Coreference Resolution Using Semantic Relatedness Information from Automatically Discovered Patterns

Xiaofeng Yang **Jian Su**
Institute for Infocomm Research
21 Heng Mui Keng Terrace, Singapore, 119613
{xiaofengy,sujian}@i2r.a-star.edu.sg

Abstract

Semantic relatedness is a very important factor for the coreference resolution task. To obtain this semantic information, corpus-based approaches commonly leverage patterns that can express a specific semantic relation. The patterns, however, are designed manually and thus are not necessarily the most effective ones in terms of accuracy and breadth. To deal with this problem, in this paper we propose an approach that can automatically find the effective patterns for coreference resolution. We explore how to automatically discover and evaluate patterns, and how to exploit the patterns to obtain the semantic relatedness information. The evaluation on ACE data set shows that the pattern based semantic information is helpful for coreference resolution.

1 Introduction

Semantic relatedness is a very important factor for coreference resolution, as noun phrases used to refer to the same entity should have a certain semantic relation. To obtain this semantic information, previous work on reference resolution usually leverages a semantic lexicon like WordNet (Vieira and Poesio, 2000; Harabagiu et al., 2001; Soon et al., 2001; Ng and Cardie, 2002). However, the drawback of WordNet is that many expressions (especially for proper names), word senses and semantic relations are not available from the database (Vieira and Poesio, 2000). In recent years, increasing interest has been seen in mining semantic relations from large text corpora. One common solution is to utilize a pattern that can represent a specific semantic relation (e.g., "*X such as Y*" for *is-a* relation, and "*X and other Y*" for *other*-relation). Instantiated with two given noun phrases, the pattern is searched in a large corpus and the occurrence number is used as a measure of their semantic relatedness (Markert et al., 2003; Modjeska et al., 2003; Poesio et al., 2004).

However, in the previous pattern based approaches, the selection of the patterns to represent a specific semantic relation is done in an ad hoc way, usually by linguistic intuition. The manually selected patterns, nevertheless, are not necessarily the most effective ones for coreference resolution from the following two concerns:

- *Accuracy*. Can the patterns (e.g., "*X such as Y*") find as many NP pairs of the specific semantic relation (e.g. *is-a*) as possible, with a high precision?
- *Breadth*. Can the patterns cover a wide variety of semantic relations, not just *is-a*, by which coreference relationship is realized? For example, in some annotation schemes like ACE, "Beijing:China" are coreferential as the capital and the country could be used to represent the government. The pattern for the common "is-a" relation will fail to identify the NP pairs of such a "capital-country" relation.

To deal with this problem, in this paper we propose an approach which can automatically discover effective patterns to represent the semantic relations

Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 528–535,
Prague, Czech Republic, June 2007. ©2007 Association for Computational Linguistics

for coreference resolution. We explore two issues in our study:

(1) *How to automatically acquire and evaluate the patterns?* We utilize a set of coreferential NP pairs as seeds. For each seed pair, we search a large corpus for the texts where the two noun phrases co-occur, and collect the surrounding words as the surface patterns. We evaluate a pattern based on its commonality or association with the positive seed pairs.

(2) *How to mine the patterns to obtain the semantic relatedness information for coreference resolution?* We present two strategies to exploit the patterns: choosing the top best patterns as a set of pattern features, or computing the reliability of semantic relatedness as a single feature. In either strategy, the obtained features are applied to do coreference resolution in a supervised-learning way.

To our knowledge, our work is the first effort that systematically explores these issues in the coreference resolution task. We evaluate our approach on ACE data set. The experimental results show that the pattern based semantic relatedness information is helpful for the coreference resolution.

The remainder of the paper is organized as follows. Section 2 gives some related work. Section 3 introduces the framework for coreference resolution. Section 4 presents the model to obtain the pattern-based semantic relatedness information. Section 5 discusses the experimental results. Finally, Section 6 summarizes the conclusions.

2 Related Work

Earlier work on coreference resolution commonly relies on semantic lexicons for semantic relatedness knowledge. In the system by Vieira and Poesio (2000), for example, WordNet is consulted to obtain the synonymy, hypernymy and meronymy relations for resolving the definite anaphora. In (Harabagiu et al., 2001), the path patterns in WordNet are utilized to compute the semantic consistency between NPs. Recently, Ponzetto and Strube (2006) suggest to mine semantic relatedness from Wikipedia, which can deal with the data sparseness problem suffered by using WordNet.

Instead of leveraging existing lexicons, many researchers have investigated corpus-based approaches to mine semantic relations. Garera and Yarowsky (2006) propose an unsupervised model which extracts hypernym relation for resloving definite NPs. Their model assumes that a definite NP and its hypernym words usually co-occur in texts. Thus, for a definite-NP anaphor, a preceding NP that has a high co-occurrence statistics in a large corpus is preferred for the antecedent.

Bean and Riloff (2004) present a system called BABAR that uses contextual role knowledge to do coreference resolution. They apply an IE component to unannotated texts to generate a set of extraction caseframes. Each caseframe represents a linguistic expression and a syntactic position, e.g. "murder of <NP>", "killed <patient>". From the caseframes, they derive different types of contextual role knowledge for resolution, for example, whether an anaphor and an antecedent candidate can be filled into co-occurring caseframes, or whether they are substitutable for each other in their caseframes. Different from their system, our approach aims to find surface patterns that can directly indicate the coreference relation between two NPs.

Hearst (1998) presents a method to automate the discovery of WordNet relations, by searching for the corresponding patterns in large text corpora. She explores several patterns for the hyponymy relation, including "*X such as Y*" "*X and/or other Y*", "*X including / especially Y*" and so on. The use of Hearst's style patterns can be seen for the reference resolution task. Modjeska et al. (2003) explore the use of the Web to do the *other*-anaphora resolution. In their approach, a pattern "*X and other Y*" is used. Given an anaphor and a candidate antecedent, the pattern is instantiated with the two NPs and forms a query. The query is submitted to the Google searching engine, and the returned hit number is utilized to compute the semantic relatedness between the two NPs. In their work, the semantic information is used as a feature for the learner. Markert et al. (2003) and Poesio et al. (2004) adopt a similar strategy for the *bridging* anaphora resolution.

In (Hearst, 1998), the author also proposes to discover new patterns instead of using the manually designed ones. She employs a bootstrapping algorithm to learn new patterns from the word pairs with a known relation. Based on Hearst's work, Pantel and Pennacchiotti (2006) further give a method which measures the reliability of the patterns based on the strength of association between patterns and instances, employing the pointwise mutual information (PMI).

3 Framework of Coreference Resolution

Our coreference resolution system adopts the common learning-based framework 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 $\boldsymbol{i}\{NP_i,\ NP_j\}$, in which NP_j is a possible anaphor and NP_i is one of its antecedent candidates. An instance is associated with a vector of features, which is used to describe the properties of the two noun phrases as well as their relationships. In our baseline system, we adopt the common features for coreference resolution such as lexical property, distance, string-matching, name-alias, apposition, grammatical role, number/gender agreement and so on. The same feature set is described in (Ng and Cardie, 2002) for reference.

During training, for each encountered anaphor NP_j, one single positive training instance is created for its closest antecedent. And a group of negative training instances is created for every intervening noun phrases between NP_j and the antecedent.

Based on the training instances, a binary classifier can be generated using any discriminative learning algorithm, like C5 in our study. For resolution, an input document is processed from the first NP to the last. For each encountered NP_j, a test instance is formed for each antecedent candidate, NP_i[1]. This instance is presented to the classifier to determine the coreference relationship. NP_j will be resolved to the candidate that is classified as positive (if any) and has the highest confidence value.

In our study, we augment the common framework by incorporating non-anaphors into training. We focus on the non-anaphors that the original classifier fails to identify. Specifically, we apply the learned classifier to all the non-anaphors in the training documents. For each non-anaphor that is classified as positive, a negative instance is created by pairing the non-anaphor and its false antecedent. These negative instances are added into the original training instance set for learning, which will generate a classifier with the capability of not only antecedent identification, but also non-anaphorically identification. The new classier is applied to the testing document to do coreference resolution as usual.

[1]For resolution of pronouns, only the preceding NPs in current and previous two sentences are considered as antecedent candidates. For resolution of non-pronouns, all the preceding non-pronouns are considered.

4 Patterned Based Semantic Relatedness

4.1 Acquiring the Patterns

To derive patterns to indicate a specific semantic relation, a set of seed NP pairs that have the relation of interest is needed. As described in the previous section, we have a set of training instances formed by NP pairs with known coreference relationships. We can just use this set of NP pairs as the seeds. That is, an instance $\boldsymbol{i}\{NP_i,\ NP_j\}$ will become a seed pair (E_i:E_j) in which NP_i corresponds to E_i and NP_j corresponds to E_j. In creating the seed, for a common noun, only the head word is retained while for a proper name, the whole string is kept. For example, instance $\boldsymbol{i}${“Bill Clinton”, “the former president”} will be converted to a NP pair (“Bill Clinton”:“president”).

We create the seed pair for every training instance $\boldsymbol{i}\{NP_i,\ NP_j\}$, except when (1) NP_i or NP_j is a pronoun; or (2) NP_i and NP_j have the same head word. We denote S+ and S- the set of seed pairs derived from the positive and the negative training instances, respectively. Note that a seed pair may possibly belong to S+ can S- at the same time.

For each of the seed NP pairs (E_i:E_j), we search in a large corpus for the strings that match the regular expression “$E_i\ *\ *\ *\ E_j$” or “$E_j\ *\ *\ *\ E_i$”, where * is a wildcard for any word or symbol. The regular expression is defined as such that all the co-occurrences of E_i and E_j with at most three words (or symbols) in between are retrieved.

For each retrieved string, we extract a surface pattern by replacing expression E_i with a mark <#t1#> and E_j with <#t2#>. If the string is followed by a symbol, the symbol will be also included in the pattern. This is to create patterns like “X * * * Y [, . ?]” where *Y*, with a high possibility, is the head word, but not a modifier of another noun phrase.

As an example, consider the pair (“Bill Clinton”:“president”). Suppose that two sentences in a corpus can be matched by the regular expressions:

(S1) " ***Bill Clinton*** *is elected* ***President*** *of the United States.*"

(S2) "*The US* ***President****, Mr* ***Bill Clinton****, today advised India to move towards nuclear non-proliferation and begin a dialogue with Pakistan to* ... ".

The patterns to be extracted for (S1) and (S2), respectively, are

P1: <#t1#> ***is elected*** <#t2#>

P2: <#t2#> ***, Mr*** <#t1#> ***,***

We record the number of strings matched by a pattern p instantiated with (E_i:E_j), noted $|(E_i, p, E_j)|$, for later use.

For each seed pair, we generate a list of surface patterns in the above way. We collect all the patterns derived from the positive seed pairs as a set of reference patterns, which will be scored and used to evaluate the semantic relatedness for any new NP pair.

4.2 Scoring the Patterns

4.2.1 Frequency

One possible scoring scheme is to evaluate a pattern based on its commonality to positive seed pairs. The intuition here is that the more often a pattern is seen for the positive seed pairs, the more indicative the pattern is to find positive coreferential NP pairs. Based on this idea, we score a pattern by calculating the number of positive seed pairs whose pattern list contains the pattern. Formally, supposing the pattern list associated with a seed pair s is PList(s), the frequency score of a pattern p is defined as

$$Freqency(p) = |\{s | s \in S+, p \in PList(s)\}| \tag{1}$$

4.2.2 Reliability

Another possible way to evaluate a pattern is based on its reliability, i.e., the degree that the pattern is associated with the positive coreferential NPs. In our study, we use pointwise mutual information (Cover and Thomas, 1991) to measure association strength, which has been proved effective in the task of semantic relation identification (Pantel and Pennacchiotti, 2006). Under pointwise mutual information (PMI), the strength of association between two events x and y is defined as follows:

$$pmi(x, y) = \log \frac{P(x, y)}{P(x)P(y)} \tag{2}$$

Thus the association between a pattern p and a positive seed pair s:(E_i:E_j) is:

$$pmi(p, (E_i : E_j)) = \log \frac{\frac{|(E_i,p,E_j)|}{|(*,*,*)|}}{\frac{|(E_i,*,E_j)|}{|(*,*,*)|}\frac{|(*,p,*)|}{|(*,*,*)|}} \tag{3}$$

where $|(E_i$,p,$E_j)|$ is the count of strings matched by pattern p instantiated with E_i and E_j. Asterisk * represents a wildcard, that is:

$$|(E_i, *, E_j)| = \sum_{p \in PList(E_i:E_j)} |(E_i, p, E_j)| \tag{4}$$

$$|(*, p, *)| = \sum_{(E_i:E_j) \in S+ \cup S-} |(E_i, p, E_j)| \tag{5}$$

$$|(*, *, *)| = \sum_{(E_i:E_j) \in S+ \cup S-; p \in Plist(E_i:E_j)} |(E_i, p, E_j)| \tag{6}$$

The reliability of pattern is the average strength of association across each positive seed pair:

$$r(p) = \frac{\sum_{s \in S+} \frac{pmi(p,s)}{max\_pmi}}{|S+|} \tag{7}$$

Here *max_pmi* is used for the normalization purpose, which is the maximum PMI between all patterns and all positive seed pairs.

4.3 Exploiting the Patterns

4.3.1 Patterns Features

One strategy is to directly use the reference patterns as a set of features for classifier learning and testing. To select the most effective patterns for the learner, we rank the patterns according to their scores and then choose the top patterns (first 100 in our study) as the features.

As mentioned, the frequency score is based on the commonality of a pattern to the positive seed pairs. However, if a pattern also occurs frequently for the negative seed pairs, it should be not deemed a good feature as it may lead to many false positive pairs during real resolution. To take this factor into account, we filter the patterns based on their accuracy, which is defined as follows:

$$Accuracy(p) = \frac{|\{s | s \in S+, p \in PList(s)\}|}{|\{s | s \in S+ \cup S-, p \in PList(s)\}|} \tag{8}$$

A pattern with an accuracy below threshold 0.5 is eliminated from the reference pattern set. The remaining patterns are sorted as normal, from which the top 100 patterns are selected as features.

		NWire			NPaper			BNews		
		R	P	F	R	P	F	R	P	F
Normal Features		54.5	80.3	64.9	56.6	76.0	64.9	52.7	75.3	62.0
+ "X such as Y"	proper names	55.1	79.0	64.9	56.8	76.1	65.0	52.6	75.1	61.9
	all types	55.1	78.3	64.7	56.8	74.7	64.4	53.0	74.4	61.9
+ "X and other Y"	proper names	54.7	79.9	64.9	56.4	75.9	64.7	52.6	74.9	61.8
	all types	54.8	79.8	65.0	56.4	75.9	64.7	52.8	73.3	61.4
+ pattern features (frequency)	proper names	58.7	75.8	66.2	57.5	73.9	64.7	54.0	71.1	61.4
	all types	59.7	67.3	63.3	57.4	62.4	59.8	55.9	57.7	56.8
+ pattern features (filtered frequency)	proper names	57.8	79.1	66.8	56.9	75.1	64.7	54.1	72.4	61.9
	all types	58.1	77.4	66.4	56.8	71.2	63.2	55.0	68.1	60.9
+ pattern features (PMI_reliability)	proper names	58.8	76.9	66.6	58.1	73.8	65.0	54.3	72.0	61.9
	all types	59.6	70.4	64.6	58.7	61.6	60.1	56.0	58.8	57.4
+ single reliability feature	proper names	57.4	80.8	**67.1**	56.6	76.2	**65.0**	54.0	74.7	**62.7**
	all types	57.7	76.4	65.7	56.7	75.9	64.9	55.1	69.5	61.5

Table 1: The results of different systems for coreference resolution

Each selected pattern p is used as a single feature, PF_p. For an instance $\boldsymbol{i}\{NP_i, NP_j\}$, a list of patterns is generated for $(E_i{:}E_j)$ in the same way as described in Section 4.1. The value of PF_p for the instance is simply $|(E_i, p, E_j)|$.

The set of pattern features is used together with the other normal features to do the learning and testing. Thus, the actual importance of a pattern in coreference resolution is automatically determined in a supervised learning way.

4.3.2 Semantic Relatedness Feature

Another strategy is to use only one semantic feature which is able to reflect the reliability that a NP pair is related in semantics. Intuitively, a NP pair with strong semantic relatedness should be highly associated with as many reliable patterns as possible. Based on this idea, we define the semantic relatedness feature (SRel) as follows:

$$SRel(\boldsymbol{i}\{NP_i, NP_j\}) = 1000 * \sum_{p \in PList(E_i:E_j)} pmi(p, (E_i : E_j)) * r(p) \quad (9)$$

where $pmi(p, (E_i{:}E_j))$ is the pointwise mutual information between pattern p and a NP pair $(E_i{:}E_j)$, as defined in Eq. 3. $r(p)$ is the reliability score of p (Eq. 7). As a relatedness value is always below 1, we multiple it by 1000 so that the feature value will be of integer type with a range from 0 to 1000. Note that among $PList(E_i{:}E_j)$, only the reference patterns are involved in the feature computing.

5 Experiments and Discussion

5.1 Experimental setup

In our study we did evaluation on the ACE-2 V1.0 corpus (NIST, 2003), which contains two data set, training and devtest, used for training and testing respectively. Each of these sets is further divided by three domains: newswire (NWire), newspaper (NPaper), and broadcast news (BNews).

An input raw text was preprocessed automatically by a pipeline of NLP components, including sentence boundary detection, POS-tagging, Text Chunking and Named-Entity Recognition. Two different classifiers were learned respectively for resolving pronouns and non-pronouns. As mentioned, the pattern based semantic information was only applied to the non-pronoun resolution. For evaluation, Vilain et al. (1995)'s scoring algorithm was adopted to compute the recall and precision of the whole coreference resolution.

For pattern extraction and feature computing, we used Wikipedia, a web-based free-content encyclopedia, as the text corpus. We collected the English Wikipedia database dump in November 2006 (refer to http://download.wikimedia.org/). After all the hyperlinks and other html tags were removed, the whole pure text contains about 220 Million words.

5.2 Results and Discussion

Table 1 lists the performance of different coreference resolution systems. The first line of the table shows the baseline system that uses only the common features proposed in (Ng and Cardie, 2002). From the table, our baseline system can

<table>
<tr><th>NO</th><th>Frequency</th><th>Frequency (Filtered)</th><th>PMI_Reliabilty</th></tr>
<tr><td>1</td><td><#t1> <#t2></td><td><#t2> | | <#t1> |</td><td><#t1> : <#t2></td></tr>
<tr><td>2</td><td><#t2> <#t1></td><td><#t1>) is a <#t2></td><td><#t2> : <#t1></td></tr>
<tr><td>3</td><td><#t1> , <#t2></td><td><#t1>) is an <#t2></td><td><#t1> . the <#t2></td></tr>
<tr><td>4</td><td><#t2> , <#t1></td><td><#t2>) is an <#t1></td><td><#t2> (<#t1>)</td></tr>
<tr><td>5</td><td><#t1> . <#t2></td><td><#t2>) is a <#t1></td><td><#t1> (<#t2></td></tr>
<tr><td>6</td><td><#t1> and <#t2></td><td><#t1> or the <#t2></td><td><#t1> (<#t2>)</td></tr>
<tr><td>7</td><td><#t2> . <#t1></td><td><#t1> (the <#t2></td><td><#t1> | | <#t2> |</td></tr>
<tr><td>8</td><td><#t1> . the <#t2></td><td><#t1> . during the <#t2></td><td><#t2> | | <#t1> |</td></tr>
<tr><td>9</td><td><#t2> and <#t1></td><td><#t1> | <#t2></td><td><#t2> , the <#t1></td></tr>
<tr><td>10</td><td><#t1> , the <#t2></td><td><#t1> , an <#t2></td><td><#t1> , the <#t2></td></tr>
<tr><td>11</td><td><#t2> . the <#t1></td><td><#t1>) was a <#t2></td><td><#t2> (<#t1></td></tr>
<tr><td>12</td><td><#t2> , the <#t1></td><td><#t1> in the <#t2> -</td><td><#t1> , <#t2></td></tr>
<tr><td>13</td><td><#t2> <#t1> ,</td><td><#t1> - <#t2></td><td><#t1> and the <#t2></td></tr>
<tr><td>14</td><td><#t1> <#t2> ,</td><td><#t1>) was an <#t2></td><td><#t1> . <#t2></td></tr>
<tr><td>15</td><td><#t1> : <#t2></td><td><#t1> , many <#t2></td><td><#t1>) is a <#t2></td></tr>
<tr><td>16</td><td><#t1> <#t2> .</td><td><#t2>) was a <#t1></td><td><#t1> during the <#t2></td></tr>
<tr><td>17</td><td><#t2> <#t1> .</td><td><#t1> (<#t2> .</td><td><#t1> <#t2> .</td></tr>
<tr><td>18</td><td><#t1> (<#t2>)</td><td><#t2> | <#t1></td><td><#t1>) is an <#t2></td></tr>
<tr><td>19</td><td><#t1> and the <#t2></td><td><#t1> , not the <#t2></td><td><#t2> in <#t1> .</td></tr>
<tr><td>20</td><td><#t2> (<#t1>)</td><td><#t2> , many <#t1></td><td><#t2> , <#t1></td></tr>
<tr><td>…</td><td>…</td><td>…</td><td>…</td></tr>
</table>

Table 2: Top patterns chosen under different scoring schemes

achieve a good precision (above 75%-80%) with a recall around 50%-60%. The overall F-measure for NWire, NPaper and BNews is 64.9%, 64.9% and 62.0% respectively. The results are comparable to those reported in (Ng, 2005) which uses similar features and gets an F-measure of about 62% for the same data set.

The rest lines of Table 1 are for the systems using the pattern based information. In all the systems, we examine the utility of the semantic information in resolving different types of NP Pairs: (1) NP Pairs containing proper names (i.e., Name:Name or Name:Definites), and (2) NP Pairs of all types.

In Table 1 (Line 2-5), we also list the results of incorporating two commonly used patterns, "X(s) such as Y" and "X and other Y(s)". We can find that neither of the manually designed patterns has significant impact on the resolution performance. For all the domains, the manual patterns just achieve slight improvement in recall (below 0.6%), indicating that coverage of the patterns is not broad enough.

5.2.1 Pattern Features

In Section 4.3.1 we propose a strategy that directly uses the patterns as features. Table 2 lists the top patterns that are sorted based on *frequency*, *filtered frequency (by accuracy)*, and *PMI_reliability*, on the NWire domain for illustration.

From the table, evaluated only based on *frequency*, the top patterns are those that indicate the appositive structure like "*X, an/a/the Y*". However, if filtered by *accuracy*, patterns of such a kind will be removed. Instead, the top patterns with both high frequency and high accuracy are those for the copula structure, like "*X is/was/are Y*". Sorted by PMI reliability, patterns for the above two structures can be seen in the top of the list. These results are consistent with the findings in (Cimiano and Staab, 2004) that the appositive and copula structures are indicative to find the *is-a* relation. Also, the two commonly used patterns "X(s) such as Y" and "X and other Y(s)" were found in the feature lists (not shown in the table). Their importance for coreference resolution will be determined automatically by the learning algorithm.

An interesting pattern seen in the lists is "*X* || *Y* |", which represents the cases when Y and X appear in the same of line of a table in Wikipedia. For example, the following text
"*American* || *United States* | *Washington D.C.* | …"
is found in the table "list of empires". Thus the pair "American:United States", which is deemed coreferential in ACE, can be identified by the pattern.

The sixth till the eleventh lines of Table 1 list the results of the system with pattern features. From the table, adding the pattern features brings the improvement of the recall against the baseline. Take the system based on *filtered frequency* as an example. We can observe that the recall increases by up to 3.3% (for NWire). However, we see the precision drops (up to 1.2% for NWire) at the same time. Overall the system achieves an F-measure better than the baseline in NWire (1.9%), while equal ($\pm$0.2%) in NPaper and BNews.

Among the three ranking schemes, simply using *frequency* leads to the lowest precision. By contrast, using *filtered frequency* yields the highest precision with nevertheless the lowest recall. It is reasonable since the low_accuracy features prone to false posi-

```
NameAlias = 1: ...
NameAlias = 0:
:..Appositive = 1: ...
   Appositive = 0:
   :..P014 > 0:
      :...P003 <= 4: 0 (3)
      :   P003 > 4: 1 (25)
      P014 <= 0:
      :..P004 > 0:...
         P004 <= 0:
         :..P027 > 0: 1 (25/7)
            P027 <= 0:
            :..P002 > 0: ...
               P002 <= 0:
               :..P005 > 0: 1 (49/22)
                  P005 <= 0:
                  :..String_Match = 1: .
                     String_Match = 0: .

 // p002: <t1> ) is a <t2>
 // P003: <t1> ) is an <t2>
 // P004: <t2> ) is an <t1>
 // p005: <t2> ) is a <t1>
 // P014: <t1> ) was an <t2>
 // p027: <t1> , ( <t2> ,
```

Figure 1: The decision tree (NWire domain) for the system using pattern features (filtered frequency) (feature *String_Match* records whether the string of anaphor NP_j matches that of a candidate antecedent NP_i)

tive NP pairs are eliminated, at the price of recall. Using *PMI_Reliability* can achieve the highest recall with a medium level of precision. However, we do not find significant difference in the overall F-measure for all these three schemes. This should be due to the fact that the pattern features need to be further chosen by the learning algorithm, and only those patterns deemed effective by the learner will really matter in the real resolution.

From the table, the pattern features only work well for NP pairs containing proper names. Applied on all types of NP pairs, the pattern features further boost the recall of the systems, but in the meanwhile degrade the precision significantly. The F-measure of the systems is even worse than that of the baseline. Our error analysis shows that a non-anaphor is often wrongly resolved to a false antecedent once the two NPs happen to satisfy a pattern feature, which affects precision largely (as an evidence, the decrease of precision is less significant when using *filtered_frequency* than using *frequency*). Still, these results suggest that we just apply the pattern based semantic information in resolving proper names which, in fact, is more compelling as the semantic information of common nouns could be more easily retrieved from WordNet.

We also notice that the patterned based semantic information seems more effective in the NWire domain than the other two. Especially for NPaper, the improvement in F-measure is less than 0.1% for all the systems tested. The error analysis indicates it may be because (1) there are less NP pairs in NPaper than in NWire that require the external semantic knowledge for resolution; and (2) For many NP pairs that require the semantic knowledge, no co-occurrence can be found in the Wikipedia corpus. To address this problem, we could resort to the Web which contains a larger volume of texts and thus could lead to more informative patterns. We would like to explore this issue in our future work.

In Figure 1, we plot the decision tree learned with the pattern features for non-pronoun resolution (NWire domain, *filtered_frequency*), which visually illustrates which features are useful in the reference determination. We can find the pattern features occur in the top of the decision tree, among the features for *name_alias*, *apposition* and *string-matching* that are crucial for coreference resolution as reported in previous work (Soon et al., 2001). Most of the pattern features deemed important by the learner are for the copula structure.

5.2.2 Single Semantic Relatedness Feature

Section 4.3.2 presents another strategy to exploit the patterns, which uses a single feature to reflect the semantic relatedness between NP pairs. The last two lines of Table 1 list the results of such a system.

Observed from the table, the system with the single semantic relatedness feature beats those with other solutions. Compared with the baseline, the system can get improvement in recall (up to 2.9% as in NWire), with a similar or even higher precision. The overall F-measure it produces is 67.1%, 65.0% and 62.7%, better than the baseline in all the domains. Especially in the NWire domain, we can see the significant (*t*-test, $p \leq 0.05$) improvement of 2.1% in F-measure. When applied on All-Type NP pairs, the degrade of performance is less significant as using pattern features. The resulting performance is better than the baseline or equal. Compared with the systems using the pattern features, it can still achieve a higher precision and F-measure (with a little loss in recall) .

There are several reasons why the single semantic relatedness feature (*SRel*) can perform better than the set of pattern features. Firstly, the feature value of *SRel* takes into consideration the information of all the patterns, instead of only the selected patterns. Secondly, since the *SRel* feature is computed based on all the patterns, it reduces the risk of false posi-

```
NameAlias = 1: ...
NameAlias = 0:
:..Appositive = 1: ...
   Appositive = 0:
   :..SRel > 28:
      :..SRel > 47: ...
      :  SRel <= 47: ...
      SRel <= 28:
      :..String_Match = 1: ...
         String_Match = 0: ...
```

Figure 2: The decision tree (Nwire) for the system using the single semantic relatedness feature

tive when a NP pair happens to satisfy one or several pattern features. Lastly, from the point of view of machine learning, using only one semantic feature, instead of hundreds of pattern features, can avoid overfitting and thus benefit the classifier learning.

In Figure 2, we also show the decision tree learned with the semantic relatedness feature. We observe that the decision tree is simpler than that with pattern features as depicted in Figure 1. After feature *name-alias* and *apposite*, the classifier checks different ranges of the *SRel* value and make different resolution decision accordingly. This figure further illustrates the importance of the semantic feature.

6 Conclusions

In this paper we present a pattern based approach to coreference resolution. Different from the previous work which utilizes manually designed patterns, our approach can automatically discover the patterns effective for the coreference resolution task. In our study, we explore how to acquire and evaluate patterns, and investigate how to exploit the patterns to mine semantic relatedness information for coreference resolution. The evaluation on ACE data set shows that the patterned based features, when applied on NP pairs containing proper names, can effectively help the performance of coreference resolution in the recall (up to 4.3%) and the overall F-measure (up to 2.1%). The results also indicate that using the single semantic relatedness feature has more advantages than using a set of pattern features.

For future work, we intend to investigate our approach in more difficult tasks like the bridging anaphora resolution, in which the semantic relations involved are more complicated. Also, we would like to explore the approach in technical (e.g., biomedical) domains, where jargons are frequently seen and the need for external knowledge is more compelling.

Acknowledgements This research is 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

D. Bean and E. Riloff. 2004. Unsupervised learning of contextual role knowledge for coreference resolution. In *Proceedings of NAACL*, pages 297–304.

P. Cimiano and S. Staab. 2004. Learning by googling. *SIGKDD Explorations Newsletter*, 6(2):24–33.

T. Cover and J. Thomas. 1991. *Elements of Information Theory*. Hohn Wiley & Sons.

N. Garera and D. Yarowsky. 2006. Resolving and generating definite anaphora by modeling hypernymy using unlabeled corpora. In *Proceedings of CoNLL* , pages 37–44.

S. Harabagiu, R. Bunescu, and S. Maiorano. 2001. Text knowledge mining for coreference resolution. In *Proceedings of NAACL*, pages 55–62.

M. Hearst. 1998. Automated discovery of wordnet relations. In Christiane Fellbaum, editor, *WordNet: An Electronic Lexical Database and Some of its Applications*. MIT Press, Cambridge, MA.

K. Markert, M. Nissim, and N. Modjeska. 2003. Using the web for nominal anaphora resolution. In *Proceedings of the EACL workshop on Computational Treatment of Anaphora*, pages 39–46.

N. Modjeska, K. Markert, and M. Nissim. 2003. Using the web in machine learning for other-anaphora resolution. In *Proceedings of EMNLP*, pages 176–183.

V. Ng and C. Cardie. 2002. Improving machine learning approaches to coreference resolution. In *Proceedings of ACL*, pages 104–111, Philadelphia.

V. Ng. 2005. Machine learning for coreference resolution: From local classification to global ranking. In *Proceedings of ACL*, pages 157–164.

P. Pantel and M. Pennacchiotti. 2006. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of ACL*, pages 113–1200.

M. Poesio, R. Mehta, A. Maroudas, and J. Hitzeman. 2004. Learning to resolve bridging references. In *Proceedings of ACL*, pages 143–150.

S. Ponzetto and M. Strube. 2006. Exploiting semantic role labeling, wordnet and wikipedia for coreference resolution. In *Proceedings of NAACL*, pages 192–199.

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

R. Vieira and M. Poesio. 2000. An empirically based system for processing definite descriptions. *Computational Linguistics*, 27(4):539–592.

M. Vilain, J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of the Sixth Message understanding Conference (MUC-6)*, pages 45–52, San Francisco, CA. Morgan Kaufmann Publishers.