parsing, the parser also performs word segmentation and POS tagging. The parser generates a full parse tree as its output. Since mentions usually correspond to noun phrases, we could simply have extracted all noun phrase chunks identified by the parser; however the boundaries of the parsed noun phrases do not usually correspond exactly with mention boundaries. In addition, since we followed the ACE conventions of only considering mentions that correspond to certain semantic types [ACE 2005], it is not too likely that all of the noun phrases are going to correspond to useful mentions. For example, the word 世界 *(world)*, although a noun phrase, is not tagged as a mention when it is not being used in the sense of a geographical location. We, therefore, used a filtering approach to identify and remove these spurious noun phrases. Filtering approaches have been successfully used by Bean and Riloff [1999], who used an unsupervised filter to construct a list of non-anaphoric phrases and NP patterns from an unannotated training corpus to identify mentions in definite noun phases. For their part, Ng and Cardie [2002a] employed a decision tree to filter out non-anaphoric phrases. Their approach achieved a large improvement in precision, but at a significant cost to recall.

The objective of filtering identified noun phrases is to identify only the noun phrases that are likely to correspond to mentions, while discarding the rest. Since the following phase, mention resolution, will work on top of these identified mentions, it is reasonable to aim for as accurate a performance on this phase as possible. The problem, however, is that precision and recall are usually inversely proportional to each other: having good precision usually means bad recall and vice-versa, and a balanced precision/recall performance usually means mediocre figures for both.

Our principle was this: the mention resolution phase will not identify additional mentions, and the ACE metric penalizes misses more heavily than false alarms. Therefore, we would go for high recall during the detection phase to minimize misses in the system output. To achieve this, we used a few simple heuristics to filter out noun phrases that are extremely unlikely to correspond to mentions. These heuristics are mostly based on the POS tags of the words, were previously developed for unrelated work in English named-entity resolution, and were not written with foreknowledge of the gold standard entities. A list of the heuristics can be found in Appendix 1.

In addition, in order to filter out spurious phrases, a stoplist was used to discard frequently occurring noun phrases such as 前提 *(the aforementioned)*, 什麼 *(what)*, 特色 *(feature)*, and 同時 *(at the same time)*. In addition, we also used a large gazetteer compiled from web sources to correct segmentation errors in proper names: *e.g*. to correct nr(埃斯特) v(拉) v(達) to (埃斯特拉達, *T. Estrada, former Cuban president*).

3.2 Mention Resolution

Once mention detection has been completed, the next step in the pipeline is that of mention tracking or resolution. In this step, the task of the system is to determine which noun phrases refer to the same entity, or are coreferent.

As defined by Trouilleux *et al*. [2000], "referential chains" are sets of expressions, or mentions, that denote the same referent. That is, given a text T, for each referential chain RC there exists a unique discourse referent DR, such that:

$$RC = \{x \mid x \text{ is an expression denoting } DR \text{ in } T\}. \qquad (2)$$

While most referential chains contain multiple elements, a referential chain may also consist of a single expression. For example, in the sentence "彼得愛加菲貓" (*Peter likes Garfield*), the set {彼得 (*Peter*)} is a referential chain. The task of coreference resolution consists of identifying these sets, which are also called "coreference chains."

Our algorithm relies on an unsupervised clustering approach for this task, which is a natural choice as it partitions the data into groups. For mention tracking, we expect the clustering algorithm to gather coreferent phrases into the same cluster, where each cluster will hopefully correspond to one coreference chain.

3.3 Modified K-Means Clustering

Most of the previous work in clustering-based noun phrase coreference resolution has centered around the use of bottom-up clustering methods [Cardie and Wagstaff 1999; Angheluta *et al*. 2004], where each noun phrase is initially assigned to a singleton cluster by itself, and clusters that are "close enough" to each other are merged.

In our system, we use a method called modified k-means clustering [Wilpon and Rabiner 1985], which takes the opposite approach and uses a top-down approach to split clusters, interleaved with a k-means iterative phase. Modified k-means clustering has been successfully applied to speech recognition. Compared with k-means clustering, modified k-means has the advantages of neither requiring a pre-set number of clusters nor being dependent upon an arbitrary starting state [Fung *et al*. 2003].

Modified k-means starts off with all of the instances in one big cluster. The system then iteratively performs the following steps:

1. For each cluster, find its centroid, defined as the instance that is the closest to all other instances in the same cluster.
2. For each instance:
   a. Calculate its distance to all of the centroids.
   b. Find the centroid with the minimum distance, and join its cluster.
3. Iterate 1-2 until instances stop moving between clusters.
4. Find the cluster with the largest intra-cluster distance, defined as the mean of the distances of all the instances in the cluster to the centroid instance. (Let this cluster be called $Cluster_{max}$ and its centroid, $Centroid_{max}$.)
   a. If the intra-cluster distance of $Cluster_{max}$ is smaller than some pre-set threshold r, stop.
5. Calculate the distances between all pairs of instances inside $Cluster_{max}$ and find the pair of instances that are the furthest apart.
   a. Add the pair of instances to the list of centroids and remove $Centroid_{max}$ from the list.
6. Repeat from Step 2.

The algorithm thus alternates traditional k-means clustering with a step that adds new clusters to the pool of existing ones. Used for coreference resolution, it splits up the instances into clusters in which the instances are more similar to each other than to instances in other clusters.

The next step is to determine a suitable threshold and a distance function with suitable parameters. As functions that check for compatibility return negative values while positive distances indicate incompatibility, a threshold of 0 would separate compatible and incompatible elements. However, since the feature extraction will not be totally accurate, we chose to be more lenient with deciding whether two phrases should be clustered together (*i.e.*, to go for recall over precision) and used a threshold of $r = 1$ to allow for possible errors.

3.4 Feature Selection

One of the advantages of using a clustering algorithm is that most clustering methods can easily incorporate both context-dependent and independent constraints into their features. This is attractive for us since we use a variety of features, which are designed both to capture the content of the phrase and its role within the sentence and document.

Most of our features give us information on a single phrase:

- **String Content** – The string of words in the phrase.
- **Head Noun** – The head noun in a phrase is the noun that is not a modifier for another noun.
- **Sentence Position** – The position of the sentence that contains the phrase, relative to the document. The first sentence is in position 1, the second in position 2, and so on.
- **Gender** – For each phrase, we use a gazetteer to assign it a gender. The possible values are male (*e.g.*, 先生, *mister*), female (*e.g.*, 小姐, *miss*), either (*e.g.*, 團長, *leader*), and neither (*e.g.*, 工廠, *factory*).
- **Number** – A phrase can be either singular (*e.g.*, 一隻貓, *one cat*), plural (*e.g.*, 兩隻狗, *two dogs*), either (*e.g.*, 產品, *product*) or neither (*e.g.*, 安全, *safety*).
- **Semantic Class** – To give the system more information on each phrase, we compiled our own gazetteer from web sources. Our gazetteer consists of 12,000 entries, each of which is labeled with the following semantic classes: person, organization, location, facility, GPE, date, money, vehicle, and weapon. Phrases in the corpus that are found in the gazetteer are given the same semantic class label; phrases not in the gazetteer are marked as *unknown*.
- **HowNet Definition** – The semantic class gazetteer covers about 80% of the phrases that are extracted. To increase the coverage of the phrases, we turned to HowNet [Dong and Dong 2000], an ontological knowledge base that encodes inter-conceptual relations and inter-attribute relations for the Chinese language. HowNet contains 120,496 entries for about 65,000 Chinese words defined with a set of 1503 sememes, which are considered atomic semantic units that cannot be reduced further. Examples of such sememes are "human," or "aValue" (attribute-value). Higher-level concepts, or definitions, are composed of subsets of these sememes, sometimes with pointers that denote certain kinds of relationships, such as "agent" or "target." For example, the word "疤" is associated with the definition "trace|疤, #disease|疾病, #wounded|受傷." As an additional feature, we labeled phrases that appeared as HowNet concepts with their sememe definitions. Phrases that do not exist in HowNet are marked as *unknown*. Overall, we found that about 66% of the extracted mentions in our corpus were covered under HowNet.
- **Proper Noun** – The part-of-speech tags "nr" (person name), "ns" (country name), "nt" (organization name), "nz" (other proper name), and a list of common proper names compiled from the Internet were used to label each noun phrase, indicating whether or not it is a proper noun.
- **Pronoun** – The part-of-speech tag "r" (pronoun) is used to determine whether the phrase is indeed a pronoun.

- **Demonstrative Noun Phrase** – A demonstrative noun phrase is a phrase that consists of a noun phrases preceded by one of the characters [此這那該] (*this/that/some*).

The following features give us information on how two phrases relate to each other:

- **Appositive** – Two noun phrases are in apposition when the first phrase is headed by a common noun, while the second one is a proper name and there no space or punctuation between the two phrases; *e.g.*, [美國總統][克林頓]上星期到朝鮮訪問 (*[US president] [Clinton] visited Pyongyang last week*). This differs from English, where two nouns are considered to be in apposition when one of them is an anaphor and separated by a comma from the other phrase, which is the most immediate proper name; *e.g.*, “Bill Gates, the chairman of Microsoft Corp”.
- **Abbreviative** – A noun phrase is an abbreviation when it is formed using part of another noun phrase; *e.g.*, 朝鮮中央通訊社 (*Pyongyang Central Communications Office*) is commonly abbreviated as 朝中社. Since name abbreviations in Chinese are often given in an ad-hoc manner, it would be infeasible to generate a list of names and abbreviations in advance. We, therefore, use the following heuristic: given two phrases, we test if one is an abbreviation of another by extracting each successive character from the shorter phrase and testing to see if it is included in the corresponding word from the longer phrase. Intuitively, we know that this is a common way of abbreviating terms; empirically, we found it to be a highly precise test: a positive result was very rarely wrong.
- **Edit Distance** – Abbreviations and nicknames are very commonly used in Chinese and even though the previous feature will work on most of them, there are some common exceptions. For example, some name-abbreviation pairs that would not get picked up are 北大西洋公約組織 (*North Atlantic Treaty Organization*) and 北約, or 奧運會 (*Olympics*) and 奧運. To make sure that those are caught as well, we introduced a Chinese-specific feature as a further test. Since abbreviations and nicknames are not usually substrings of the original strings but will still share some common characters, we measure the Levenshtein distance, defined as the number of character insertions, deletions, and substitutions, between every potential antecedent-anaphor pair.

To calculate the distance between two noun phrases, a set of functions is defined over the features. For features that give information on a single mention, functions compare the value of the same feature over a pair of phrases. For features defined relative to two mentions such as *edit distance* and *appositive*, the function simply returns the value of the feature itself.

The idea behind the functions is this: some features are indicators of whether two phrases are compatible with each other, with respect to coreferentiality. These features are *string* *content*, *head noun*, *demonstrative*, *appositive*, *abbreviation*, and *edit distance*. If two phrases match on this particular feature (for example, if the *head noun* feature for NP_i and NP_j are identical), then this is a strong indicator that these two phrases are coreferential. However, if they do not match, this does not necessarily mean that the two phrases are non-coreferential. Hence, these functions return negative values (decreasing the distance) when the two phrases match, but 0 (neutral) when they do not.

Table 1. Features and Functions Used for Clustering.

Feature *f*	**Function (*$Incompatibility_f(NP_i, NP_j)$*)**
String Match	-1 if the string of NP_i matches the string of NP_j; else 0
Head Noun Match	-1 if the head noun of NP_i matches the head noun of NP_j; else 0
Sentence Distance	0 if NP_i and NP_j are in the same sentence; For non-pronouns: 1/10 if they are one sentence apart; and so on with a maximum value of 1; For pronouns: if more than two sentences apart, then 1
Gender Agreement	1 if they do not match in gender; else 0
Number Agreement	1 if they do not match in number; else 0
Semantic Agreement	1 if they do not match in semantic class; else 0
HowNet Definition	1 if neither phrase is labeled as *unknown* and all of the sememes do not match, else 0.
Proper Name Agreement	1 if both are proper names, but mismatch on every word; else 0
Pronoun Agreement	1 if either NP_i or NP_j is a pronoun and the two mismatch in gender or number; else (e.g. if either one is unknown, or either one is not a pronoun), 0
Demonstrative Noun Phrase	-1 if NP_i is demonstrative and NP_i contains NP_j; else 0
Appositive	-1 if NP_i and NP_j are in an appositive relationship; else 0
Abbreviation	-1 if NP_i and NP_j are in an abbreviative relationship; else 0
Edit Distance	-1 if NP_i and NP_j are the same, -1/(length of longer string) if one edit is needed to transform one to another, and so on.

On the other hand, there are some features where a mismatch would strongly indicate that the two NPs are non-compatible and are not likely to refer to the same entity. The *gender*, *number*, *semantic*, *HowNet*, *proper name*, *pronoun*, and *sentence distance* features are all indicators of non-compatibility; hence, their associated functions return positive values, increasing the distance and making it less likely that the two phrases will be grouped into the same cluster.

Table 1 presents details of the features and the corresponding functions that were used in our system. Combining the values of all these functions gives us the distance between two phrases, with greater distances indicating greater incompatibility. For our system, we borrowed a simple distance metric from Cardie and Wagstaff [1999] that sums up the results of a series of functions over the two phrases:

$$dist(NP_i, NP_j) = \sum_{f \in F} w_f * incompatibility_f(NP_i, NP_j) \qquad (3)$$

where w_f is the weight of that particular feature (all features carry equal weight for us), and $incompatibility_f(NP_i, NP_j)$ is the result of the function corresponding to that feature when those two noun phrases are considered.

To summarize our efforts thus far, we have proposed an approach that adapts an unsupervised machine learning method for Chinese coreference resolution under limited resources. We have proposed a new methodology for mention detection and designed new features for mention resolution that are specifically geared towards our task.

4. Experimental Setup

To validate our algorithm, two data sets are used for evaluation. The first data set is an annotated version of TDT3 Chinese corpus, which was created by selecting 30 documents from the TDT3 corpus and then having it annotated by a native Chinese speaker following the MUC-7 [Hirschman and Chinchor 1997] and ACE Chinese entity guidelines [NIST 2005a]. We annotated proper nouns, nominal nouns, and pronouns, and according to MUC-7 guidelines, each phrase participates in exactly one entity, and all phrases in the same entity are coreferent. Using the MUC and ACE guidelines, we annotated noun phrases of the following nine types of entities, which are a combined set of those used in MUC and ACE:

- **Person** – Humans.
- **Organization** – Corporations and groups of people defined by an organizational structure.
- **Location** – Geographical areas, landmasses, and bodies of water.
- **Geopolitical entity (GPE)** – Comprised of a population, a government, a physical location, and a notion.
- **Facility** – Buildings and man-made structures.
- **Vehicle** – Physical devices designed to move an object from one location to another.
- **Weapon** – Physical devices used as instruments for physically harming or destroying.

- **Date** – Numbered days with a combination of the name of the day, the month, and the year.
- **Money** – Amounts of cash or currency.

The second corpus comes from the Chinese data in the ACE05 Entity Detection and Recognition evaluation task, which is similar to coreference resolution. This task requires that seven types of entities that are mentioned in the source data be detected and that the selected noun phrase about these entities be organized into a unified representation. The seven types of entities are *facility*, *GPE*, *location*, *organization*, *person*, *vehicle*, and *weapon*. The source data consists of three domains: newswire, broadcast news, and weblogs. For our experiments, we used the newswire and broadcast news domains. Table 2 shows some statistics from the corpora.

Table 2. Corpus Statistics

	Annotated TDT3	**ACE05 nwire**	**ACE05 bnews**
Documents	30	69	73
Character	23k	36k	32k
Entity	592	2044	1632
Mention	2997	4347	3678
Semantic Classes	32.7% person, 33.9% GPE, 13.5% organization, 7.7% facility, 3.9% location, 2.7% vehicle, 3.9% weapon, 1.1% date, 0.5% money	40.8% person, 30.7% GPE, 17.2% organization, 3.6% facility, 5.7% location, 1.7% vehicle, 0.3% weapon	44.5% person, 24.3% GPE, 17.9% organization, 7.7% facility, 4.6% location, 2.0% vehicle, 0.9% weapon

5. Evaluation

Since our algorithm breaks down the coreference resolution task into two subtasks, we will evaluate them separately and also investigate how or whether mistakes made in one subtask affect performance in the other.

5.1 Mention Detection

The subtask of mention detection is similar to that of noun phrase chunking, and we will evaluate it in the same fashion. We compare the output of the algorithm with the gold standard mentions, and count the number of mentions that are correctly identified. As an evaluation measure, we use the usual precision, recall, and f-measure metrics:

$$F = \frac{2PR}{P + R}. \quad (4)$$

Table 3 shows the results of the mention detection subtask achieved by our system on the TDT and ACE corpora, respectively.

Table 3. Mention Detection Results

	Recall	**Precision**	**F-Measure**
Annotated TDT3	88.5	48.4	62.6
ACE05 nwire	77.5	65.5	70.8
ACE05 bnews	73.8	64.0	68.5

5.2 Mention Resolution

As described in the section on Previous Work, both of the most commonly used noun phrase coreference resolution metrics have their detractors. In our work, we chose to use the ACE metric, which is currently the most widely accepted metric for this task.

Table 4 presents the performance of the second phase of our algorithm – the mention detection subtask – as measured by the official ACE05 scoring program. The entry “Our Algorithm” corresponds to the performance of our algorithm for each of the separate corpora. To get a sense of the difficulty of the task, we present a baseline system that simply assumes that mentions are coreferent if the “String Match” function (the most indicative feature) tests true. From the results, it can be seen that our system achieves a performance gain of over 20% on both the TDT3 and ACE05 newswire corpora, and over 10% on the ACE05 broadcast news corpora.

Table 4. Coreference Resolution Performance

Corpus	**Experiment**	**ACE value**
TDT3	Our Algorithm Baseline (string match only) Gold Standard Entities (upper bound)	52.5 43.7 77.0
ACE05 nwire	Our Algorithm Baseline (string match only) Gold Standard Entities (upper bound)	55.3 46.3 75.6
ACE05 bnews	Our Algorithm Baseline (string match only) Gold Standard Entities (upper bound)	55.1 49.0 77.2

Another point of comparison can be made when we compare the results obtained by our entire algorithm against the performance obtained *if we had performed the mention detection on gold standard entities*. The performance for this experiment is illustrated in the “Gold Standard Mentions” entry, and it gives us an idea of the upper bound that we could potentially achieve if we got 100% accuracy on the mention detection subtask. From the figures, it can be seen that there is substantial degradation of the overall performance of the algorithm as a result of errors in the first subtask cascading down the second subtask. This propagation of errors in pipelined systems is well known and documented.

6. Analysis

One interesting question to ask about the results is the contribution of any given individual feature to the result of the overall system. We have already investigated the effect of mention detection on the overall performance, and in this section we take a look at the features for the clustering algorithm used in the mention tracking subtask.

Table 5. Analysis: Contribution of each feature

Feature Removed	**ACE score (TDT3)**	**Change**	**ACE score (ACE05bn)**	**Change**
String Match	**71.9**	**-5.1**	**68.2**	**-9.0**
Head Noun Match	74.8	-2.2	75.5	-1.7
Sentence Distance	74.0	-3.0	75.1	-2.1
Gender Agreement	75.9	-1.1	74.1	-3.1
Number Agreement	75.5	-1.5	76.8	-0.4
Semantic Agreement	**71.1**	**-5.9**	**69.4**	**-7.8**
Proper Name Agreement	*76.7*	*-0.3*	*76.9*	*-0.3*
Pronoun Agreement	*76.6*	*-0.4*	*77.0*	*-0.2*
Demonstrative Noun Phrase	*76.0*	*-1.0*	*76.7*	*-0.5*
Appositive	73.2	-3.8	73.9	-3.3
Abbreviation	75.1	-1.9	76.5	-0.7
Edit Distance	**72.7**	**-4.3**	**71.7**	**-5.5**
HowNet Class	73.5	-3.5	74.3	-2.9
None (All Features)	77.0	--	77.2	--

In order to get a result that reflects the contribution of each feature alone, and to ensure that any conclusions we draw are extendable to other corpora, we performed a series of experiments of the mention tracking subtask on the gold standard entities of the TDT3 and the broadcast news portion of the ACE05 corpora. The first experiment was performed using all the features that were available to us, and then, one at a time, features were removed from the clustering algorithm.

Table 5 presents the results of the experiments. The last entry in the table shows the results of the full system; the drop in performance when a feature is removed is indicative of its contribution.

Judging from the results, the three features that contribute the most to performance are the *string match*, *semantic agreement*, and *edit distance* features. Two out of the three, *string match* and *edit distance*, operate on lexical information. The importance of string matching to coreference resolution is consistent with the findings in previous studies [Yang *et al*. 2004a], which arrived at the same conclusion for English. Edit distance, which captures certain phenomena not covered by string match, has also been found to be effective by Strube *et al*. [2002] for English coreference resolution, though their results focus on words rather than characters. The fact that *string match* and *edit distance* represent some overlapping information is not a problem for the k-means clustering algorithm, as it does not assume independent features.

Of our features, those that contribute the least to the overall performance of the system are the *proper name agreement* and *pronoun agreement* features. The reason for this is that the information of these features is already covered by *string match* and *head noun match*; thus, there are not enough distinct examples for them to make any significant impact.

In addition to feature coverage, another factor in determining the performance of the system is accuracy – both in the mention detection subtask as well as in feature generation. We have already seen the drop in system performance as a result of incorrectly identified mentions. For feature generation, we know that some of our features are always going to be generated correctly (for example, *string match* or *edit distance*), while others, such as *number agreement*, are generated using heuristics or gazetteers; therefore, even the values of the features themselves will be prone to errors.

To get a better sense of the source of errors, we randomly selected two documents in our corpus for closer examination. This revealed to us the reason why the *gender*, *number*, and *semantic class* features were not as useful as we had first thought they would be. Table 6 shows some of the statistics from our examination. Over 80% of the identified mentions are tagged with the correct value for the aforementioned features, which is a positive sign. However, the ability of the features to determine whether two mentions refer to the same entity is decreased by the coarse resolution of the feature values: about 50% of the mentions are tagged as *neither* for *Gender*, over 60% are tagged as *singular* for *Number*, and almost 70% of the mentions are either tagged as *person* or *GPE*.

Table 6. Automatic Feature Generation Statistics from Sampled Documents

Feature	Remarks
Gender agreement	50.4% neither, 22.8% either, 22.2% male, 4.6% female
Number agreement	66.0% singular, 14.4% neither, 11.1% either, 8.5% plural
Semantic agreement	50.3% person, 17.6% GPE, 12.4% organization, 7.8% location, 4.5 % unknown, 7.4% others
HowNet	41.0% unknown
Abbreviation	75% correct
Appositive	84.4% correct

The same table also shows why the *HowNet* feature does not contribute much to the performance of the system: its coverage is very limited, as only about 41% of the mentions exist as concepts in HowNet and thus receive a feature value.

On the positive side, it is heartening to see that our heuristics for checking for abbreviations and apposition work well: *abbreviation* was correctly tagged 75% of the time, and *apposition* achieved an accuracy of almost 85%.

The investigation also revealed the extent of segmentation errors upon our system performance. Upon examination, it was found that 35.2% of the missing link errors and 24.0% of the spurious link errors had been caused by segmentation errors. This finding illustrates the importance of the preprocessing step, and it also demonstrates the difficulty involved with working with relatively resource-poor languages or genres.

The length of the mentions in the examined documents also provides us with clues as to where our system could be improved. Our system relies heavily on lexical features, which work best with long strings of many characters. However, the mentions in the documents average a little over two characters in length. The result is that the lexical features have limited usefulness, at least in our document.

Another apparent problem with our approach is that almost all our features are designed to describe intra-mention information. The problem with this approach is that determining coreference resolution uses quite a lot of contextual information. For example, one of the entities in our two randomly-sampled documents was the one referring to 陳水扁總統 (*Taiwanese president Chen Shui-bian*). The mentions referring to this entity include 總統 (*president*), 陳總統 (*president Chen*), 我 (*myself*), 陳 (*Chen*), 他 (*him*), 一個台南小孩 (*a child from Tainan*), 導游 (*tour guide*), as well as 陳水扁 (*Chen Shui-bian*). While intra-mention information can (and does) distinguish 總統 (*president*), 陳總統 (*president Chen*), 陳 (*Chen*), and 陳水扁 (*Chen Shui-bian*) as referring to the same entity, it is not possible to realize that the other mentions also refer to this entity without using contextual information. The result is that these other mentions end up being separated out into singleton entities – entities with just one mention in them. This is a direction that we are definitely planning to work on in the future.

Table 7. Comparison of our system with results reported in previous work.

	ACE05 nwire	**ACE05 bnews**	**ACE03**
Our hybrid approach	**55.3**	**55.1**	
Florian *et al*. [Florian *et al*. 2004]	--	--	58.8
Zhou *et al*. [Zhou *et al*. 2005]	--	--	63.3
IBM	70.5	69.6	--
BBN Technologies	67.9	70.1	--
New York University	64.3	69.9	--
University of Colorado	64.9	57.4	--
Hong Kong Polytechnic University	51.3	50.2	--
XIAMEN University	44.8	51.0	--
Harbin Institute of Technology	44.1	48.0	--
Basis Technology, Inc.	3.0	4.7	--

To our knowledge, this is the first published result on unsupervised Chinese coreference resolution. To get a general idea of the performances achieved by other systems, Table 7 shows the performance of our system together with other previously reported results, some of which are from published reports while others are from the official evaluation of Entity Detection and Recognition task on Chinese [NIST 2005b]. It shows that our system achieves numerical results comparable to those from previous systems.

7. Conclusions and Future Work

In this paper, we have presented an unsupervised approach to Chinese coreference resolution. Our approach performs resolution by clustering, with the advantage that no annotated training data is needed. We evaluated our approach using an annotated version of TDT3 corpus and the ACE05 Chinese data, and found that our system achieves results comparable to the official results of using an unsupervised approach. We also analyzed the performance of our system by investigating the contribution of individual features to our system. The analysis illustrates the contribution of the new language-specific features, and also demonstrates that a reasonable coreference resolution system can be implemented quickly and efficiently through the use of readily-available resources.

While the results produced by our system are impressive, it is noted that all of our features consider only intra-mention information, which our in-depth analysis shows to be inadequate for coreference resolution. In future work, we plan to investigate the use of more sophisticated features, including contextual clues, to improve the performance of our system and implement entity-based clustering.

Acknowledgements

The authors would like to thank the Hong Kong Research Grants Council for supporting this work through research grant PolyU5191/04E, and also to the three anonymous reviewers for their insightful comments and suggestions.

References

ACE, The ACE Evaluation Plan, http://www.nist.gov/speech/tests/ace/ace05/index.htm, 2005.

Angheluta, R., P. Jeuniaux, M. Rudradeb, and M.F. Moens, “Clustering Algorithms for Noun Phrase Coreference Resolution,” *Proceedings of the 7th International Conference on the Statistical Analysis of Textual Data*, 2004, Louvain La Neuve, Belgium, pp. 60-70.

Baldwin, B., T. Morton, A. Bagga, J. Baldridge, R. Chandraseker, A. Dimitriadis, K. Snyder, and M. Wolska, “Description of the UPENN CAMP System as Used for Coreference.” In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, 1998, Fairfax, Virginia.

Bean, D., and E. Riloff, “Corpus-Based Identification of Non-Anaphoric Noun Phrases,” In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL-99)*, 1999, Maryland, USA, pp. 373-380.

Bean, D. and E. Riloff, “Unsupervised Learning of Contextual Role Knowledge for Coreference Resolution,” In *Proceedings of the Human Language Technology Conference / North American Chapter of the Association for Computational Linguistics Annual Meeting (HLT/NAACL-04)*, 2004, Boston, Massachusetts, pp. 297-304.

Cardie, C. and K. Wagstaff, “Noun phrase coreference as clustering,” In *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999, Maryland, USA, pp. 82-89.

Dong, Z.D., and Q. Dong, HowNet. http://www.keenage.com, 2000.

Florian, R., H. Hassan, A. Ittycheriah, H. Jing, N. Kambhatla, X. Luo, N. Nicolov, and S. Roukos, “Statistical Model for Multilingual Entity Detection and Tracking,” In *Proceedings of the 2004 annual meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2004)*, 2004, Boston, Massachusetts, pp. 1-8.

Fung, P., G. Ngai, and C. Cheung, “Combining Optimal Clustering and Hidden Markov Models for Extractive Summarization,” *Workshop on Multilingual Summarization and Question Answering, ACL-2003 Workshop*, 2003, Sapporo, pp. 21-28.

Hirschman, L., and N. Chinchor, MUC7 Coreference Task Definition, http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/co_task.html, 1997.

Luo, X., “On coreference resolution performance metrics,”. In *Proc. of HLT/EMNLP*, 2005, Vancouver, Canada, pp. 25-32.

Mccallum, A., and B. Wellner, “Conditional models of identity uncertainty with application to noun coreference”. In *Proceedings of NIPS-17*, 2004, Vancouver, Canada.

Ng, V., and C. Cardie, “Identifying Anaphoric and Non-Anaphoric Noun Phrases to Improve Coreference Resolution,” *Proceedings of the 19th International Conference on Computational Linguistics (COLING-2002)*, 2002, Taipei, Taiwan.

Ng, V., and C. Cardie, “Improving machine learning approaches to coreference resolution,” In *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics*, 2002, Philadelphia, pp. 104-111.

NIST, ACE Chinese Annotation Guidelines for Entities, http://www.ldc.upenn.edu/Projects/ACE/, 2005.

NIST, NIST 2005 Automatic Content Extraction Evaluation Official Results, http://www.nist.gov/speech/tests/ace/ace05/doc/ace05eval_official_results_20060110.htm, 2005.

Soon, W., H. Ng, and D. Lim, “A machine learning approach to coreference resolution of noun phrases,” *Computational Linguistics*, 27(4), 2001, pp. 521-544.

Strube, M., S. Rapp, and C. Muller, “The influence of minimum edit distance on reference resolution,” In *Proc. of the 2002 Conference on Empirical Methods in Natural Language Processing*, 2002, Philadelphia, pp. 312-319.

Trouilleux, F., E. Gaussier, G. G. Bes, and A. Zaenen, “Coreference resolution evaluation based on descriptive specificity,” In *Proceedings of the LREC 2000 Workshop on Linguistic Coreference*, 2000, Athens.

Vilain, M., J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman, “A model-theoretic coreference scoring scheme,” In *Proc. of the Sixth Message Understanding Conference (MUC-6)*, 1995, San Francisco, CA. Morgan Kaufmann, pp. 45-52

Wilpon, J., and L. Rabiner, “A modified K-means clustering algorithm for use in isolated word recognition,” In *IEEE Transactions on Acoustics, Speech, Signal Processing*. ASSP-33(3), 1985, pp. 587-594.

Yang, X., G. Zhou, J. Su, and C. L. Tan, “Improving noun phrase coreference resolution by matching strings,” In *Proc. of the 1st Int’l Joint Conference on Natural Language Processing*, 2004, Hainan, China, pp. 22-31.

Yang, X., G. Zhou, J. Su, and C. L. Tan, “An NP-Cluster Based Approach to Coreference Resolution,” *Proceedings of the 20th International Conference on Computational Linguistics*, 2004, Geneva, Switzerland, pp. 226-232.

Zelenko, D., Scoring Problems, http://cio.nist.gov/esd/emaildir/lists/ace_list/msg00849.html, 2005.

Zhang, H., Q. Liu, K. Zhang, G. Zou, and S. Bai, “Statistical Chinese Parser ICTPROP”, Technical Report No. 2003.06, Institute of Computing Technology, Chinese Academy of Sciences, 2003.

Zhou, Y., C. Huang, J. Gao, and L. Wu, “Transformation Based Chinese Entity Detection and Tracking,” *Proceedings of the Second International Joint Conference on Natural Language Processing*, 2005, Korea, pp. 232-237.

Zhou, G., and J. Su, “A High-Performance Coreference Resolution System using a Constraint-based Multi-Agent Strategy,” *Proceedings of the 20th International Conference on Computational Linguistics (COLING-2004)*, 2004, Geneva, Switzerland, pp. 522-528.

Appendix

List of Heuristics used in Mention Detection

- Keep all non-recursive noun phrases
  e.g. 韓國外交部 (*Korean Foreign Service*), 官員 (*officials*) from NP (NP (ns(韓國 *Korea*) nt(外交部 *Foreign Service*)) NP (n(官員 *officials*))).
- Keep all quantifier phrases.
  e.g. 一名旅客 (*a certain traveler*) from NP (QP (m(一 *one*) CLP (q(名))) NP (n(旅客 *traveler*))).
- Keep all determiner phrases.
  e.g. 這次旅遊(*this tour*) from DP (r(這次 *this*)) NP (vn(旅遊 *tour*)))
- Keep all pronouns.
  e.g. r(他們 *they*), r(他 *he*), r(自己 *myself*), r(我們 *we*), r(您 *you*).
- Keep all proper noun sequences.
  e.g. 小淵惠三 *(Obuchi Keizo)* from NP (nr(小淵) nr(惠)) NP (nr(三))
- Keep all noun sequences.
  e.g. 核子設施 *(nuclear facilities)* from NP (n(核子 *nuclear*) n(設施 *facilities*)))
- Keep frequently appearing proper nouns from the gazetteer.
  e.g. 埃斯特拉達 *(T. Estrada, former Cuban president)* from nr(埃斯特) v(拉) v(達)
- Keep all sequences matching certain regular expression-like patterns.
  e.g. mq.*n: m(五 *five*) q(天 *day*) dec(的 *'s*) n(國事訪問 *official visit*); r.*n: r(其他 *other*) ns(中國 *Chinese*) n(官員 *officials*)
  (Notation: '*' is the Kleene star operator, '.' is a wildcard corresponding to a single POS tag, other characters correspond to POS tags.)
- Keep two noun phrases with POS tagging pattern *noun-propernoun-propernoun*.
  e.g. n(記者 *journalist*) and nr(陳占杰 *Chen Chanchieh*) from NP(n(記者 *journalist*) nr(陳 *Chen*) nr(占杰 *Chanchieh*))
- Keep two noun phrases with POS tagging pattern *noun-dec-noun*.
  e.g. ns(中國 *China*) and n(政策 *policy*) from NP (ns(中國 *China*) dec(的 *'s*) n(政策 *policy*))
- Keep two noun phrases with POS tagging pattern *noun-conjunctive-noun*.
  e.g. ns(中國 *China*) and ns(美國 *USA*) from NP (ns(中國 *China*) c(和 *and*) ns(美國 *USA*))
- Keep all proper nouns with POS tagging pattern *nr ns nt nz*.
  e.g. ns(新疆 *Xinjiang*) and nz(維吾爾 *Uygur*) from NP (ns(新疆 *Xinjiang*) nz(維吾

爾 *Uygur*) n(自治區 *Autonomous Region*) n(領導 *leader*)).

- Discard noun phrases with POS tag *t* inside.
  e.g. NP(t(兩日 *two days*) t(下午 *afternoon*)); NP (t(目前 *present*))
- Discard noun phrases with only quantifier characters
  e.g. NP(m(兩 *two*) q(名 *persons*)); NP (m(十 *ten*) q(年 *years*))
- Discard noun phrases starting with prepositions.
  e.g. NP (p(對 *to*) ns(中國 *China*) n(人民 *people*))
- Discard noun phrases that contain verbs or punctuations.
  e.g. NP (n(總統 *presidential*) vn(大選 *election*)); NP (ns(日本 *Japan*) w、(、) ns(香港 *Hong Kong*))
- Discard single character noun phrases excepting those that have been tagged as proper nouns or pronouns
  e.g. n(字 *character*); n(月 *moon*)
- Discard noun phrases that are found in the stoplist.
  e.g. 前提 *(the aforementioned),* 什麼 *(what),* 特色 *(feature),* 同時 *(at the same time).*
- Discard noun phrases with stopwords appearing inside them: i.e. those with 的 *(dec),* 說 *(say),* 經 *(after),* 爲 *(for).*
  e.g. NP (n(車廂 *compartment*)) f(內 *inside*)) dec(的)); NP (r(他 *he*) v(說 *says*)); NP (p(經 *after*) vn(大賽 *competition*) n(評委會 *committee*)); NP(vl(爲 *for*) n(我國 *our country*))