Improving Noun Phrase Coreference Resolution by Matching Strings

Xiaofeng Yang[1,2], Guodong Zhou[1], Jian Su[1], and Chew Lim Tan[2]

[1] Institute for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore, 119613
{xiaofengy,zhougd,sujian}@i2r.a-star.edu.sg
[2] Department of Computer Science, National University of Singapore, Singapore, 117543
{yangxiao,tancl}@comp.nus.edu.sg

Abstract. In this paper we present a noun phrase coreference resolution system which aims to enhance the identification of the coreference realized by string matching. For this purpose, we make two extensions to the standard learn-ing-based resolution framework. First, to improve the recall rate, we introduce an additional set of features to capture the different matching patterns between noun phrases. Second, to improve the precision, we modify the instance selection strategy to allow non-anaphors to be included during training instance generation. The evaluation done on MEDLINE data set shows that the combination of the two extensions provides significant gains in the F-measure.

1 Introduction

Noun phrase coreference resolution is the process of determining whether or not two noun phrases in a document refer to the same entity. In recent years, supervised machine learning approaches have been applied to this problem and achieved reasonable success [1–5].

The previous work has reported that three features contribute most to noun phrase coreference resolution, namely, string match, name alias and apposition. Among them, string match is of the most importance. In the system by Soon et al. [3], for example, simply using head-match feature can achieve a recall as high as 56.4% and 55.2% for MUC-6 [6] and MUC-7 [7] data set, respectively. Indeed, in most of genres, there are large numbers of cases when the coreference between noun phrases is realized by string matching. Therefore, we can expect a good overall performance if high accuracy of string matching can be obtained.

Unfortunately, in contrast to name alias and apposition which are comparatively easy for a shallow system to resolve, the cases of matching of strings are more complicated. The types of the noun phrase modifiers and their matching patterns have considerable influence on coreference determination. For example, two phrases containing different adjective modifiers, such as "the red apple", "the green apple", usually refer to different entities. Also, some special modifiers, such as the superlative adjective or relative clauses, indicate that a noun phrase is a discourse-new description [8] and do not refer to any preceding noun phrase even if they are full-string matched. Therefore, the simple head-string matching or full-string matching check is not sufficient for coreference resolution; the former will lead to a low precision, and the latter may guarantee

the precision, but nevertheless with significant loss in recall (over 10% as in Soon et al.'s system).

String matching tasks have been explored by a number of communities including statistics, database and artificial intelligence communities. Various string distance metrics have been proposed to measure the matching degree of noun phrases [9]. However, string matching in coreference resolution task is comparatively complicated in that many contextual factors have to be considered. So far, several researchers have dealt with string matching in coreference resolution by heuristic methods (e.g. [10]) or using similarity features such as Minimum Edit Distance [11] or LCS [12] (See "Related Work" for further discussion).

In this paper, we present a NP coreference resolution system which investigates the coreference realized by string matching. We make two extensions to the standard learning-based approach framework to improve the recall and the precision of the resolution. First, we incorporate a set of features that is supposed to capture the various matching patterns between noun phrases. In calculating the matching degree of strings, we explore several similarity metrics, together with two different weighting schemes. Second, we modify the training instance selection strategy. Traditionally, training instances are formed based on one anaphor and its possible antecedents. However, non-anaphors are also informative in that they can effectively help the anaphoricity determination. In our approach we make use of non-anaphor in generating the training instances, which provides us significant gains in coreference resolution precision. The experimental results show that combination of the above two modifications boost the performance in F-measure compared with the baseline system.

2   Data Corpus

Our coreference resolution system is a component of our information extraction system in biomedical domain. For this purpose, we have built an annotated coreference corpus which consists of 200 MEDLINE[1] documents from GENIA data set[2]. The documents are all from biomedical literature with an average length of 244 words. The distribution of different types of markables is summarized in Table 1.

Table 1. Distribution of different types of markables in the 200 MEDLINE data set.

	Total Number	Percentage
Anaphoric Markables		
Non-pron	3561	29.1%
Pron	131	1%
Non-Anaphoric Markable		
Non-pron	8272	67.6%
Pron	259	2.1%
Total	12223	100%

---

[1] http://www.medstract.org

[2] http://www-tsujii.is.s.u-tokyo.ac.jp/~genia/index.html

To determine the boundary of the noun phrases, and to provide the necessary information for subsequent processes, a pipeline of Nature Language Processing components is applied to an input raw text. Among them, named entity recognition, part-of-speech tagging and text chunking adopt the same Hidden Markov Model (HMM) based engine with error-driven learning capability [13, 14]. The named entity recognition component trained on GENIA corpus [15] can recognizes up to 23 common Biomedical entity types (i.e. Virus, Tissue, RNA, DNA, Protein, etc) with an overall performance of 66.1 F-measure (P=66.5% R=65.7%)

3   The Framework of the Baseline Coreference Resolution System

We built a baseline coreference resolution system which adopts the standard learning-based framework employed in the work by Soon et al. [3].

During training, for each anaphor NP_j in a given text, a positive instance is generated by pairing NP_j with its closest non-pronominal antecedent. A set of negative instances is also formed by NP_j and each of the non-pronominal markables occurring between NP_j and NP_i.

A training instance is associated with a feature vector which, as described in Table 2, consists of 8 features. Here two string match features are tried in the system exclusively, i.e., *FullStrMatch* and *HeadStrMatch*. They represent the tightest and the loosest matching criterion, respectively.

Table 2. Feature set for baseline coreference resolution system.

1.	ante_Type	the type of NP_i (definite np, indefinite np, pronoun, ProperNP...)
2.	ana_Type	the type of NP_j (definite np, indefinite np, pronoun, ProperNP...)
3.	Appositive	1 if NP_i and NP_j are in an appositive structure; else 0
4.	NameAlias	1 if NP_i and NP_j are in an alias of the other; else 0
5.	GenderAgree	1 if NP_i and NP_j agree in gender; else 0
6.	NumAgree	1 if NP_i and NP_j agree in number; else 0
7.	SemanticAgree	1 if NP_i and NP_j agree in semantic class; else 0
8.	HeadStrMatch	1 if NP_i and NP_j contain the same head string; else 0
8'.	FullStrMatch	1 if NP_i and NP_j contain the same string after discarding determiners; else 0

When the training instances are ready, a classifier is learned by C5.0 algorithm [16]. During resolution, each encountered noun phrase, NP_j, is paired in turn with each preceding noun phrase, NP_i, from right to left. Each pair is associated with a feature vector as during training, and then presented to the coreference classifier. The classifier returns a positive or negative result indicating whether or not NP_i is coreferential to NP_j. The process terminates once an antecedent is found for NP_j, or the beginning of the text is reached. In the former case, NP_j is to be linked into the coreferential chain where the antecedent occurs.

4   New String Matching Features

4.1   String Matching Factors

Noun phrases preceded or followed by modifiers are common in numbers of genres. Generally, the modifiers of a noun phrase carry important information for coreference resolution. Two noun phrases with the same head string may probably refer to distinct entities if their modifiers fail to be matched. For example: "activation of T lymphocytes" - "activation of the proenkephalin promoter", "the first candidate" - "the second candidate", "the B cells" - "the Hela cells", and so on.

Also, the presence of some special modifiers, such as superlative adjective or relative clause, indicates that the modified noun phrase is a discourse-new description[3] and do not refer to any previously mentioned entity. For example:

(e1) *She jumps at the slightest noise.*
(e2) *Great changes have taken place in the town where he lived.*

In addition to modifiers, the head of a noun phrase itself provides clues of coreference. Typically, a group of definite noun phrase such as "the morning" and "the fact", refers to time or a larger situation and may not be used as anaphors. In such cases, noun phrases can not be linked together even if their modifiers are all matched well.

In our system, the above factors that influence coreference determination are incorporated in terms of features. Specifically, given a noun phrase, we first extract the information of its head and modifiers. Then, we measure the matching degree of their modifiers and keep the results in the features. We will introduce the detailed processing in the following subsections.

4.2   Noun Phrase Processing

To facilitate matching, for each noun phrase, we keep the information of its head-string, full-string and modifiers into a case structure as shown in Table 3. The value of each attribute is a bagging of word tokens.

During matching modifiers of noun phrases, it is possible that one string is name alias to the other, or two words in the strings are morphological variants to each other. In these cases, even though the modifiers contain different tokens, they can still be well matched. Therefore, in order to improve the recall rate of the resolution, we apply the following three actions to the attribute values of a noun phrase.

1. Expand the attribute values. If an attribute contains an acronym, replace the acronym with its corresponding definition[4].
2. Remove from the attributes values those stop-words, that is, non-informative words, including the prepositions (e.g. "of", "to", "in", etc), the articles (e.g. "a", "an", "the"), and all kinds of punctuation marks (e.g. "[", "]", "-", etc.).

---

[3] Vieira and Poesio [10] gave a detail introduction to the discourse-new description.
[4] In our system we use a heuristic method to extract the acronym list from the documents in collection.

Table 3. Attribute List of a noun phrase, NP_i.

NP_i	
Head	The head of NP_i
EntireNP	The entire string of NP_i
NUM	The number modifier of NP_i
VERB	The nonfinite modifier (*verb+ed*, *verb+ing*) of NP_i
PrepObj	The object of the preposition
ADJ_J	The Adj (normal form) modifier
ADJ_R	The comparative Adj modifier
ADJ_S	The superlative Adj modifier
ProperNP	The proper noun modifier
OtherNP	The normal nominal modifier

3. Stem the remaining words in the attribute values. In most cases, morphological variants of words have similar semantic interpretations and can be considered as equivalent. Currently we just use simple rules to truncate words, e.g. "terminal" is stemmed as "termin".

As an example, suppose the definition of the acronym of "LTR" and "HIV-1" is "long terminal repeat" and "human immunodeficiency virus type 1", respectively. The noun phrase "LTR of HIV-1" will be converted into:

Table 4. An example: Structure of "LTR of HIV-1".

Input NP: LTR of HIV-1
NP.Head = { repeat }
NP.EntireNP = { long termin repeat human immunodeficien virus type 1 }
NP.NUM = { 1 }
NP.PrepOBj = { human immunodeficien virus type 1}
NP.ADJ_J = { long termin }
NP.ProperNP = { human immunodeficien virus type 1}

4.3   Feature Definition

In addition to the features used in the baseline system, we introduce a set of features which aim to capture the matching patterns of the modifiers between noun phrases. All features are listed in Table 5 together with their respective possible values.

Features 9 - 30 record the matching degree, which we will discuss in the next subsection, between the attribute values of two noun phrases. For example, the feature *ante_ana_Prep* is to keep the matching degree of NP_i.PrepObj and NP_j.PrepObj. Note that ante_ana_[attribute] is different from ana_ante_[attribute]; the former is the matching degree of the possible antecedent NP_i against the possible anaphor, NP_j, while the latter is that of NP_j against NP_i. The values may be not equal to each to under some degree metrics.

Table 5. New string matching features of our coreference resolution system.

9.	ante_Relative	1 if NP$_i$ is modified by a relative clause; else 0
10.	ante_specialNP	1 if NP$_i$ is a special definite np which acts as a non-anaphor; else 0
11.	ana_Relative	1 if NP$_j$ is modified by a relative clause; else 0
12.	ana_specialNP	1 if NP$_j$ is a special definite np which acts as a non-anaphor; else 0
13.	ante_ana_(EntireNP, Number,	Matching degree of
~	Verb, Prep, AdjJ, AdjR,	NPi.(EntireNP, . . . , CommonNP) against
21	AdjS, ProperNP, CommonNP)	NPj.(EntireNP, . . . , CommonNP)
22.	ana_ante_(EntireNP, Number,	Matching degree of
~	Verb, Prep, AdjJ, AdjR,	NPj.(EntireNP, . . . , CommonNP) against
30	AdjS, ProperNP, CommonNP)	NPi.(EntireNP, . . . , CommonNP)

4.4 String Similarity Metrics

The matching degree of the attributes is measured in terms of string similarity. Three similarity metrics have been explored in our system:

– **Contain**

$$Contain(S1, S2) = \begin{cases} 1 & : & \text{if S1 is contained in S2} \\ 0 & : & \text{otherwise} \end{cases} \qquad (1)$$

The *Contain* metric checks whether the tokens in S1 is completely contained in S2. The intuition behind it is that if a possible anaphor contains less information than a possible antecedent, they are probably coreferential to each other.

– **ContainRatio**

$$ContainRatio(S1, S2) = 100 \times \frac{\sum_{t \in S1 \cap S2} w1_t}{\sum_{t \in S1} w1_t} \qquad (2)$$

where $w1_t$ is the weight of token t in S1.
ContainRatio measures the ratio of the number of common tokens between S1 and S2. It provides a smooth variant of function *Contain* in evaluating the degree that one string is contained in the other.

– **COS-Similarity**

$$COS - Similarity(S1, S2) = 100 \times \frac{\sum_{t \in S1 \cap S2} w1_t \times w2_t}{\sqrt{\sum_{t \in S1} w1_t{}^2} \times \sqrt{\sum_{t \in S2} w2_t{}^2}} \qquad (3)$$

The *COS-similarity* metric is widely used in Information Retrieval systems to calculate the similarity of documents or sentences. Note that for this is a symmetric metric, that is, $COS - Similarity(S1, S2) == COS - Similarity(S2, S1)$. This however does not hold truth on the metrics *Contain* and *ContainRatio*.

4.5 Weighting Schemes

In the metrics *ContainRatio* and *Cos-Similarity*, we use weight to reflect the importance of a token. Two weighting schemes are explored in our study:

- ***Binary Scheme.*** This simplest scheme assigns weight 1 to a token if the token occurs in the current string, or 0 if otherwise.
- ***TFIDF Scheme.*** Well known in the information retrieval community, this scheme takes into account the frequency factor of a token throughout all documents. The weight of a token t in a document d can be defined as:

$$w_{dt} = f_{dt} \times \log \frac{N}{df_t} \qquad (4)$$

where f_{dt} is the frequency of token t in document d, while N is the number of documents in the date set (e.g., 200 in our system) and d_{ft} is the number of documents containing token t.

5   New Training Instance Selection Strategy

In the traditional learning-based coreference resolution system, the training instances are formed by an anaphor and its possible antecedent. However, non-anaphors are also informative in that they provide important information for anaphoricity determination. As in the example sentence (e1) and (e2) in section 4.1, indefinite noun phrases, or definite noun phrase modified by superlative adjective, give us clues that they are a discourse-new entity and do not refer to any preceding noun phrase, even they match in the full string. However, such information can not be obtained if non-anaphors are absent in the training instances. As a result, the generated classifier would probably fail in the cases of non-anaphors, and thus degrade the precision rate of the resolution.

To improve the ability of anaphoricity determination, in our system we make use of non-anaphors to generate training instances. Specifically, for each non-anaphor NP_j, we

- Search for the first noun phrase NP_i which contains the same head string as NP_j from backwards.
- If such NP_i exists, generate a training instance by pairing NP_i and NP_j. Naturally, the instance is labeled as negative.

6   Results and Discussions

Our approach was evaluated on the MEDLINE data set introduced in Section 2. Five-fold cross-evaluation was done with each bin containing 40 documents from the data set. The performance of different coreference resolution systems was evaluated according to the scoring scheme proposed by Vilain et al. [17].

The first two lines of Table 6 list the performance of the baseline systems described in section 3. Here *HeadStrMatch* is the system using feature 8, i.e. *HeadStrMatch*, while *FullStrMatch* is the system using feature 8', i.e. *FullStrMatch*. The two baselines achieve an F-measure of 60.9% and 58.4%, respectively. *HeadStrMatch*, which performs the loosest matching check, gets a high recall 71.4%, but comparatively low precision 53.1%. By contrast, *FullStrMatch*, which performs the tightest matching check, obtains a high precision (68.5%) at a price of significantly low recall (51.0%).

Table 6. Experimental results on the Medline data set using C5.0 (the *ed systems use *Contain-Ration* metric with Binary weighting scheme).

	Recall	Precision	F-measure
HeadStrMatch	**71.4**	53.1	60.9
FullStrMatch	51.0	68.5	58.4
NewFeature*	70.5	63.8	66.9
NonAnaphor+NewFeature*	68.1	**69.7**	**68.9**

The third line of the table summarizes the performance of the system *NewFeature*, which adopts our new string matching features as described in Section 4. Compared with *HeadStrMatch*, *NewFeature* achieves a significant increase of 10.7% in the precision rate with only a small loss (0.9%) in recall. On the other hand, compared with *FullStrMatch*, the recall rate of *NewFeature* improves significantly (about 20%), while the precisions drops only 4.7%. As a whole, our new features produce gains of about 6% and 8.5% in F-measure over *HeadStrMatch* and *FullStrMatch*, respectively.

Results on the modification to the training instance selection strategies are shown in the last line of Table 6. Compared with *NewFeature*, the inclusion of non-anaphors in the training instance generation gives an increase of about 6% in the precision. The precision is even higher than that of *FullStrMatch*. While the recall drops a little (2.4%), we see a further increase of 2% in F-measure. The drop in recall is reasonable since the learned classifiers become stricter in checking non-anaphoric markables. The degrade in recall was also reported by Ng and Cardie [18], where they use a separate anaphoricity determination module to improve the coreference resolution.

In our experiments, we also explore the influence of the three string similarity metrics (i.e., *Contain*, *ContainRatio*, *Cos-Similarity*) and the two weighting schemes (i.e., *Binary* and *TF.IDF*), on the performance of coreference resolution. The results are summarized in Table 7. From the comparisons shown in the table we can find the tradeoffs between the recall and precision when applying different similairty metrics. For example, in the system *NewFeature* (with *Binary* weight), the metric *Cos-Similarity* leads to the highest recall (72.8%), while *ContainRatio* produces the highest precision (63.8%). However, from the overall evaluation, the metric *ContainRatio* outperforms all the other two competitors in the F-measure.

Table 7. Influence of different string similarity metrics and token weighting schemes on the resolution.

Strategy	Similarity Metric	Binary Weight			TFIDF Weight		
		R	P	F	R	P	F
NewFeature	Contain	70.6	61.7	65.8	-	-	-
	ContainRatio	70.5	63.8	67.0	72.1	61.3	66.2
	Cos-Similairty	**72.8**	60.2	65.9	69.3	62.9	65.9
NonAnaphor+NewFeature	Contain	66.5	71.4	68.8	-	-	-
	ContainRatio	68.1	69.7	**68.9**	65.2	69.9	67.4
	Cos-Similairty	66.7	70.0	68.3	63.7	**72.5**	67.8

In comparing the two different weighting schemes, it is interesting to note that the systems using *TF.IDF* does not perform better than those using *Binary* in the F-measure. *TF.IDF* scheme may improve precision, especially for *Cos-Similarity* metric (2.7% higher). Nevertheless, the contribution of the frequency information to precision rate is not significant enough to compensate the loss in recall. We see that the recall drops (over 3.0% for *Cos-Similarity* metric) at the same time (the exception is *NewFeature+ContainRation*, where *TFIDF* gets a higher recall but lower precision than *Binary*). In fact, in determining the coreference relationship between two noun phrases, each token in the modifiers, no matter how many times it occurs throughout the current document and the entire data set, may likely provide an important clue. That is may be why *Binary* weighting scheme seems to be superior to *TF.IDF* scheme.

7   Related Work

Several work has been done on the resolution of coreference realized by string matching. (e.g. [2, 10, 3, 4, 11, 12, 19]). Compared to existing approaches, our approach has the following advantages:

- Our approach can deal with all types of nouns. In contrast, the study by Vieira and Poesio[10] focuses only on definite noun phrases. Also, the conditional model by McCallum and Wellner [19] is mainly for Proper noun coreference resolution.
- Our feature set can capture various matching patterns between noun phrases. In contrast, for the feature MED used by Strube et al. [11] and LCS by Castano et al. [12], the matching is restricted only on the full strings of noun phrases.

Ng and Cardie [18] proposed an anaphoricity determination module to improve the coreference resolution. In their approach, a multiple anaphoricity classifier has to be trained and applied in the coreference resolution. In contrast, the anaphoricity determination function is integrated seamlessly in our coreference classification, attributed to our training instance selection strategy.

8   Conclusion

In the paper we presented a system which aims to address the coreference realized by string matching. We improve the performance of the baseline resolution system in two ways. First, we proposed an extensive feature set to capture the matching information between noun phrase modifiers. To improve the recall rate, techniques such as expansion, stemming, and stopping words removal were applied to the original strings. Different matching degree metrics and weighting schemes have been tried to obtain the feature values. Second, we modified the selection strategy for training instances. Non-anaphors now are also included in the training instance generation. This enhances the anaphoricity identification ability of the classifier, and thus improves the precision.

While the experimental results show that combination of the above two modifications boost the system performance, there is still room for improvement. For example, in calculating the matching degree of two strings, the semantic compatibility between words, e.g., hypernym or synonym, have influence on the string matching. We would like to take this factor into account for our future work.

References

1. Aone, C., Bennett, S.W.: Evaluating automated and manual acquistion of anaphora resolution strategies. In: Proceedings of the 33rd Annual Meeting of the Association for Compuational Linguistics. (1995) 122–129
2. McCarthy, J., Lehnert, Q.: Using decision trees for coreference resolution. In: Proceedings of the 14th International Conference on Artificial Intelligences. (1995) 1050–1055
3. Soon, W., Ng, H., Lim, D.: A machine learning approach to coreference resolution of noun phrases. Computational Linguistics **27** (2001) 521–544
4. Ng, V., Cardie, C.: Improving machine learning approaches to coreference resolution. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia (2002) 104–111
5. Yang, X., Zhou, G., Su, J., Tan, C.: Coreference resolution using competition learning approach. In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, Japan (2003)
6. MUC-6: Proceedings of the Sixth Message Understanding Conference. Morgan Kaufmann Publishers, San Francisco, CA (1995)
7. MUC-7: Proceedings of the Seventh Message Understanding Conference. Morgan Kaufmann Publishers, San Francisco, CA (1998)
8. Poesio, M., Vieira, R.: A corpus-based investigation of definite description use. Computational Linguistics **24** (1998) 183–261
9. Cohen, W., Ravikumar, P., Fienberg, S.: A comparison of string distance metrics for name-matching tasks. In: Procedings of IJCAI-03 Workshop on Information Integration on the Web. (2003)
10. Vieira, R., Poesio, M.: An empirically based system for processing definite descriptions. Computational Linguistics **27** (2001) 539–592
11. Strube, M., Rapp, S., Muller, C.: The influence of minimum edit distance on reference resolution. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Philadelphia (2002) 312–319
12. Castano, J., Zhang, J., Pustejovsky, J.: Anaphora resolution in biomedical literature. In: International Symposium on Reference Resolution, Alicante, Spain (2002)
13. Zhou, G., Su, J.: Error-driven HMM-based chunk tagger with context-dependent lexicon. In: Proceedings of the Joint Conference on Empirical Methods on Natural Language Processing and Very Large Corpus, Hong Kong (2000)
14. Zhou, G., Su, J.: Named Entity recognition using a HMM-based chunk tagger. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia (2002)
15. Shen, D., Zhang, J., Zhou, G., Su, J., Tan, C.: Effective adaptation of hidden markov model-based named-entity recognizer for biomedical domain. In: Proceedings of ACL03 Workshop on Natural Language Processing in Biomedicine, Japan (2003)
16. Quinlan, J.R.: C4.5: Programs for machine learning. Morgan Kaufmann Publishers, San Francisco, CA (1993)
17. Vilain, M., Burger, J., Aberdeen, J., Connolly, D., Hirschman, L.: A model-theoretic coreference scoring scheme. In: Proceedings of the Sixth Message understanding Conference (MUC-6), San Francisco, CA, Morgan Kaufmann Publishers (1995) 45–52
18. Ng, V., Cardie, C.: Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution. In: Proceedings of the 19th International Conference on Computational Linguistics (COLING02). (2002)
19. McCallum, A., Wellner, B.: Toward conditional models of identity uncertainty with application to proper noun coreference. In: Procedings of IJCAI-03 Workshop on Information Integration on the Web. (2003) 79–86