

Other-Anaphora Resolution in Biomedical Texts with Automatically Mined Patterns

Chen Bin[#], Yang Xiaofeng^{\$}, Su Jian[^] and Tan Chew Lim^{*}

^{#*}School of Computing, National University of Singapore

^{\$^}Institute for Infocomm Research, A-STAR, Singapore

{[#]chenbin, ^{*}tancl}@comp.nus.edu.sg

{^{\$}xiaofengy, [^]sujiang}@i2r.a-star.edu.sg

Abstract

This paper proposes an *other*-anaphora resolution approach in bio-medical texts. It utilizes automatically mined patterns to discover the semantic relation between an anaphor and a candidate antecedent. The knowledge from lexical patterns is incorporated in a machine learning framework to perform anaphora resolution. The experiments show that machine learning approach combined with the auto-mined knowledge is effective for *other*-anaphora resolution in the biomedical domain. Our system with auto-mined patterns gives an accuracy of 56.5%., yielding 16.2% improvement against the baseline system without pattern features, and 9% improvement against the system using manually designed patterns.

1 Introduction

The last decade has seen an explosive growth in the amount of textual information in biomedicine. There is a need for an effective and efficient text-mining system to gather and utilize the knowledge encoded in the biomedical literature.

For a correct discourse analysis, a text-mining system should have the capability of understanding the reference relations among different expressions in texts. Hence, anaphor resolution, the task of resolving a given text expression to its referred expression in prior texts, is important for an intelligent text processing system.

In linguistics, an expression that points back to a previously mentioned expression is called an anaphor, and the expression being referred to by the anaphor is called its antecedent. Most previous work on anaphora resolution aims at *identity*-anaphora in which both an anaphor and its antecedent are mentions of the same entity.

In this paper, we focus on a special type of anaphora resolution, namely, *other*-anaphora resolution, in which an anaphor to be resolved has a prefix modifier “other” or “another”. The antecedent of an *other*-anaphor is a complement expression to the anaphor in a super set. In other words, an *other*-anaphor is a set of elements excluding the element(s) specified by the antecedent. If the modifier “other” or “another” is removed, an anaphor becomes the super set including the antecedent. Thus, *other*-anaphora in fact represents a “part-whole” relation. Consider the following text

“*IL-10 inhibits nuclear stimulation of nuclear factor kappa B (NF kappa B).*”

Several other transcription factors including NF-IL-6, AP-1, AP-2, GR, CREB, Oct-1, and Sp-1 are not affected by IL-10.”

Here, the expression “*other transcription factors*” is an *other*-anaphor, while the “*NF kappa B*” is its antecedent. The anaphor refers to any transcription factors except the antecedent. By removing the lexical modifier “other”, we can get a super set “*transcription factors*” that includes the antecedent. The anaphor and antecedent thus have a “part-whole” relation¹.

Other-anaphora resolution is an important sub-task in information extraction for biomedical

domain. It also contributes to biomedical ontology building as it targeted at a “part-whole” relation which is in the same hierarchical orders as in ontology. Furthermore, *other*-anaphora resolution is a first-step exploration in the resolution of bridging anaphora. Furthermore, other-anaphora resolution is a first-step exploration in the resolution of bridging, a special anaphora phenomenon in which the semantic relation between an anaphor and its antecedent is more complex (e.g. part-whole) than co-reference.

Previous work on *other*-anaphora resolution relies on knowledge resources, for example, ontology like WordNet to determine the “part-whole” relation. However, in the biomedical domain, a document is full of technical terms which are usually missing in a general-purpose ontology. To deal with this problem, pattern-based approaches have been widely employed, in which a pattern that represents the “part-whole” relation is designed. Two expressions are connected with the specific pattern and form a query. The query is searched in a large corpus for the occurrence frequency which would indicate how likely the two given expressions have the part-whole relation. The solution can avoid the efforts of constructing the ontology knowledge for the “part-whole” relation. However, the pattern is designed in an ad-hoc method, usually from linguistic intuition and its effectiveness for *other*-anaphora resolution is not guaranteed.

In this paper, we propose a method to automatically mine effective patterns for *other*-anaphora resolution in biomedical texts. Our method runs on a small collection of seed word pairs. It searches a large corpus (e.g., PubMed abstracts as in our system) for the texts where the seed pairs co-occur, and collects the surrounding words as the surface patterns. The automatically found patterns will be used in a machine learning framework for *other*-anaphora resolution. To our knowledge, our work is the first effort of applying the pattern-base technique to *other*-anaphora resolution in biomedical texts.

The rest of this paper is organized as follows. Section 2 introduces previous related work. Section 3 describes the machine learning framework for *other*-anaphora resolution. Section 4 presents in detail our method for automatically pattern mining. Section 5 gives experiment results and has some discussions. Finally, Section 6 concludes the paper and shows some future work.

2 Related Work

Previous work on *other*-anaphora resolution commonly depends on human engineered knowledge and/or deep semantic knowledge for the “part-whole” relation, and mostly works only in the news domain.

Markert *et al.*, (2003) presented a pattern-based algorithm for *other*-anaphora resolution. They used a manually designed pattern “ANTECEDENT *and/or other* ANAPHOR”. Given two expressions to be resolved, a query is formed by instantiating the pattern with the two given expressions. The query is searched in the Web. The higher the hit number returned, the more likely that the anaphor and the antecedent candidate have the “part-whole” relation. The anaphor is resolved to the candidate with the highest hit number. Their work was tested on 120 *other*-anaphora cases extracted from Wall Street Journal. The final accuracy was 52.5%.

Modjeska *et al.*, (2003) also presented a similar pattern-based method for *other*-anaphora resolution, using the same pattern “ANTECEDENT *and/or other* ANAPHOR”. The hit number returned from the Web is used as a feature for a Naïve Bayesian Classifier to resolve *other*-anaphors. Other features include surface words, substring matching, distance, gender/number agreement, and semantic tag of the NP. They evaluated their method with 500 *other*-anaphora cases extracted from Wall Street Journal, and reported a result of 60.8% precision and 53.4% recall.

Markert and Nissim (2005) compared three systems for *other*-anaphora resolution, using the same data set as in (Modjeska *et al.*, 2003).

The first system consults WordNet for the part-whole relation. The WordNet provides information on meronym/holonym (part-of relation) and hypernym/ hyponym (type-of relation). Their system achieves a performance of 56.8% for precision and 37.0% for recall.

The second and third systems employ the pattern based approach, employing the same manual pattern “ANTECEDENT *and/or other* ANAPHOR”. The second system did search in British Nation Corpus, giving 62.6% precision and 26.2% recall. The third system did search in the Web as in (Markert *et al.*, 2003), giving 53.8% precision and 51.7% recall.

3 Anaphora Resolution System

3.1 Corpus

In our study, we used the GENIA corpus² for our *other*-anaphora resolution in biomedical texts. The corpus consists of 2000 MEDLINE abstracts (around 440,000 words). From the GENIA corpus, we extracted 598 *other*-anaphora cases. The 598 cases do not contain compound prepositions or idiomatic uses of “other”, like “on the other hand” and “other than”. And all these anaphors have their antecedents found in the current and previous two sentences of the *other*-anaphor. On average, there are 15.33 candidate antecedents for each anaphor to be resolved.

To conduct *other*-anaphora resolution, an input document is preprocessed through a pipeline of NLP components, including tokenization, sentence boundary detection, part-of-speech (POS) tagging, noun phrase (NP) chunking, and named-entity recognition (NER). These preprocessing modules are aimed to determine the boundaries of each NP in a text, and to provide necessary information of an NP for subsequent processing. In our system, we employed the tool-kits built by our group for these components. The POS tagger was trained and tested on the GENIA corpus (version 2.1) and achieved an accuracy of 97.4%. The NP-chunking module, evaluated on UPEN WSJ TreeBank, produced 94% F-measure. The NER module, trained on GENIA corpus (version 3.0), achieved 71.2% F-measure covering 22 entity types (e.g., Virus, Protein, Cell, DNA, etc).

3.2 Learning Framework

Our *other*-anaphora resolution system adopts the common learning-based model for *identity*-anaphora resolution, as employed by (Soon *et al.*, 2001) and (Ng and Cardie, 2002).

In the learning framework, a training or testing instance has the form of $fv(cand_i, ana)$ where $cand_i$ is the j^{th} candidates of the antecedent of anaphor ana . An instance is labelled as positive if $cand_i$ is the antecedent of ana , or negative if $cand_i$ is not the antecedent of ana . An instance is associated with a feature vector which records different properties and relations between ana and $cand_i$. The features used in our system will be discussed later in the paper.

During training, for each *other*-anaphor, we consider as the candidate antecedents the preceding NPs in its current and previous two sentences.

A positive instance is formed by pairing the anaphor and the correct antecedent. And a set of negative instances is formed by pairing the anaphor and each of the other candidates.

Based on these generated training instances, we can train a binary classifier using any discriminative learning algorithm. In our work, we employed support vector machine (SVM) due to its good performance in high dimensional feature vector spaces.

During the resolution process, for each *other*-anaphor encountered, all of the preceding NPs in a three-sentence window are considered. A test instance is created for each of the candidate antecedents. The feature vector is presented to the trained classifier to determine the *other*-anaphoric relation. The candidate with highest SVM outcome value is selected as the antecedent.

3.3 Baseline Features

Knowledge is usually represented as features for machine learning. In our system, we used the following groups of features for *other*-anaphora resolution

- **Word Distance Indicator**

This feature measures the word distance between an anaphor and a candidate antecedent, with the assumption that the candidate closer to the anaphor has a higher preference to be the antecedent.

- **Same Sentence Indicator**

This feature is either 0 or 1 indicating whether an anaphor and a candidate antecedent are in the same sentence. Here, the assumption is that the candidate in the same sentence as the anaphor is preferred for the antecedent.

- **Semantic Group Indicators**

A named-entity can be classified to a semantic category such as “DNA”, “RNA”, “Protein” and so on³. Thus we use a set of features to record the category pair of an anaphor and a candidate antecedent. For example, “DNA-DNA” is generated for the case when both anaphor and candidate are DNAs. And “DNA-Protein” is generated if an anaphor is a DNA and a candidate is a protein. These features indicate whether a semantic group can refer to another.

Note that an anaphor and its antecedent may possibly belong to different semantic categories. For example, in the GENIA corpus we found that

² <http://www-tsujii.is.s.u-tokyo.ac.jp/~genia/topics/Corpus/>

³ In our study, we followed the semantic categories defined in the annotation scheme of the GENIA corpus.

in some cases an expression of a protein name actually denotes the gene that encodes the protein. Thus for a given anaphor and a candidate under consideration, it is necessary to record the pair-wise semantic groups, instead of using a single feature indicating whether two expressions are of the same group.

The semantic group for a named entity is given by our preprocessing NER. For the common NPs produced from the NP chunker, we classify the semantic group by looking for the words inside NPs. For example, an NP ending with “cells” is classified to “Cell” group while an NP ending with “gene” or “allele” is classified to “DNA” group.

- **Lexical Pattern Indicators**

In some cases, the surrounding words of an anaphor and a candidate antecedent strongly indicate the “part-whole” relation. For example, in “...*asthma and other hypereosinophilic diseases*”, the reference between “*other hypereosinophilic diseases*” and “*asthma*” is clear if the in-between words “*and other*” are taken into consideration. Another example of such a hint pattern is “*one... the other ...*”. The feature is 1 if the specific patterns are present for the current anaphor and candidate pair. A candidate with such a feature is preferred to be the antecedent.

- **Hierarchical Name Indicator**

This feature indicates whether an antecedent candidate is a substring of an anaphor or vice versa. This feature is used to capture cases like “*Jun*” and “*JunB*” (“*Jun*” is a family of protein while “*JunB*” is a member of this family). In many cases, an expression that is a super set comes with certain postfix words, for example, “family members” in

“*Fludarabine caused a specific depletion of STAT1 protein (and mRNA) but not of other STAT family members.*”

This kind of phenomenon is more common in bio-medical texts than in news articles.

3.4 SVM Training and Classification

In our system, we utilized the open-source software SVM-Light⁴ for the classifier training and testing. SVM is a robust statistical model which has been applied to many NLP tasks. SVM tries to learn a separating line to separate the positive instances from negative instances. Kernel transformations are applied for non-linear separable

cases (Vapnik, 1995). In our study, we just used the default learning parameters provided by SVM-Light with the linear kernel. A more sophisticated kernel may further improve the performance.

4 Using Auto-mined Pattern Features

The baseline features listed in Section 3.3 only rely on shallow lexical, position and semantic information about an anaphor and a candidate antecedent. It could not, nevertheless, disclose the “part-whole” relation between two given expressions. In section 2, we have shown some existing pattern-based solutions that mine the “part-whole” relation in a large corpus with some patterns that can represent the relation. However, these manually designed patterns are usually selected by heuristics, which may not necessarily lead to a high coverage with a good accuracy in different domains. To overcome this shortcoming, we would like to use an automatic method to mine effective patterns from a large data set. First, we create a set of seed pairs of the “part-whole” relation. And then, we use the seed pairs to discover the patterns that encode the “part-whole” relation from a large data set (PubMed as in our system). Such a solution is supposed to improve the coverage of lexical patterns, while still retain the desired “part-whole” relation for *other-anaphora* resolution.

The overview of our system with the automatic mined patterns is illustrated in figure 1.

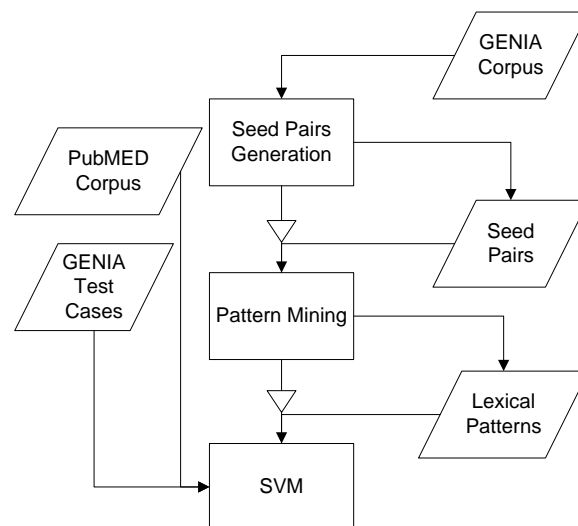


Figure 1: System Overview

There are three major parts in our system, namely, seed-pairs generation, pattern mining and SVM learning and classification. In the subsequent subsections, we will discuss each of the three parts in details.

⁴ <http://svmlight.joachims.org/>

4.1 Seed Pairs Preparation

A seed pair is a pair of phrases/words following “part-whole” order, for example,

“*integrin alpha*” - “*adhesion molecules*”

where “*integrin alpha*” is a kind of “*adhesion molecules*”.

We extracted the seed pairs automatically from the GENIA corpus. The auto-extracting procedure makes use of some lexical clues like “A, such as B, C and D”, “A (e.g. B and C)”, “A including B” and etc. The capital letter A, B, C and D refer to a noun phrase such as “*integrin alpha*” and “*adhesion molecules*”. For each occurrence of “A such as B, C and D”, the program will generate seed pairs “B-A”, “C-A” and “D-A”.

Consider the following example,

“*Mouse thymoma line EL-4 cells produce cytokines such as interleukin (IL) -2, IL-3, IL-4, IL-10, and granulocyte-macrophage colony-stimulating factor in response to phorbol 12-myristate 13-acetate (PMA).*”

We can extract the following seed pairs,

“*interleukin (IL) -2*” - “*cytokines*”

“*IL -3*” - “*cytokines*”

“*IL -4*” - “*cytokines*”

“*IL -10*” - “*cytokines*”

“*granulocyte-macrophage colony-stimulating factor*” - “*cytokines*”

A similar action is taken for other lexical clues. Totally, we got 909 distinct seed pairs extracted from the GENIA corpus.

After the seed pairs have been extracted, an automatic verification of the seed pairs is performed. The first purpose of the verification is to correct chunking errors. For example, “*HLA Class II Gene*” may likely be wrongly split into “*HLA Class*” and “*II Gene*”. This kind of errors is repaired by several simple syntactic rules. The second purpose of the verification is to remove the inappropriate seed pairs. In our system, we abandoned the seed pairs containing pronouns like “those”, “they”, or nouns like “element”, “member” and “agent”. Such seed pairs may either find no patterns, or lead to meaningless patterns because “those” or “elements” have no specific semantics and could refer to anything.

4.2 Pattern Mining

Having obtained the set of seed pairs, we will use them to mine patterns for the “part-whole” relation. For each seed pair “*antecedent - anaphor*” (*anaphor* represents the NP for the “whole”,

while *antecedent* represents the NP for the “part”), our system will search in a large data set for two queries: “*antecedent * anaphor*” and “*anaphor * antecedent*” where the “*” denotes any sequence of words or symbols. For a returned search results, the text in between “*antecedent*” and “*anaphora*” is extracted as a pattern.

In our study, we used PubMed 2007 data set for the pattern mining. The data set contains about 52,000 abstracts with around 9,400,000 words, and is an ideal large-scale resource for pattern mining.

Consider, as an example, a seed pair “*NK kappa B*” - “*transcription factor*”. Suppose that a returned sentence for the query “*NK kappa B * transcription factor*” is

“...*NK kappa B family transcription factors*...”

And a returned sentence for the query “*transcription factor * NK kappa B*” is

“...*transcription factors, including NF kappa B*...”

We can extract a pattern,

“ANTECEDENT *family* ANAPHOR” from the first sentence and a pattern

“ANAPHOR, *including* ANTECEDENT” from the second sentence.

We restrict the patterns so that no pattern span across two or more sentences. In other words, the pattern shall not contain the symbol “.”. The violated patterns will be removed.

The count that a pattern occurs in the PubMed for a seed pair is recorded. As a pattern could be reduced by different seed pairs, we define the occurrence frequency of a pattern as the sum of the counts of the pattern for all the seed pairs, using following formula:

$$C_{pat_i} = \sum_{s_j \in S} Occ(pat_i, s_j) \quad Eq(1)$$

where C_{pat_i} is the frequency of pattern pat_i ; s_j is a seed pair; S is the set of all seed pairs. $Occ(pat_i, s_j)$ is the count of the pattern pat_i for s_j .

All the mined patterns are sorted according to its frequency as defined in $Eq(1)$.

4.3 Pattern Application

For classifier training and testing, the patterns with high frequency are used as features. In our system, we used the top 40 patterns, while we also examined the influence the number of the patterns on the performance. (See Section 5.2)

Given an instance $fv(candi_j, ana)$ and a pattern feature pat_i , a query is constructed by in-

stantiating with *candi*, *and ana*. For example, for an instance *fv("NF Kappa B", "transcription factors")* and a pattern feature "ANAPHOR, *including* ANTECEDENT", we can get a query "transcription factors, *including* NF kappa B". The query is searched in the PubMed data set. The count of the query is recorded. The value of the pattern feature of a candidate is calculated by normalizing the occurrence frequency among all the candidates of the anaphor.

For demonstration, suppose we have an anaphor "other transcription factors" with two antecedent candidates "IL-10" and "NF kappa B". Given a pattern feature "ANAPHOR, *including* ANTECEDENT", the count of the query "transcription factors, *including* IL-10" is 100 while that for "transcription factors, *including* NF-Kappa B" is 300. Then the values of the pattern feature for "IL-10" and "NF kappa B" are 0.25 ($\frac{100}{100+300}$) and 0.75 ($\frac{300}{100+300}$), respectively.

The value of a pattern feature can be interpreted as a degree of belief that an anaphor and a candidate antecedent have the "part-whole" relation, with regard to the specific pattern. Since the value of a pattern feature is normalized among all the candidates, it could indicate the preference of a candidate against other competing candidates.

5 Experiment Results

5.1 Experiments Setup

In our experiments, we conducted a 3-fold cross validation to evaluate the performances. The total 598 other-anaphora cases were divided into 3 sets of size 200, 199 and 199 respectively. For each experiment, two sets were used for training while the other set was used for testing.

For evaluation, we used the accuracy as the performance metric, which is defined as the correctly resolved other-anaphors divided by all the testing other-anaphors, that is,

$$accuracy = \frac{\# \text{ of correctly resolved anaphors}}{\# \text{ of total anaphors}}$$

5.2 Experiments Results

Table 1 shows the performance of different other-anaphora resolution systems. The first line is for the baseline system with only the normal features as described in Section 3.3. From the table, we can find that the baseline system only achieves around 40% accuracy. A performance is lower than a similar system in news domain by

Modjeska *et al.*, (2003) where they reported 51.6 % precision with 40.6% recall. This difference is probably because they utilized more semantic knowledge such as hypernymy and meronymy acquired from WordNet. Such knowledge, nevertheless, is not easily available in the biomedical domain.

Sys	Fold-1	Fold-2	Fold-3	Overall
Baseline	42.0 %	38.2 %	40.7 %	40.3 %
No Pattern	84/200	76/199	81/199	241/598
Manual Pattern	49.0 %	45.7 %	47.7 %	47.5 %
	98/200	91/199	95/199	284/598
Auto-mined Pattern	59.0 %	53.8 %	56.8 %	56.5 %
	118/200	107/199	113/199	338/598

Table 1: Performance Comparisons

In our experiments, we tested the system with manually designed pattern features. We tried 10 patterns that can represent the "part-whole" relation. Table 2 summarizes the patterns used in the system. Among them, the pattern "Anaphor *such as* Antecedent" and "Antecedent *and other* Anaphor" are commonly used in previous pattern based approaches (Markert *et al.*, 2003; Modjeska *et al.*, 2003).

Pattern
ANTECEDENT <i>is a kind of</i> ANAPHOR
ANTECEDENT <i>is a type of</i> ANAPHOR
ANTECEDENT <i>is a member of</i> ANAPHOR
ANTECEDENT <i>is a part of</i> ANAPHOR
ANAPHOR <i>such as</i> ANTECEDENT
ANTECEDENT <i>and other</i> ANAPHOR
ANTECEDENT <i>within</i> ANAPHOR
ANTECEDENT <i>is a component of</i> ANAPHOR
ANTECEDENT <i>is a sort of</i> ANAPHOR
ANTECEDENT <i>belongs to</i> ANAPHOR

Table 2: Manually Selected Patterns

The second line of Table 1 shows the results of the system with the manual pattern features. We can find that adding these pattern features produces an overall accuracy of 47%, yielding an increase of 7% accuracy against the baseline system without the pattern features.

The improvement in accuracy is consistent with previous work using the pattern-based approaches in the news domain (Modjeska *et al.*, 2003). However, we found the performance in the biomedical domain is worse than that in the news domain. For example, Modjeska *et al.* (2003) reported a precision around 53%. This difference of performance suggests that the ma-

nally designed patterns may not necessarily work equally well in different domains.

The last system we examined in the experiment is the one with the automatically mined pattern features. Table 3 summarizes the top mined patterns ranked based on their occurrence frequency. Some of the patterns are intuitively good representation of the “part-whole” relation. For example, “ANAPHOR, *including* ANTECEDENT”. “ANAPHOR, *such as* ANTECEDENT” and “ANAPHOR *and other* ANTECEDENT” which are in the manually designed pattern list, are generated.

The last line of Table 1 lists the result of the system with automatically mined pattern features. It outperforms the baseline system (up to 16% accuracy), and the system with manually selected patterns (9% accuracy). These results prove that our pattern features are effective for the *other-anaphora* resolution.

Pattern	Freq
ANAPHOR, <i>including</i> ANTECEDENT	1213
ANAPHOR <i>including</i> ANTECEDENT	726
ANTECEDENT <i>family</i> ANAPHOR	583
ANAPHOR <i>such as</i> ANTECEDENT	542
ANTECEDENT <i>transcription</i> ANAPHOR	439
ANAPHOR, <i>such as</i> ANTECEDENT	295
ANTECEDENT <i>and other</i> ANAPHOR	270
ANAPHOR <i>and</i> ANTECEDENT	250
ANTECEDENT, <i>dendritic</i> ANAPHOR	246
ANTECEDENT <i>and</i> ANAPHOR	238
ANTECEDENT <i>human</i> ANAPHOR	223
ANAPHOR (<i>e.g.</i> , ANTECEDENT	213
ANTECEDENT/ <i>rel</i> ANAPHOR	188
ANTECEDENT- <i>like</i> ANAPHOR	188
ANAPHOR <i>against</i> ANTECEDENT	163

Table 3: Auto-Mined Patterns

To further compare the manually designed patterns and the automatically discovered patterns. We examined the coverage rate of the two pattern sets. The coverage rate measures the capability that a set of patterns could lead to positive anaphor-antecedent pairs. An *other-anaphor* is said to be **covered** by a pattern set, if the anaphor and its antecedent could be hit (i.e., the corresponding query has a non-zero hit number) by at least one pattern in the list. Thus the coverage rate could be defined as

$$\text{Coverage}(P) = \frac{\text{\#anaphors covered by the pattern set } P}{\text{\# total anaphors}}$$

The coverage rates of the two pattern sets are tabulated in table 4. It is apparent that the auto-

mined patterns have a significantly higher coverage (more than twice) than the manually designed patterns.

Patterns	Coverage Rate
Manually Designed	36.0 %
Auto-Mined	92.1 %

Table 4: Coverage Comparison

In our experiments we were also concerned about the usefulness of each individual pattern. For this purpose, we examined the loss of the accuracy when withdrawing a pattern feature from the feature list. The top 10 patterns with the largest accuracy loss are summarized in table 5.

Pattern	Acc Loss
ANAPHOR, <i>including</i> ANTECEDENT	4.18%
ANAPHOR <i>including</i> ANTECEDENT	3.18%
ANAPHOR <i>such as</i> ANTECEDENT	2.84%
ANTECEDENT <i>transcription</i> ANAPHOR	2.17%
ANTECEDENT <i>and other</i> ANAPHOR	2.01%
ANAPHOR, <i>such as</i> ANTECEDENT	1.84%
ANTECEDENT <i>family</i> ANAPHOR	1.84%
ANAPHOR (<i>e.g.</i> , ANTECEDENT	1.51%
ANTECEDENT- <i>like</i> ANAPHOR	1.17%
ANTECEDENT/ <i>rel</i> ANAPHOR	1.17%

Table 5: Usefulness of Each Pattern

The process of automatic pattern mining would generate numerous surface patterns. It is not reasonable to use all the patterns as features. As mentioned in section 4.3, we rank the pattern based on their occurrence frequency and select the top ones as the features. It would be interesting to see how the number of patterns influences the performance of anaphora resolution. In figure 2, we plot the accuracy under different number top pattern features. We can find by using more patterns, the coverage keeps increasing. The accuracy also increases, but it reaches the peak with around 40 patterns. With more patterns, the accuracy remains at the same level. This is because the low frequency patterns usually are not that indicative of the “part-whole” relation. Including these pattern features would bring noises but not help the performance. The flat curve after the peak point suggests that the machine learning algorithm can effectively identify the importance of the pattern features for the resolution decision, and therefore including non-indicative patterns would not damage the performance.

In our experiment, we also interested to compare the utility of PubMed with other general data sets. Thus, we tested pattern mining by us-

ing the Google-5-grams corpus⁵ which lists the hit number of all the queries of five words or less in the Web. Unfortunately, we found that the performance is worse than using PubMed. The patterns mined from the Web corpus only gives an accuracy of around 41%, almost the same as the baseline system without using any pattern features. The bad performance is due to the fact that most of bio-medical names are quite long (2~4 words) and occur infrequently in the non-technique data set. Consequently, a query formed by a biomedical seed pair usually cannot be found in the Web corpus (We found the coverage of the auto-mined patterns mined from the corpus is only about 20%).

Performance with various No. of Patterns

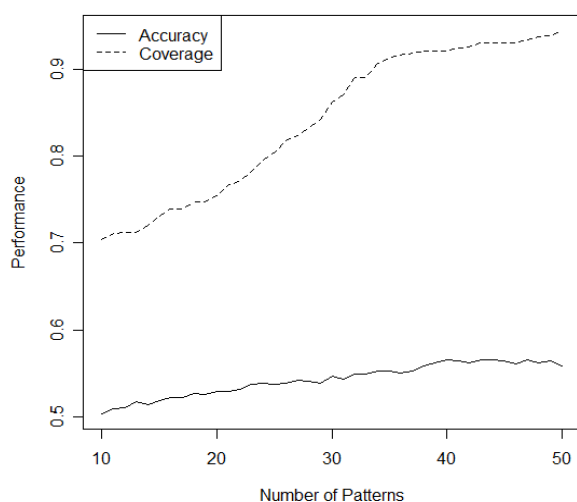


Figure 2: Performance of Various No. of Patterns

6 Conclusion & Future Works

In this paper, we have presented how to automatically mined pattern features for learning-based other-anaphora resolution in bio-medical texts. The patterns that represent the “part-whole” relations are automatically mined from a large data set. They are used as features for a SVM-based classifier learning and testing. The results of our experiments show a reasonably good performance with 56.5% accuracy). It outperforms (16% in accuracy) the baseline system without the pattern features, and also beats (9%) the system with manually designed pattern features.

There are several directions for future work. We would like to employ a pattern pruning process to remove those less indicative patterns such as “ANAPHOR, ANTECEDENT”. And we also plan to perform pattern normalization which integrates two similar or literally identical pat-

terns into a single one. By doing so, the useful patterns may come to the top of the pattern list. Also we would like to explore ontology resources like MESH and Genes Ontology, which can provide enriched hierarchies of bio-medical terms and thus would benefit other-anaphora resolution.

Acknowledgements

This study on co-reference resolution is partially supported by a Specific Targeted Research Project (STREP) of the European Union's 6th Framework Programme within IST call 4, Bootstrapping of Ontologies and Terminologies STrategic REsearch Project (BOOTStrep).

References

- Castano J, Zhang J and Pustejovsky J. Anaphora Resolution in Biomedical Literature. Submitted to *International Symposium on Reference Resolution 2002*, Alicante, Spain
- Clark H. Bridging. In *Thinking. Readings in Cognitive Science*. Johnson-Laird and Wason edition. Cambridge. Cambridge University Press; 1977.411–420
- Gasparin C and Vieira R. Using Word Similarity Lists for Resolving Indirect Anaphora. In *Proceedings of ACL Workshop on Reference Resolution and Its Application. 30 June 2004; Barcelona*. 2004.40-46
- Girju R, Badulescu A and Moldovan D. Automatic Discovery of Part-Whole Relations. *Computational Linguistics*, 2006, 32(2):83-135
- Bernauer J.. Analysis of Part-Whole Relation and Subsumption in Medical Domain. *Data Knowledge Engineering* 1996, 20:405-415
- Markert K. and Nissim M. Comparing Knowledge Sources for Nominal Anaphora Resolution. *Computational Linguistics*, 2005, 31(3):367-402
- Markert K, Modjeska N and Nissim M. Using the Web for Nominal Anaphora Resolution. In *Proceedings of EACL Workshop on the Computational Treatment of Anaphora. 14 April 2003; Budapest*. 2003.39-46
- Mitkov R. Anaphor Resolution. The State of The Art. Working Paper, University of Wolverhampton, UK, 1999
- Modjeska N, Markert K and Nissim M. Using the Web in Machine Learning for Other-anaphor Resolution. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing. July 2003, Sapporo*.176-183
- Soon WM, Ng HT and Lim CY. A Machine Learning Approach to Coreference Resolution of Noun Phrases. *Computational Linguistics*, 2001, 27(4).521-544
- Vapnik, V. Chapter 5 Methods of Pattern Recognition. In *The Nature of Statistical Learning Theory*. New York. Springer-Verlag, 1995.123-167
- Varzi C. Parts, Wholes, and Part-whole Relation. The Prospects of the Mereotopology. *Data & Knowledge Engineering*, 1996, 20.259-286
- Vieira R, Bick E, Coelho J, Muller V, Collovini S, Souza J and Rino L. Semantic Tagging for Resolution of Indirect Anaphora. In *Proceedings of 7th SIGdial Workshop on Discourse and Dialogue. July 2006; Sydney*.76-79
- Burges C. A Tutorial on Supporting Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery* 1998, 2:121-167
- Ng V. and Cardie C. Improving machine learning approaches to coreference resolution. In *Proceedings of Annual Conference for Association of Computational Linguistics 2002*, Philadelphia.104-111

⁵ <http://www ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2006T13>