Coling 2010

23rd International Conference on Computational Linguistics

Proceedings of the

Second Workshop on NLP Challenges in the Information Explosion Era (NLPIX 2010)

28 August 2010
Beijing International Convention Center
Beijing, China

Produced by
Chinese Information Processing Society of China

To order the CD of Coling 2010 and its Workshop Proceedings, please contact:

Chinese Information Processing Society of China
No.4, Southern Fourth Street
Haidian District, Beijing, 100190
China
Tel: +86-010-62562916
Fax: +86-010-62562916
cips@iscas.ac.cn

Introduction

A long-standing problem in Natural Language Processing has been a lack of large-scale knowledge for computers. The emergence of the Web and the rapid increase of information on the Web brought us to what could be called the ”information explosion era,” and drastically changed the environment of NLP. The Web is not only a marvelous target for NLP, but also a valuable resource from which knowledge could be extracted for computers. Motivated by the desire to have a very first opportunity to discuss early approaches to those issues and to share the state-of-the-art technologies at that time, the first International Workshop on NLP Challenges in the Information Explosion Era (NLPIX 2008) was successfully held in conjunction with WWW 2008 in Beijing.

Since the discussion of the first workshop, research and development activities on large-scale text processing and large-scale knowledge acquisition become much more popular these days. The large-scale NLP naturally requires large-scale infrastructures, such as neatly-prepared huge corpora, robust morpho-syntactic tools, and high-performance computing environments. However, such infrastructures can not be prepared by individual researchers nor research groups alone in general, although of course we know some exceptions. Based on this motivation, towards much larger-scale NLP, activities aiming at constructing and sharing the infrastructures have continued. Although we have found many publications presented in recent conferences/workshops including the above mentioned workshop, we still do not have opportunities to compare latest approaches, share analysis on advantages/disadvantages, and discuss possible directions towards further improvement and innovation.

Furthermore, beyond the success of large-scale NLP and knowledge acquisition, we are starting to face a new problem: how to manage and use the automatically acquired knowledge (AAK in short). We are still not confident that those large-scale AAK can actually solve real world problems. How to incorporate the AAK into existing NLP frameworks and how to manage them are yet unsolved issues. One approach could be some bootstrapping of extracting knowledge and enhancing NLP based on the knowledge. The representation and standardization of AAK are also emerging important issues. One of the most highly demanded applications for AAK-based NLP is a semantic search to cope with the information explosion on the Web. Though our daily life heavily depends on the Web information, our diversified needs have not been sufficiently satisfied by the existing search engines. AAK-based NLP can be a key technology to realize a new-generation semantic search, which incorporates enhanced information access, analysis and organization.

The aim of the second workshop of the series of International Workshop on NLP Challenges in the Information Explosion Era (NLPIX) is to bring researchers and practitioners together in order to discuss large-scale and sharable NLP infrastructures, and furthermore to discuss emerging NEW issues beyond them. The program committee accepted 9 papers that cover wide variety of topics such as lexical acquisition, lexical semantics, coreference, and information access, many of which are based on very large scale Web text data.

The invited talks were given by Hang Li (Microsoft Research Asia) and Hoifung Poon (University of Washington).

We are grateful to Info-plosion (New IT Infrastructure for the Information-explosion Era, Grant-in-Aid for Scientific Research on Priority Areas, MEXT (Ministry of Education, Culture, Sports, Science and Technology, Japan)) for partially sponsoring the workshop. We would like to thank all the authors who submitted papers. We express our deepest gratitude to the committee members for their timely reviews. We also thank the COLING 2010 organizers for their help with administrative matters.

Sadao Kurohashi and Takehito Utsuro

Co-Organizers

Organizers:

Sadao Kurohashi, Kyoto University (Japan)
Takehito Utsuro, University of Tsukuba (Japan)

Program Committee:

Pushpak Bhattacharyya, IIT (India)
Thorsten Brants, Google (USA)
Eric Villemonte de la Clergerie, INRIA (France)
Atsushi Fujii, Tokyo Institute of Technology (Japan)
Julio Gonzalo, UNED (Spain)
Kentaro Inui, Tohoku University (Japan)
Noriko Kando, NII (Japan)
Daisuke Kawahara, NICT (Japan)
Jun'ichi Kazama, NICT (Japan)
Adam Kilgarriff, Lexical Computing Ltd. (UK)
Gary Geunbae Lee, POSTECH (Korea)
Hang Li, Microsoft (China)
Dekang Lin, Google (USA)
Tatsunori Mori, Yokohama National University (Japan)
Satoshi Sekine, New York University (USA)
Kenjiro Taura, University of Tokyo (Japan)
Kentaro Torisawa, NICT (Japan)
Marco Turchi, European Commission - Joint Research Centre (Italy)
Yunqing Xia, Tsinghua University (China)

Additional Reviewer:

Wei Wu, Microsoft (China)

Invited Speakers:

Hang Li, Microsoft (China)
Hoifung Poon, University of Washington (USA)

Table of Contents

Workshop Program

Saturday, August 28, 2010

9:30 *Opening*

9:40–10:30 **Invited Talk I**

Query Understanding in Web Search - by Large Scale Log Data Mining and Statistical Learning
Hang Li

10:30–11:00 *Tea Break*

11:00–12:15 **Session I: Information Access**

Exploiting Term Importance Categories and Dependency Relations for Natural Language Search
Keiji Shinzato and Sadao Kurohashi

Summarizing Search Results using PLSI
Jun Harashima and Sadao Kurohashi

Automatic Classification of Semantic Relations between Facts and Opinions
Koji Murakami, Eric Nichols, Junta Mizuno, Yotaro Watanabe, Hayato Goto, Megumi Ohki, Suguru Matsuyoshi, Kentaro Inui and Yuji Matsumoto

12:15–13:30 *Lunch*

13:30–14:20 **Invited Talk II**

Statistical Relational Learning for Knowledge Extraction from the Web
Hoifung Poon

14:20–15:35 **Session II: Lexical Acquisition**

Even Unassociated Features Can Improve Lexical Distributional Similarity
Kazuhide Yamamoto and Takeshi Asakura

A Look inside the Distributionally Similar Terms
Kow Kuroda, Jun'ichi Kazama and Kentaro Torisawa

Utilizing Citations of Foreign Words in Corpus-Based Dictionary Generation
Reinhard Rapp and Michael Zock

Saturday, August 28, 2010 (continued)

15:35–16:00 *Tea Break*

16:00–17:15 **Session III: Coreference and Semantics**

Large Corpus-based Semantic Feature Extraction for Pronoun Coreference
Shasha Liao and Ralph Grishman

Mining Coreference Relations between Formulas and Text using Wikipedia
Minh Nghiem Quoc, Keisuke Yokoi, Yuichiroh Matsubayashi and Akiko Aizawa

Adverse-Effect Relations Extraction from Massive Clinical Records
Yasuhide Miura, Eiji Aramaki, Tomoko Ohkuma, Masatsugu Tonoike, Daigo Sugihara, Hiroshi Masuichi and Kazuhiko Ohe

Query Understanding in Web Search
- by Large Scale Log Data Mining and Statistical Learning

Hang Li
Microsoft Research Asia, China

Abstract

Query understanding is an important component of web search, like document understanding, query document matching, ranking, and user understanding. The goal of query understanding is to predict the user's search intent from the given query. Needless to say, search log mining and statistical learning are fundamental technologies to address the task of query understanding. In this talk, I will first introduce a large-scale search log mining platform which we have developed at MSRA. I will then explain our approach to query understanding, as well as document understanding, query document matching, and user understanding. After that, I will describe in details about our methods for query understanding based on statistical learning. They include query refinement using CRF, named entity recognition in query using topic model, context aware query topic prediction using HMM.

This is joint work with Gu Xu, Daxin Jiang and other collaborators.

Proceedings of the Second Workshop on NLP Challenges in the Information Explosion Era (NLPIX 2010), page 1,
Beijing, August 2010

Exploiting Term Importance Categories and Dependency Relations for Natural Language Search

Keiji Shinzato
Graduate School of Informatics,
Kyoto University
shinzato@i.kyoto-u.ac.jp

Sadao Kurohashi
Graduate School of Informatics,
Kyoto University
kuro@i.kyoto-u.ac.jp

Abstract

In this paper, we propose a method that clearly separates terms (words and dependency relations) in a natural language query into important and other terms, and differently handles the terms according to their importance. The proposed method uses three types of term importance: necessary, optional, and unnecessary. The importance are detected using linguistic clues. We evaluated the proposed method using a test collection for Japanese information retrieval. Performance was resultantly improved by differently handling terms according to their importance.

1 Introduction

Currently, search engines that receive a couple of keywords reflecting users' information needs predominate. These keyword-based searches have been focused on evaluation conferences for information retrieval (IR) such as TREC and NTCIR. Search engines based on keywords, however, have a crucial problem that it is difficult for their users to represent complex needs, such as "*I want to know what Steve Jobs said about the iPod.*" A natural language sentence can more adeptly accommodate such information needs than a couple of keywords because users can straightforwardly present their needs. We call a query represented by a sentence a natural language query (NLQ).

The other advantage of NLQs is that search engines can leverage dependency relations between words in a given query. Dependency relations allow search engines to retrieve documents with a similar linguistic structure to that of the query. Search performance improvement can be expected through the use of dependency relations.

For handling an NLQ, we can consider a conjunctive search (AND search) that retrieves documents that include all terms in the query, a simple methodology similar to real-world Web searches. This methodology, however, often leads to insufficient amounts of search results. In some instances, no documents match the query. This problem occurs because the amount of search results is inversely proportional to the number of terms used in a search; and an NLQ includes many terms. Hence, a conjunctive search simply using all terms in an NLQ is problematic.

Apart from this, we can consider conventional IR methodology. This approach performs a disjunctive search (OR search), and then ranks retrieved documents according to scores that are computed by term weights derived from retrieval models. The methodology attempts to use term weights to distinguish important terms and other items. However, a problem arises in that irrelevant documents are more highly ranked than relevant ones when giving NLQs. This is because an NLQ tends to contain some important terms and many noisy (redundant) terms and document relevancy is calculated from the combinations of these term weights.

Avoiding the above problems, we define three discrete categories of term importance: *necessary*; *optional*, and *unnecessary*, and propose a method that classifies words and dependency relations in an NLQ into term importance, and then, when performing document retrieval, differently handles the terms according to their importance. The necessary type includes expressions in Named Enti-

Proceedings of the Second Workshop on NLP Challenges in the Information Explosion Era (NLPIX 2010), pages 2–11,
Beijing, August 2010

ties (NEs) and compound nouns, the optional includes redundant verbs and the unnecessary includes expressions that express inquiries such as "*I want to find*." The process of IR consists of two steps: document collecting and document scoring. The proposed method uses only necessary terms for document collecting and necessary and optional terms for document scoring.

We evaluated the proposed method using the test collections built at the NTCIR-3 and NTCIR-4 conferences for evaluating Japanese IR. Search performance was resultantly improved by differently handling terms (words and dependency relations) according to their importance.

This paper is organized as follows. Section 2 shows related work, and section 3 describes how to leverage dependency relations in our retrieval method. Section 4 presents term importance categories, and section 5 gives methodology for detecting such categories. Experiment results are shown in section 6.

2 Related Work

A large amount of the IR methodology that has been proposed (Robertson et al., 1992; Ponte and Croft, 1998) depends on retrieval models such as probabilistic and language models. Bendersky and Croft (Bendersky and Croft, 2008), for instance, proposed a new language model in which important noun phrases can be considered.

IR methodology based on important term detection has also been proposed (Callan et al., 1995; Allan et al., 1997; Liu et al., 2004; Wei et al., 2007). These previous methods have commonly focused on noun phrases because the methods assumed that a document relates to a query if the two have common noun phrases. Liu et al. (Liu et al., 2004) classified noun phrases into four types: proper nouns, dictionary phrases (e.g., computer monitor), simple phrases, and complex phrases, and detected them from a *keyword-based* query by using named entity taggers, part-of-speech patterns, and dictionaries such as WordNet. The detected phrases were assigned different window sizes in a proximity operator according to their types. Wei et al. (Wei et al., 2007) extended Liu's work for precisely detecting noun phrases. Their method used hit counts obtained from Google and Wikipedia in addition to clues used in Liu's work. The differences between the proposed method and these methods are (i) the proposed method focuses on an NLQ while the previous methods focus on a keyword-based query, (ii) the proposed method needs no dictionaries, and (iii) while the previous methods retrieve documents by proximity searches of words in phrases, the proposed method retrieves them by dependency relations in phrases. Therefore, the proposed method does not need to adjust window size, and naturally performs document retrieval based on noun phrases by using dependency relations.

Linguistically motivated IR research pointed out that dependency relations did not contribute to significantly improving performance due to low accuracy and robustness of syntactic parsers (Jones, 1999). Current state-of-the-art parsers, however, can perform high accuracy for real-world sentences. Therefore, dependency relations are remarked in IR (Miyao et al., 2006; Shinzato et al., 2008b). For instance, Miyao et al. (Miyao et al., 2006) proposed an IR system for a biomedical domain that performs deep linguistic analysis on a query and each document. Their system represented relations between words by a predicate-argument structure, and used ontological databases for handling synonyms. Their experiments using a small number of *short* queries showed that their proposed system significantly improved search performance versus a system not performing deep linguistic analysis. Shinzato et al. (Shinzato et al., 2008b) proposed a Web search system that handles not only words but also dependency relations as terms; yet they did not discuss the effectiveness of dependency relations. This paper reveals the effectiveness of dependency relations through experiments using test collections for Japanese Web searches.

3 Exploitation of Dependency Relation

One of the advantages of an NLQ is leveraging dependency relations between words in the query. We can expect that search performance improves because the dependency relations allow systems to retrieve documents that have similar linguistic structure to that of the query. Therefore the proposed method exploits dependency relations for

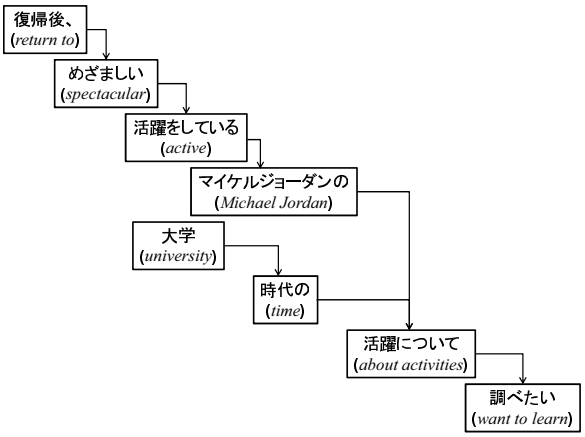

Figure 1: Parsing result of an NLQ.

retrieving documents. Though a dependency relation is generally a relation between two clauses, we regard a relation between two content words as a dependency relation. More precisely, we represent a dependency relation by a *directed* binary relation of content words, and discard the case marker between content words. Also, (compound) functional words such as "について (*about*)" and "に従って (*according to*)" are attached to the former content word. Figure 1 shows the parsing result of the query "復帰後，めざましい活躍をしているマイケルジョーダンの大学時代の活躍について調べたい.[1]" The pair of content words ⟨大学 (*university*), 時代 (*time*)⟩ is extracted as a dependency relation from the parsing result. Note that the pair of content words ⟨ 時代 (*time*), 大学 (*university*)⟩ is not extracted as a dependency relation because a dependency relation is represented by a directed binary relation.

We used Okapi BM25 (Robertson et al., 1992) for estimating relevancy between a query and a document, which is how it is used in most case, though we slightly extend this measure for estimating relevancy for dependency relations. We denote a set of words in a query q as $T_{q_{word}}$, and also denote a set of dependency relations in q as $T_{q_{dpnd}}$. The relevancy between query q and document d is as follows:

$$R(q,d) = (1-\beta)\sum_{t \in T_{q_{word}}} BM(t,d) + \beta \sum_{t \in T_{q_{dpnd}}} BM(t,d),$$

where β is a parameter for adjusting the ratio of a score calculated from dependency relations. The score $BM(t,d)$ is defined as:

$$BM(t,d) = w \times \frac{(k_1+1)F_{dt}}{K+F_{dt}} \times \frac{(k_3+1)F_{qt}}{k_3+F_{qt}},$$
$$w = \log \frac{N-n+0.5}{n+0.5}, K = k_1((1-b)+b\frac{l_d}{l_{ave}}).$$

Here, F_{dt} is the frequency with which t appears in document d, F_{qt} is the frequency that t appears in q, N is the number of documents being searched, n is the document frequency of t, l_d is the length of document d (words), and l_{ave} is the average document length. Finally, k_1, k_3, and b, are Okapi parameters, for which we use values $k_1 = 1$, $k_3 = 0$ and $b = 0.6$.

4 Term Importance Category

Conventional IR methodology regards weights estimated by retrieval models, such as probabilistic and language models, as term importance. The methods depending on the term weights, however, cause a problem in that irrelevant documents are more highly ranked than relevant ones when an NLQ is given. This is because (i) NLQs tend to contain some important terms and a large quantity noise (redundant terms) and (ii) document relevancy is estimated by the combinations of these term weights.

Avoiding this problem, term importance is clearly separated, instead of representing by weights. We propose three term-importance categories and methodology that differently handles terms according to their importance categories. These categories are defined as follows:

Necessary: Terms that must be in retrieved documents. We can also consider a proximity constraint so that all retrieved documents must contain necessary terms within N words.

Optional: Terms preferable for inclusion in retrieved documents.

Unnecessary: Terms for which it does not matter if they are included in retrieved documents.

In this paper, terms in *necessary*, *optional* and *unnecessary* categories are referred to as *necessary terms*, *optional terms*, and *unnecessary terms*, respectively.

[1] This means that Michael Jordan's performance has been spectacular since his return to NBA, and I want to learn about his activities when he was a university student.

IR methodology consists of two steps: document collecting and document scoring. In the proposed method, document collecting is performed using only necessary terms, document scoring is performed using both necessary and optional terms, and neither step uses unnecessary terms.

As mentioned, the proposed method retrieves documents exploiting not only words but also dependency relations. Though a conjunctive search with words and dependency relations can be considered, the proposed method basically only uses words. In short, words are handled as *necessary* terms, while dependency relations are handled as *optional* terms. This is because the number of documents that include all dependency relations tends to be small. Importance of words and dependency relations is, however, revised depending on whether they can be regarded as important expressions. The revision methodology is described in the next section.

5 Revision of Term Importance

The proposed method basically deals with words and dependency relations as necessary terms and optional terms, respectively. However, the term importance of the following words and dependency relations are revised.

1. Dependency relations in NEs and strongly connected compound nouns.
2. Redundant verbs, verbs whose meaning can be inferred from surrounding nouns.
3. Words and dependency relations in inquiry expressions and functional expressions.

This section describes how to recognize the above expressions and revise the term importance of the recognized expressions.

5.1 Named Entity and Strongly Connected Compound Noun

The term importance of all dependency relations in Named Entities (NEs) is revised to a *necessary* category. We believe that a user entering a search engine query including an NE expects to obtain documents that include the NE. For instance, if a user's query includes "*American Bank*," the user prefers documents that include "*American Bank*" to those with the individual words "*American*" and "*Bank*." That is why the proposed method revises the term importance of all dependency relations in an NE to a necessary category. This revision guarantees that search engine users will obtain documents including the NEs in a query.

In addition to NEs, for some compound nouns a search engine user prefers to obtain documents that include the compound noun rather than the individual words in the compound noun. We refer to this as a **Strongly Connected Compound Noun** (SCCN). An example of an SCCN is "*information science*." In the same way as "*American Bank*," a user whose search engine query contains "*information science*" expects to obtain documents that include "*information science*" rather than with the individual words "*information*" and "*science*."

On the other hand, there are also compound nouns, such as "*Kyoto sightseeing*", that do not need to be included in retrieved documents as a *single phrase*. For these, a user approves of retrieved documents that include "*Kyoto*" and "*sightseeing*" separately. We therefore need criteria for distinguishing such compound nouns and SCCNs.

The problem is how to compute the connection strength of words in a compound noun N (i.e., $w_1, ..., w_{|N|}$). For computing the connection strength among words in N, we assumed that words in an SCCN are unlikely to occur in documents as "w_i の w_{i+1} (w_{i+1} *of* w_i)". This assumption reflects the observation that "*Kyoto sightseeing*" is likely to be expressed as "*sightseeing of Kyoto*" and that "*information science*" is unlikely to be expressed by "*science of information*." In line with this assumption, the connection strength is calculated as follows:

$$Score_{strength}(N) = \frac{1}{|N|-1} \sum_{i=1}^{|N|-1} \frac{DF(w_i \ w_{i+1})}{DF(w_{i+1} の w_i)}.$$

Here, $DF(X)$ is the document frequency of X computed from hundreds of millions Japanese Web pages (Shinzato et al., 2008a). The proposed method regards a compound noun N as an SCCN if the value of $Score_{strength}(N)$ exceeds a threshold T_p. We used the value of 300 as the threshold. In addition to dependency relations in NEs,

the term importance of dependency relations in an SCCN is also revised from an *optional* category to a *necessary* category.

5.2 Redundant Verb

The proposed method deals with a verb whose meaning is inferable from the surrounding nouns as an optional term. We refer to such a verb a *redundant verb*.

Consider the following two expressions:

(A) 作家 (*author*) の (*of*) 書いた (*wrote*) 本 (*book*)
(*A book written by an author*)

(B) 作家 (*author*) の (*of*) 本 (*book*)
(*A book of an author*)

The expression (A) is often paraphrased as the expression (B) which omits the verb "*write*." However, we can recognize that (A) is equivalent to (B). This is because the meaning of the verb "*write*" can be inferred from the noun "*author*." In other words, the noun "*author*" can be considered to imply the meaning of the verb "*write*." According to this observation, we assumed that a verb whose meaning is inferable from the surrounding nouns does not need to be included in retrieved documents.

For computing redundancy of verbs, we made the assumption that a noun n implies the meaning of a verb v if a syntactic dependency relation between a noun n and a verb v frequently occurs in corpora. We defined the following score function according to the assumption.

$$Score_{cooc}(n, v) = P(n, v) \cdot \log_2 \frac{P(n, v)}{P(n) \cdot P(v)},$$

where $P(n)$ and $P(v)$ indicate the probabilities of a noun n and a verb v respectively. $P(n, v)$ is the probability of a dependency relation between a noun n and a verb v. These probabilities were estimated from 1.6 billion Japanese sentences extracted from the hundreds of millions of Japanese pages used for computing $DF(X)$ in the previous section.

For each noun n that is the parent-of or child-of dependency relation of a verb v, the above score is calculated. We consider that the meaning of a verb v can be inferred from a noun n if the value

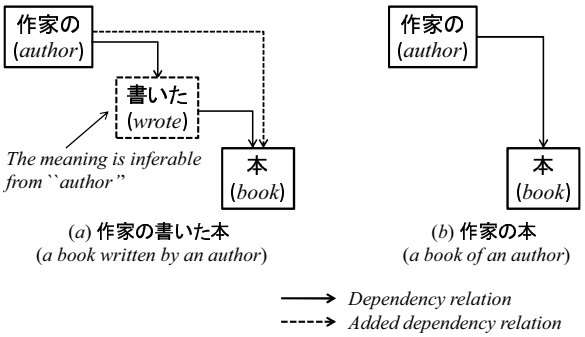

Figure 2: Structural difference between "作家の書いた本 (*a book written by an author*)" and "作家の本 (*a book of an author*)".

of $Score_{cooc}(n, v)$ exceeds a threshold T_v. The value of the threshold is used 1×10^{-6} which was decided empirically. For instance, the nouns *author* and *book* in Figure 2 (a) are used for computing the above score with respect to the verb *wrote*, and then *wrote* is regarded as a *redundant verb* if either one exceeds the threshold.

When a verb v is regarded as an optional term (i.e., v is a redundant verb), the proposed method appends a new dependency relation consisting of the parent-of and child-of dependency relation of the redundant verb v. Figure 2 (a) shows the parsing result of the expression (A). A new dependency relation between "*author*" and "*book*" is depicted by a *dashed arrow*. Figure 2 (b) shows the parsing result of the expression (B). Though there is a structural gap between the expressions (A) and (B), this gap is bridged by the new dependency relation because the dependency relation (`author`, `book`) is contained in the both expressions.

5.3 Inquiry Expressions and Functional Words

An NLQ tends to contain expressions, such as "*I want to find*" and "*I want to know*," and such expressions almost never relate to users' information needs. Therefore we regard words and dependency relations in these expressions as unnecessary terms. To do so, we crafted the *inquiry pattern* shown in Figure 3. The importance of words and dependency relations in the matched expressions is revised to an *unnecessary* category if expressions in a query matched the pattern. The spelling variations of words, such as "探す (find)"

INQUIRY PATTERN:
<EPITHET>?<EXPOSITION>? <DOC>?(について (*about*))?<PREDICATE>;

<EPITHET>: [詳しい (*in detail*) | 詳細だ (*in detail*)];

<EXPOSITION>: [説明 (*explain*) | 書く (*write*) | 記述 (*describe*) | 記載 (*mention*) | 記す (*write down*) | 述べる (*express*)][する (*do*)]? [(いる (*be*) | ある (*be*) | れる (*reru*) | られる (*rareru*)]?;

<DOC>: [ウェブ (*Web*) | ＷＥＢ (*Web*)]? [文書 (*document*) | ページ (*page*) | ＨＰ (*homepage*) | 情報 (*information*) | 文章 (*sentences*) | テキスト (*text*)];

<PREDICATE>: [知る (*know*) | 探す (*look for*) | 調べる (*find*) | 見る (*watch*) | 見つける (*find out*) | 読む (*read*)][たい (*tai*) | いる (*iru*)];

Figure 3: Inquiry patterns. The notation [A | B] indicates A*or* B and the symbol '?' indicates that an expression in front of the symbol may be omitted. The words *reru, rareru, tai* and *iru* are Japanese functional words.

and "さがす (find)" are properly handled when matching an inquiry pattern.

In addition to the inquiry expressions, we can consider that content words that play a role like functional words, such as ある (*be*), なる (*become*), and 使う (*use*), are unnecessary for retrieving documents. To detect these words we constructed an unnecessary content word list.

6 Experiments

6.1 Settings

We evaluated the proposed method by using the test collections built at the NTCIR-3 (Eguchi et al., 2003) and NTCIR-4 (Eguchi et al., 2004) conferences. These share a target document set, which consists of 11,038,720 Japanese Web pages. For the evaluation, we used 127 informational topics defined in the test collections (47 from NTCIR-3 and 80 from NTCIR-4). An example of the informational topic definition is shown in Figure 4. <DESC> includes a sentence reflecting the user's information needs; the sentence can be regarded as an NLQ. Therefore, we used only <DESC> as a query in the experiments. The relevance of each document with respect to a topic was judged as *highly relevant*, *relevant*, *partially relevant*, *irrelevant* or *unjudged*. We regarded the highly relevant, relevant, and partially relevant documents as *correct* answers.

The process of IR consists of two steps: document collecting and document scoring. In both steps, the proposed method considered synonyms automatically extracted from ordinary dictionaries and Web pages (Shibata et al., 2008). For calculating the scores, we selected the value of 0.2 as the parameter β. This value was estimated using the dry-run data set of NTCIR-3.

```
<TOPIC><NUM> 0008 </NUM><TITLE> Salsa,
learn, methods </TITLE><DESC> I want to
find out about methods for learning how
to dance the salsa </DESC> .. </TOPIC>
```

Figure 4: Example of a search topic.

For each topic, we retrieved 1,000 documents and then assessed search performance according to MRR, P@10, R-prec, MAP, DCG_N (Jarvelin and Kekalainen, 2002), and Q-Measure (QM) (Sakai, 2004). We calculated these scores for each topic then averaged them. Note that *unjudged* documents were treated as *irrelevant* when computing the scores. As the graded relevance for DCG_N and QM, we mapped *highly relevant*, *relevant* and *partially relevant* to 3, 2 and 1, respectively.

The proposed method often leads to an insufficient number of search results because the method performs a conjunctive search using necessary terms. Therefore, evaluation measures, such as QM, which utilize low-ranked search results for computing their scores, give low scores in the proposed method. To avoid this problem we combine the proposed method with an OR (dpnd) search, which is described in the next section. More precisely, let $R(d)$ denote the rank given by the proposed method for a document d, and $R_{OR}(d)$ denote the rank given by the OR(dpnd) search. The final score for a document d is defined as:

$$S(d) = \frac{1}{R(d)} + \frac{1}{R_{OR}(d)}$$

The documents collected by the proposed method and the OR(dpnd) search are sorted according to values of $S(d)$, and then the top 1,000 of the sorted documents are regarded as the search result of the proposed method. Note that the search result of the OR(dpnd) search is dealt with fusing the proposed method when the number of search results of the proposed method is zero.

All NLQs extracted from <DESC> were an-

Table 1: Comparison between the proposed method and alternative methods.

Methods	AND		OR		OR (dpnd)			AND$_{prox}$+ OR (dpnd)			Proposed method				
Prox. & Terms	Prox.	Word	Prox.	Word	Prox.	Word	Dpnd.	Prox.	Word	Dpnd.	Prox.	Word		Dpnd.	
												Normal	RV	Normal	NEs & SCCNs
Search conditions	No	◯	No	△	No	△	△	Yes	◯	△	Yes	◯	△	△	◯
								No	△	△	No	△	△	△	△
MRR	0.533		0.538		0.503			**0.547**			0.537				
P@10	0.328		0.337		0.352			0.352			**0.357**				
DCG$_{10}$	3.469		3.497		3.583			3.634			**3.713**				
DCG$_{100}$	7.191		8.898		9.167			9.045			**9.280**				
DCG$_{1000}$	8.956		16.221		16.553			16.678			**16.866**				
R-prec	0.174		0.207		0.212			0.217			**0.221**				
MAP	0.120		0.151		0.158			0.161			**0.164**				
QM	0.095		0.168		0.175			0.179			**0.183**				

Prox: Proximity, Dpnd: Dependency relation, RV: Redundant verb.

alyzed by the JUMAN[2], Japanese morphological analyzer and KNP[3], Japanese syntactic parser which implemented the named entity recognition feature proposed by Sasano and Kurohashi (Sasano and Kurohashi, 2008). All documents were also analyzed by JUMAN and KNP, and then words and dependency relations in the documents were indexed as index terms. For instance, the dependency relation (`university`, `time`) shown in Figure 1 is indexed as `university` → `time`.

6.2 Comparison with Alternative Searches

We first investigated the effectiveness of clear boundaries of term importance and differently handling of terms according to their importance. We compared the proposed method with the following alternative search methods (see Table 1):

AND: Conjunctive search only using words. We do nothing even if the number of retrieved documents is less than 1,000. Retrieved documents are ranked according to Okapi BM25 scores. This is the same equation when the parameter β is regarded as zero in $R(q, d)$. The Prox. column in Table 1 indicates whether a proximity operator is imposed. The symbol ◯ in the Word column means that words in a query are handled as necessary terms.

OR: Disjunctive search only using words. Retrieved documents are ranked according to Okapi BM25 scores. The symbol △ in the Word column means that words in a query are handled as optional terms.

[2]http://nlp.kuee.kyoto-u.ac.jp/nl-resource/juman.html

[3]http://nlp.kuee.kyoto-u.ac.jp/nl-resource/knp.html

OR (dpnd): Disjunctive search using both words and dependency relations. Retrieved documents are ranked according to scores of $R(q, d)$. We used the value of 0.2 as the parameter β.

AND$_{prox}$+OR(dpnd): In the same way as the proposed method, this search consists of conjunctive search and OR search. The conjunctive search uses only words with a proximity operator. Retrieved documents must contain words in a search query within 75 words (regardless of order). The parameter value was decided by the results of pilot studies. Retrieved documents are ranked according to Okapi BM25 scores. These scores are calculated by both words and dependency relations. On the other hand, the OR(dpnd) search described above is used as an OR search. Let $R_{prox}(d)$ denote the rank given by the conjunctive search, and $R_{OR}(d)$ denote the rank given by the OR(dpnd) search, and the final score for a document d is defined as:

$$S(d) = \frac{1}{R_{prox}(d)} + \frac{1}{R_{OR}(d)}.$$

The documents collected by the conjunctive and OR(dpnd) searches are sorted according to the above values, then the top 1,000 documents are regarded as the search result of this search.

In the above methods, the unnecessary expressions described in Section 5.3 are not used.

The proposed method exploits dependency relations in NEs and SCCNs as necessary terms, and the other dependency relations are handled as optional terms. Redundant verbs are handled as optional terms and the others are necessary terms. The proposed method imposes the same proximity operator as the AND$_{prox}$+OR (dpnd) search.

Table 2: Comparison with systems in NTCIR3

(a) For MRR and P@10.

System	MRR	P@10
GRACE	**0.502**	**0.330**
UAIFI5	0.383	0.289
NAICR	0.468	0.249
Ours	0.431	0.313

(b) For R-prec and MAP.

System	R-prec	MAP
GRACE	**0.230**	**0.208**
OKSAT	0.156	0.190
NAICR	0.115	0.180
Ours	0.208	0.156

Table 3: Comparison with systems in NTCIR4.

System	MRR	P@10	R-prec	MAP
GRACE	**0.645**	**0.501**	**0.278**	**0.216**
DBLAB	0.613	0.435	0.254	0.212
SSTUT	0.562	0.370	0.189	0.132
Ours	0.600	0.383	0.229	0.169

Table 1 shows performance of the proposed method and alternative methods. We can see that the proposed method outperforms not only AND and OR searches which are simple and conventional methodology but also the AND$_{prox}$+OR(dpnd) search. A small number of documents is returned by the AND search since the documents must include all necessary terms in a query. Because of this, the AND search indicates the worst performance in almost all evaluation measures. Though the proposed method also retrieves documents that must include all necessary terms in a query, the method achieves high performance because of its combination with the OR(dpnd) search.

From the difference between the OR and OR (dpnd) searches, we can see that dependency relations improve the performance of the OR search.

6.3 Comparison with Systems in NTCIR

Next we compared the search performance of the proposed method and that of systems participated in NTCIR 3 and NTCIR 4. In NTCIR 3, the measures MRR and P@10 and measures MAP and R-prec were used in different tasks. Therefore we selected the top three systems for each evaluation measure. In NTCIR 4, we selected the top three systems according to MAP.

Tables 2 and 3 show the comparison results for NTCIR3 and 4. Note that although GRACE, DBLAB and SSTUT in the tables used pseudo-relevance feedback, the proposed method did not. Tables 2 (a) and (b) show that the proposed method achieves the close performance of GRACE, the best system in NTCIR 3, in terms of P@10 and R-prec.

On the other hand, Table 3 shows that the proposed method outperforms SSTUT, the third system in NTCIR 4. The difference between the performance of the proposed method and that of GRACE and DBLAB is derived from pseudo-relevance feedback. We expect that the proposed method achieves similar performance to GRACE and DBLAB if it utilizes pseudo-relevance feedback. Usage of of pseudo-relevance feedback is our future work.

6.4 Effectiveness of Dependency Relation in Document Scoring

We investigated the optimized value of the parameter β used to regulate the extent to which dependency relations are used in the document scoring. For estimating the value, we investigated the performance when changing the value of β from 0.0 to 0.9 at increments of 0.1.

The performance is shown in Table 4. The "0.0" row means that document scoring is performed without using dependency relations. We can see that dependency relations contribute to improved search performance. In particular, maximum values of most evaluation measure are indicated when the value of β is 0.2.

6.5 Influence of Redundant Verb

Next we classified all verbs in queries into redundant verbs and other verbs, then examined the search performance when changing their term importance. The result is shown in Table 5. The proposed method deals with redundant verbs as optional terms, and the others as necessary terms (Normal: ◯, Redundant: △ in the table). The proposed method outperforms methods that handle all verbs as necessary terms (Normal: ◯, Redundant: ◯).

An example of a query that includes a redundant verb and contributes to improved search performance is "*I want to find shops that make bread with natural yeast.*" In this query, the proposed method found a document that describes "*... is a well-known bakery. Bread with natural yeast is a popular item.*" Though this document did not include the verb "*make*," we were able to find it because the redundant verb detection procedure de-

Table 4: Changes in search performance, when varying the parameter β in document scoring.

β	MRR	P@10	DCG_{10}	DCG_{100}	DCG_{1000}	R-prec	MAP	QM
0.0	**0.548**	0.341	3.528	9.108	17.209	0.208	0.151	0.170
0.1	0.529	0.350	3.619	9.265	**17.454**	0.214	0.155	0.173
0.2	0.537	**0.357**	**3.713**	**9.280**	16.866	**0.221**	**0.164**	**0.183**
0.3	0.497	0.338	3.446	9.174	17.418	0.209	0.152	0.171
0.4	0.507	0.339	3.335	8.791	17.038	0.199	0.145	0.164
0.5	0.486	0.320	3.150	8.307	16.482	0.191	0.136	0.154
0.6	0.467	0.303	2.988	7.793	15.645	0.174	0.126	0.143
0.7	0.458	0.292	2.873	7.384	14.777	0.166	0.118	0.133
0.8	0.456	0.278	2.790	7.059	14.216	0.157	0.110	0.124
0.9	0.447	0.263	2.646	6.681	13.569	0.148	0.104	0.117

scribed in Section 5.2 judged that the meaning of "*make*" is inferable from "*bread*."

The highest performance, however, was achieved when regarding all verbs as *optional* terms (Normal: △, Redundant: △). In this setting, the example of a query that contributes to improved search performance is "*I want to find out how the heliocentric theory of Copernicus was accepted by Christian society.*" The redundant verb detection procedure judged that the meaning of "*accept*" is not inferable from "*society*." Consequently, the verb "*accept*" is handled as a necessary term. Though this judgement is correct, the handling of verbs as necessary terms means that the possibility of the same event being expressed by different expressions such as synonyms is discarded. In general, a verb has multiple synonyms, and multiple expressions can be considered for describing the identical event. The handling of verbs as necessary terms can thereby be a cause of decreased search performance. We cope with the side effect of verbs by expanding synonym databases.

6.6 Influence of Dependency Relation Usage

Finally we investigated search performance when changing importance of dependency relations.

Table 6 shows that scores of all evaluation measures are close to each other when we simply used all dependency relations as *necessary*, *optional* or *unnecessary* terms. On the other hand, the proposed method handles dependency relations in NEs and SCCNs as necessary terms, and handles the other dependency relations as optional terms. This setting achieves relatively higher performance than the other settings. This means that the different handling of dependency relations according to their categories improves search performance.

7 Conclusion

In this paper, we defined three term importance categories: *necessary*; *optional* and *unnecessary*, and proposed a method that classifies terms in an NLQ into a category. The term importance is detected by word co-occurrence frequencies estimated from large-scale Web documents and NE recognition. The proposed method also handles dependency relations in a query as terms for achieving high performance.

We evaluated the proposed method using the NTCIR-3 and NTCIR-4 test collections for Japanese information retrieval. The search performance resultantly improved by regarding terms (words and dependency relations) in the named entities and compound nouns as *necessary* terms. Moreover, the performance was partially improved by regarding redundant verbs as *optional*.

References

Allan, James, Jamie Callan, W. Bruce Croft, Lisa Ballesteros, John Broglio, Jinxi Xu, and Hongmin Shu. 1997. Inquery at trec-5. In *NIST*, pages 119–132.

Bendersky, Michael and W. Bruce Croft. 2008. Discovering key concepts in verbose queries query. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval 2008*, pages 491–498.

Callan, James P., W. Bruce Croft, and John Broglio. 1995. Trec and tipster experiments with inquery. *Inf. Process. Manage.*, 31(3):327–343.

Eguchi, Koji, Keizo Oyama, Emi Ishida, Noriko Kando, and Kazuko Kuriyama. 2003. The web retrieval task and its evaluation in the third ntcir workshop. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Table 5: Changes in search performance, when varying term importance of verbs.

Verbs		MRR	P@10	DCG_{10}	DCG_{100}	DCG_{1000}	R-prec	MAP	QM
Normal	Redundant								
○	○	0.525	0.352	3.640	9.110	16.734	0.217	0.161	0.180
○	△	**0.537**	0.357	3.713	9.280	16.866	**0.221**	0.164	0.183
○	×	0.534	0.354	3.664	9.273	16.832	**0.221**	0.164	0.183
△	△	**0.537**	**0.360**	**3.755**	**9.404**	**17.053**	**0.221**	**0.165**	**0.184**
△	×	0.534	0.357	3.709	9.399	17.019	**0.221**	**0.165**	**0.184**
×	×	0.533	0.356	3.703	9.401	17.018	**0.221**	**0.165**	**0.184**

Table 6: Changes in search performance, when varying the importance of dependency relations.

Dependency relations		MRR	P@10	DCG_{10}	DCG_{100}	DCG_{1000}	R-prec	MAP	QM
Outside of NEs & SCCNs	Inside of NEs & SCCNs								
○	○	0.513	0.338	3.474	8.987	16.650	0.211	0.155	0.174
△	○	0.537	0.357	3.713	**9.280**	**16.866**	**0.221**	**0.164**	**0.183**
×	○	**0.561**	0.349	3.642	9.072	16.547	0.213	0.159	0.177
△	△	0.552	0.347	3.647	9.073	16.565	0.215	0.159	0.177
×	△	0.539	**0.359**	**3.725**	9.223	16.827	**0.221**	**0.164**	0.182
×	×	**0.561**	0.344	3.655	9.059	16.545	0.214	0.159	0.177

Eguchi, Koji, Keizo Oyama, Akiko Aizawa, and Haruko Ishikawa. 2004. Overview of web task at the fourth ntcir workshop. In *Proceedings of the Fourth NTCIR Workshop on Research in Information Access Technologies Information Retrieval, Question Answering and Summarization*.

Jarvelin, Kalervo and Jaana Kekalainen. 2002. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems*, 20:422–446.

Jones, Karen Sparck. 1999. What is the role of nlp in text retrieval? In Strzalkowski, T., editor, *Natural language information retrieval*, pages 1–24. Kluwer Academic Publishers.

Liu, Shuang, Fang Liu, Clement Yu, and Weiyi Meng. 2004. An effective approach to document retrieval via utilizing wordnet and recognizing phrases. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 266–272.

Miyao, Yusuke, Tomoko Ohta, Katsuya Masuda, Yoshimasa Tsuruoka, Kazuhiro Yoshida, Takashi Ninomiya, and Jun'ichi Tsujii. 2006. Semantic retrieval for the accurate identification of relational concepts in massive textbases. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, pages 1017–1024.

Ponte, Jay M. and W. Bruce Croft. 1998. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281.

Robertson, Stephen E., Steve Walker, Micheline Hancock-Beaulieu, Aarron Gull, and Marianna Lau. 1992. Okapi at TREC. In *Text REtrieval Conference*, pages 21–30.

Sakai, Tetsuya. 2004. New performance metrics based on multigrade relevance: Their application to question answering. In *Proceedings of the Fourth NTCIR Workshop Meeting*.

Sasano, Ryohei and Sadao Kurohashi. 2008. Japanese named entity recognition using structural natural language processing. In *Proceedings of Third International Joint Conference on Natural Language Processing*, pages 607–612.

Shibata, Tomohide, Michitaka Odani, Jun Harashima, Takashi Oonishi, and Sadao Kurohashi. 2008. SYNGRAPH: A flexible matching method based on synonymous expression extraction from an ordinary dictionary and a web corpus. In *Proc. of IJCNLP2008*, pages 787–792.

Shinzato, Keiji, Daisuke Kawahara, Chikara Hashimoto, and Sadao Kurohashi. 2008a. A large-scale web data collection as a natural language processing infrastructure. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC08)*.

Shinzato, Keiji, Tomohide Shibata, Daisuke Kawahara, Chikara Hashimoto, and Sadao Kurohashi. 2008b. TSUBAKI: An open search engine infrastructure for developing new information access methodology. In *Proc. of IJCNLP2008*, pages 189–196.

Wei, Zhang, Liu Shuang, Yu Clement, Sun Chaojing, Liu Fang, and Meng Weiyi. 2007. Recognition and classification of noun phrases in queries for effective retrieval. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 711–720.

Summarizing Search Results using PLSI

Jun Harashima* **and Sadao Kurohashi**
Graduate School of Informatics
Kyoto University
Yoshida-honmachi, Sakyo-ku,
Kyoto, 606-8501, Japan
{harashima,kuro}@nlp.kuee.kyoto-u.ac.jp

Abstract

In this paper, we investigate generating a set of query-focused summaries from search results. Since there may be many topics related to a given query in the search results, in order to summarize these results, they should first be classified into topics, and then each topic should be summarized individually. In this summarization process, two types of redundancies need to be reduced. First, each topic summary should not contain any redundancy (we refer to this problem as redundancy within a summary). Second, a topic summary should not be similar to any other topic summary (we refer to this problem as redundancy between summaries). In this paper, we focus on the document clustering process and the reduction of redundancy between summaries in the summarization process. We also propose a method using PLSI to summarize search results. Evaluation results confirm that our method performs well in classifying search results and reducing the redundancy between summaries.

1 Introduction

Currently, the World Wide Web contains vast amounts of information. To make efficient use of this information, search engines are indispensable. However, search engines generally return only a long list containing the title and a snippet of each of the retrieved documents. While such lists are effective for navigational queries, they are not helpful to users with informational queries. Some systems (e.g., Clusty[1]) present keywords related to a given query together with the search results. It is, however, difficult for users to understand the relation between the keywords and the query, as the keywords are merely words or phrases out of context. To solve this problem, we address the task of generating a set of query-focused summaries from search results to present information about a given query using natural sentences.

Since there are generally many topics related to a query in the search results, the task of summarizing these results is one of, so to speak, multi-topic multi-document summarization. Studies on multi-document summarization typically address summarizing documents related to a single topic (e.g., TAC[2]). However we need to address summarizing documents related to multiple topics when considering the summarization of search results.

To summarize documents containing multiple topics, we first need to classify them into topics. For example, if a set of documents related to *swine flu* contains topics such as the outbreaks of *swine flu*, the measures to treat *swine flu*, and so on, the documents should be divided into these topics and summarized individually. Note that a method for soft clustering should be employed in this process, as one document may belong to several topics.

*Research Fellow of the Japan Society for the Promotion of Science (JSPS)

[1] http://clusty.com/

[2] http://www.nist.gov/tac/

Proceedings of the Second Workshop on NLP Challenges in the Information Explosion Era (NLPIX 2010), pages 12–20,
Beijing, August 2010

In the summarization process, two types of redundancies need to be addressed. First, each topic summary should not contain any redundancy. We refer to this problem as redundancy within a summary. This problem is well known in the field of multi-document summarization (Mani, 2001) and several methods have been proposed to solve it, such as Maximum Marginal Relevance (MMR) (Goldstein et al., 2000) (Mori et al., 2004), using Integer Linear Programming (ILP) (Filatova and Hatzivassiloglou, 2004) (McDonald, 2007) (Takamura and Okumura, 2009), and so on.

Second, no topic summary should be similar to any of the other topic summaries. We refer to this problem as redundancy between summaries. For example, to summarize the abovementioned documents related to *swine flu*, the summary for outbreaks should contain specific information about outbreaks, whereas the summary for measures should contain specific information about measures. This problem is characteristic of multi-topic multi-document summarization. Some methods have been proposed to generate topic summaries from documents (Radev and Fan, 2000) (Haghighi and Vanderwende, 2009), but to the best of our knowledge, the redundancy between summaries has not yet been addressed in any study.

In this paper, we focus on the document clustering process and the reduction of redundancy between summaries in the summarization process. Furthermore, we propose a method using PLSI (Hofmann, 1999) to summarize search results. In the proposed method, we employ PLSI to estimate the membership degree of each document to each topic, and then classify the search results into topics using this information. In the same way, we employ PLSI to estimate the membership degree of each keyword to each topic, and then extract the important sentences specific to each topic using this information in order to reduce the redundancy between summaries. The evaluation results show that our method performs well in classifying search results and successfully reduces the redundancy between summaries.

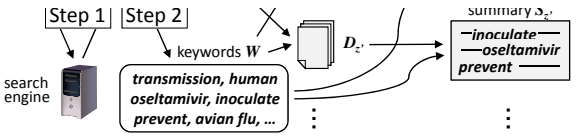

Figure 1: Overview of the proposed method.

2 Proposed Method

2.1 Overview

Figure 1 gives an overview of the proposed method, which comprises the following four steps.

Step 1. Acquisition of Search Results Using a search engine, obtain the search results for a given query.

Step 2. Keyword Extraction Extract the keywords related to the query from the search results using the method proposed by Shibata et al. (2009).

Step 3. Document Clustering Estimate the membership degree of each document to each topic using PLSI, and classify the search results into topics.

Step 4. Summarization For each topic, generate a summary by extracting the important sentences specific to each topic from each document cluster.

In the following subsections, we describe each step in detail.

2.2 Step 1. Acquisition of Search Results

First, we obtain the search results for a given query using a search engine. To be more precise, we obtain the top N' documents of the search engine results. Next, we remove those documents that should not be included in the summarization, such as link collections, using a simple filtering method. For example, we regard any document that has too many links as a link collection, and remove it.

In this paper, we write D to denote the search results after the filtering, and let $N = |D|$.

2.3 Step 2. Keyword Extraction

We extract the keywords related to a query from D using the method proposed by Shibata et al. (2009), which comprises the following four steps.

Step 2-1. Relevant Sentence Extraction For each document in D, extract the sentences containing the query and the sentences around the query as relevant sentences.

Step 2-2. Keyword Candidate Extraction For each relevant sentence, extract compound nouns and parenthetic strings as keyword candidates.

Step 2-3. Synonymous Candidate Unification Find the paraphrase pairs and the orthographic variant pairs in the keyword candidates, and merge them.

Step 2-4. Keyword Selection Score each keyword candidate, rank them, and select the best M as the keywords related to the query.

In this paper, we write W to denote the extracted keywords.

2.4 Step 3. Document Clustering

We classify D into topics using PLSI. In PLSI, a document d and a word w are assumed to be conditionally independent given a topic z, and the joint probability $p(d, w)$ is calculated as follows.

$$p(d, w) = \sum_z p(z)\ p(d|z)\ p(w|z) \tag{1}$$

$p(z)$, $p(d|z)$, and $p(w|z)$ are estimated by maximizing the log-likelihood function L, which is calculated as

$$L = \sum_d \sum_w freq(d, w) \log p(d, w), \tag{2}$$

where $freq(d, w)$ represents the frequency of word w in document d. L is maximized using the EM algorithm, in which the E-step and M-step are given below.

E-step

$$p(z|d, w) = \frac{p(z)\ p(d|z)\ p(w|z)}{\sum_{z'} p(z')\ p(d|z')\ p(w|z')} \tag{3}$$

M-step

$$p(z) = \frac{\sum_d \sum_w freq(d, w)\ p(z|d, w)}{\sum_d \sum_w freq(d, w)} \tag{4}$$

$$p(d|z) = \frac{\sum_w freq(d, w)\ p(z|d, w)}{\sum_{d'} \sum_w freq(d', w)\ p(z|d', w)} \tag{5}$$

$$p(w|z) = \frac{\sum_d freq(d, w)\ p(z|d, w)}{\sum_d \sum_{w'} freq(d, w')\ p(z|d, w')} \tag{6}$$

The EM algorithm iterates through these steps until convergence.

First, we give PLSI the number of topics K, the search results D, and the keywords W as input, and estimate $p(z)$, $p(d|z)$, and $p(w|z)$, where z is a topic related to the query, d is a document in D, and w is a keyword in W. There is, however, no way of knowing the value of K; that is, we do not know in advance how many topics related to the query there are in the search results. Hence, we perform PLSI for several values of K, and select the K that has the minimum Akaike Information Criterion (AIC) (Akaike, 1974), calculated as follows.

$$AIC = -2L + 2K(N + M) \tag{7}$$

Furthermore, we select $p(z)$, $p(d|z)$, and $p(w|z)$ estimated using the selected K as the result of PLSI.

Next, we calculate the membership degree of each document to each topic. The membership degree of document d to topic z, denoted $p(z|d)$, is calculated as

$$p(z|d) = \frac{p(d|z)\ p(z)}{\sum_{z'} p(d|z')}. \tag{8}$$

Finally, for each topic, we collect those documents whose membership degree to the topic is larger than the threshold α. If there is a document whose membership degree to multiple topics is larger than the threshold, we classify the document into each topic.

In this paper, D_z denotes the documents classified into topic z.

2.5 Step 4. Summarization

For each topic, we extract the important sentences specific to that topic from each document

Figure 2: Algorithm for summarization.

Input: A set of K document clusters $\{D_z\}(z \in Z)$
Output: A set of K summaries $\{S_z\}(z \in Z)$
Procedure:
1: **for** all $z \in Z$
2: **while** $|S_z| < num(z)$
3: **for** all $s \in D_z$
4: calculate $s\_score(z, s, S_z)$
5: $s_{max} = argmax_{s \in D_z \setminus S_z}\ s\_score(z, s, S_z)$
6: $S_z = S_z \cup \{s_{max}\}$
7: **return** S_z

cluster. Figure 2 gives the algorithm for summarization. When we generate the summary S_z for topic z, we calculate the importance of sentence s to topic z, denoted as $s\_score(z, s, S_z)$, for each sentence in D_z (lines 3-4). Then we extract the sentence s_{max} with the maximum importance as an important sentence, and include s_{max} in S_z (lines 5-6). When we extract the next important sentence, we recalculate the importance $s\_score(z, s, S_z)$ for each sentence in D_z except the sentence in S_z (lines 3-4). Then we extract the sentence s_{max} with the maximum importance as an important sentence, and add s_{max} to S_z (lines 5-6). We continue this process until the number of important sentences composing the summary, denoted $|S_z|$, reaches the number of important sentences extracted for topic z, denoted $num(z)$ (line 2).

$s\_score(z, s, S_z)$ is calculated as follows:

$$s\_score(z, s, S_z) = \sum_{w \in W_s} \Big(w\_score(z, w) \times c\_score(w, S_z, s) \Big) \quad (9)$$

where W_s represents the keywords in sentence s.

$w\_score(z, w)$ is a function to reduce the redundancy between summaries, and represents the importance of keyword w to topic z. We can use the probability of w given z, denoted $p(w|z)$, as the $w\_score(z, w)$. This approach fails, however, because if there are keywords with a high probability in both topic z and another topic z', the sentences containing such keywords are extracted as the important sentences in both topics, and it follows that the generated summaries will contain redundancy. To solve this problem, we use the membership degree of keyword w to topic z, denoted $p(z|w)$, as $w\_score(z, w)$. We use $p(z)$ and $p(w|z)$ estimated using PLSI in Section 2.4 to calculate $p(z|w)$.

$$p(z|w) = \frac{p(w|z)\ p(z)}{\sum_{z'} p(w|z')} \quad (10)$$

Keywords with high probability in several topics should have a low membership degree to each topic. Thus, using $p(z|w)$ as the $w\_score(z, w)$ prevents extracting sentences containing such keywords as important sentences, and it follows that the similarity between the summaries is reduced. Furthermore, the keywords which are specific to a topic are supposed to have a high membership degree to that topic. Thus, using $p(z|w)$ as $w\_score(z, w)$ makes it easier to extract sentences containing such keywords as important sentences, and with the result that each summary is specific to the particular topic.

Table 1: Values of $c\_score(w, S_z, s)$.

	w is contained in S_z	w is not contained in S_z
w is the subject of s	2	-2
otherwise	0	1

$c\_score(w, S_z, s)$ is a function to reduce the redundancy within a summary, and represents the importance of a keyword w in a sentence s under the condition that there is a set of extracted important sentences S_z. The value of $c\_score(w, S_z, s)$ is determined mainly by whether or not w is contained in S_z. Table 1 gives the values of $c\_score(w, S_z, s)$. For example, if w is contained in S_z, we set $c\_score(w, S_z, s) = 0$, else we set $c\_score(w, S_z, s) = 1$. In this way, we can extract the sentences containing the keywords that are not contained in S_z as important sentences, and reduce the redundancy within the summary. Note that we make some exceptions to generate a coherent summary. For example, even if w is contained in S_z, we set $c\_score(w, S_z, s) = 2$ as long as w is the subject of s. In the same way, even if w is not contained in S_z, we set $c\_score(w, S_z, s) = -2$ as long as w is the subject of s. These values for $c\_score(w, S_z, s)$ are empirically determined.

Finally, using $p(z)$ we determine the number of important sentences extracted for topic z, denoted as $num(z)$.

$$num(z) = \begin{cases} \lfloor\ I \times p(z)\ \rfloor & (\ p(z) \geq \beta\ ) \\ I_{min} & (\ p(z) < \beta\ ) \end{cases} \quad (11)$$

where I represents the parameter that controls the total number of important sentences extracted for each topic. The higher the probability a topic has, the more important sentences we extract. Note that no matter how low $p(z)$ is, we extract at least I_{min} important sentences.

3 Experiments

3.1 Overview

To evaluate the proposed method, we recruited 48 subjects, mainly IT workers, and asked them to fill in a questionnaire. We prepared a system implemented according to our method, and asked the subjects to use our system to evaluate the following four aspects of our method.

- Validity of the number of topics
- Precision of document clustering
- Degree of reduction in redundancy between summaries
- Effectiveness of the method for presenting information through summaries

We allowed the subjects to create arbitrary queries for our system.

3.2 System

Figure 3 shows the system results for the query *swine flu*. Our system presents a separate summary for each topic related to a given query. In Fig.3, the colored words in the summaries are keywords specific to each topic. If a user clicks on a keyword, the system presents a list of documents containing that keyword at the bottom of the browser.

The configuration of our system was as follows. In the acquisition process, the system obtained the search results for a given query using the search engine TSUBAKI (Shinzato et al., 2008b). Setting $N' = 1,000$, we obtained the top $1,000$ documents in the search results for the query. In the keyword extraction process, we set $M = 100$, and extracted 100 keywords related to the query from the search results. In the document clustering process, we performed PLSI for $K = 3, 4, 5$, and selected the K with the minimum AIC. We set the initial value of $p(z) = 1/K$, and the initial values of $p(d|z)$ and $p(w|z)$ to random values. The EM algorithm continued until the increase in L reached just below 1 to achieve convergence. We set $\alpha = 1/K$. In the summarization process, we set $I = 10$, since the number of important sentences able to be presented in a browser is about 10. We set $I_{min} = 2$ and $\beta = 0.2$, and extracted at least two important sentences, even if $p(z)$ was very low.

3.3 Validity of the Number of Topics

First, we investigated how well the proposed method determined the number of topics. In our method, the number is determined using AIC. Ideally, we should have manually counted the topics in the search results, and compared this with the number determined using AIC. It was, however, difficult to count the topics, because the search results contained $1,000$ documents. Furthermore, it was impossible to count the number of topics for each query given by each subject. Thus, in this investigation, we simply asked the subjects whether they felt the number of topic summaries presented to them was appropriate or not, and investigated our method in terms of usability.

Table 2 gives the results. According to Table 2, 60.4% of the subjects agreed that the number of topic summaries presented by our system was acceptable. The average of the number of topics determined by our method was 3.18 per 1 query. On the other hand, 33.3% of the subjects said the number of topic summaries was too low or somewhat too low. According to these results, it seems that users are satisfied with the system presenting about 3 or 4 topic summaries, and our method determined the desirable number of topics in terms of usability.

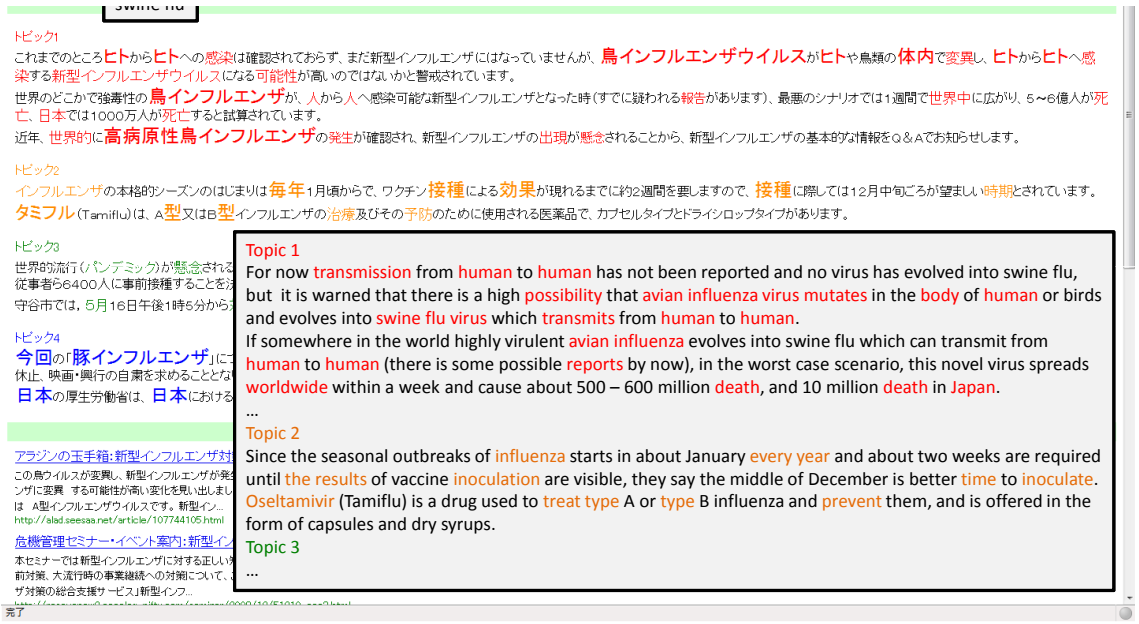

Figure 3: System results for the query *swine flu*.

Table 2: Validity of the number of topics.

options	# subjects	(%)
(a) definitely too many	0	(0.0)
(b) somewhat too many	3	(6.3)
(c) acceptable	29	(60.4)
(d) somewhat too few	11	(22.9)
(e) definitely too few	5	(10.4)

3.4 Precision of Document Clustering

Second, we investigated how precisely the proposed method classified the search results into topics. To be more precise, we evaluated the reliability of the membership degree $p(z|d)$ used in the document clustering process. It is generally difficult to evaluate clustering methods. In our case, we did not have any correct data and could not even create these since, as mentioned previously, the number of topics is not known. Furthermore, it is not possible to classify by hand search results containing $1,000$ documents. Consequently, we did not evaluate our method directly by comparing correct data with the clustering result from our method, but instead evaluated it indirectly by investigating the reliability of the membership degree $p(z|d)$ used in the document clustering process.

The evaluation process was as follows. First, we presented the subjects with a document d, which was estimated by our system to have a high membership degree to a topic z. Strictly speaking, we selected as d, a document with a membership degree of about 0.9. Next, we presented two documents to the subjects. One was a document d' whose membership degree to z was also about 0.9, and another was a document d'' whose membership degree to z was about 0.1. Finally, we asked them which document was more similar to d[3].

Table 3 gives the results. According to this table, 60.5% of the subjects said d' was more similar or somewhat more similar. On the other hand, only 12.6% of the subjects said d'' was more similar or somewhat more similar. We see from these results that the ability to recognize topics in our system is in agreement to some extent with

[3]Naturally, we did not tell them that d' had a similar membership degree to d, whereas d'' did not.

Table 3: Precision of the estimation $p(z|d)$.

options	# subjects	(%)
(a) d' is definitely more similar	14	(29.2)
(b) d' is somewhat more similar	15	(31.3)
(c) undecided	13	(27.1)
(d) d'' is somewhat more similar	3	(6.3)
(e) d'' is definitely more similar	3	(6.3)

the subjects' ability for recognizing topics; that is, our method was able to estimate a reliable membership degree $p(z|d)$. Thus, it seems that our method using $p(z|d)$ is able to classify search results into topics to some extent.

3.5 Degree of Reduction in Redundancy between Summaries

Third, we investigated how well the proposed method reduced the redundancy between summaries. To be more precise, we used three measures as $w\_score(z, w)$ to generate summaries and investigated which measure generated the least redundant summaries. Generally, methods for reducing redundancy are evaluated using ROUGE (Lin, 2004), BE (Hovy et al., 2005), or Pyramid (Nenkova and Passonneau, 2004). However, the use of these methods require that ideal summaries are created by humans, and this was not possible for the same reason as mentioned previously. Thus, we did not perform a direct evaluation using the methods such as ROUGE, but instead evaluated how well our method performed in reducing redundancy between summaries using the membership degree $p(z|w)$ as $w\_score(z, w)$.

The evaluation process was as follows. We used three measures as $w\_score(z, w)$, and generated three sets of summaries.

Summaries A This set of summaries was generated using $dfidf(w)$ as $w\_score(z, w)$, with $dfidf(w)$ calculated as $ldf(w) \times log(100\,million/gdf(w))$, $ldf(w)$ representing the document frequency of keyword w in the search results, and $gdf(w)$ representing the document frequency of keyword w in the TSUBAKI document collection (Shinzato et al., 2008a) comprising about 100 million documents.

Table 4: Comparison of $dfidf(w)$, $p(w|z)$ and $p(z|w)$.

options	# subjects	(%)
(a) B is definitely less redundant	5	(10.4)
(b) B is somewhat less redundant	16	(33.3)
(c) undecided	15	(31.3)
(d) A is somewhat less redundant	6	(12.5)
(e) A is definitely less redundant	6	(12.5)
options	# subjects	(%)
(a) C is definitely less redundant	16	(33.3)
(b) C is somewhat less redundant	14	(29.2)
(c) undecided	6	(12.5)
(d) A is somewhat less redundant	8	(16.7)
(e) A is definitely less redundant	4	(8.3)
options	# subjects	(%)
(a) C is definitely less redundant	15	(31.3)
(b) C is somewhat less redundant	8	(16.7)
(c) undecided	10	(20.8)
(d) B is somewhat less redundant	6	(12.5)
(e) B is definitely less redundant	9	(18.8)

Summaries B This set of summaries was generated using $p(w|z)$ as $w\_score(z, w)$.

Summaries C This set of summaries was generated using $p(z|w)$ as $w\_score(z, w)$.

We then presented the subjects with three pairs of summaries, namely a pair from A and B, a pair from A and C, and a pair from B and C, and asked them which summaries in each pair was less redundant[4].

The results are given in Tables 4. Firstly, according to the comparison of A and B and the comparison of A and C, A was more redundant than B and C. The value of $dfidf(w)$ to keyword w was the same for all topics. Thus, using $dfidf(w)$ as $w\_score(z, w)$ made summaries redundant, as each summary tended to contain the same keywords with high $dfidf(w)$. On the other hand, as the value of $p(w|z)$ and $p(z|w)$ were dependent on the topic, the summaries generated using these measures were less redundant.

Second, the comparison of B and C shows that 48.0% of the subjects considered C to be less redundant or somewhat less redundant. $p(w|z)$ was a better measure than $dfidf(w)$, but even using $p(w|z)$ generated redundancy between sum-

[4]Naturally, we did not tell them which summaries were generated using which measures

Table 5: Comparison of summaries and keywords.

options	# subjects	(%)
(a) X is definitely more helpful	25	(52.1)
(b) X is somewhat more helpful	10	(20.8)
(c) undecided	3	(6.3)
(d) Y is somewhat more helpful	8	(16.7)
(e) Y is definitely more helpful	2	(4.2)

maries. Because common keywords to a query have high $p(w|z)$ for several topics, sentences containing these keywords were extracted as the important sentences for those topics, and thus the summaries were similar to one another. On the other hand, the keywords' value for $p(z|w)$ was low, allowing us to extract the important sentences specific to each topic using $p(z|w)$ as $w\_score(z, w)$, thereby reducing the redundancy between summaries.

3.6 Effectiveness of the Method for Presenting Information Using Summaries

We also investigated the effectiveness of the method for presenting information through summaries. We asked the subjects to compare two different ways of presenting information and to judge which way was more effective in terms of usefulness for collecting information about a query. One of the methods presented the search results with topic summaries generated by our system (method X), and while the another method presented the search results with the keywords included in each topic summary (method Y).

Table 5 gives the results. 72.9% of the subjects considered the method using summaries to be more effective or somewhat more effective. From these results, it appears that the method of presenting information through summaries is effective in terms of usefulness for collecting information about a query.

4 Conclusion

In this paper, we focused on the task of generating a set of query-focused summaries from search results. To summarize the search results for a given query, a process of classifying them into topics related to the query was needed. In the proposed method, we employed PLSI to estimate the membership degree of each document to each topic, and then classified search results into topics using this metric. The evaluation results showed that our method estimated reliable degrees of membership. Thus, it seems that our method is able to some extent to classify search results into topics.

In the summarization process, redundancy within a summary and redundancy between summaries needs to be reduced. In this paper, we focused on the reduction of the latter redundancy. Our method made use of PLSI to estimate the membership degree of each keyword to each topic, and then extracted the important sentences specific to each topic using this metric. The evaluation results showed that our method was able to reduce the redundancy between summaries using the membership degree.

In future, we will investigate the use of more sophisticated topic models. Although our method detected the topics related to a query using a simple topic model (i.e., PLSI), we believe that more sophisticated topic models such as LDA (Blei et al., 2003) allow us to improve our method.

References

Akaike, Hirotugu. 1974. A new look at the statistical model identification. *IEEE Transactions on Automation Control*, 19(6):716–723.

Blei, David M., Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022.

Filatova, Elena and Vasileios Hatzivassiloglou. 2004. A formal model for information selection in multi-sentence text extraction. In *Proceedings of COLING 2004*, pages 397–403.

Goldstein, Jade, Vibhu Mittal, Jaime Carbonell, and Mark Kantrowitz. 2000. Multi-document summarization by sentence extraction. In *Proceedings of ANLP/NAACL 2000 Workshop on Automatic Summarization*, pages 40–48.

Haghighi, Aria and Lucy Vanderwende. 2009. Exploring content models for multi-document summarization. In *Proceedings of HLT-NAACL 2009*, pages 362–370.

Hofmann, Thomas. 1999. Probabilistic latent semantic indexing. In *Proceedings of SIGIR 1999*, pages 50–57.

Hovy, Eduard, Chin-Yew Lin, and Liang Zhou. 2005. Evaluating duc 2005 using basic elements. In *Proceedings of DUC 2005*.

Lin, Chin-Yew. 2004. Rouge: A package for automatic evaluation of summaries. In *Proceedings of ACL 2004 Workshop on Text Summarization Branches Out*, pages 74–81.

Mani, Inderjeet. 2001. *Automatic Summarization*. John Benjamins Publishing Company.

McDonald, Ryan. 2007. A study of global inference algorithms in multi-document summarization. In *Proceedings of ECIR 2007*, pages 557–564.

Mori, Tatsunori, Masanori Nozawa, and Yoshiaki Asada. 2004. Multi-answer-focused multi-document summarization using a question-answering engine. In *Proceedings of COLING 2004*, pages 439–445.

Nenkova, Ani and Rebecca Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *Proceedings of NAACL-HLT 2004*.

Radev, Dragomir R. and Weiguo Fan. 2000. Automatic summarization of search engine hit lists. In *Proceedings of ACL 2000 Workshop on Recent advances in NLP and IR*, pages 1361–1374.

Shibata, Tomohide, Yasuo Bamba, Keiji Shinzato, and Sadao Kurohashi. 2009. Web information organization using keyword distillation based clustering. In *Proceedings of WI 2009*, pages 325–330.

Shinzato, Keiji, Daisuke Kawahara, Chikara Hashimoto, and Sadao Kurohashi. 2008a. A large-scale web data collection as a natural language processing infrastructure. In *Proceedings of LREC 2008*, pages 2236–2241.

Shinzato, Keiji, Tomohide Shibata, Daisuke Kawahara, Chikara Hashimoto, and Sadao Kurohashi. 2008b. TSUBAKI: An Open Search Engine Infrastructure for Developing New Information Access Methodology. In *Proceedings of IJCNLP 2008*, pages 189–196.

Takamura, Hiroya and Manabu Okumura. 2009. Text summarization model based on maximum coverage problem and its variant. In *Proceedings of EACL 2009*, pages 781–789.

Automatic Classification of Semantic Relations between Facts and Opinions

Koji Murakami† **Eric Nichols**‡ **Junta Mizuno**†‡ **Yotaro Watanabe**‡
Hayato Goto† **Megumi Ohki**† **Suguru Matsuyoshi**† **Kentaro Inui**‡ **Yuji Matsumoto**†
†Nara Institute of Science and Technology
‡Tohoku University
{kmurakami,matuyosi,hayato-g,megumi-o,matsu}@is.naist.jp
{eric-n,junta-m,inui}@ecei.tohoku.ac.jp

Abstract

Classifying and identifying semantic relations between facts and opinions on the Web is of utmost importance for organizing information on the Web, however, this requires consideration of a broader set of semantic relations than are typically handled in Recognizing Textual Entailment (RTE), Cross-document Structure Theory (CST), and similar tasks. In this paper, we describe the construction and evaluation of a system that identifies and classifies semantic relations in Internet data. Our system targets a set of semantic relations that have been inspired by CST but that have been generalized and broadened to facilitate application to mixed fact and opinion data from the Internet. Our system identifies these semantic relations in Japanese Web texts using a combination of lexical, syntactic, and semantic information and evaluate our system against gold standard data that was manually constructed for this task. We will release all gold standard data used in training and evaluation of our system this summer.

1 Introduction

The task of organizing the information on the Internet to help users find facts and opinions on their topics of interest is increasingly important as more people turn to the Web as a source of important information. The vast amounts of research conducted in NLP on automatic summarization, opinion mining, and question answering are illustrative of the great interest in making relevant information easier to find. Providing Internet users with thorough information requires recognizing semantic relations between both facts and opinions, however the assumptions made by current approaches are often incompatible with this goal. For example, the existing semantic relations considered in Recognizing Textual Entailment (RTE) (Dagan et al., 2005) are often too narrow in scope to be directly applicable to text on the Internet, and theories like Cross-document Structure Theory (CST) (Radev, 2000) are only applicable to facts or second-hand reporting of opinions rather than relations between both.

As part of the STATEMENT MAP project we proposed the development of a system to support information credibility analysis on the Web (Murakami et al., 2009b) by automatically summarizing facts and opinions on topics of interest to users and showing them the evidence and conflicts for each viewpoint. To facilitate the detection of semantic relations in Internet data, we defined a sentence-like unit of information called the *statement* that encompasses both facts and opinions, started compiling a corpus of *statements* annotated with semantic relations (Murakami et al., 2009a), and begin constructing a system to automatically identify semantic relations between statements.

In this paper, we describe the construction and evaluation of a prototype semantic relation identification system. We build on the semantic relations proposed in RTE and CST and in our previous work, refining them into a streamlined set of semantic relations that apply across facts and opinions, but that are simple enough to make automatic recognition of semantic relations between statements in Internet text possible.Our semantic relations are [AGREEMENT], [CONFLICT], [CONFINEMENT], and [EVIDENCE]. [AGREEMENT] and [CONFLICT] are expansions of the [EQUIVALENCE] and [CONTRADICTION]

Proceedings of the Second Workshop on NLP Challenges in the Information Explosion Era (NLPIX 2010), pages 21–30,
Beijing, August 2010

relations used in RTE. [CONFINEMENT] and [EVIDENCE] are new relations between facts and opinions that are essential for understanding how statements on a topic are inter-related.

Our task differs from opinion mining and sentiment analysis which largely focus on identifying the polarity of an opinion for defined parameters rather than identify how facts and opinions relate to each other, and it differs from web document summarization tasks which focus on extracting information from web page structure and contextual information from hyperlinks rather than analyzing the semantics of the language on the webpage itself.

We present a system that automatically identifies semantic relations between *statements* in Japanese Internet texts. Our system uses *structural alignment* to identify *statement* pairs that are likely to be related, then classifies semantic relations using a combination of lexical, syntactic, and semantic information. We evaluate cross-statement semantic relation classification on sentence pairs that were taken from Japanese Internet texts on several topics and manually annotated with a semantic relation where one is present. In our evaluation, we look closely at the impact that each of the resources has on semantic relation classification quality.

The rest of this paper is organized as follows. In Section 2, we discuss related work in summarization, semantic relation classification, opinion mining, and sentiment analysis, showing how existing classification schemes are insufficient for our task. In Section 3, we introduce a set of cross-sentential semantic relations for use in the opinion classification needed to support information credibility analysis on the Web. In Section 4, we present our cross-sentential semantic relation recognition system, and discuss the algorithms and resources that are employed. In Section 5, we evaluate our system in a semantic relation classification task. In Section 6, we discuss our findings and conduct error analysis. Finally, we conclude the paper in Section 7.

2 Related Work

2.1 Recognizing Textual Entailment

Identifying logical relations between texts is the focus of Recognizing Textual Entailment, the task of deciding whether the meaning of one text is entailed from another text. A major task in the RTE Challenge (Recognizing Textual Entailment Challenge) is classifying the semantic relation between a Text (T) and a Hypothesis (H) into [ENTAILMENT], [CONTRADICTION], or [UNKNOWN]. Over the last several years, several corpora annotated with thousands of (T,H) pairs have been constructed for this task. In these corpora, each pair was tagged indicating its related task (e.g. Information Extraction, Question Answering, Information Retrieval or Summarization).

The RTE Challenge has successfully employed a variety of techniques in order to recognize instances of textual entailment, including methods based on: measuring the degree of lexical overlap between bag of words (Glickman et al., 2005; Jijkoun and de Rijke, 2005), the alignment of graphs created from syntactic or semantic dependencies (Marsi and Krahmer, 2005; MacCartney et al., 2006), statistical classifiers which leverage a wide range of features (Hickl et al., 2005), or reference rule generation (Szpektor et al., 2007). These approaches have shown great promise in RTE for entailment pairs in the corpus, but more robust models of recognizing logical relations are still desirable.

The definition of contradiction in RTE is that T contradicts H if it is very unlikely that both T and H can be true at the same time. However, in real documents on the Web, there are many pairs of examples which are contradictory in part, or where one statement confines the applicability of another, as shown in the examples in Table 1.

2.2 Cross-document Structure Theory

Cross-document Structure Theory (CST), developed by Radev (2000), is another task of recognizing semantic relations between sentences. CST is an expanded rhetorical structure analysis based on Rhetorical Structure Theory (RST: (William and Thompson, 1988)), and attempts to describe the semantic relations that exist between two or more sentences from different source documents that are related to the same topic, as well as those that come from a single source document. A corpus of cross-document sentences annotated with CST relations has also been constructed (The CSTBank Corpus: (Radev et al., 2003)). CSTBank is organized into clusters of topically-related articles. There are 18 kinds of semantic relations in this corpus, not limited to [EQUIVALENCE] or [CONTRADICTION], but also including [JUDGEMENT], [ELABORATION], and [RE-

Query	Matching sentences	Output
キシリトールは虫歯予防に効果がある	キシリトールの含まれている量が多いほどむし歯予防の効果は高いようです	同意
	The cavity-prevention effects are greater the more Xylitol is included.	[AGREEMENT].
	キシリトールがお口の健康維持や虫歯予防にも効果を発揮します	同意
Xylitol is effective at preventing cavities.	Xylitol shows effectiveness at maintaining good oral hygiene and preventing cavities.	[AGREEMENT]
	キシリトールの虫歯抑制効果についてはいろいろな意見がありますが実際は効果があるわけではありません	対立
	There are many opinions about the cavity-prevention effectiveness of Xylitol, but it is not really effective.	[CONFLICT]
還元水は健康に良い	弱アルカリ性のアルカリイオン還元水があなたと家族の健康を支えます	同意
	Reduced water, which has weak alkaline ions, supports the health of you and your family.	[AGREEMENT]
	還元水は活性酸素を除去すると言われ健康を維持してくれる働きをもたらす	同意
Reduced water is good for the health.	Reduced water is said to remove active oxygen from the body, making it effective at promoting good health.	[AGREEMENT]
	美味しくても酸化させる水は健康には役立ちません	対立
	Even if oxidized water tastes good, it does not help one's health.	[CONFLICT]
イソフラボンは健康維持に効果がある	大豆イソフラボンをサプリメントで過剰摂取すると健康維持には負の影響を与える結果となります	限定
Isoflavone is effective at maintaining good health.	Taking too much soy isoflavone as a supplement will have a negative effect on one's health	[CONFINEMENT]

Table 1: Example semantic relation classification.

FINEMENT]. Etoh *et al.* (Etoh and Okumura, 2005) constructed a Japanese Cross-document Relation Corpus, and they redefined 14 kinds of semantic relations in their corpus.

CST was designed for objective expressions because its target data is newspaper articles related to the same topic. Facts, which can be extracted from newspaper articles, have been used in conventional NLP research, such as Information Extraction or Factoid Question Answering. However, there are a lot of opinions on the Web, and it is important to survey opinions in addition to facts to give Internet users a comprehensive view of the discussions on topics of interest.

2.3 Cross-document Summarization Based on CST Relations between Sentences

Zhang and Radev (2004) attempted to classify CST relations between sentence pairs extracted from topically related documents. However, they used a vector space model and tried multi-class classification. The results were not satisfactory. This observation may indicate that the recognition methods for each relation should be developed separately. Miyabe *et al.* (2008) attempted to recognize relations that were defined in a Japanese cross-document relation corpus (Etoh and Okumura, 2005). However, their target relations were limited to [EQUIVALENCE] and [TRANSITION]; other relations were not targeted. Recognizing [EVIDENCE] is indispensable for organizing information on the Internet. We need to develop satisfactory methods of [EVIDENCE] recognition.

2.4 Opinion Mining and Sentiment Analysis

Subjective statements, such as opinions, have recently been the focus of much NLP research including review analysis, opinion extraction, opinion question answering, and sentiment analysis. In the corpus constructed in the Multi-Perspective Question Answering (MPQA) Project (Wiebe et al., 2005), individual expressions are tagged that correspond to explicit mentions of private states, speech event, and expressive subjective elements.

The goal of opinion mining to extract expressions with polarity from texts, not to recognize semantic relations between sentences. Sentiment analysis also focus classifying subjective expressions in texts into positive/negative classes. In comparison, although we deal with sentiment information in text, our objective is to recognize semantic relations between sentences. If a user's query requires positive/negative information, we will also need to extract sentences including sentiment expression like in opinion mining, however, our semantic relation, [CONFINEMENT], is more precise because it identifies the condition or scope of the polarity. Queries do not necessarily include sentiment information; we also accept queries that are intended to be a statement of fact. For example, for the query "Xylitol is effective at preventing cavities." in Table 1, we extract a variety of sentences from the Web and recognize semantic relations between the query and many kinds of sentences.

3 Semantic Relations between Statements

In this section, we define the semantic relations that we will classify in Japanese Internet texts as well as their corresponding relations in RTE and CST. Our goal is to define semantic relations that are applicable over both fact and opinions, making them more appropriate for handling Internet texts. See Table 1 for real examples.

3.1 [AGREEMENT]

A bi-directional relation where statements have equivalent semantic content on a shared topic. Here we use *topic* in a narrow sense to mean that the semantic contents of both statements are relevant to each other.

The following is an example of [AGREEMENT] on the topic of *bio-ethanol environmental impact*.

(1) a. Bio-ethanol is good for the environment.
b. Bio-ethanol is a high-quality fuel, and it has the power to deal with the environment problems that we are facing.

Once relevance has been established, [AGREEMENT] can range from strict logical entailment or identical polarity of opinions.

Here is an example of two statements that share a broad topic, but that are not classified as [AGREEMENT] because *preventing cavities* and *tooth calcification* are not intuitively relevant.

(2) a. Xylitol is effective at preventing cavities.
b. Xylitol advances tooth calcification.

3.2 [CONFLICT]

A bi-directional relation where statements have negative or contradicting semantic content on a shared topic. This can range from strict logical contradiction to opposite polarity of opinions. The next pair is a [CONFLICT] example.

(3) a. Bio-ethanol is good for our earth.
b. There is a fact that bio-ethanol further the destruction of the environment.

3.3 [EVIDENCE]

A uni-directional relation where one statement provides justification or supporting evidence for the other. Both statements can be either facts or opinions. The following is a typical example:

(4) a. I believe that applying the technology of cloning must be controlled by law.
b. There is a need to regulate cloning, because it can be open to abuse.

The *statement* containing the evidence consists of two parts: one part has a [AGREEMENT] or [CONFLICT] with the other *statement*, the other part provides support or justification for it.

3.4 [CONFINEMENT]

A uni-directional relation where one statement provides more specific information about the other or quantifies the situations in which it applies. The pair below is an example, in which one *statement* gives a condition under which the other can be true.

(5) a. Steroids have side-effects.
b. There is almost no need to worry about side-effects when steroids are used for local treatment.

4 Recognizing Semantic Relations

In order to organize the information on the Internet, we need to identify the [AGREEMENT], [CONFLICT], [CONFINEMENT], and [EVIDENCE] semantic relations. Because identification of [AGREEMENT] and [CONFLICT] is a problem of measuring semantic similarity between two *statements*, it can be cast as a sentence alignment problem and solved using an RTE framework. The two sentences do not need to be from the same source.

However, the identification of [CONFINEMENT] and [EVIDENCE] relations depend on contextual information in the sentence. For example, conditional statements or specific discourse markers like "because" act as important cues for their identification. Thus, to identify these two relations across documents, we must first identify [AGREEMENT] or [CONFLICT] between sentences in different documents and then determine if there is a [CONFINEMENT] or [EVIDENCE] relation in one of the documents.

Furthermore, the surrounding text often contains contextual information that is important for identifying these two relations. Proper handling of surrounding context requires discourse analysis and is an area of future work, but our basic detection strategy is as follows:

1. Identify a [AGREEMENT] or [CONFLICT] relation between the Query and Text
2. Search the Text sentence for cues that identify [CONFINEMENT] or [EVIDENCE]

3. Infer the applicability of the [CONFINEMENT] or [EVIDENCE] relations in the Text to the Query

4.1 System Overview

We have finished constructing a prototype system that detects semantic relation between *statements*. It has a three-stage architecture similar to the RTE system of MacCartney *et al.* (2006):

1. Linguistic analysis
2. Structural alignment
3. Feature extraction for detecting [EVIDENCE] and [CONFINEMENT]
4. Semantic relation classification

However, we differ in the following respects.

First, our relation classification is broader than RTE's simple distinction between [ENTAILMENT], [CONTRADICTION], and [UNKNOWN]; in place of [ENTAILMENT] and [CONTRADICTION, we use broader [AGREEMENT] and [CONFLICT] relations. We also consider cover gradations of applicability of statements with the [CONFINEMENT] relation.

Second, we conduct structural alignment with the goal of aligning semantic structures. We do this by directly incorporating dependency alignments and predicate-argument structure information for both the user query and the Web text into the alignment scoring process. This allows us to effectively capture many long-distance alignments that cannot be represented as lexical alignments. This contrasts with MacCartney *et al.* (2006), who uses dependency structures for the Hypothesis to reduce the lexical alignment search space but do not produce structural alignments and do not use the dependencies in detecting entailment.

Finally, we apply several rich semantic resources in alignment and classification: extended modality information that helps align and classify structures that are semantically similar but divergent in tense or polarity; and lexical similarity through ontologies like WordNet.

4.2 Linguistic Analysis

In order to identify semantic relations between the user *query* (Q) and the sentence extracted from Web *text* (T), we first conduct syntactic and semantic linguistic analysis to provide a basis for alignment and relation classification.

For syntactic analysis, we use the Japanese dependency parser CaboCha (Kudo and Matsumoto, 2002) and the predicate-argument structure analyzer ChaPAS (Watanabe et al., 2010). CaboCha splits the Japanese text into phrase-like chunks and represents syntactic dependencies between the chunks as edges in a graph. ChaPAS identifies predicate-argument structures in the dependency graph produced by CaboCha.

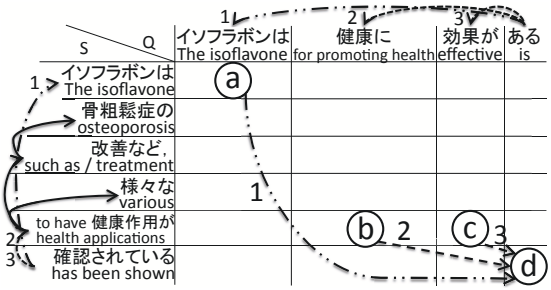

Figure 1: An example of structural alignment

We also conduct extended modality analysis using the resources provided by Matsuyoshi *et al.* (2010), focusing on tense, modality, and polarity, because such information provides important clues for the recognition of semantic relations between *statements*.

4.3 Structural Alignment

In this section, we describe our approach to structural alignment. The structural alignment process is shown in Figure 1. It consists of the following two phases:

1. lexical alignment
2. structural alignment

We developed a heuristic-based algorithm to align chunk based on lexical similarity information. We incorporate the following information into an alignment confidence score that has a range of 0.0-1.0 and align chunk whose scores cross an empirically-determined threshold.

- surface level similarity: identical content words or cosine similarity of chunk contents
- semantic similarity of predicate-argument structures
  - **predicates** we check for matches in predicate entailment databases (Hashimoto et al., 2009; Matsuyoshi et al., 2008) considering the default case frames reported by ChaPAS
  - **arguments** we check for synonym or hypernym matches in the Japanese WordNet (2008) or the Japanese hypernym collection of Sumida *et al.* (2008)

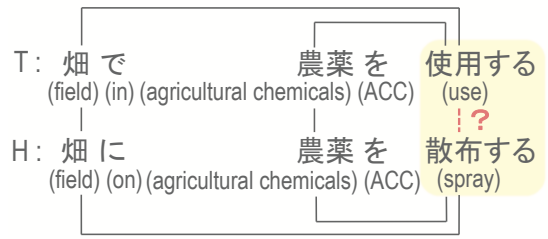

Figure 2: Determining the compatibility of semantic structures

We compare the predicate-argument structure of the query to that of the text and determine if the argument structures are compatible. This process is illustrated in Figure 2 where the T(ext) "Agricultural chemicals are used in the field." is aligned with the H(ypothesis) "Over the field, agricultural chemicals are sprayed." Although the verbs *used* and *sprayed* are not directly semantically related, they are aligned because they share the same argument structures. This lets up align predicates for which we lack semantic resources. We use the following information to determine predicate-argument alignment:

- the number of aligned children
- the number of aligned case frame arguments
- the number of possible alignments in a window of n chunk
- predicates indicating existence or quantity. E.g. *many*, *few*, *to exist*, etc.
- polarity of both parent and child chunks using the resources in (Higashiyama et al., 2008; Kobayashi et al., 2005)

We treat structural alignment as a machine learning problem and train a Support Vector Machine (SVM) model to decide if lexically aligned chunks are semantically aligned.

We train on gold-standard labeled alignment of 370 sentence pairs. This data set is described in more detail in Section 5.1. As features for our SVM model, we use the following information:

- the distance in edges in the dependency graph between parent and child for both sentences
- the distance in chunks between parent and child in both sentences
- binary features indicating whether each chunk is a predicate or argument according to ChaPAS
- the parts-of-speech of first and last word in each chunk
- when the chunk ends with a case marker, the case of the chunk , otherwise *none*
- the lexical alignment score of each chunk pair

4.4 Feature Extraction for Detecting Evidence and Confinement

Once the structural alignment system has identified potential [AGREEMENT] or [CONFLICT] relations, we need to extract contextual cues in the Text as features for detecting [CONFINEMENT] and [EVIDENCE] relations. Conditional statements, degree adverbs, and partial negation, which play a role in limiting the scope or degree of a *query*'s contents in the *statement*, can be important cues for detecting the these two semantic relations. We currently use a set of heuristics to extract a set of expressions to use as features for classifying these relations using SVM models.

4.5 Relation Classification

Once the structural alignment is successfully identified, the task of semantic relation classification is straightforward. We also solve this problem with machine learning by training an SVM classifier. As features, we draw on a combination of lexical, syntactic, and semantic information including the syntactic alignments from the previous section. The feature set is:

alignments We define two binary function, $ALIGN_{word}(q_i, t_m)$ for the lexical alignment and $ALIGN_{struct}((q_i, q_j), (t_m, t_k))$ for the structural alignment to be true if and only if the node $q_i, q_j \in Q$ has been semantically and structurally aligned to the node $t_m, t_k \in T$. Q and T are the (Q)uery and the (T)ext, respectively. We also use a separate feature for a score representing the likelihood of the alignment.

modality We have a feature that encodes all of the possible polarities of a predicate node from modality analysis, which indicates the utterance type, and can be *assertive*, *volitional*, *wish*, *imperative*, *permissive*, or *interrogative*. Modalities that do not represent opinions (i.e. *imperative*, *permissive* and *interrogative*) often indicate [OTHER] relations.

antonym We define a binary function $ANTONYM(q_i, t_m)$ that indicates if the pair is identified as an antonym. This information helps identify [CONFLICT].

Relation	Measure	3-class		Cascaded 3-class		Δ
[AGREEMENT]	precision	0.79	(128 / 162)	0.80	(126 / 157)	+0.01
[AGREEMENT]	recall	0.86	(128 / 149)	0.85	(126 / 149)	-0.01
[AGREEMENT]	f-score	0.82		0.82		-
[CONFLICT]	precision	0	(0 / 5)	0.36	(5 / 14)	+0.36
[CONFLICT]	recall	0	(0 / 12)	0.42	(5 / 12)	+0.42
[CONFLICT]	f-score	0		0.38		+0.38
[CONFINEMENT]	precision	0.4	(4 / 10)	0.8	(4 / 5)	+0.4
[CONFINEMENT]	recall	0.17	(4 / 23)	0.17	(4 / 23)	-
[CONFINEMENT]	f-score	0.24		0.29		+0.05

Table 2: Semantic relation classification results comparing 3-class and cascaded 3-class approaches

negation To identify negations, we primarily rely on a predicate's *Actuality* value, which represents epistemic modality and existential negation. If a predicate pair $ALIGN_{word}(q_i, t_m)$ has mismatching actuality labels, the pair is likely a [OTHER].

contextual cues This set of features is used to mark the presence of any contextual cues that identify of [CONFINEMENT] or [EVIDENCE] relations in a chunk . For example, "ので (because)" or "ため (due to)" are typical contextual cues for [EVIDENCE], and "とき (when)" or "ならば (if)" are typical for [CONFINEMENT].

5 Evaluation

5.1 Data Preparation

In order to evaluate our semantic relation classification system on realistic Web data, we constructed a corpus of sentence pairs gathered from a vast collection of webpages (2009a). Our basic approach is as follows:

1. Retrieve documents related to a set number of topics using the Tsubaki[1] search engine
2. Extract real sentences that include major subtopic words which are detected based on TF/IDF in the document set
3. Reduce noise in data by using heuristics to eliminate advertisements and comment spam
4. Reduce the search space for identifying sentence pairs and prepare pairs, which look feasible to annotate
5. Annotate corresponding sentences with [AGREEMENT], [CONFLICT], [CONFINEMENT], or [OTHER]

[1] http://tsubaki.ixnlp.nii.ac.jp/

Although our target semantic relations include [EVIDENCE], they difficult annotate consistently, so we do not annotate them at this time. Expanding our corpus and semantic relation classifier to handle [EVIDENCE] remains and area of future work.

The data that composes our corpus comes from a diverse number of sources. A hand survey of a random sample of the types of domains of 100 document URLs is given below. Half of the URL domains were not readily identifiable, but the known URL domains included governmental, corporate, and personal webpages. We believe this distribution is representative of information sources on the Internet.

type	count
academic	2
blogs	23
corporate	10
governmental	4
news	5
press releases	4
q&a site	1
reference	1
other	50

We have made a partial release of our corpus of sentence pairs manually annotated with the correct semantic relations[2]. We will fully release all the data annotated semantic relations and with gold standard alignments at a future date.

5.2 Experiment Settings

In this section, we present results of empirical evaluation of our proposed semantic relation classification system on the dataset we constructed in the previous section. For this experiment, we use SVMs as described in Section 4.5

[2] http://stmap.naist.jp/corpus/ja/index.html (in Japanese)

to classify semantic relations into one of the four classes: [AGREEMENT], [CONFLICT], [CONFINEMENT], or [OTHER] in the case of no relation. As data we use 370 sentence pairs that have been manually annotated both with the correct semantic relation and with gold standard alignments. Annotations are checked by two native speakers of Japanese, and any sentence pair where annotation agreement is not reached is discarded. Because we have limited data that is annotated with correct alignments and semantic relations, we perform five-fold cross validation, training both the structural aligner and semantic relation classifier on 296 sentence pairs and evaluating on the held out 74 sentence pairs. The figures presented in the next section are for the combined results on all 370 sentence pairs.

5.3 Results

We compare two different approaches to classification using SVMs:

3-class semantic relations are directly classified into one of [AGREEMENT], [CONFLICT], and [CONFINEMENT] with all features described in 4.5

cascaded 3-class semantic relations are first classified into one of [AGREEMENT], [CONFLICT] without contextual cue features. Then an additional judgement with all features determines if [AGREEMENT] and [CONFLICT] should be reclassified as [CONFINEMENT]

Initial results using the **3-class** classification model produced high f-scores for [AGREEMENT] but unfavorable results for [CONFLICT] and [CONFINEMENT]. We significantly improved classification of [CONFLICT] and [CONFINEMENT] by adopting the **cascaded 3-class** model. We present these results in Table 2 and successfully recognized examples in Table 1.

6 Discussion and Error Analysis

We constructed a prototype semantic relation classification system by combining the components described in the previous section. While the system developed is not domain-specific and capable of accepting queries on any topic, we evaluate its semantic relation classification on several queries that are representative of our training data.

Figure 3 shows a snapshot of the semantic relation classification system and the various semantic relations it recognized for the query.

	Baseline	Structural Alignment	Upper-bound
Precision	0.44 (56/126)	0.52 (96/186)	0.74 (135/183)
Recall	0.30 (56/184)	0.52 (96/184)	0.73 (135/184)
F1-score	0.36	0.52	0.74

Table 3: Comparison of lexical, structural, and upper-bound alignments on semantic relation classification

In the example (6), recognized as [CONFINEMENT] in Figure 3, our system correctly identified negation and analyzed the description "Xylitol alone can not completely" as playing a role of requirement.

(6) a. キシリトールは虫歯予防に効果がある
(Xylitol is effective at preventing cavities.)
b. キシリトールだけでは完全な予防は出来ません
(Xylitol alone can not completely prevent cavities.)

Our system correctly identifies [AGREEMENT] relations in other examples about reduced water from Table 1 by structurally aligning phrases like "promoting good health" and "supports the health" to "good for the health."

These examples show how resources like (Matsuyoshi et al., 2010) and WordNet (Bond et al., 2008) have contributed to the relation classification improvement of structural alignment over them baseline in Table 3. Focusing on similarity of syntactic and semantic structures gives our alignment method greater flexibility.

However, there are still various examples which the system cannot recognized correctly. In examples on cavity prevention, the phrase "effective at preventing cavities" could not be aligned with "can prevent cavities" or "good for cavity prevention," nor can "cavity prevention" and "cavity-causing bacteria control."

The above examples illustrate the importance of the role played by the alignment phase in the whole system's performance.

Table 3 compares the semantic relation classification performance of using lexical alignment only (as the baseline), lexical alignment and structural alignment, and, to calculate the maximum possible precision, classification using correct alignment data (the upper-bound). We can

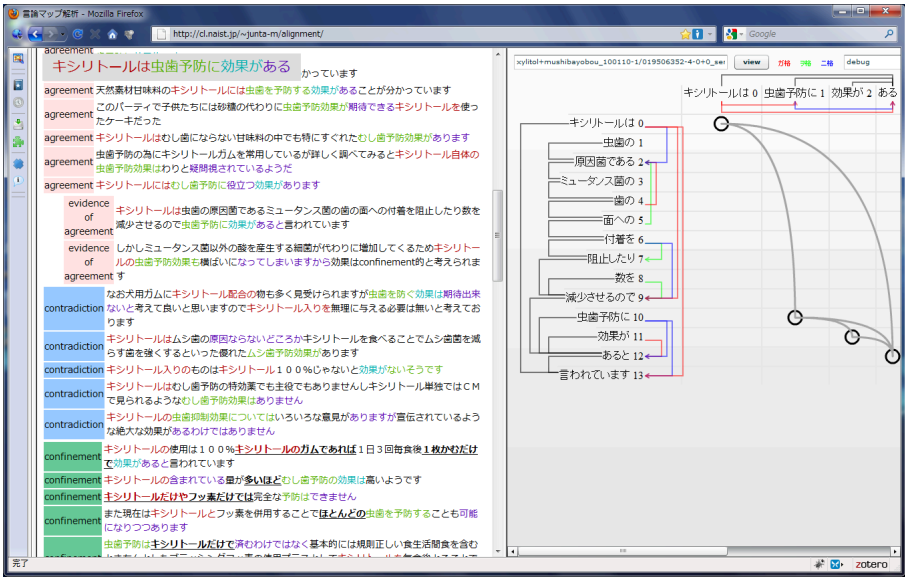

Figure 3: Alignment and classification example for the query *"Xylitol is effective at preventing cavities."*

see that structural alignment makes it possible to align more words than lexical alignment alone, leading to an improvement in semantic relation classification. However, there is still a large gap between the performance of structural alignment and the maximum possible precision. Error analysis shows that a big cause of incorrect classification is incorrect lexical alignment. Improving lexical alignment is a serious problem that must be addressed. This entails expanding our current lexical resources and finding more effective methods of apply them in alignment.

The most serious problem we currently face is the feature engineering necessary to find the optimal way of applying structural alignments or other semantic information to semantic relation classification. We need to conduct a quantitative evaluation of our current classification models and find ways to improve them.

7 Conclusion

Classifying and identifying semantic relations between facts and opinions on the Web is of utmost importance to organizing information on the Web, however, this requires consideration of a broader set of semantic relations than are typically handled in RTE, CST, and similar tasks. In this paper, we introduced a set of cross-sentential semantic relations specifically designed for this task that apply over both facts and opinions. We presented a system that identifies these semantic relations in Japanese Web texts using a combination of lexical, syntactic, and semantic information and evaluated our system against data that was manually constructed for this task. Preliminary evaluation showed that we are able to detect [AGREEMENT] with high levels of confidence. Our method also shows promise in [CONFLICT] and [CONFINEMENT] detection. We also discussed some of the technical issues that need to be solved in order to identify [CONFLICT] and [CONFINEMENT].

Acknowledgments

This work is supported by the National Institute of Information and Communications Technology Japan.

References

Bond, Francis, Hitoshi Isahara, Kyoko Kanzaki, and Kiyotaka Uchimoto. 2008. Boot-strapping a wordnet using multiple existing wordnets. In *Proc. of the 6th International Language Resources and Evaluation (LREC'08)*.

Dagan, Ido, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Proc. of the PASCAL Challenges Workshop on Recognising Textual Entailment*.

Etoh, Junji and Manabu Okumura. 2005. Cross-document relationship between sentences corpus.

In *Proc. of the 14th Annual Meeting of the Association for Natural Language Processing*, pages 482–485. (in Japanese).

Glickman, Oren, Ido Dagan, and Moshe Koppel. 2005. Web based textual entailment. In *Proc. of the First PASCAL Recognizing Textual Entailment Workshop*.

Hashimoto, Chikara, Kentaro Torisawa, Kow Kuroda, Masaki Murata, and Jun'ichi Kazama. 2009. Large-scale verb entailment acquisition from the web. In *Conference on Empirical Methods in Natural Language Processing (EMNLP2009)*, pages 1172–1181.

Hickl, Andrew, John Williams, Jeremy Bensley, Kirk Roberts, Bryan Rink, and Ying Shi. 2005. Recognizing textual entailment with lcc's groundhog system. In *Proc. of the Second PASCAL Challenges Workshop*.

Higashiyama, Masahiko, Kentaro Inui, and Yuji Matsumoto. 2008. Acquiring noun polarity knowledge using selectional preferences. In *Proc. of the 14th Annual Meeting of the Association for Natural Language Processing*.

Jijkoun, Valentin and Maarten de Rijke. 2005. Recognizing textual entailment using lexical similarity. In *Proc. of the First PASCAL Challenges Workshop*.

Kobayashi, Nozomi, Kentaro Inui, Yuji Matsumoto, Kenji Tateishi, and Toshikazu Fukushima. 2005. Collecting evaluative expressions for opinion extraction. *Journal of natural language processing*, 12(3):203–222.

Kudo, Taku and Yuji Matsumoto. 2002. Japanese dependency analysis using cascaded chunking. In *Proc of CoNLL 2002*, pages 63–69.

MacCartney, Bill, Trond Grenager, Marie-Catherine de Marneffe, Daniel Cer, and Christopher D. Manning. 2006. Learning to recognize features of valid textual entailments. In *Proc. of HLT/NAACL 2006*.

Marsi, Erwin and Emiel Krahmer. 2005. Classification of semantic relations by humans and machines. In *Proc. of ACL-05 Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pages 1–6.

Matsuyoshi, Suguru, Koji Murakami, Yuji Matsumoto, and Kentaro Inui. 2008. A database of relations between predicate argument structures for recognizing textual entailment and contradiction. In *Proc. of the Second International Symposium on Universal Communication*, pages 366–373, December.

Matsuyoshi, Suguru, Megumi Eguchi, Chitose Sao, Koji Murakami, Kentaro Inui, and Yuji Matsumoto. 2010. Annotating event mentions in text with modality, focus, and source information. In *Proc. of the 7th International Language Resources and Evaluation (LREC'10)*, pages 1456–1463.

Miyabe, Yasunari, Hiroya Takamura, and Manabu Okumura. 2008. Identifying cross-document relations between sentences. In *Proc. of the 3rd International Joint Conference on Natural Language Processing (IJCNLP-08)*, pages 141–148.

Murakami, Koji, Shouko Masuda, Suguru Matsuyoshi, Eric Nichols, Kentaro Inui, and Yuji Matsumoto. 2009a. Annotating semantic relations combining facts and opinions. In *Proceedings of the Third Linguistic Annotation Workshop*, pages 150–153, Suntec, Singapore, August. Association for Computational Linguistics.

Murakami, Koji, Eric Nichols, Suguru Matsuyoshi, Asuka Sumida, Shouko Masuda, Kentaro Inui, and Yuji Matsumoto. 2009b. Statement map: Assisting information credibility analysis by visualizing arguments. In *Proc. of the 3rd ACM Workshop on Information Credibility on the Web (WICOW 2009)*, pages 43–50.

Radev, Dragomir, Jahna Otterbacher, and Zhu Zhang. 2003. CSTBank: Cross-document Structure Theory Bank. http://tangra.si.umich.edu/clair/CSTBank.

Radev, Dragomir R. 2000. Common theory of information fusion from multiple text sources step one: Cross-document structure. In *Proc. of the 1st SIGdial workshop on Discourse and dialogue*, pages 74–83.

Sumida, Asuka, Naoki Yoshinaga, and Kentaro Torisawa. 2008. Boosting precision and recall of hyponymy relation acquisition from hierarchical layouts in wikipedia. In *Proc. of the 6th International Language Resources and Evaluation (LREC'08)*.

Szpektor, Idan, Eyal Shnarch, and Ido Dagan. 2007. Instance-based evaluation of entailment rule acquisition. In *Proc. of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 456–463.

Watanabe, Yotaro, Masayuki Asahara, and Yuji Matsumoto. 2010. A structured model for joint learning of argument roles and predicate senses. In *Proceedings of the 48th Annual Meeting of the Association of Computational Linguistics (to appear)*.

Wiebe, Janyce, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2-3):165–210.

William, Mann and Sandra Thompson. 1988. Rhetorical structure theory: towards a functional theory of text organization. *Text*, 8(3):243–281.

Zhang, Zhu and Dragomir Radev. 2004. Combining labeled and unlabeled data for learning cross-document structural relationships. In *Proc. of the Proceedings of IJC-NLP*.

Statistical Relational Learning for Knowledge Extraction from the Web

Hoifung Poon
University of Washington, USA

Abstract

Extracting knowledge from unstructured text has been a long-standing goal of NLP. The advent of the Web further increases its urgency by making available billions of online documents. To represent the acquired knowledge that is complex and heterogeneous, we need first-order logic. To handle the inherent uncertainty and ambiguity in extracting and reasoning with knowledge, we need probability. Combining the two has led to rapid progress in the emerging field of statistical relational learning. In this talk, I will show that statistical relational learning offers promising solutions for conquering the knowledge-extraction quest. I will present Markov logic, which is the leading unifying framework for representing and reasoning with complex and uncertain knowledge, and has spawned a number of successful applications for knowledge extraction from the Web. In particular, I will present OntoUSP, an end-to-end knowledge extraction system that can read text and answer questions. OntoUSP is completely unsupervised and benefits from jointly conducting ontology induction, population, and knowledge extraction. Experiments show that OntoUSP extracted five times as many correct answers compared to state-of-the-art systems, with a precision of 91%.

Proceedings of the Second Workshop on NLP Challenges in the Information Explosion Era (NLPIX 2010), page 31,
Beijing, August 2010

Even Unassociated Features Can Improve Lexical Distributional Similarity

Kazuhide Yamamoto and **Takeshi Asakura**
Department of Electrical Engineering
Nagaoka University of Technology
{yamamoto, asakura}@jnlp.org

Abstract

This paper presents a new computation of lexical distributional similarity, which is a corpus-based method for computing similarity of any two words. Although the conventional method focuses on emphasizing features with which a given word is associated, we propose that even unassociated features of two input words can further improve the performance in total. We also report in addition that more than 90% of the features has no contribution and thus could be reduced in future.

1 Introduction

Similarity calculation is one of essential tasks in natural language processing (1990; 1992; 1994; 1997; 1998; 1999; 2005). We look for a semantically similar word to do corpus-driven summarization, machine translation, language generation, recognition of textual entailment and other tasks. In task of language modeling and disambiguation we also need to semantically generalize words or cluster words into some groups. As the amount of text increases more and more in the contemporary world, the importance of similarity calculation also increases concurrently.

Similarity is computed by roughly two approaches: based on thesaurus and based on corpus. The former idea uses thesaurus, such as WordNet, that is a knowledge resource of hierarchical word classification. The latter idea, that is the target of our work, originates from Harris's distributional hypothesis more than four decades ago (1968), stating that semantically similar words tend to appear in similar contexts. In many cases a context of a word is represented as a feature vector, where each feature is another expression that co-occurs with the given word in the context.

Over a long period of its history, in particular in recent years, several works have been done on distributional similarity calculation. Although the conventional works have attained the *fine* performance, we attempt to further improve the quality of this measure. Our motivation of this work simply comes from our observation and analysis of the output by conventional methods; Japanese, our target language here, is written in a mixture of four scripts: Chinese characters, Latin alphabet, and two Japanese-origin characters. In this writing environment some words which have same meaning and same pronunciation are written in two (or more) different scripts. This is interesting in terms of similarity calculation since these two words are completely same in semantics so the similarity should be ideally 1.0. However, the reality is, as far as we have explored, that the score is far from 1.0 in many *same* word pairs. This fact implies that the conventional calculation methods are far enough to the goal and are expected to improve further.

The basic framework for computing distributional similarity is same; for each of two input words a context (i.e., surrounding words) is extracted from a corpus, a vector is made in which an element of the vector is a value or a weight, and two vectors are compared with a formula to compute similarity. Among these processes we have focused on features, that are elements of

Proceedings of the Second Workshop on NLP Challenges in the Information Explosion Era (NLPIX 2010), pages 32–39,
Beijing, August 2010

the vector, some of which, we think, adversely affect the performance. That is, traditional approaches such as Lin (1998) basically use all of observed words as context, that causes noise in feature vector comparison. One may agree that the number of the characteristic words to determine the meaning of a word is some, not all, of words around the target word. Thus our goal is to detect and reduce such noisy features.

Zhitomirsky-Geffet and Dagan (2009) have same motivation with us and introduced a bootstrapping strategy that changes the original features weights. The general idea here is to promote the weights of features that are common for associated words, since these features are likely to be most characteristic for determining the word's meaning. In this paper, we propose instead a method to using features that are both unassociated to the two input words, in addition to use of features that are associated to the input.

2 Method

The lexical distributional similarity of the input two words is computed by comparing two vectors that express the context of the word. In this section we first explain the feature vector, and how we define initial weight for each feature of the vector. We then introduce in Subsection 2.3 the way to compute similarity by two vectors. After that, we emphasize some of the features by their association to the word, that is explained in Subsection 2.4. We finally present in Subsection 2.5 feature reduction which is our core contribution of this work. Although our target language is Japanese, we use English examples in order to provide better understanding to the readers.

2.1 Feature Vector

We first explain how to construct our feature vector from a text corpus.

A word is represented by a feature vector, where features are collection of syntactically dependent words co-occurred in a given corpus. Thus, we first collect syntactically dependent words for each word. This is defined, as in Lin (1998), as a triple (w, r, w'), where w and w' are words and r is a syntactic role. As for definition of *word*, we use not only words given by a morphological analyzer but also compound words. Nine case particles are used as syntactic roles, that roughly express subject, object, modifier, and so on, since they are easy to be obtained from text with no need of semantic analysis. In order to reduce noise we delete triples that appears only once in the corpus.

We then construct a feature vector out of collection of the triples. A feature of a word is an another word syntactically dependent with a certain role. In other words, given a triple (w, r, w'), a feature of w corresponds to a dependent word with a role (r, w').

2.2 (Initial) Filtering of Features

There are several weighting functions to determine a value for each feature element. As far as we have investigated the literature the most widely used feature weighting function is pointwise mutual information (MI), that is defined as follows:

$$MI(w, r, w') = \log_2 \frac{freq(w, r, w')S}{freq(w)freq(r, w')} \quad (1)$$

where $freq(r, w')$ is the frequency of the co-occurrence word w' with role r, $freq(w)$ is the independent frequency of a word w, $freq(w, r, w')$ is the frequency of the triples (w, r, w'), and S is the number of all triples.

In this paper we do not discuss what is the best weighting functions, since this is out of target. We use mutual information here because it is most widely used, i.e., in order to compare performance with others we want to adopt the standard approach.

As other works do, we filter out features that have a value lower than a minimal weight thresholds α. The thresholds are determined according to our preliminary experiment, that is explained later.

2.3 Vector Similarity

Similarity measures of the two vectors are computed by various measures. Shibata and Kurohashi (2009) have compared several similarity measures including Cosine (Ruge, 1992), (Lin,

(input word) w: boy
(feature) v: $\underline{\text{guard}_{\text{OBJ}}}$
(synonyms of w, shown with its similarity to w) $Syn(w)$ =
{ child(0.135), girl(0.271), pupil(0.143), woman(0.142), young people(0.147) }

(feature vectors V):
V(boy) = { $\text{parents}_{\text{MOD}}$, $\text{runaway}_{\text{SUBJ}}$, $\text{reclaim}_{\text{OBJ}}$, $\text{father}_{\text{MOD}}$, $\underline{\text{guard}_{\text{OBJ}}}$, $\cdots$ }

V(child) = { $\underline{\text{guard}_{\text{OBJ}}}$, look_{OBJ}, $\text{bring}_{\text{OBJ}}$, $\text{give birth}_{\text{OBJ}}$, care_{OBJ} , $\cdots$ }
V(girl) = { $\text{parents}_{\text{MOD}}$, $\underline{\text{guard}_{\text{OBJ}}}$, $\text{father}_{\text{MOD}}$, $\text{testify}_{\text{SUBJ}}$, look_{OBJ}, $\cdots$ }
V(pupil) = { $\text{target}_{\text{OBJ}}$, $\underline{\text{guard}_{\text{OBJ}}}$, care_{OBJ}, aim_{OBJ}, $\text{increase}_{\text{SUBJ}}$, $\cdots$ }
V(woman) = { name_{MOD}, $\text{give birth}_{\text{OBJ}}$, $\text{group}_{\text{MOD}}$, together+with, $\text{parents}_{\text{MOD}}$, $\cdots$ }
V(young people) = { $\text{harmful}_{\text{TO}}$, $\text{global}_{\text{MOD}}$, $\text{reclaim}_{\text{OBJ}}$, $\text{wrongdoing}_{\text{MOD}}$, $\cdots$ }

(words that has feature v) $Asc(v)$ = {boy, child, girl, pupil, $\cdots$}

$weight(w, v) = weight$ (boy, $\text{guard}_{\text{OBJ}}$) $= \sum_{w_f \in Asc(v) \cap Syn(w)} sim(w, w_f)$
$= 0.135 + 0.271 + 0.143 = 0.549$

Figure 1: Example of feature weighting for word *boy*.

1998), (Lin, 2002), Simpson, Simpson-Jaccard, and conclude that Simpson-Jaccard index attains best performance of all. Simpson-Jaccard index is an arithmetic mean of Simpson index and Jaccard index, defined in the following equation:

$$sim(w_1, w_2) = \frac{1}{2}(sim_J(w_1, w_2) + sim_S(w_1, w_2)) \quad (2)$$

$$sim_J(w_1, w_2) = \frac{|V_1 \cap V_2|}{|V_1 \cup V_2|} \quad (3)$$

$$sim_S(w_1, w_2) = \frac{|V_1 \cap V_2|}{min(|V_1|, |V_2|)} \quad (4)$$

where V_1 and V_2 is set of features for w_1 and w_2, respectively, and $|A|$ is the number of set A. It is interesting to note that both Simpson and Jaccard compute similarity according to degree of overlaps of the two input sets, that is, the reported best measure computes similarity by ignoring the weight of the features. In this paper we adopt Simpson-Jaccard index, sim, which indicates that the weight of features that is explained below is only used for feature reduction, not for similarity calculation.

2.4 Feature Weighting by Association

We then compute weights of the features of the word w according to the degree of semantic association to w. The weight is biased because all of the features, i.e., the surrounding words, are not equally characteristic to the input word. The core idea for feature weighting is that a feature v in w is more weighted when more synonyms (words of high similarity) of w also have v.

Figure 1 illustrates this process by examples. Now we calculate a feature $\text{guard}_{\text{OBJ}}$ for a word *boy*, we first collect synonyms of w, denoted by $Syn(w)$, from a thesaurus. We then compute similarities between w and each word in $Syn(w)$ by Equation 2. The weight is the sum of the similarities of words in $Syn(w)$ that have feature v, defined in Equation 5.

$$weight(w, v) = \sum_{w_f \in Asc(v) \cap Syn(w)} sim(w, w_f) \quad (5)$$

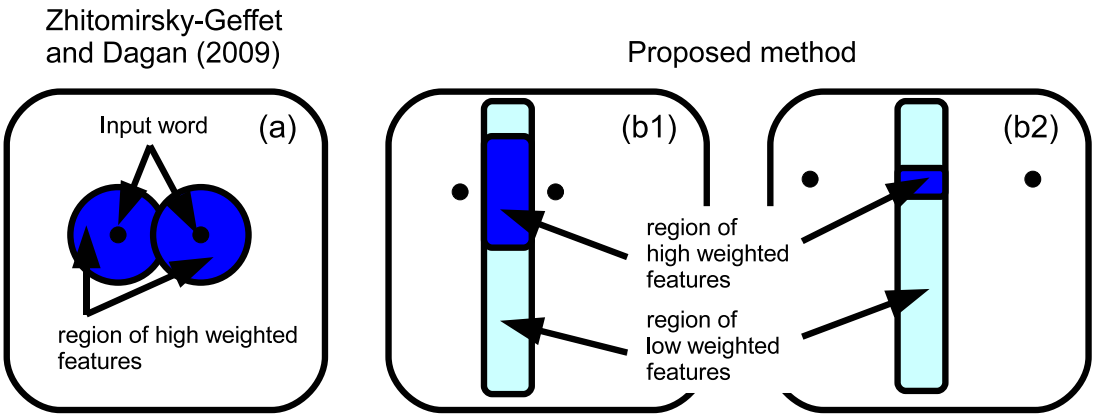

Figure 2: An illustration of similarity calculation of Zhitomirsky-Geffet and Dagan (2009) (a) and the proposed method (b1 and b2) in feature space. In order to measure the distance of the two words (shown in black dots) they use only associated words, while we additionally use unassociated words in which the distances to the words are similar.

2.5 Feature Reduction

We finally reduce features according to the difference of weights of each feature in words we compare. In computing similarity of two words, w_1 and w_2, a feature v satisfying Equation 6 is reduced.

$$abs(weight(w_1, v) - weight(w_2, v)) > \beta \quad (6)$$

where $abs()$ is a function of absolute value, and β is a threshold for feature reduction.

Figure 2 illustrates our idea and compares the similar approach proposed by Zhitomirsky-Geffet and Dagan (2009). Roughly speaking, Zhitomirsky-Geffet and Dagan (2009) compute similarity of two words, shown as black dots in (a), mainly according to *associated* features (dark-colored circle), or features that has high weights in Equation 5. And the associated features are determined word by word independently.

In contrast, the proposed method relatively reduces features, depending on *location* of input two words. At (b1) in the figure, not only associated (high-colored area) but unassociated features (light-colored area) are used for similarity computation in our method. As Equation 6 shows, regardless of how much a feature is associated to the word, the feature is not reduced when it has similar weight to both w_1 and w_2, located at the middle area of the two words in the figure.

This idea seems to work more effectively, compared with Zhitomirsky-Geffet and Dagan (2009), in case that input two words are not so similar, that is shown at (b2) of the figure. As they define associated features independently, it is likely that the overlapped area is little or none between the two words. In contrast, our method uses features at the *middle* area of two input words, where there is always certain features provided for similarity computation, shown in case (b2). Simplified explanation is that our similarity is computed as the ratio of the associated area to the unassociated area in the figure. We will verify later if the method works better in low similarity calculation.

2.6 Final Similarity

The final similarity of two words are calculated by two shrunk vectors (or feature sets) and Equation 2, that gives a value between 0 and 1.

3 Evaluation

3.1 Evaluation Method

In general it is difficult to answer how similar two given words are. Human have no way to judge correctness if computed similarity of two words is, for instance, 0.7. However, given two word pairs, such as (w, w_1) and (w, w_2), we may answer which of two words, w_1 or w_2, is more similar to w than the other one. That is, degree of similarity is defined relatively hence accuracy of similarity measures is evaluated by way of relative comparisons.

In this paper we employ an automatic evaluation method in order to reduce time, human labor, and individual variations. We first collect four levels of similar word pairs from a thesaurus[1]. Thesaurus is a resource of hierarchical words classification, hence we can collect several levels of similar word pairs according to the depth of common parent nodes that two words have. Accordingly, we constructed four levels of similarity pairs, Level 0, 1, 2, and 3, where the number increases as the similarity increases. Each level includes 800 word pairs that are randomly selected. The following examples are pairs with word *Asia* in each Level.

Example: Four similarity levels for pair of *Asia*.

Level 3(high):	Asia vs. Europe
Level 2:	Asia vs. Brazil
Level 1:	Asia vs. my country
Level 0(low):	Asia vs. system

We then combine word pairs of adjacent similarity Levels, such as Level 0 and 1, that is a test set to see low-level similarity discrimination power. The performance is calculated in terms of how clearly the measure distinguishes the different levels. In a similar fashion, Level 1 and 2, as well as 2 and 3, are combined and tested for middle-level and high-level similarity discrimination, respectively. The number of pairs in each test set is 1,600 as two Levels are combined.

[1]In this experiment we use *Bunrui Goi Hyo* also for evaluation. Therefore, this experimental setting is a kind of closed test. However, we see that the advantage to use the same thesaurus in the evaluation seems to be small.

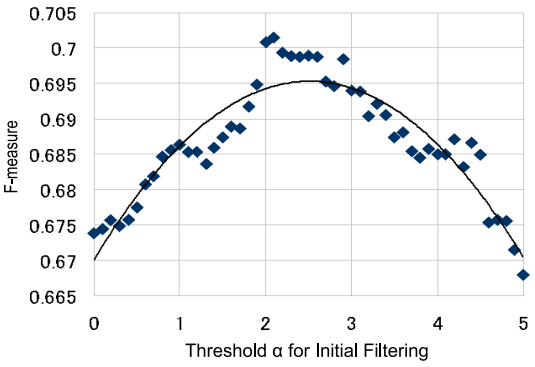

Figure 3: Relation between threshold α and performance in F-measures for Level 3+2 test set.

3.2 Experimental Setting

The corpus we use in this experiment is all the articles in *The Nihon Keizai Shimbun Database*, a Japanese business newspaper corpus covering the years 1990 through 2004. As morphological analyzer we use *Chasen* 2.3.3 with IPA morpheme dictionary. The number of collected triples is 2,584,905, that excludes deleted ones due to one time appearance and words including some symbols.

In Subsection 2.4 we use *Bunrui Goi Hyo*, a Japanese thesaurus for synonym collection. The potential target words are all content words, except words that have less than twenty features. The number of words after exclusion is 75,530. Moreover, words that have four or less words in the same category in the thesaurus are regarded as out of target in this paper, due to limitation of $Syn(w)$ in Subsection 2.4. Also, in order to avoid word sense ambiguity, words that have more than two meanings, i.e., those classified in more than two categories in the thesaurus, also remain to be solved.

3.3 Threshold for Initial Filtering

Figure 3 shows relation between threshold α and the performance of similarity distinction that is drawn in F-measures, for Level 3+2 test set. As can be seen, the plots seem to be concave down

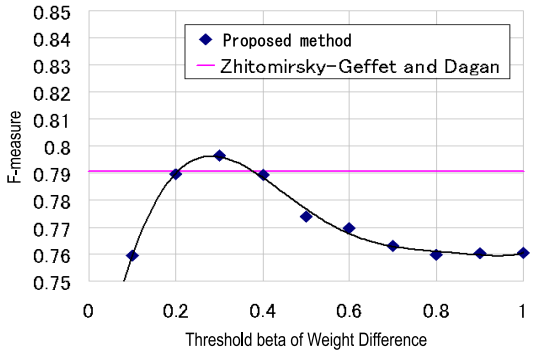

Figure 4: Threshold vs. accuracy in Level 3+2 set.

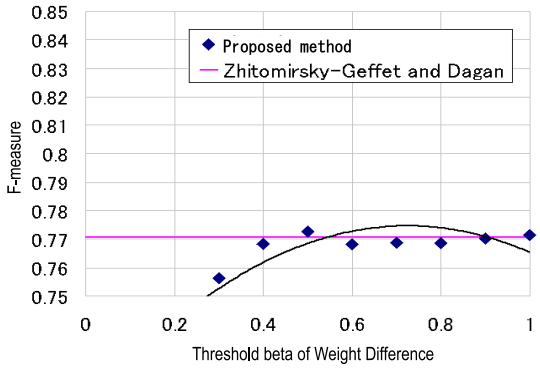

Figure 5: Threshold vs. accuracy in Level 2+1 set.

and there is a clear peak when α is between 2 and 3.

In the following experiments we set α the value where the best performance is given for each test set. We have observed similar phenomena in other test sets. The thresholds we use is 2.1 for Level 3+2, 2.4 for Level 2+1, and 2.4 for Level 1+0.

3.4 Threshold for Weighting Function

Figure 4, 5, and 6 show relation between threshold β and performance in Level 3+2, 2+1, 1+0 test set, respectively. The threshold at the point where highest performance is obtained greatly depends on Levels: 0.3 in Level 3+2, 0.5 in Level 2+1, and 0.9 in Level 1+0. Comparison of these three figures indicates that similarity distinction

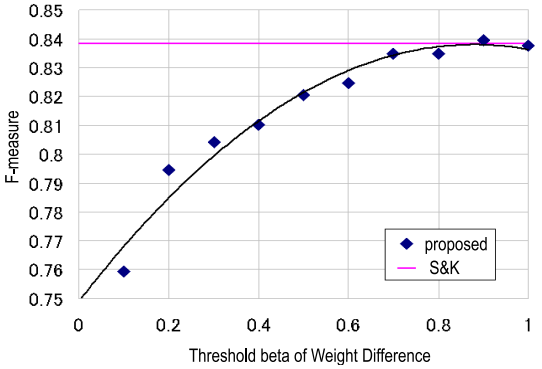

Figure 6: Threshold vs. accuracy in Level 1+0 set.

Table 1: Performance comparison of three methods in each task (in F-measures).

Level	S&K	ZG&D	proposed
Lvl.3+Lvl.2	0.702	0.791	**0.797**
Lvl.2+Lvl.1	0.747	0.771	**0.773**
Lvl.1+Lvl.0	0.838	0.789	**0.840**

power in higher similarity region requires lower threshold, i.e., fewer features. In contrast, conducting fine distinction in lower similarity level requires higher threshold, i.e., a lot of features most of which may be unassociated ones.

3.5 Performance

Table 1 shows performance of the proposed method, compared with Shibata and Kurohashi (2009) (S&K in the table) and Zhitomirsky-Geffet and Dagan (2009) (ZG&D)[2]. The method of Shibata and Kurohashi (2009) here is the best one among those compared. It uses only initial filtering described in Subsection 2.2. The method of Zhitomirsky-Geffet and Dagan (2009) in addition emphasize associated features as explained in Subsection 2.4. All of the results in the table are the best ones among several threshold settings.

The result shows that the accuracy is 0.797 (+0.006) in Level 3+2, 0.773 (+0.002) in Level

[2]The implementations of providing associated words and the bootstrapping are slightly different to Zhitomirsky-Geffet and Dagan (2009).

2+1, and 0.840 (+0.001) in Level 1+0, where the degree of improvement here are those compared with best ones except our proposed method. This confirms that our method attains equivalent or better performance in all of low, middle, and high similarity levels.

We also see in the table that S&K and ZG&D show different behavior according to the Level. However, it is important to note here that our proposed method performs equivalent or outperforms both methods in all Levels.

4 Discussions

4.1 Behavior at Each Similarity Level

As we have discussed in Subsection 2.5, our method is expected to perform better than Zhitomirsky-Geffet and Dagan (2009) in distinction in lower similarity area. Roughly speaking, we interpret the results as follows. Shibata and Kurohashi (2009) always has many features that degrades the performance in higher similarity level, since the ratio of noisy features may throw into confusion. Zhitomirsky-Geffet and Dagan (2009) reduces such noise that gives better performance in higher similarity level and is stable in all levels. And our proposed method maintains performance of Zhitomirsky-Geffet and Dagan (2009) in higher level while improves performance that is close to Shibata and Kurohashi (2009) in lower level, utilizing fewer features. We think our method can include advantages over the two methods.

4.2 Error Analysis

We overview the result and see that the major errors are NOT due to lack of features. Table 2 illustrates the statistics of words with a few features (less than 50 or 20). This table clearly tells us that, in the low similarity level (Level 1+0) in particular, there are few pairs in which the word has less than 50 or 20, that is, these pairs are considered that the features are erroneously reduced.

4.3 Estimation of Potential Feature Reduction

It is interesting to note that we may reduce 81% of features in Level 3+2 test set while keeping the performance, if we can reduce them properly. In a same way, 87% of features in Level 2+1 set, and 52% of features in Level 1+0 set, may also be reduced. These numbers are given at the situation in which F-measure attains best performance. Here, it is not to say that we are sure to reduce them in future, but to estimate how many features are really effective to distinguish the similarity.

Table 2: Relation of errors and words with a few features. In the table, (h) and (l) shows pairs that are judged higher (lower) by the system. Column of < 50 (< 20) means number of pairs each of which has less than 50 (20) features.

Level	#errs	< 50 fea.	< 20 fea.
Lvl.3+2 (h)	125	76 (61%)	32 (26%)
Lvl.3+2 (l)	220	150 (68%)	60 (27%)
Lvl.2+1 (h)	137	75 (55%)	32 (23%)
Lvl.2+1 (l)	253	135 (53%)	52 (21%)
Lvl.1+0 (h)	149	23 (15%)	4 (3%)
Lvl.1+0 (l)	100	17 (17%)	3 (3%)

Here we have more look at the statistics. The number of initial features on average is 609 in Level 3+2 test set. If we decrease threshold by 0.1, we can reduce 98% of features at the threshold of 0.8, where the performance remains best (0.791). This is a surprising fact for us since only 12 ($\fallingdotseq 609 \times (1 - 0.98)$) features really contribute the performance. Therefore, we estimate that there is a lot to be reduced further in order to purify the features.

5 Conclusion and Future Work

This paper illustrates improvement of lexical distributional similarity by not only associated features but also utilizing unassociated features. The core idea is simple, and is reasonable when we look at machine learning; in many cases we use training instances of not only something positive but something negative to make the distinction of the two sides clearer. Similarly, in our task we use features of not only associated but unassociated to make computation of similarity (or *distance* in semantic space) clearer. We as-

sert in this work that a feature that has similar weight to two given words also plays important role, regardless of how much it is associated to the given words.

Among several future works we need to further explore reduction of features. It is reported by some literature such as Hagiwara et al. (2006) that we can reduce so many features while preserving the same accuracy in distributional similarity calculation. This implies that, some of them are still harmful and are expected to be reduced further.

List of Tools and Resources

1. Chasen, a morphological analyzer, Ver.2.3.3. Matsumoto Lab., Nara Institute of Science and Technology. http://chasen-legacy.sourceforge.jp/

2. IPADIC, a dictionary for morphological analyzer. Ver.2.7.0. Information-Technology Promotion Agency, Japan. http://sourceforge.jp/projects/ipadic/

3. Bunrui Goihyo, a word list by semantic principles, revised and enlarged edition. The National Institute for Japanese Language. http://www.kokken.go.jp /en/publications/bunrui_goihyo/

4. Nihon Keizai Shimbun Newspaper Corpus, years 1990-2004, Nihon Keizai Shimbun, Inc.

References

Dagan, Ido, Lillian Lee, and Fernando Pereira. 1999. Similarity-based Models of Co-occurrence Probabilities. *Machine Learning*, 34(1-3):43–69.

Grefenstette, Gregory. 1994. Exploration in Automatic Thesaurus Discovery. Kluwer Academic Publishers. Norwell, MA.

Hagiwara, Masato, Yasuhiro Ogawa, Katsuhiko Toyama. 2006. Selection of Effective Contextual Information for Automatic Synonym Acquisition. *In Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pp.353–360.

Harris, Zelig S. 1968. Mathematical Structures of Language. Wiley, New Jersey.

Hindle, Donald. 1990. Noun Classification from Predicate-Argument Structures. *In Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, pp.268–275.

Lee, Lillian. 1997. Similarity-Based Approaches to Natural Language Processing. Ph.D. thesis, Harvard University, Cambridge, MA.

Lee, Lillian. 1999. Measures of distributional similarity. *In Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pp. 25–32, College Park, MD.

Lin, Dekang. 1998. Automatic Retrieval and Clustering of Similar Words. *In Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pp.768–774. Montreal.

Lin, Dekang and and Patrick Pantel. 2002. Concept Discovery from Text. *In Proceedings of 19th International Conference on Computational Linguistics*, pp.577–583. Taipei.

Ruge, Gerda. 1992. Experiments of Linguistically-based Term Associations. *Information Processing & Management*, 28(3):317–332.

Shibata, Tomohide and Sadao Kurohashi. 2009. Distributional similarity calculation using very large scale Web corpus. *In Proceedings of Annual Meeting of Association for Natural Language Processing*. pp. 705–708.

Weeds, Julie and David Weir. 2005. Co-occurrence retrieval: A Flexible Framework for Lexical Distributional Similarity. *Computational Linguistics*. 31(4):439–476.

Zhitomirsky-Geffet, Maayan and Ido Dagan. 2009. Bootstrapping Distributional Feature Vector Quality. *Computational Linguistics*, 35(3):435–461.

A Look inside the Distributionally Similar Terms

Kow Kuroda kuroda@nict.go.jp
Jun'ichi Kazama kazama@nict.go.jp
Kentaro Torisawa torisawa@nict.go.jp
National Institute of Information and Communications Technology (NICT)

Abstract

We analyzed the details of a Web-derived distributional data of Japanese nominal terms with two aims. One aim is to examine if distributionally similar terms can be in fact equated with "semantically similar" terms, and if so to what extent. The other is to investigate into what kind of semantic relations constitute (strongly) distributionally similar terms. Our results show that over 85% of the pairs of the terms derived from the highly similar terms turned out to be semantically similar in some way. The ratio of "classmate," synonymous, hypernym-hyponym, and meronymic relations are about 62%, 17%, 8% and 1% of the classified data, respectively.

1 Introduction

The explosion of online text allows us to enjoy a broad variety of large-scale lexical resources constructed from the texts in the Web in an unsupervised fashion. This line of approach was pioneered by researchers such as Hindle (1990), Grefenstette (1993), Lee (1997) and Lin (1998). At the heart of the approach is a crucial working assumption called "distributional hypothesis," as with Harris (1954). We now see an impressive number of applications in natural language processing (NLP) that benefit from lexical resources directly or indirectly derived from this assumption. It seems that most researchers are reasonably satisfied with the results obtained thus far.

Does this mean, however, that the distributional hypothesis was proved to be valid? Not necessarily: while we have a great deal of confirmative results reported in a variety of research areas, but we would rather say that the hypothesis has never been fully "validated" for two reasons. First, it has yet to be tested under the precise definition of "semantic similarity." Second, it has yet to be tested against results obtained at a truly large scale.

One of serious problems is that we have seen no agreement on what "similar terms" mean and should mean. This paper intends to cast light on this unsolved problem through an investigation into the precise nature of lexical resources constructed under the distributional hypothesis. The crucial question to be asked is, Can distributionally similar terms really be equated with semantically similar terms or not? In our investigation, we sought to recognize what types of semantic relations can be found for pairs of terms with high distributional similarity, and see where the equation of distributional similarity with semantic similarity fails. With this concern, this paper tried to factor out as many components of semantic similarity as possible. The effort of factorization resulted in the 18 classes of semantic (un)relatedness to be explained in §2.3.1. Such factorization is a necessary step for a full validation of the hypothesis. To meet the criterion of testing the hypothesis at a very large scale, we analyzed 300,000 pairs of distributionally similar terms. Details of the data we used are given in §2.2.

This paper is organized as follows. In §2, we present our method and data we used. In §3, we present the results and subsequent analysis. In §4, we address a few remaining problems. In §5, we state tentative conclusions.

Proceedings of the Second Workshop on NLP Challenges in the Information Explosion Era (NLPIX 2010), pages 40–49,
Beijing, August 2010

2 Method and Data

2.1 Method

The question we need to address is how many subtypes of semantic relation we can identify in the highly similar terms. We examined the question in the following procedure:

(1) a. Select a set of "base" terms B.
b. Use a similarity measure M to construct a list of n terms $T = [t_{i,1}, t_{i,2}, \ldots, t_{i,j}, \ldots, t_{i,n}]$ where $t_{i,j}$ denotes the j-th most similarity term in T against $b_i \in B$. $P(k)$ are pairs of b_i and $t_{i,k}$, i.e., the k-th most similar term to b_i.
c. Human raters classify a portion Q of the pairs in $P(k)$ with reference to a classification guideline prepared for the task.

Note that the selection of base set B can be independent of the selection of T. Note also that T is indexed by terms in B. To encode this, we write: $T[b_i] = [t_{i,1}, t_{i,2}, \ldots, t_{i,j}, \ldots, t_{i,n}]$.

2.2 Data

For T, we used Kazama's nominal term clustering (Kazama and Torisawa, 2008; Kazama et al., 2009). In this data, base set B for T is one million terms defined by the type counts of dependency relations, which is roughly equated with the "frequencies" of the terms. Each base term in B is associated with up to 500 of the most distributionally similar terms. This defines T.

For M, we used the Jensen-Shannon divergence (JS-divergence) base on the probability distributions derived by an EM-based soft clustering (Kazama and Torisawa, 2008). For convenience, some relevant details of the data construction are described in Appendix A, but in a nutshell, we used dependency relations as distributional information. This makes our method comparable to that used in Hindle (1990). The statistics of the distributional data used were as follows: roughly 920 million types of dependency relations[1)] were automatically acquired from a large-scale Japanese Web-corpus called the *Tsubaki* corpus (Shinzato et al., 2008) which consists of roughly 100 million Japanese pages with six billion sentences. After excluding hapax nouns, we had about 33 million types of nouns (in terms of string) and 27 million types of verbs. These nouns were ranked by type count of the two context triples, i.e., (t, p, v) and (n^*, p^*, t). B was determined by selecting the top one million terms with the most variations of context triples.

2.2.1 Sample of $T[b]$

For illustration, we present examples of the Web-derived distributional similar terms. (2) shows the 10 most distributionally similar terms (i.e., $[t_{1070,1}, t_{1070,2}, \ldots, t_{1070,10}]$ in $T(b_{1070})$) where b_{1070} = "ピアノ" (piano) is the 1070-th term in B. Likewise, (3) shows the 10 most distributionally similar terms $[t_{38555,1}, t_{38555,2}, \ldots, t_{38555,10}]$ in $T(b_{38555})$) where b_{38555} = "チャイコフスキー" (Tchaikovsky) is the 38555-th term in B.

(2) 10 most similar to "ピアノ"

1. エレクトーン (Electone; electronic organ) [-0.322]
2. バイオリン (violin) [-0.357]
3. ヴァイオリン (violin) [-0.358]
4. チェロ (cello) [-0.358]
5. トランペット (trumpet) [-0.377]
6. 三味線 (shamisen) [-0.383]
7. サックス (saxophone) [-0.39]
8. オルガン (organ) [-0.392]
9. クラリネット (clarinet) [-0.394]
10. 二胡 (erh hu) (-0.396)

(3) 10 most similar to "チャイコフスキー"

1. ブラームス (Brahms) [-0.152]
2. シューマン (Schumann) [-0.163]
3. メンデルスゾーン (Mendelssohn) [-0.166]
4. ショスタコーヴィチ (Shostakovich) [-0.178]
5. シベリウス (Sibelius) [-0.18]

[1)] The 920 million types come in two kinds of context triples: 590 million types of (t, p, v) and 320 million types of (t, p^*, n^*), where t denotes the target nominal term, p a postposition, v a verb, and n^* a nominal term that follows t and p^*, i.e., "*t-no*" analogue to the English "*of t*."

6. ハイドン (Haydn) [-0.181]
7. ヘンデル (Händel) [-0.181]
8. ラヴェル (Ravel) [-0.182]
9. シューベルト (Schubert) [-0.187]
10. ベートーヴェン (Beethoven) [-0.19]

For construction of $P(k)$, we had the following settings: i) $k = 1, 2$; and ii) for each k, we selected the 150,000 most frequent terms (out of one million terms) with some filtering specified below. Thus, Q was 300,000 pairs whose base terms are roughly the most frequent 150,000 terms in B with filtering and targets are terms $k = 1$ or $k = 2$.

2.2.2 Filtering of terms in B

For filtering, we excluded the terms of B with one of the following properties: a) they are in an invalid form that could have resulted from parse errors; b) they have regular ending (e.g., -こと , -事 [event], -時 [time or when], -もの [thing or person], -物 [thing], -者 [person]). The reason for the second is two-fold. First, it was desirable to reduce the ratio of the class of "classmates with common morpheme," which is explained in §2.3.2, whose dominance turned out to be evident in the preliminary analysis. Second, the semantic property of the terms in this class is relatively predictable from their morphology. That notwithstanding, this filtering might have had an undesirable impact on our results, at least in terms of representativeness. Despite of this, we decided to place priority on collecting more varieties of classes.

The crucial question is, again, whether distributionally similar terms can really be equated with semantically similar terms. Put differently, what kinds of terms can we find in the sets constructed using distributionally similarity? We can confirm the hypothesis if the most of the term pairs are proved to be semantically similar for most sets of terms constructed based on the distributional hypothesis. To do this, however, we need to clarify what constitutes semantic similarity. We will deal with this prerequisite.

2.3 Classification

2.3.1 Factoring out "semantic similarity"

Building on lexicographic works like Fellbaum (1998) and Murphy (2003), we assume that the following are the four major classes of semantic relation that contribute to semantic similarity between two terms:

(4) a. "synonymic" relation (one can substitute for another on an identity basis). Examples are (*Microsoft*, *MS*).
b. "hypernym-hyponym" relation between two terms (one can substitute for another on un underspecification/abstraction basis). Examples are (*guys*, *players*)
c. "meronymic" (part-whole) relation between two terms (one term can be a substitute for another on metonymic basis). Examples are (*bodies*, *players*) [cf. *All the players have strong bodies*]
d. "classmate" relation between two terms, t_1 and t_2, if and only if (i) they are not synonymous and (ii) there is a concrete enough class such that both t_1 and t_2 are instances (or subclasses).[2)] For example, (*China*, *South Korea*) [cf. *(Both) China and South Korea are countries in East Asia*], (*Ford*, *Toyota*) [cf. *(Both) Ford and Toyota are top-selling automotive companies*] and (*tuna*, *cod*) [cf. *(Both) tuna and cod are types of fish that are eaten in the Europe*] are classmates.

For the substitution, the classmate class behaves somewhat differently. In this case, one term cannot substitute for another for a pair of terms. It is hard to find out the context in which pairs like (*China*, *South Korea*), (*Ford*, *Toyota*) and (*tuna*, *cod*) can substitute one another. On the other hand, substitution is more or less possible in the other three types. For example, a synonymic pair of (*MS*, *Microsoft*) can substitute for one another in contexts like *Many people regularly complain*

[2)] The proper definition of classmates is extremely hard to form. The authors are aware of the incompleteness of their definition, but decided not to be overly meticulous.

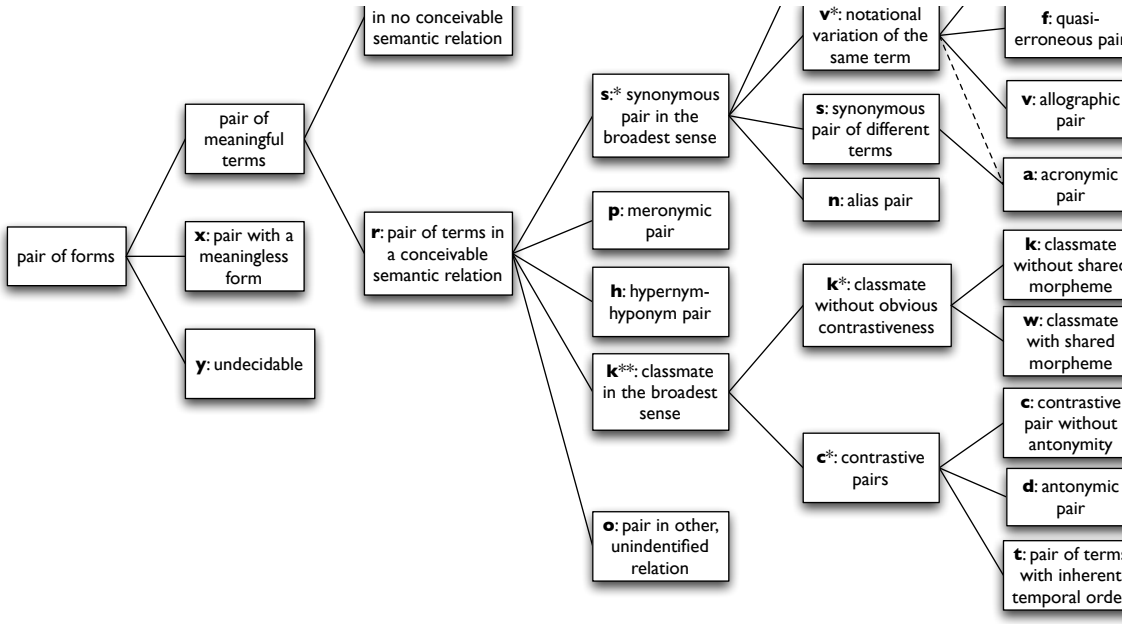

Figure 1: Classification tree for semantic relations used

about products { *i. MS; ii. Microsoft* }. A hypernym-hyponym pair of (*guys*, *players*) can substitute in contexts like *We have dozens of excellent* { *i. guys; ii. players* } *on our team*. A meronymic pair of (*bodies*, *players*) can substitute for each other in contexts like *They had a few of excellent* { *i. bodies; ii. players* } *last year*.

2.3.2 Classification guidelines

The classification guidelines were specified based on a preliminary analysis of 5,000 randomly selected examples. We asked four annotators to perform the task. The guidelines were finalized after several revisions. This revision process resulted in a hierarchy of binary semantic relations as illustrated in Figure 1, which subsumes 18 types as specified in (5). The essential division is made at the fourth level where we have **s*** (pairs of synonyms in the broadest sense) with two subtypes, **p** (pairs of terms in the "part-whole" relation), **h** (pairs of terms in the "hypernym-hyponym" relation), **k**** (pairs of terms in the "classmate" relation), and **o** (pairs of terms in any other relation). Note that this system includes the four major types described in (4). The following are some example pairs of Japanese terms with or without English translations:

(5) **s**: synonymic pairs (subtype of **s***) in which the pair designates the same entity, property, or relation. Examples are: (根元, 株元) [both mean *root*], (サポート会員, 協力会員) [(*supporting member*, *cooperating member*)], (呼び出し元, 親プロセス) [(*invoker of the process*, *parent process*)], (ベンチャービジネス, ベンチャー) [(*venture business*, *venture*)], (相手投手, 相手ピッチャー) [(*opposing hurler*, *opposing pitcher*)], (病歴, 既往歴) [(*medical history*, *anamneses*)],

n: alias pairs (subtype of **s***) in which one term of the pair is the "alias" of the other term. Examples are (*Steve Jobs*, *founder of Apple, Inc.*), (*Barak Obama*, *US President*), (ノグチ, イサム・ノグチ), (侑一郎, うにっ子)

a: acronymic pair (subtype of **s***) in which one term of the pair is the acronym of of the other term. Examples are: (*DEC*, *Digital Equipment*), (*IBM*, *International Business Machine*) (Microsoft 社, MS 社), (難関大, 難関大学), (配置転換, 配転),

v: allographic pairs (subtype of **s***) in which the pair is the pair of two forms of the same term. Examples are: (*convention centre*, *convention center*), (*colour terms*, *color terms*), (乙女ゲーム, 乙女ゲー), (アカスリ, あかすり), (コンピュータシステム, コンピューターシステム), (廻り, 回り), (Ｓｏｌｏ, ｓｏｌｏ), (かっこ, 括弧), (消化器癌, 消化器がん), (坐薬, 座薬), (踏みつけ, 踏み付け)

h: hypernym-hyponym pair in which one term of the pair designates the "class" of the other term. Examples (order is irrelevant) are: (thesaurus, Roget's), (検索ツール, 検索ソフト) [(*search tool*, *search software*)], (失業対策, 雇用対策) [(*unemployment measures*, *employment measures*)], (景況, 雇用情勢) [(*business conditions*, *employment conditions*)], (フェスティバル, 音楽祭) [(*festival*, *music festival*)], (検査薬, 妊娠検査薬) [(*test agent*, *pregnancy test*)], (シンビジウム, 洋ラン) [(*cymbidium*, *orchid*)], (企業ロゴ, ロゴマーク) [(*company logo*, *logo*)], (神秘体験, 臨死体験) [(*mystical experiences*, *near-death experiences*)]

p: meronymic pair in which one term of the pair designates the "part" of the other term. Examples (order is irrelevant) are: (ちきゅう, うみ) [(*earth*, *sea*)], (確約, 了解) [(*affirmation*, *admission*)], (知見, 研究成果) [(*findings*, *research progress*)], (ソーラーサーキット, 外断熱工法) [(*solar circuit system*, *exterior thermal insulation method*)], (プロバンス, 南仏) [(*Provence*, *South France*)],

k: classmates not obviously contrastive without common morpheme (subtype of **k***). Examples are: (自分磨き, 体力作り) [(*self-culture*, *training*)], (所属機関, 部局) [(*sub-organs*, *services*)], (トンパ文字, ヒエログリフ) [(*Dongba alphabets*, *hieroglyphs*)], (*Tom*, *Jerry*)

w: classmates not obviously contrastive with common morpheme (subtype of **k***). Examples are: (ガス設備, 電気設備) [(*gas facilities*, *electric facilities*)], (他社製品, 本製品) [(*products of other company*, *aforementioned products*)], (系列局, 地方局) [(*affiliate station*, *local station*)], (新潟市, 和歌山市) [(*Niigata City*, *Wakayama City*)], (シナイ半島, マレー半島) [(*Sinai Peninsula*, *Malay Peninsula*)],

c: contrastive pairs without antonymity (subtype of **c***). Examples are: (ロマン主義, 自然主義) [(*romanticism*, *naturalism*)], (携帯電話ユーザー, インターネットユーザー) [(*mobile user*, *internet user*)], (海外版, PS2 版), [(*bootleg edition*, PS2 *edition*)]

d: antonymic pairs = contrastive pairs with antonymity (subtype of **c***). Examples are: (接着, 分解) [(*bonding*, *disintegration*)], (砂利道, 舗装路) [(*gravel road*, *pavement*)], (西壁, 東壁) [(*west walls*, *east walls*)], (娘夫婦, 息子夫婦) [(*daughter and son-in-law*, *son and daughter-in-law*)], (外税, 内税) [(*tax-exclusive prices*, *tax-inclusive prices*)], (リアブレーキ, フロントブレーキ) [(*front brake*, *rear brake*)], (タッグマッチ, シングルマッチ) [(*tag-team match*, *solo match*)], (乾拭き, 水拭き) [(*wiping with dry materials*, *wiping with wet materials*)], (ノースリーブ, 長袖) [(*sleeveless*, *long-sleeved*)]

t: pairs with inherent temporal order (subtype of **c***). Examples are: (稲刈り, 田植え) [(*harvesting of rice*, *planting of rice*)], (ご到着日, ご出発日) [(*day of departure*, *day of arrival*)], (進路決定, 進路選択) [(*career decision*, *career selection*)], (居眠り, 夜更かし) [(*catnap*, *stay up*)], (密猟, 密輸) [(*poaching*, *con-*

traband trade)], (投降, 出兵) [(*surrender*, *dispatch of troops*)], (二回生, 三回生) [(*2nd-year student*, *3rd-year student*)]

e: erroneous pairs are pairs in which one term of the pair seems to suffer from character-level input errors, i.e. "mistypes." Examples are: (筋線維, 筋繊維), (発砲スチロール, 発泡スチロール), (太宰府, 大宰府)

f: quasi-erroneous pair is a pair of terms with status somewhat between **v** and **e**. Examples (order is irrelevant) are: (スポイト, スポイド) [(*supoi***t***o*, *supoi***d***o*)], (ゴルフバッグ, ゴルフバック) [(*gorufuba***gg***u*, *gorufuga***kk***u*)], (ビックバン, ビッグバン) [(*bi***gg***u ban*, *bi***kk***u ban*)],

m: misuse pairs in which one term of the pair seems to suffer from "mistake" or "bad memory" of a word (**e** is caused by mistypes but **m** is not). Examples (order is irrelevant) are: (氷漬け, 氷付け), (積み下ろし, 積み降ろし), (開講, 開校), (恋愛観, 恋愛感), (平行, 並行)

o: pairs in other unidentified relation in which the pair is in some semantic relation other than **s***, **k****, **p**, **h**, and **u**. Examples are: (下心, 独占欲) [(*ulterior motives*, *possessive feeling*)], (理論的背景, 基本的概念) [(*theoretical background*, *basic concepts*)], (アレクサンドリア, シラクサ) [(*Alexandria*, *Siracusa*)],

u: unrelated pairs in which the pair is in no promptly conceivable semantic relation. Examples are: (非接触, 高分解能) [(*noncontact*, *high resolution*)], (模倣, 拡大解釈) [(*imitation*, *overinterpretation*)],

x: nonsensical pairs in which either of the pair is not a proper term of Japanese. (but it can be a proper name with very low familiarity). Examples are: (わったん, まる赤), (セルディ, 瀬璃), (チル, エルダ), (ウーナ, 香瑩), (ma, ジョージア)

y: unclassifiable under the allowed time limit.[3] Examples are: (場所網, 無規準ゲーム), (fj, スラド), (反力, 断力),

Note that some relation types are symmetric and others are asymmetric: **a**, **n**, **h**, **p**, and **t** (and **e**, **f**, and **m**, too) are asymmetric types. This means that the order of the pair is relevant, but it was not taken into account during classification. Annotators were asked to ignore the direction of pairs in the classification task. In the finalization, we need to reclassify these to get them in the right order.

2.3.3 Notes on implicational relations

The overall implicational relation in the hierarchy in Figure 1 is the following:

(6) a. **s**, **k****, **p**, **h**, and **o** are supposed to be mutually exclusive, but the distinction is sometimes obscure.[4]
b. **k**** has two subtypes: **k*** and **c***.
c. **k** and **w** are two subtypes **k***.
d. **c**, **d** and **t** three subtypes of **c***.

To resolve the issue of ambiguity, priority was set among the labels so that **e**, $\mathbf{f} < \mathbf{v} < \mathbf{a} < \mathbf{n} < \mathbf{p} < \mathbf{h} < \mathbf{s} < \mathbf{t} < \mathbf{d} < \mathbf{c} < \mathbf{w} < \mathbf{k} < \mathbf{m} < \mathbf{o} < \mathbf{u} < \mathbf{x} < \mathbf{y}$, where the left label is more preferred over the right. This guarantees preservation of the implicational relationship among labels.

2.3.4 Notes on quality of classification

We would like to add a remark on the quality. After a quick overview, we reclassified **o** and **w**, because the first run of the final task ultimately produced a resource of unsatisfactory quality.

Another note on inter-annotator agreement: originally, the classification task was designed and run as a part of a large-scale language resource development. Due to its overwhelming size, we tried to make our development as efficient as possible. In the final phase, we asked

[3] We did not ask annotators to check for unknown terms.

[4] To see this, consider pairs like (*large bowel*, *bowel*), (*small bowel*, *bowel*). Are they instances of **p** or **h**? The difficulty in the distinction between **h** and **p** becomes harder in Japanese due to the lack of plurality marking: cases like (*Mars*, *heavenly body*) (a case of **h**) and (*Mars*, *heavenly bodies*) (a **p** case) cannot be explicitly distinguished. In fact, the Japanese term 天体 can mean both "heavenly body" (singular) and "heavenly bodies" (plural).

Table 1: Distribution of relation types

rank	count	ratio (%)	cum. (%)	class	label
1	108,149	36.04	36.04	classmates without common morpheme	**k**
2	67,089	22.35	58.39	classmates with common morpheme	**w**
3	26,113	8.70	67.09	synonymic pairs	**s**
4	24,599	8.20	75.29	hypernym-hyponym pairs	**h**
5	20,766	6.92	82.21	allographic pairs	**v**
6	18,950	6.31	88.52	pairs in other "unidentified" relation	**o**
7	12,383	4.13	92.65	unrelated pairs	**u**
8	8,092	2.70	95.34	contrastive pairs without antonymity	**c**
9	3,793	1.26	96.61	pairs with inherent temporal order	**t**
10	3,038	1.01	97.62	antonymic pairs	**d**
11	2,995	1.00	98.62	meronymic pairs	**p**
12	1,855	0.62	99.23	acronymic pairs	**a**
13	725	0.24	99.48	alias pairs	**n**
14	715	0.24	99.71	erroneous pairs	**e**
15	397	0.13	99.85	misuse pairs	**m**
16	250	0.08	99.93	nonsensical pairs	**x**
17	180	0.06	99.99	quasi-erroneous pairs	**f**
18	33	0.01	100.00	unclassified	**y**

17 annotators to classify the data with no overlap. Ultimately we obtained results that deserve a detailed report. This history, however, brought us to an undesirable situation: no inter-annotator agreement is calculable because there was no overlap in the task. This is why no inter-rater agreement data is now available.

3 Results

Table 1 summarizes the distribution of relation types with their respective ranks and proportions. The statistics suggests that classes of **e**, **f**, **m**, **x**, and **y** can be ignored without risk.

3.1 Observations

We noticed the following. Firstly, the largest class is the class of classmates, narrowly defined or broadly defined. The narrow definition of the classmates is the conjunction of **k** and **w**, which makes 58.39%. The broader definition of classmates, **k****, is the union of **k**, **w**, **c**, **d** and **t**, which makes 62.10%. This confirms the distributional hypothesis.

The second largest class is the narrowly defined synonymous pairs **s**. This is 8.7% of the total, but the general class of synonymic pairs, **s*** as the union of **s**, **a**, **n**, **v**, **e**, **f**, and **m**, makes 16.91%. This comes next to **h** and **w**. Notice also that the union of **k**** and **s*** makes 79.01%.

The third largest is the class of terms in hypernym-hyponym relations. This is 8.20% of the total. We are not sure if this is large or small.

These results look reasonable and can be seen as validation of the distributional hypothesis. But there is something uncomfortable about the the fourth and fifth largest classes, pairs in "other" relation and "unrelated" pairs, which make 6.31% and 4.13% of the total, respectively. Admittedly, 6.31% are 4.13% are not very large numbers, but it does not guarantee that we can ignore them safely. We need a closer examination of these classes and return to this in §4.

3.2 Note on allography in Japanese

There are some additional notes: the rate of allographic pairs [**v**] (6.92%) is rather high.[5)] We suspect that this ratio is considerably higher than the similar results that are to be expected in other

[5)] Admittedly, 6.92% is not a large number in an absolute value, but it is quite large for the rate of allographic pairs.

languages. In fact, the range of notational variations in Japanese texts is notoriously large. Many researchers in Japanese NLP became to be aware of this, by experience, and claim that this is one of the causes of Japanese NLP being less efficient than NLP in other (typically "segmented") languages. Our result revealed only the allography ratio in nominal terms. It is not clear to what extent this result is applied to the notional variations on predicates, but it is unlikely that predicates have a lesser degree of notational variation than nominals. At the least, informal analysis suggests that the ratio of allography is more frequent and has more severe impacts in predicates than in nominals. So, it is very unlikely that we had a unreasonably high rate of allography in our data.

3.3 Summary of the results

Overall, we can say that the distributional hypothesis was to a great extent positively confirmed to a large extent. Classes of classmates and synonymous pairs are dominant. If the side effects of filtering described in §2.2.2 are ignored, nearly 88% (all but **o**, **u**, **m**, **x**, and **y**) of the pairs in the data turned out to be "semantically similar" in the sense they are classified into one of the regular semantic relations defined in (5). While the status of the inclusion of hypernym-hyponym pairs in classes of semantically similar terms could be controversial, this result cannot be seen as negative.

One aspect somewhat unclear in the results we obtained, however, is that highly similar terms in our data contain such a number of pairs in unidentifiable relation. We will discuss this in more detail in the following section.

4 Discussion

4.1 Limits induced by parameters

Our results have certain limits. We specify those here.

First, our results are based on the case of $k = 1, 2$ for $P(k)$. This may be too small and it is rather likely that we did not acquire results with enough representativeness. For more complete results, we need to compare the present results under larger k, say $k = 4, 8, 16, \ldots$. We did not do this, but we have a comparable result in one of the preliminary studies. In the preparation stage, we classified samples of pairs whose base term is at frequency ranks 13–172, 798–1,422 and 12,673–15,172 where k = 1, 2, 3, …, 9, 10.[6)] Table 2 shows the ratios of relation types for this sample (k = 1, 2, 4, 8, 10).

Table 2: Similarity rank = 1, 2, 4, 8, 10

rank	1	2	4	8	10
v	18.13	10.48	3.92	2.51	1.04
o	17.08	21.24	26.93	28.24	29.56
w	13.65	13.33	14.30	12.19	12.75
s	11.74	9.14	7.05	4.64	4.06
u	11.07	16.48	17.63	20.79	20.87
h	10.50	10.29	11.17	12.96	10.20
k	7.82	8.38	7.84	7.74	8.22
d	2.58	2.00	1.57	1.16	0.85
p	2.00	1.14	1.08	1.35	1.79
c	1.43	1.05	1.27	1.35	1.89
a	1.05	1.33	0.88	0.39	0.57
x	1.05	1.14	1.27	1.64	2.08
t	0.29	0.19	0.20	0.39	0.47
f	0.10	0.10	0.00	0.10	0.09
m	0.00	0.10	0.20	0.00	0.19
#item	1,048	1,050	1,021	1,034	1,059

From Table 2, we notice that: as similarity rank decreases, (i) the ratios of **v**, **s**, **a**, and **d** decrease monotonically, and the ratios of **v** and **s** decrease drastically; (ii) the ratios of **o**, **u**, and **x** increases monotonically, and the ratio of **o** and **u** increases considerably; and while (iii) the ratios of **h**, **k**, **p**, **w**, **m**, and **f** seem to be constant. But it is likely that the ratios of **h**, **k**, **p**, **w**, **m**, and **f** change at larger k, say 128, 256.

Overall, however, this suggests that the difference in similarity rank has the greatest impact on **s*** (recall that **s** and **v** are subtypes of **s***), **o**, and **u**, but not so much on others. Two tendencies can be stated: first, terms at lower similarity ranks become less synonymous. Second,

[6)] The frequency/rank in B was measured in terms of the count of types of dependency relation.

the relationships among terms at lower similarity ranks become more obscure. Both are quite understandable.

There are, however, two caveats concerning the data in Table 2, however. First, the 15 labels used in this preliminary task are a subset of the 18 labels used in the final task. Second, the definitions of some labels are not completely the same even if the same labels are used (this is why we have this great of a ratio of **o** in Table 2. We must admit, therefore, that no direct comparison is possible between the data in Tables 1 and 2.

Second, it is not clear if we made the best choices for clustering algorithm and distributional data. For the issue of algorithm, there are too many clustering algorithms and it is hard to reasonably select candidates for comparison. We do, however, plan to extend our evaluation method to other clustering algorithms. Currently, one of such options is Bayesian clustering. We are planning to perform some comparisons.

For the issue of what kind of distributional information to use, many kinds of distributional data other than dependency relation are available. For example, simple co-occurrences within a "window" are a viable option. With a lack of comparison, however, we cannot tell at the present what will come about if another kind of distributional data was used in the same clustering algorithm.

4.2 Possible overestimation of hypernyms

A closer look suggests that the ratio of hypernym-hyponym pairs was somewhat overestimated. This is due to the algorithm used in our data construction. It was often the case that head nouns were extracted as bare nouns from complex, much longer noun phrases, sometimes due to the extraction algorithms or parse errors. This resulted in accidental removal of modifiers being attached to head nouns in their original uses. We have not yet checked how often this was the case. We are aware that this could have resulted in the overestimation of the ratio of hypernymic relations in our data.

4.3 Remaining issues

As stated, the fourth largest class, roughly 6.31% of the total, is that of the pairs in the "other" unidentified relation [**o**]. In our setting, "other" means that it is in none among the synonymous, classmate, part-whole or hypernym-hyponym relation. A closer look into some examples of **o** suggest that they are pairs of terms with extremely vague association or contrast.

Admittedly, 6.31% is not a large number, but its ratio is comparable with that of the allographic pairs [**v**], 6.92%. We have no explanation why we have this much of an unindenfiable kind of semantic relation distinguished from unrelated pairs [**u**]. All we can say now is that we need further investigation into it.

u is not as large as **o**, but it has a status similar to **o**. We need to know why this much amount of this kind of pairs. A possible answer would be that they are caused by parse errors, directly or indirectly.

5 Conclusion

We analyzed the details of the Japanese nominal terms automatically constructed under the "distributional hypothesis," as in Harris (1954). We had two aims. One aim was to examine to see if what we acquire under the hypothesis is exactly what we expect, i.e., if distributional similarity can be equated with semantic similarity. The other aim was to see what kind of semantic relations comprise a class of distributionally similar terms.

For the first aim, we obtained a positive result: nearly 88% of the pairs in the data turned out to be semantically similar under the 18 criteria defined in (5), which include hypernym-hyponym, meronymic, contrastive, and synonymic relations. Though some term pairs we evaluated were among none of these relations, the ratio of **o** and **u** in sum is about 14% and within the acceptable range.

For the second aim, our result revealed that the ratio of the classmates, synonymous, relation, hypernym-hyponym, and meronymic relations are respectively about 62%, 17%, 8% and 1% of the classified data.

Overall, these results suggest that automatic acquisition of terms under the distributional hypothesis give us reasonable results.

A Clustering of one million nominals

This appendix provides some details on how the clustering of one million nominal terms was performed.

To determine the similarity metric of a pair of nominal terms (t_1, t_2), Kazama et al. (2009) used the Jensen-Shannon divergence (JS-divergence) $D_{JS}(p||q) = \frac{1}{2}D(p||M) + \frac{1}{2}D(q||M)$, where p and q are probability distributions, and $D = \sum_i p(i)\log\frac{p(i)}{q(i)}$ (Kullback-Leibler divergence, or KL-divergence) of p and q, and $M = \frac{1}{2}(p+q)$. We obtained p and q in the following way.

Instead of using raw distribution, Kazama et al. (2009) applied smoothing using EM algorithm (Rooth et al., 1999; Torisawa, 2001). In Torisawa's model (2001), the probability of the occurrence of the dependency relation $\langle v, r, n\rangle$ is defined as:

$$P(\langle v,r,t\rangle) =_{\text{def}} \sum_{a\in A} P(\langle v,r\rangle|a)\,P(t|a)\,P(a),$$

where a denotes a hidden class of $\langle v, r\rangle$ and term t. In this equation, the probabilities $P(\langle v,r\rangle|a)$, $P(t|a)$, and $P(a)$ cannot be calculated directly because class a is not observed in a given dependency data. The EM-based clustering method estimates these probabilities using a given corpus. In the E-step, the probability $P(a|\langle v,r\rangle)$ is calculated. In the M-step, the probabilities $P(\langle v,r\rangle|a)$, $P(t|a)$, and $P(a)$ are updated until the likelihood is improved using the results of the E-step. From the results of this EM-based clustering method, we can obtain the probabilities $P(\langle v,r\rangle|a)$, $P(t|a)$, and $P(a)$ for each $\langle v,r\rangle$, t, and a. Then, $P(a|t)$ is calculated by the following equation:

$$P(a|t) = \frac{P(t|a)\,P(a)}{\sum_{a\in A} P(t|a)\,P(a)}.$$

The distributional similarity between t_1 and t_2 was calculated by the JS divergence between $P(a|t_1)$ and $P(a|t_2)$.

References

Fellbaum, C., ed. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Grefenstette, G. 1993. Automatic thesaurus generation from raw text using knowledge-poor techniques. In *In Making Sense of Words: The 9th Annual Conference of the UW Centre for the New OED and Text Research*.

Harris, Z. S. 1954. Distributional structure. *Word*, 10(2-3):146–162. Reprinted in Fodor, J. A and Katz, J. J. (eds.), *Readings in the Philosophy of Language*, pp. 33–49. Englewood Cliffs, NJ: Prentice-Hall.

Hindle, D. 1990. Noun classification from predicate-argument structures. In *Proceedings of ACL-90*, pp. 268–275, Pittsburgh, PA.

Kazama, J. and K. Torisawa. 2008. Inducing gazetteers for named entity recognition by large-scale clustering of dependency relations. In *Proceedings of ACL-2008: HLT*, pp. 407–415.

Kazama, J., S. De Saeger, K. Torisawa, and M. Murata. 2009. Generating a large-scale analogy list using a probabilistic clustering based on noun-verb dependency profiles. In *Proceedings of the 15th Annual Meeting of the Association for Natural Language Processing*. [in Japanese].

Lee, L. 1997. *Similarity-Based Approaches to Natural Language Processing*. Unpublished Ph.D. thesis, Harvard University.

Lin, D. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of COLING/ACL-98, Montreal, Canda*, pages 768–774.

Murphy, M. L. 2003. *Semantic Relations and the Lexicon*. Cambridge University Press, Cambridge, UK.

Rooth, M., S. Riezler, D. Presher, G. Carroll, and F. Beil. 1999. Inducing a semantically annotated lexicon via em-based clustering. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pp. 104–111.

Shinzato, K., T. Shibata, D. Kawahara, C. Hashimoto, and S. Kurohashi. 2008. *TSUBAKI*: An open search engine infrastructure for developing new information access. In *Proceedings of IJCNLP 2008*.

Torisawa, K. 2001. An unsupervised method for canonicalization of Japanese postpositions. In *Proceedings of the 6th Natural Language Processing Pacific Rim Symposium (NLPRS)*, pp. 211–218.

Utilizing Citations of Foreign Words in Corpus-Based Dictionary Generation

Reinhard Rapp
University of Tarragona
GRLMC
reinhardrapp@gmx.de

Michael Zock
Laboratoire d'Informatique Fondamentale
CNRS Marseille
Michael.Zock@lif.univ-mrs.fr

Abstract

Previous work concerned with the identification of word translations from text collections has been either based on parallel or on comparable corpora of the respective languages. In the case of comparable corpora basic dictionaries have been necessary to form a bridge between the languages under consideration. We present here a novel approach to identify word translations from a single monolingual corpus without necessarily requiring dictionaries, although, as will be shown, a dictionary can still be useful for improving the results. Our approach is based on the observation that for various reasons monolingual corpora typically contain many foreign words (for example citations). Relying on standard newsticker texts, we will show that their co-occurrence-based associations can be successfully used to identify word translations.

1 Introduction

The web has popularized information access. As a consequence, the information put on the web evolved, expanding from mainly technical documents in one language (English) to topics concerning nearly any aspect of life in many languages. For this reason it cannot be expected anymore that all web users speak English. Yet users speaking only one of the minority languages will be penalized, finding only a small fraction of web content accessible. Hence they can make only very limited use of what is available. In order to increase information access independently of the users' mother tongue, automatic translation is desirable.

Recognizing this need, Google, among others, is providing free machine translation services for any pair of currently 50 languages.[1] However, with 6800 living languages, of which 600 also use a written form, offering comprehensive translation services remains a challenge.

The statistical approach to machine translation (SMT), as adopted by Google, relies on parallel corpora, i.e. large collections of existing translations. But it is a daunting task trying to acquire parallel corpora for all possible language pairs. Therefore, it appears that for some languages Google has combined SMT with an interlingua approach. This allows optimal exploitation of languages for which parallel corpora are easily obtained. These languages are then used as pivots. Note that in phrase-based SMT an interlingua approach may operate at the level of the phrase table, which facilitates matters while speeding up the process. At the downside it must be noted that a phrase table derived via a pivot language is generally of lower quality than a phrase table directly compiled from parallel texts (provided the corpus size is similar). Hence, just as for other interlingua approaches, translation quality is severely compromised.

An alternative approach that has been suggested is to try to generate the required dictionaries from other sources than parallel corpora. Bear in mind that statistical machine translation requires a *language model* and a *translation model*. To generate the language model only monolingual corpora of the target language are required which, for example, can be acquired from the web. If only few such documents exist, one may well conclude that there is probably no real need

[1] http://www.google.de/language_tools?hl=de as of April 22, 2010.

Proceedings of the Second Workshop on NLP Challenges in the Information Explosion Era (NLPIX 2010), pages 50–59,
Beijing, August 2010

for translation involving this particular language. So the main bottleneck are the parallel corpora required to generate a translation model. But the purpose of the translation model is in essence the creation of a bilingual dictionary, be it a dictionary of individual words or a dictionary of phrases. For this reason, if we can find other ways to generate dictionaries for lesser used languages, this will be beneficial not only for the users of these languages but also for the solution of the overall problem of machine translation.

In other words, an important challenge is the generation of dictionaries. Since comparable corpora are a far more common resource than parallel corpora, attempts to exploit them for dictionary construction have received considerable attention recently.[2]

One approach is to mine parallel sentences from comparable corpora. Roughly speaking, this can be done by automatically translating a corpus from one language (source language) to another (target language), and then searching in a large corpus of the target language for sentences similar to the translations. The advantage of this procedure is that the sentences retrieved this way are correct sentences as they were produced by humans, whereas the sentences translated by a machine tend to be garbled and of lower quality. However, the big problem with this approach is to ensure that the retrieved sentence pairs are indeed translations of each other. While there is no perfect solution to this problem, several studies have shown that such data can be useful for building or supplementing translation models in SMT (see e. g. Munteanu & Marcu, 2005; Wu & Fung, 2005).

Another approach for exploiting comparable corpora in dictionary generation is based on the observation that word co-occurrence patterns between languages tend to be similar (Fung & McKeown, 1997; Rapp, 1995; Chiao et al., 2004). If, for example, two words X and Y co-occur more often than expected by chance in a corpus of language A, then their translated equivalents should also co-occur more frequently than expected in a corpus of language B. A great number of variants of this approach has been proposed, e.g. emphasizing aspects of corpus selection or expanding it to collocations or short phrases (Babych et al., 2007).

What is common to these studies is that they consider the source and the target language as two distinct semantic spaces, without any links at the beginning. Therefore, in order to connect the two, a base dictionary is required, and the purpose of the system is to expand this base dictionary. Building a dictionary from scratch is not possible this way or at least computationally unfeasible (see Rapp, 1995).

Whether the assumption of two completely distinct semantic spaces is realistic remains an open issue. Are separate lexical networks really a reasonable model for the processing of different languages by people?

One could say this is a plausible model, assuming a person lived for some years in one country, and then for some more years in another country, assuming further that this person never looked at a dictionary or another multilingual document and never communicated with a person mixing both languages.

It is known that this can work. The reason is probably the following: Many words of the basic dictionary assumed above correspond to items of the physical world. These items generally have names in natural languages which can serve as mediators. That the extrapolation to more abstract notions is possible has been claimed by Rapp (1999).

Still, although persons proceeding this way can easily understand and, after some years, even think in each of the two languages, experience shows that they tend to have some difficulties when making translations, especially literal translations.

So, although the above scenario is possible, we do not think that it is a typical one for our modern times. There are certainly good reasons why there are so many language courses, and why there is such an abundance of dictionaries. It is a matter of commonsense that the person trying to acquire a new language will look at a multilingual dictionary. He or she will also communicate with other persons who mix languages, for example, relatives, other people from the com-

[2] There is also the approach of identifying orthographically similar words (Koehn & Knight, 2002) which does not even require a corpus as simple word lists will suffice. However, this approach is promising only for closely related languages but appears to have limited scope otherwise. For this reason we will not further discuss it here.

munity of foreigners coming from the same country, teachers in language classes, etc. In many cases there will also be multilingual documents around: leaflets, explanations in a museum, or signs in a public area (e.g. airport).

Hence the spoken and written "corpus" (input) on which such a person's language acquisition process is based is not solely monolingual. While the corpus may be mainly monolingual, it surely will contain some multilingual elements.

If we agree on this, our next step could be to acquire transcripts of language teaching classes with bilingual teachers and try to exploit these for dictionary generation. Since obtaining such transcripts in large enough quantities should be much more difficult than obtaining parallel corpora, this approach will probably not solve the data acquisition bottleneck which is the practical problem we were about to solve in the first place.

The current study is therefore based on newsticker texts which is a text type very similar to standard newspaper texts. At least for some languages it is available in large quantities. However, this type of text is probably not ideally suited for our purpose. Surprisingly, the reason is that newsticker and newspaper texts tend to be very well edited. This means that the author will typically avoid foreign words, and if ever some remain the respective passages are likely be rephrased in order to make sure that the text uses familiar vocabulary, easily understandable by the readers. However, this is problematic for our approach which is based on the occurrences of foreign words in a monolingual text. So this is one of the rare cases where noisy corpora should yield better results than perfectly clean data.

On the other hand, as this study suggests a (to our knowledge) novel approach, we consider it important to use a corpus that is generally known and available, and which has not been compiled with this particular purpose in mind. Only this way our results can convincingly give an idea concerning the baseline performance of the suggested algorithm. At this stage we consider this more important than optimizing results by compiling corpora specifically suited for the purpose, even though this will be a logical next step.

2 Approach and Language Resources

Starting from the observation that monolingual dictionaries typically include a large number of foreign words, we consider the most significant co-occurrences of them as potential translation candidates. This implies that the underlying corpus corresponds to the target language, and that it can be utilized for any source language for which it contains a sufficient number of word citations. As this paper is written in English, we chose an English corpus as this should make judging our results convenient for most readers. However, being the world's most widely spoken language, English tends to be rather self-contained in comparison to other languages, which may use foreign words more frequently. In particular, as a side effect of globalization, the use of English terminology is popular in many other languages. Therefore, in order to identify, for example, German–English word translations, it is better to look at occurrences of English words in a German corpus rather than at occurrences of German words in an English corpus.[3]

Nevertheless, the corpus we use here is the latest release of the English Gigaword Corpus (Fourth Edition) provided by the Linguistic Data Consortium (Parker et al., 2009). It consists of newswire texts of the time between 1995 and 2008 from the following news agencies:

- Agence France-Presse, English Service
- Associated Press Worldstream, English Service
- Central News Agency of Taiwan, English Service
- Los Angeles Times/Washington Post Newswire Service
- New York Times Newswire Service
- Xinhua News Agency, English Service

Altogether, the corpus comprises about 3 billion words. Since we are not interested in the translation of function words, and in order to reduce the computational load, we removed all function words that were included in a stop word list for English comprising about 200 items. The stop words had been manually selected from a corpus-derived list of high frequency words.

In the resulting corpus associations between words need to be identified, something that is usually done on the basis of co-occurrences. In

[3] Note that the results of both directions may be combined. This is something we leave for future work.

order to count the co-occurrences between pairs of words, a text window comprising the ten words preceding and following a given foreign word is considered. On the resulting co-occurrence counts a standard association metric like the log-likelihood ratio (Dunning, 1993) is applied.

Note that the above mentioned window size of ±10 words from the given word relates to the preprocessed corpus from which function words have already been removed. Since in English roughly every second word tends to be a function word, the effective window size is about ±20 words. This window size is somewhat larger than what we typically find in other studies. However, the reason for this is quite obvious: As citations of foreign words are rare, we have a severe problem of data sparseness, and by looking at a relatively large window we try to somewhat compensate for this.[4]

Despite its simplicity, this procedure of computing associations to foreign words already works well for identifying word translations. We simply assume that the strongest association is the best translation. We used this approach for words from three languages: French, German, and Spanish. The results are presented in the next section. In order to measure the quality of our results, for all source words of a language we counted the number of times where the expected English target word obtained the highest association score.

As our gold standard for evaluation we used an existing list of translations as described in Rapp & Zock (2010), i.e. a resource that had not been compiled with the current application in mind. The data consists of 1079 word equations in three languages: English, French, and German. It has been extracted from the respective editions of the Collins GEM dictionaries, whereby when looking up a word only the first entry in the list of possible translations was taken into account. As in the current study we are also interested in Spanish, we manually looked up the main translations at the leo.dict.org website[5] and added another column to this resource. Table 1 shows a few sample entries of the resulting list of *word equations* which were used for evaluating our approach.

We should mention that the term *word equation* is a bit problematic, as most words tend to be ambiguous, and ambiguities tend to vary with language. For this reason, we should, at least in principle, disambiguate all words in our corpus and map them to unambiguous concepts. Next we should use a gold standard using such concepts rather than words. Unfortunately, the current state of the art does not allow doing this with sufficient accuracy. Anyhow, addressing this problem is well beyond the scope of this paper.

SOURCE LANGUAGES			TARGET LANG.
FRENCH	GERMAN	SPANISH	ENGLISH
britannique	britisch	británico	British
Pâques	Ostern	Pascua	Easter
capable	fähig	capaz	able
accent	Akzent	acento	accent
accident	Unfall	accidente	accident
accordéon	Akkordeon	acordeón	accordion
acide	Säure	ácido	acid
gland	Eichel	bellota	acorn
action	Handlung	acción	action
avantage	Vorteil	ventaja	advantage

Table 1. Some sample entries from the gold standard of word equations.

So far, for identifying the translations of the 1079 French words, we assumed the following approach: We first computed their associations and then conducted an evaluation by checking for how many words the top association was identical to the English translation found in the gold standard. The same approach was also used for the other languages, namely German and French. Hence, the three source languages were treated completely independently of each other.

[4] In preliminary experiments we also experimented with other window sizes. However, as we noticed that changes within a reasonable range of e.g. 5 to 20 words have only little effect, we do not consider them here.

[5] This is a manually edited high quality online dictionary. Although it can be used for free, in our view for many purposes is as good as or even better than conventional printed dictionaries.

However, there are several problems with this approach, in particular:

a) Several correct translations
b) Data sparseness
c) Homograph trap

Let us discuss these issues point by point.

a) Several correct translations

Suppose we tried to identify the translation of the German word *Straße* and our gold standard listed *street* as the correct translation. If, however, our system produced *road* this would be considered just as much of an error as if it had produced a very remote word such as *volcano*. Hence, considering only a single word as being correct, which is the consequence of using as gold standard the resource exemplified in Table 1, implies that performance figures are artificially low, giving us only the lower bound of the true performance.

Despite this shortcoming, we will nevertheless do so for the following reasons: 1) This is a pilot study presenting a new approach. For this reason, clarity has priority over performance. 2) The number of translations listed in a dictionary typically depends on the size of the resource. Hence, there is no absolute difference between correct and incorrect translations. Rather, we need to set a threshold somewhere, and truncating after the first word listed is arguably the clearest and simplest way of doing so. 3) This is the main reason. We want to extend our approach to the multilingual case by (simultaneously) looking at several source languages. Given the fact that each language tends to have its own (i.e. idiosyncratic) ambiguities, we are already satisfied if words from the various source languages have the same main translation. That all possible translations are identical is very unlikely.

b) Data sparseness

What will happen if a source word does not occur at all in the corpus, or only once or twice? We mentioned already that an appropriate choice of text genre, corpus size, and window size can somewhat reduce the problem of data sparseness. We also mentioned that by reversing source and target languages we can look at the problem from two perspectives, which may yield further improvement. Nevertheless, these suggestions are limited in scope. Hence, given the nature of our approach, data sparseness will remain the core problem.

Fortunately, there is another possibility which is more promising than the ones mentioned above, provided that we manage to solve the ambiguity problem. The solution consists in considering several source languages concurrently. Suppose that rather than starting from scratch we use existing dictionaries for various languages.[6] In this case we can easily generate word equations such as the ones shown in Table 1. We do this by considering as a single item all words appearing in a given row (excluding the target language word), and by computing the associations to this aggregated artificial unit. (This is a simplified proposal. We shall see later how to improve it.) If, for example, we have 10 source languages, then it does not matter that 8 source words do not occur in the corpus, as long as the other two are well represented.

c) The homograph trap

By this we mean that a word form from the source language also exists in the target language, but with a different meaning. For example, let us assume that we wanted to translate the word *can* (house) from Catalan to English. Suppose further that we are lucky and have ten Catalan citations with this word in our English corpus. But this will not help us because the word *can* happens to also belong to English, meaning something completely different. Moreover, *can* is a high frequency word, occurring millions of times in a large corpus. Of course, if we had a perfect word sense disambiguator, we could separate the Catalan and the English occurrences of *can*, thereby solving the problem.[7] Unfortunately, existing tools are not powerful enough to do the job. What is worse, such collisions are not

[6] Which, for example, by using open source tools such as Moses and Giza++ (see www.statmt.org) can be easily generated from parallel corpora, e.g. from the Europarl corpus (Koehn, 2005) or the JRC Acquis corpus (Steinberger et al., 2006).

[7] If we assume that foreign words typically occur in clusters, we could also use language identification software.

uncommon between languages using the same script. So what can we do? Our suggestion is exactly the same as above for the problem of data sparseness, i.e. to look at several source languages in parallel.

But it is clear that collapsing all source words into a single item does not work. If only one of them happens to be also a common word in the target language, it is very likely that its co-occurrences will override the co-occurrences of the foreign words we are interested in. So there is little chance to come up with a correct result.

We propose a relatively simple solution to this problem, which possibly may well be novel in this context. Let us develop the idea.

In preliminary experiments we have tried several possibilities. Collapsing the source words would be equivalent to adding the respective co-occurrence vectors. This is apparently not adequate because, as mentioned above, the vector of a very frequent word would dominate all others. An alternative would be to sum up the association vectors. By the term association vector we mean the co-occurrence vectors after application of an association measure (in our case the log-likelihood ratio). It turns out that this somewhat reduces the problem without solving it entirely. Another possibility would be vector multiplication. Multiplication is considerably better than addition as a property of multiplication is that moderate but coinciding support for a particular target word from several source words leads to a higher product than strong support by only a few. This is a highly desirable property as it helps us avoiding the homograph trap, and because all values are subject to considerable sampling errors.

Unfortunately, there is yet another problem. Our association measure of choice, namely the log-likelihood ratio, as typical for ratios, has a skewed value characteristic. Since otherwise our previous experiences with the log-likelihood ratio are very good,[8] and since it seems reasonably well suited for sparse data (Dunning, 1993), we suggest to multiply log-likelihood ranks rather than log-likelihood scores. This proposal is based on the observation (Dunning, 1993) that rankings of association strengths as produced by the log-likelihood ratio tend to be highly accurate even at higher ranks. Let us call this procedure the *product-of-rank* algorithm

This algorithm works as follows: Starting from a vocabulary of target language words (which are the translation candidates), for each of these words an association vector is computed. Next, for each association vector the ranks of all words in the source language word tuple under consideration are determined. Hence, if we have three languages (e.g. English, French and German) we would get three values. These values are multiplied with each other, and finally all target language words are sorted according to the resulting products. As small ranks stand for strong associations, the word obtaining the smallest value is considered to be the translation of the source language tuple. This algorithm turned out to lead to highly plausible rankings and to be robust with regard to sampling errors.[9] It is also quite effective in eliminating the homograph problem.

3 Experimental Results and Evaluation

Let us first try to see whether the basic assumption underlying our approach is sound, namely that we will find a sufficient number of foreign words in our corpus. To check this claim, we have listed in Table 2 for each of the four languages the number of words from the gold standard falling into particular frequency categories. For example, the value of 70 in the field belonging to the row *6-10* and the column *Spanish* means that out of the 1079 Spanish words in our gold standard 70 have a corpus frequency between 6 and 10 in the 4th edition of the English Gigaword Corpus. Apparently, words with zero occurrences or with a very low corpus frequency are problematic because of data sparseness. Yet words with very high frequencies are not less problematic, as they may turn out to have homographs in the target language. As there is no generally accepted definition of what the vocabulary of a given language is, we cannot give precise figures concerning the number of homographs in our gold standard for each language pair. Never-

[8] To the best of our knowledge no other measure could consistently beat it over a wide range of NLP applications.

[9] A further improvement is possible by giving words with identical association strengths not arbitrary ranking positions within this group, but an average rank which is to be assigned to all of them.

theless, we believe that Table 2 gives a fair impression. By taking a look at the high frequency source language words one can see that the pair French–English has the greatest number of homographs, followed by German–English, and finally Spanish–English.

Corpus frequency	Source languages			Targ. lang.
	German	French	Spanish	English
0	449	329	317	0
1	64	85	43	0
2	26	52	25	0
3	24	39	23	0
4	17	34	27	0
5	7	26	15	0
6-10	32	71	70	0
11-20	50	59	86	0
21-50	63	52	129	0
51-100	50	37	95	1
101-200	52	10	75	3
201-500	50	25	74	6
501-1000	43	18	31	19
1001-10000	100	71	37	245
above 10000	52	171	32	805

Table 2: Corpus frequencies of the words occurring in the gold standard.

As to be expected, the corpus frequencies of the language of the corpus, namely English, are orders of magnitude higher than those of the other languages. But the table also gives a good idea concerning the presence of French, German, and Spanish word citations in written English. However, we should not be misled by the overwhelming presence of French words in the high frequency ranges, as this mainly reflects the amount of homography. Although pronunciation rules are very different between English and French, spelling tends to be similar, which is why there are lots of homographs. In contrast, Spanish and German usually use different spelling even for words having the same historical roots, which is why homography is far less common.[10]

[10] As an example for such spelling conversions, let's mention that the grapheme *c* in English is almost consistently replaced by *k* in German, e.g. *class* → *Klasse* and *clear* → *klar*.

From the figures of Table 2 one may conclude that identifying word translations from a monolingual corpus is not easy because of data sparseness. Nevertheless it seems possible, at least to some extent. Let us therefore take a look at some results.

In our experimental work we first identified word translations for stimulus words from a single source language, then for stimulus words from two source languages, and finally for stimulus words from three source languages.

a) One source language

We started by conducting separate runs for each of the three source languages (French, German, Spanish) and determined the number of times the algorithm was able to come up with the expected English translation as the top ranked association for the 3 * 1079 source words. Note, however, that hereby we did not consider the full range of possible target words present in the English Gigaword corpus as this would include many foreign words. Instead, we restricted the number of target words to the 1079 English words present in the gold standard.

The respective figures are 163 (15.1%) for French, 85 (7.9%) for German, and 97 (9.0%) for Spanish. As can be seen, French clearly performed best, which confirms previous studies that the lexical agreement between French and English is surprisingly high. Nevertheless, on average, only 10.7% of the translations were identified correctly, which does not look very good. However, remember that these figures can be considered as a lower bound as we do not take alternative translations into account and as the underlying corpus has not been prepared specifically for this purpose. Note also that the *product-of-ranks* algorithm has no effect in the case when only a single source language is considered. (If there is only one value, no multiplication takes place.)

b) Two source languages

Our next step was to combine pairs of source languages. There are three possible pairs, namely French–German, French–Spanish, and German–Spanish. Their respective performance figures are as follows: 217 (21.0%), 225 (20.9%), and 145 (13.4%). Computing the mean of these re-

sults yields an average of 18.4%, which is a nice improvement over the initial 10.7% which we had for single source languages. This lends support to our hypothesis that the product-of-ranks algorithm works effectively in this context.

c) Three source languages

Finally, all three source languages were combined, resulting in the correct translation of 248 of the altogether 1079 test items, which corresponds to a performance of 23.0%. This further improvement is consistent with our hypothesis that performance should increase when more source languages are considered.

Let us take a closer look at these performance gains. At the beginning we increased the number of source languages by 100% (from 1 to 2), yielding a relative performance increase of 72% (the absolute performance improved from 10.7% to 18.4%). Next we increased the number of source languages by 50% (from 2 to 3) which yielded a relative performance increase of 25% (absolute performance had improved from 18.4% to 23%). This means that the behavior is worse than linear, as in the linear case we should have obtained a further improvement of 72%/2 = 36%. But of course when combining statistics in NLP, hardly ever a linear behavior can be observed, and the above findings seem satisfactory. Nevertheless they should be supported by looking at further languages, see Section 4.[11]

For the case of looking at three source languages in parallel, let us provide data concerning the rank distribution of the expected translations (see the middle column of Table 3). Overall, in 357 of the 1079 cases (33.9%) the expected translation ranks among the top five, and in 392 cases (36.3%) it is among the top ten associations. These results are based on a window size of ±10 words when counting the co-occurrence frequencies. To give an idea that the procedure is robust in this respect, we provide analogous values for a window size of ±20 words in the third column of Table 3. As can be seen, apart from the usual statistical fluctuations the difference is hardly noticeable.

Rank	Number of items with the respective rank	
	window size ±10	window size ±20
rank could not be computed (all source words unknown)	11	10
1	248	247
2	55	51
3	32	36
4	15	19
5	7	8
6	16	8
7	7	6
8	3	5
9	3	5
10	6	4
above 10	676	680

Table 3: Ranks of the expected translations when all three source languages are combined.

EXAMPLE 1

Given word French: tablier [7]
Given word German: Schürze [0]
Given word Spanish: delantal [4]

Expected translation into English according to the gold standard: apron [3059]

Top 5 translations as computed:

1	apron	[3059]
2	sausage	[9954]
3	sauce	[49139]
4	appetite	[24682]
5	mustard	[13477]

Table 4: Sample results.

[11] Another important question, which we have not dealt with yet, is to what extend the observed gain in performance when increasing the number of source languages is a side effect of a higher likelihood that at least one of the source words happens to be identical to the target word (with the same or a similar meaning). In such cases (which might be common when considering related languages), predicting the correct translation is rather easy.

EXAMPLE 2

Given word French: carton [2671]
Given word German: Karton [22]
Given word Spanish: cartón [0]

Expected translation into English
according to gold standard: cardboard [13714]

Top 5 translations as computed:

1	cardboard	[13714]
2	cigarette	[54583]
3	fold	[43682]
4	milk	[85426]
5	egg	[42948]

Table 5: Sample results.

Having looked at the quantitative results, some sample output may also be of interest. For this purpose, Tables 4 and 5 show sample results for triplets of source language words. Hereby, the numbers in square brackets refer to the corpus frequencies of the respective words in the English Gigaword Corpus.

4 Summary and Future Work

In this paper we made an attempt to solve the problem of identifying word translations on the basis of a single monolingual corpus where the same corpus is supposed to be used for several language pairs. The basic idea underlying our work is to look at citations of foreign words, to compute their co-occurrence-based associations, and to consider these as translations of the respective words.

We pointed out some difficulties with this approach, namely the problem of data sparseness and the homograph trap, but were able to suggest and implement at least partial solutions. Using the product-of-ranks algorithm, our main suggestion was to look at several source languages in parallel, which at least in theory has the potential to solve the experienced problems.

We did not have very high expectations when starting this work and were positively surprised by the resulting performance of up to 25% correctly predicted test items. As pointed out, in order to avoid raising unjustified expectations, we presented somewhat conservative figures which should leave room for improvements.

Obvious extensions of the current work are to increase the number of considered languages and to also use other large monolingual corpora. For example, we could use the web corpora provided by the web-as-a-corpus (WaCky) initiative (Baroni et al., 2009). A few such corpora have already been made available recently, and as they are based on a largely automatic acquisition procedure there are probably more to come. This reflects a tendency towards extremely large corpora. Processing in the current framework turns out to be unproblematic if sparse matrices are used, as foreign word occurrences are implicitly of low frequency.

Although web corpora should be very noisy in comparison to the carefully edited newsticker texts used here, the interesting thing is that according to the hypothesis formulated in the introduction the current approach seems to provide one of the rare occasions where noisy data is better than perfectly clean data, and we hope that future work will prove this prediction correct.

Another possibility for future work is to look at second rather than first order associations, i.e. to consider those words as potential translations of a given foreign word which show similar context words. This might be promising in so far as the sparse data problem is less salient in this case.

Finally, let us come back to our speculative question from the introduction whether or not people speaking different languages have separate lexico-semantic networks in their mind. Aparently our experiments did not provide evidence for either assumption. But the most straightforward assumption would probably be that our mind does not attach language labels to the words we perceive, and simply treats them all equally. At the lexical level, our mind's unknown inner workings may be in effect analogous to clustering words according to their observed co-occurrence patterns. The likely result is that in some cases there will be many interconnections between clusters, and in other cases few. Depending on the language environment experienced by a person, we cannot rule out that some of the larger clusters might exactly correspond to languages. But what the current research does

tell us is that there can be a multitude of statistically significant co-occurrences even at non-obvious places. So what we possibly should rule out is that, even across languages, there are separate clusters without any interconnections.

Acknowledgments

Part of this research was supported by a Marie Curie Intra European Fellowship within the 7th European Community Framework Programme. We thank the Linguistic Data Consortium for making available the English Gigaword Corpus, and Lourdes Callau, Maria Dolores Jimenez Lopez, and Lilica Voicu for their support in acquiring it.

References

Babych, Bogdan; Sharoff, Serge; Hartley, Anthony; Mudraya, Olga (2007). Assisting Translators in Indirect Lexical Transfer. *Proceedings of the 45th International Conference of the Association for Computational Linguistics ACL 2007, Prague*, 136–143.

Baroni, Marco; Bernardini, Silvia; Ferraresi, Adriano, Zanchetta, Eros (2009). The WaCky Wide Web: A collection of very large linguistically processed Web-crawled corpora. *Journal of Language Resources and Evaluation* 43 (3): 209–226.

Chiao, Yun-Chuang; Sta, Jean-David; Zweigenbaum, Pierre (2004). A novel approach to improve word translations extraction from non-parallel, comparable corpora. In: *Proceedings of the International Joint Conference on Natural Language Processing*, Hainan, China. AFNLP.

Dunning, Ted (1993). Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1), 61–74.

Fung, P.; McKeown, K. (1997). Finding terminology translations from non-parallel corpora. *Proceedings of the 5th Annual Workshop on Very Large Corpora,* Hong Kong, 192–202.

Fung, P.; Yee, L. Y. (1998). An IR approach for translating new words from nonparallel, comparable texts. In: *Proceedings of COLING-ACL 1998,* Montreal, Vol. 1, 414–420.

Koehn, Philipp; Knight, Kevin (2002). Learning a translation lexicon from monolingual corpora. In: Unsupervised Lexical Acquisition. Proceeding of the ACL SIGLEX Workshop, 9–16.

Koehn, Philipp (2005). Europarl: A parallel corpus for statistical machine translation. *Proceedings of MT Summit*, Phuket, Thailand, 79–86.

Munteanu, Dragos Stefan; Marcu, Daniel (2005). Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics* 31(4), 477–504.

Rapp, Reinhard (1995). Identifying word translations in non-parallel texts. In: *Proceedings of the 33rd Meeting of the Association for Computational Linguistics.* Cambridge, Massachusetts, 320–322.

Rapp, Reinhard. (1999). Automatic identification of word translations from unrelated English and German corpora. In: *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics 1999,* College Park, Maryland. 519–526.

Rapp, Reinhard; Zock, Michael (2010). Automatic dictionary expansion using non-parallel corpora. In: Andreas Fink, Berthold Lausen, Wilfried Seidel Alfred Ultsch (Eds.) *Advances in Data Analysis, Data Handling and Business Intelligence. Proceedings of the 32nd Annual Meeting of the GfKl, 2008*. Heidelberg: Springer.

Steinberger, Ralf; Pouliquen, Bruno; Widiger, Anna; Ignat, Camelia; Erjavec, Tomaž; Tufiş, Dan; VARGA, Dániel (2006). The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. *Proceedings of the 5thLREC***,** Genoa, Italy.

Wu, Dekai; Fung, Pascale (2005). Inversion transduction grammar constraints for mining parallel sentences from quasi-comparable corpora. Proceedings of the *Second International Joint Conference on Natural Language Processing (IJCNLP-2005)*. Jeju, Korea.

Parker, Robert, Graff, David; Kong, Junbo; Chen, Ke; Maeda, Kazuaki (2009). *English Gigaword. Fourth Edition.* Linguistic Data Consortium, Philadelphia.

Large Corpus-based Semantic Feature Extraction for Pronoun Coreference

Shasha Liao
Dept. of Computer Science
New York University
liaoss@cs.nyu.edu

Ralph Grishman
Dept. of Computer Science
New York University
grishman@cs.nyu.edu

Abstract

Semantic information is a very important factor in coreference resolution. The combination of large corpora and 'deep' analysis procedures has made it possible to acquire a range of semantic information and apply it to this task. In this paper, we generate two statistically-based semantic features from a large corpus and measure their influence on pronoun coreference. One is contextual compatibility, which decides if the antecedent can be used in the anaphor's context; the other is role pair, which decides if the actions asserted of the antecedent and the anaphor are likely to apply to the same entity. We apply a semantic labeling system and a baseline coreference system to a large corpus to generate semantic patterns and convert them into features in a MaxEnt model. These features produce an absolute gain of 1.5% to 1.7% in resolution accuracy (a 6% reduction in errors). To understand the limitations of these features, we also extract patterns from the test corpus, use these patterns to train a coreference model, and examine some of the cases where coreference still fails. We also compare the performance of patterns extracted from semantic role labeling and syntax.

1 Introduction

Coreference resolution is the task of determining whether two phrases refer to the same entity. Coreference is critical to most NLP tasks, yet even the sub-problem of pronoun coreference remains very challenging. In principle, we need several types of information to identify the right antecedent. First, number and gender agreement constraints can narrow the candidate set. If multiple candidates remain, we would next use some sequence or syntactic features, like position, word, word salience and discourse focus. For example, whether an antecedent is in subject position might be helpful because the subject is more likely to be referred to; or an entity that has been referred to repeatedly is more likely to be referred to again. However, these features do not suffice to pick the correct antecedent, and sometimes similar syntactic structures might have quite different coreference solutions. For example, for the following two sentences:

(1) *The terrorist shot a 13-year-old boy;* ***he*** *was arrested after the attack.*

(2) *The terrorist shot a 13-year-old boy;* ***he*** *was fatally wounded in the attack.*

it is likely that "he" refers to "*terrorist*" in (1) and "*boy*" in (2). However, we cannot get the right antecedent using the features we mentioned above because the examples share the same antecedent words and syntactic structure. People can still resolve these correctly because "*terrorist*" is more likely to be arrested than "*boy*", and because the one shooting is more likely to be arrested than the one being shot.

In such cases, semantic constraints and preferences are required for correct coreference resolution. Methods for acquiring and using such knowledge are receiving increasing attention in

Proceedings of the Second Workshop on NLP Challenges in the Information Explosion Era (NLPIX 2010), pages 60–68,
Beijing, August 2010

recent work on anaphora resolution. Dagan and Itai (1990), Bean and Riloff (2004), Yang and Su (2007), and Ponzetto and Strube (2006) all explored this task.

However, this task is difficult because it requires the acquisition of a large amount of semantic information. Furthermore, there is not universal agreement on the value of these semantic preferences for pronoun coreference. Kehler et al. (2004) reported that such information did not produce apparent improvement in overall pronoun resolution.

In this paper, we will extract semantic features from a semantic role labeling system instead of a parse tree, and explore whether pronoun coreference resolution can benefit from such knowledge, which is automatically extracted from a large corpus. We studied two features: the contextual compatibility feature which has been demonstrated to work at the syntactic level by previous work; and the role pair feature, which has not previously been applied to general domain pronoun co-reference. In addition, to obtain a rough upper bound on the benefits of our approach and understand its limitations, we conducted a second experiment in which the semantic knowledge is extracted from the evaluation corpus.

We will use the term *mention* to describe an individual referring phrase. For most studies of coreference, mentions are noun phrases and may be headed by a name, a common noun, or a pronoun. We will use the term *entity* to refer to a set of coreferential mentions.

2 Related Work

Contextual compatibility features have long been studied for pronoun coreference: Dagan and Itai (1990) proposed a heuristics-based approach to pronoun resolution. It determined the preference of candidates based on predicate-argument frequencies.

Bean and Riloff (2004) present a system, which uses contextual role knowledge to aid coreference resolution. They used lexical and syntactic heuristics to identify high-confidence coreference relations and used them as training data for learning contextual role knowledge. They got substantial gains on articles in two specific domains, terrorism and natural disasters.

Yang et al. (2005) use statistically-based semantic compatibility information to improve pronoun resolution. They use corpus-based and web-based extraction strategies, and their work shows that statistically-based semantic compatibility information can improve coreference resolution.

In contrast, Kehler et al. (2004) claimed that the contextual compatibility feature does not help much for pronoun coreference: existing learning-based approaches already performed well; such statistics are simply not good predictors for pronoun interpretation; data is sparse in the collected predicate-argument statistics.

The role pair feature has not been studied for general, broad-domain pronoun co-reference, but it has been used for other tasks: Pekar (2006) built pairs of 'templates' which share an 'anchor' argument; these correspond closely to our role pairs. Association statistics of the template pairs were used to acquire verb entailments. Abe et al. (2008) looked for pairs appearing in specific syntactic patterns in order to acquire finer-grained event relations. Chambers and Jurafsky (2008) built narrative event chains, which are partially ordered sets of events related by a common protagonist. They use high-precision hand-coded rules to get coreference information, extract predicate arguments that link the mentions to verbs, and link the arguments of the coreferred mentions to build a verb entailment model.

Bean and Riloff (2004) used high-precision hand-coded rules to identify coreferent mention pairs, which are then used to acquire role pairs that they refer to as *Caseframe Network* features. They use these features to improve coreference resolution for two domain-specific corpora involving terrorism and natural disasters. Their result raises the natural question as to whether the approach (which may capture domain-specific pairs such as “kidnap—release” in the terrorism domain) can be successfully extended to a general news corpus. We address this question in the experiments reported here.

3 Corpus Analysis

In order to extract semantic features from our large training corpus, we apply a sequence of analyzers. These include name tagging, parsing, a baseline coreference analyzer, and, most important, a semantic labeling system that can generate the logical grammatical and predicate-argument representation automatically from a

parse tree (Meyers et al. 2009). We use semantic labeling because it provides more general and meaningful patterns, with a "deeper" analysis than parsed text. The output of the semantic labeling is the dependency representation of the text, where each sentence is a graph consisting of nodes (corresponding to words) and arcs. Each arc captures up to three relations between two words: (1) a SURFACE relation, the relation between a predicate and an argument in the parse of a sentence; (2) a LOGIC1 (grammatical logical) relation which regularizes for lexical and syntactic phenomena like passive, relative clauses, and deleted subjects; and (3) a LOGIC2 (predicate-argument) relation corresponding to relations in PropBank and NomBank. It is designed to be compatible with the Penn TreeBank (Marcus et al., 1994) framework and therefore, Penn TreeBank-based parsers, while incorporating Named Entities, PropBank, and NomBank.

Because nouns and verbs provide the most relevant contexts and capture the events in which the entities participate, we generate *semantic patterns* (triples) only for those arcs with verb or noun heads. We use the following relations:

- Logic2 relations: We use in particular the Arg0 relation (which corresponds roughly to *agent*) and Arg1 relation (which corresponds roughly to *patient*).
- Logic1 relations: We use in particular the Sbj and Obj relations, representing the logical subject and object of a verb (regularizing passive, relative clauses, deleted subjects)
- Surface relations: T-pos relation is particularly used, which captures the head noun – determiner relation for possessive constructs such as "bomber's attack" and "his responsibility".

For example, for the sentence:

`John is hit by Tom's brother.`

we generate the semantic patterns

<Arg1 hit John>
<Arg0 hit brother>
<T-pos brother Tom>

We apply this labeling system to all the data we use, and to generate the semantic pattern, we take first its predicate-argument role; if that is null, we take its logical grammatical role; if both are null, we take its surface role.

To reduce data sparseness, all inflected words are changed to their base form (e.g. "attackers"→"attacker"). All names are replaced by their ACE types (person, organization, location, etc.). Only patterns with noun arguments are extracted because we only consider noun phrases as possible antecedents.

4 Semantic Features

4.1 Contextual Compatibility Patterns

Pronouns, especially neutral pronouns ("it", "they"), carry little semantics of their own, so examining the compatibility of the context of a pronoun and its candidate antecedents is a good way to improve antecedent selection. Specifically, we want to determine whether the predicate, which is applied to the anaphor, can be applied to the antecedents. We take the semantic pattern with the anaphor in third position. Then, each candidate antecedent is substituted for the anaphor to see if it is suitable for the context. For example, consider the sentence

The company issued a statement that it bought G.M.

which would generate the semantic patterns

<Arg0 issue company>
<Arg1 issue statement>
<Arg0 buy it>
<Arg1 buy Organization>

(here "G.M" is a name of type *organization* and so is replaced by the token Organization). The relevant context of the anaphor is the semantic pattern <Arg0 buy it>. Suppose there are two candidate antecedents for "it": "company" and "statement". We would generate the two semantic patterns <Arg0 buy company> and <Arg0 buy statement>. Assuming <Arg0 buy company> is more highly ranked than <Arg0 buy statement>, we can infer that the anaphor is more likely to refer to "company". (We describe the specific metric we use for ranking below, in section 4.3.) As further examples consider:

(3) *The suspect's lawyer, Chifumu Banda, told the court **he** had advised Chiluba not to appear in court Friday.*

(4) *Foreign military analysts said it would be highly unusual for an accident to kill a whole submarine crew and* ***they*** *suggested possible causes to a disaster…*

For (3), if we know that a lawyer is more likely to give advice than a suspect, we could link "*he*" to "*lawyer*" instead of "*suspect*" in the first sentence. For (4), if we know that analysts are more likely to "*suggest*" than crew, we can link "*they*" to "analysts" in the second sentence.

4.2 Role Pair Patterns

The role pair pattern is a new feature in general pronoun co-reference. The original intuition for introducing it into coreference is that there are pairs of actions involving the same entity that are much more likely to occur together than would be true if one assumed statistical independence. The second action may be a rephrasing or elaboration of the first, or the two might be actions that are part of a common 'script'. For example:

(5) *Prime Minister Mahathir Mohamad sacked the former deputy premier in 1998, who was sentenced to a total of 15 years in jail after being convicted of corruption and sodomy.* ***He*** *was released after four years because….*

(6) *The robber attacked the boy with a knife;* ***he*** *was bleeding heavily and died in the hospital the next day.*

For (5), if we know that the person who was sentenced is more likely to be released than the person who sacked others, we would know "*he*" refers to "*deputy premier*" instead of "*prime minister*". And in (6), because someone being attacked is more likely to die than the attacker, we can infer that "*he*" refers to "*boy*".

To acquire such information, we need to identify those pairs of predicates which are likely to apply to the same entity. We collect this data from a large corpus. The basic process is: apply a baseline coreference system to produce mentions and entities for a large corpus. For every entity, record the predicates for every mention, and then the pairs of predicates for successive mentions within each entity.

Although the performance of the baseline coreference is not very high, and individual documents may yield many idiosyncratic pairs, we can gather many significant role pairs by collecting statistics from a large corpus and filtering out the low frequency patterns; this process can eliminate much of the noise due to coreference errors.

Here is an example of the extracted role pairs involving "attack":

Arg0 attack *x* ↔	Obj volley *x*
	Arg0 bombard *x*
	Obj barrage *x*
	Arg0 snatch *x*
	Sbj attack *x*
	Arg0 pound *x*
	Obj reoccupy *x*
	Arg1 halt *x*
	Arg0 assault *x*
	Arg1 bombard *x*

Table1. Top 10 role pairs associated with "Arg0 attack x"

4.3 Contextual Compatibility Scores

To properly compare the patterns involving alternative candidate antecedents, we need to normalize the raw frequencies first. We followed Yang et al. (2005)'s idea, which normalizes the pattern frequency by the frequency of the candidates, and use a relative score that is normalized by the maximum score of all its candidates:

$$CompScore(P_{context,Cand}) = \frac{CompFreq(P_{context,Cand})}{Max_{Ci \in Set(cands)} CompFreq(P_{context,Ci})}$$

and $$CompFreq(P_{context,Cand}) = \frac{freq(P_{context,Cand})}{freq(Cand)}$$

where $P_{context,Cand}$ is the contextual compatibility pattern built from the context of the pronoun and the base form of the candidate.

In contrast to Yang's work, which used contextual compatibility on the *mention* level, we consider the contextual compatibility of an *entity* to an anaphor: we calculate the contextual information of all the mentions and choose the one with highest score as the contextual compatibility score for this entity[1]:

[1] Note that all the mentions in the entity are generated by the overall coreference system. Also, the ACE entity type of names is determined by the system. No key annotations are considered in the entire coreference phase.

$$freq(context,entity) = Max_{mention_i \in entity} freq(P_{context,mention_i})$$

4.4 Role Pair Scores

Unlike the contextual compatibility feature, we only take the role pair of the successive mentions in the candidate entity and the anaphor, because they are more reliably coreferential than arbitrary pairs of mentions within an entity:

$$PairFreq(p_{ana}, p_{cand}) = \frac{freq(p_{ana}, p_{cand})}{freq(p_{cand})}$$

where p_{ana} and p_{cand} are the contextual patterns of the anaphor and the last mention in the candidate entity.

For a set of possible candidates, we compute a relative score:

$$PairScore(p_{ana},p_{cand}) = \frac{PairFreq(p_{ana},p_{cand})}{Max_{pi \in Set(cands)} PairFreq(p_{ana},pi)}$$

Both scores are quantized (binned) in intervals of 0.1 for use as MaxEnt features.

5 Experiment

Our coreference solution system uses ACE annotated data and follows the ACE 2005 English entity guidelines.[2] The baseline coreference system to compare with is the same one used for extracting semantic features from the large corpus. It employs an entity-mention (rather than a mention-pair) model.

Besides entity and mention information, which (as mentioned above) is system output, the semantic information is also automatically extracted by a semantic labeling system. As a result, we report results in section 5.4 which involve no information from the reference (key) annotation.

5.1 Baseline System Description

The baseline system first applies processes like parsing, semantic labeling, name tagging, and entity mention tagging, producing a set of mentions to which coreference analysis is then applied. The coreference phase deals with coreference among mentions that might be pronouns, names or proper nouns, and generates entities when it is finished. The whole is a one-pass process, resolving coreference in the order in which mentions appear in the document. In the pronoun coreference process, every pronoun mention is assigned to one of the candidate entities.

Features	Description
Hobbs_ Distance	Hobbs distance between the last mention in the entity and the anaphor
Head_Pro	Combined word features of the head of the last mention in the entity and anaphor
Is_Subject	True if the last mention in the entity is a subject of the sentence
Last_Cat	Whether the last mention in the entity is a noun phrase, a pronoun or a name
Co_Prior	Number of prior references to this entity

Table 2. Features used in baseline system

The baseline co-reference system has separate, quite elaborate, primarily rule-based systems to handle names, nominals, headless NP's, and adverbs ("here", "there") which may be anaphoric, as well as first- and second-person pronouns. The MaxEnt model under study in this paper is only responsible for third-person pronouns. Also, gender, number, and human/non-human are handled separately outside of the MaxEnt model, and the model only resolves mentions that satisfy these constraints.[3] In the MaxEnt model, 5 basic features (described in table 2) are used. Thus, while the set of features used in the model is relatively small in comparison to many current statistically based reference resolvers, these are the primary features relevant to the limited task

[2] Automatic Content Extraction evaluation, http://projects.ldc.upenn.edu/ace/

[3] Gender information is obtained from a dictionary of gender-specific nouns and from first-name lists from the US Census. Number information comes from large syntactic dictionaries, corpus annotation of collective nouns (syntactically singular nouns which may take plural anaphors), and name tagger information (some organizations and political entities may take plural anaphors).

of the MaxEnt model, and its performance is still competitive[4].

5.2 Corpus Description

There are two kinds of corpora used in our experiment, a small coreference-annotated corpus used for training and evaluating (in cross-validation) the pronoun coreference model, and a large raw-text corpus for extracting semantic information.

For model training and evaluation, we assembled two small corpora from the available ACE data. One consists of news articles (460 documents) from ACE 2005 (330 documents) and ACE 2003 (130 documents), which together contain 3934 pronouns. The other is the full ACE 2005 training set (592 documents), which includes newswire, broadcast news, broadcast conversations (interviews and discussions), web logs, web forums, and Fisher telephone transcripts, and contains 5659 pronouns.

In evaluation, we consider a pronoun to be correctly resolved if its antecedent in the system output (the most recent prior mention in the entity to which the pronoun is assigned) matches the antecedent in the key. We report accuracy (percentage of pronouns which are correctly resolved).

We used a large corpus to extract semantic information, consisting of five years of AFP newswire from the LDC English Gigaword corpus (1996, 2002, 2004, 2005 and 2006), a total of 907,368 documents. We omit news articles written in 1998, 2000 and 2003 to insure there is no overlap between the ACE data and Gigaword data. We pre-processed each document (parsing, name identification, and semantic labeling) and ran the baseline coreference system, which automatically identified all the mentions (including name mentions and nominal mentions) and built a set of entities for each document.

[4]For example, among papers reporting a pronoun accuracy metric, Kehler et al. (2004), testing on a 2002 ACE news corpus, get a pronoun accuracy (without semantic features) of 75.7%; (Yang et al. 2005), testing on the MUC coreference corpora (also news) get for their single-candidate baseline (without semantic features) 75.1% pronoun accuracy. Although the testing conditions in each case are different, these are comparable to our baseline performance.

5.3 Semantic Information Extraction from Large Corpus

In order to remove noise, we only keep contextual compatibility patterns that appear more than 5 times; and only keep role pair patterns which appear more than 15 times, and appear in more than three different years to avoid random pairs extracted from repeated stories. We automatically extracted 626,008 contextual compatibility patterns and 4,736,359 role pairs. Note that we extract fewer patterns than Yang (2005), who extracted in total 2,203,203 contextual compatibility patterns, from a much smaller corpus (173,252 Wall Street Journal articles). This might be for two reasons: first, we pruned low frequency patterns; second, we used a semantic labeling system instead of shallow parsing. Section 5.6 gives a comparison of pattern extraction based on different levels of analysis.

5.4 Results

	News Corpus		2005 Corpus	
	Accu	**SignTest (p <=)**	**Accu**	**SignTest (p <=)**
baseline	75.54		72.04	
context	76.59	0.025	73.35	0.002
role pair	76.28	0.031	73.03	0.003
combine	77.02	0.0005	73.72	0.0015

Table 3. Accuracy of 5-fold cross-validation with statistics-based semantic features

We did a 5-fold cross validation to test the contribution from statistically-based semantic features, and report an average accuracy. All the mentions and their features are obtained from system output; as a result, if the correct antecedent is not correctly discovered and analyzed from the previous phases, we will not be able to co-refer the pronoun correctly. Experiments on the news articles show that each feature provides approximately 1% gain by itself, and contributes to a substantial overall gain of 1.45%. For the 2005 corpus, the baseline is lower because the documents come from different genres, and we get more gain from each semantic feature. We also computed the significance over the baseline using the sign test[5].

[5] In applying the sign test, we treated each pronoun as an independent sample, which is either correctly resolved or incorrectly resolved. Where the individual observations are

5.5 Self-Extracted Bound

To better understand the potential maximum contribution of our semantic features, we constructed an approximation to the most favorable possible semantic features for each test set. We did this by using perfect coreference knowledge and by collecting patterns for each test set *from the test set itself*. For each corpus used for cross-validation, we first collect all the contextual compatibility and role pair patterns corresponding to the correct antecedents (we ignore the patterns corresponding to the wrong antecedents, because we can not get this negative information when we extract them from a large corpus), and score these patterns to produce semantic features for the MaxEnt Model, both training and testing. We then use these features in the model and do a cross-validation as before. Also, as before, we rely on system output to identify and analyze potential antecedents; if the prior phases do not do so correctly, coreference analysis may well fail. This experiment shows that we can get about 3 to 4% gain from each feature type separately; 4.5 to 5.5% gain is achieved from the two features together.

	News Corpus		2005 Corpus	
	Accu	**SignTest (p <=)**	**Accu**	**SignTest (p <=)**
baseline	75.54		72.04	
context	79.23	7e-14	76.04	9e-27
role pair	78.85	6e-13	75.95	1e-26
combine	79.97	4e-16	77.50	2e-38

Table 4. Accuracy of 5-fold cross-validation with self-extracted semantic features

5.6 Comparison between Semantic and Syntax Patterns

To better understand the difference between semantic role labeling and syntactic relations, we did a comparison between patterns extracted from the syntax level and those extracted from semantic role labeling:

Experiments show that using semantic roles (such as Arg0 and Arg1) works better. This may be because the "deeper" representation provides more generalization of relations. For example, the phrases "weapon's use" and "use weapon" share the same semantic relation <Arg1 use weapon>, while they yield different grammatical relations: <T-pos use weapon> and <Obj use weapon>.

	News Corpus		2005 Corpus	
	semantic	**syntax**	**semantic**	**syntax**
baseline	75.54		72.04	
context	79.23	77.73	76.04	75.83
role pair	78.85	76.87	75.95	74.17
combine	79.97	78.42	77.50	76.76

Table 5. Accuracy of 5-fold cross-validation with self-extracted semantic features based on different levels of syntactic/semantic relations

5.7 Error Analysis

We analyzed the errors in the self-extracted results, to see why such corpus-specific semantic features do not produce an even greater reduction in errors. For the contextual compatibility feature, we find cases where an incorrect candidate is equally compatible with the context of the anaphor; for example, if all the candidates are person names, they will share the same context feature because they generate the same ACE type. In other cases, the context does not provide enough information. For example, in a context tuple <Arg0 get *x*>, *x* can be almost any noun, because "get" is too vague to predicate the compatible subjects. There are similar limitations with the role pair feature; for example, <Arg0 get they> can be associated with a lot of other actions.

To quantify this problem, we counted the patterns that appear in both positive examples (correct antecedents) and negative examples (incorrect antecedents). For contextual compatibility patterns, 39.5% of the patterns which appear with positive examples also appear in the negative sample, while for role pair patterns, 19% of the patterns which appear with positive examples also appear in the negative sample. So we see that, even with a pattern set highly tuned to the test set, many patterns do not by themselves serve to distinguish correct from incorrect coreference.

We analyzed some of the cases where the semantic information does not help, or even harms the analysis. In some cases all the antecedent

(changes in) binary outcomes, the sign test provides a suitably sensitive significance test. (In particular, it is comparable to performing a paired t-test over counts of correct resolutions, aggregated over documents.)

scores are very low, either because the patterns are very rare or the antecedent is a common word that appears in a lot of patterns. In other cases, several antecedents have a high compatibility score but the correct one does not have the top score. In these cases, the contextual compatibility is not reliable, as was pointed out by Kehler et al. (2004):

(7) *The model for a republic, adopted over bitter objections from those advocating direct election of a president, is for presidential nominations to be made with public input and the winning candidate decided by a two-thirds majority of Parliament. Former prime minister Paul Keating, who put the republic issue in the spotlight in* ***his*** *unsuccessful 1996 campaign for re-election, welcomed the result.*

Here adding semantic features leads "*his*" to be incorrectly resolved to "*president*" rather than the entity with mentions "*prime minister*" and "*Paul Keating*"; all the relevant patterns are common, but the score for <Arg0 campaign president> is higher (around 0.0012) than for <Arg0 campaign minister> (0.0004) or <Arg0 campaign Person> (0.0006).

Another problem is that the patterns do not capture enough context information, for example:

(8) *The U.S. administration has been pressing the Security Council to adopt a statement condemning Pyongyang for failing to meet* ***its*** *obligations.*

If we can get the semantic context of "*fail to meet its obligations*" instead of "*its obligations*", we might get better solutions for (8).

The role pair information raises similar problems. Some verbs are very vague, like "*get*", "*take*", "*have*", and role pairs with these verbs might not be very useful. Here is an example:

(9) *The retired Greek officer tried to get Ocalan to the Netherlands, home to a large Kurdish community.* ***He*** *claimed he had been manipulated by the government.*

In this sentence, the role pair information is very vague and it is hard to select a proper antecedent by connecting the subject of "*try*" or "*get*" or the object of "*get*" to the subject of "*claim*".

5.8 Limitations of Semantic Features

The availability of very large corpora coupled with improved pre-processing (e.g., faster parsers, accurate semantic labelers) is making it easier to extract large sets of semantic patterns. However, results on "perfect" semantic information show that even if we can get very good semantic features, there are at least two concerns to address:

- How to best capture the context information: larger context patterns may suffer from data sparseness; simple patterns may be insufficiently selective, appearing in both positive and negative samples.
- In some cases, the baseline features are sufficient to select the antecedent and the semantic features only do harm. If we are able to better gauge our confidence in the decisions based on the baseline features and on the semantic features, we may be able to combine these two sources more effectively.

6 Conclusions and Future Work

We have presented two ways to incorporate semantic features into a MaxEnt model-based pronoun coreference system, where these features have been extracted from a large corpus using a baseline IE (Information Extraction) system and a semantic labeling system, with no specific domain information.

We also estimated the maximal benefit of these features and did some error analysis to identify cases where this semantic knowledge did not suffice. Our experiments show the value of these semantic features for pronoun coreference, but also the limitations of our current context representation and reference resolution models.

Last, we compared the features extracted from different levels of analysis, and showed that 'deeper' representations worked better.

References

Shuya Abe, Kentaro Inui, and Yuji Matsumoto. 2008. *Two-phased event relation acquisition: coupling the relation-oriented and argument-oriented approaches.* Proc. 22nd Int'l Conf. on Computational Linguistics (COLING 2008).

David Bean and Ellen Riloff. 2004. *Unsupervised Learning of Contextual Role Knowledge for*

Coreference Resolution. Proc. HLT-NAACL 2004.

Nathanael Chambers and Dan Jurafsky. 2008. *Unsupervised Learning of Narrative Event Chains*. Proc. ACL-08: HLT.

I. Dagan and A. Itai. 1990. *Automatic processing of large corpora for the resolution of anaphora references*. Proc. 13th International Conference on Computational Linguistics (COLING 1990).

J. Hobbs. 1978. *Resolving pronoun references*. Lingua, 44:339–352.

A. Kehler, D. Appelt, L. Taylor, and A. Simma. 2004. *The (non)utility of predicate-argument frequencies for pronoun interpretation*. Proc. HLT-NAACL 2004.

A. Meyers, M. Kosaka, N. Xue, H. Ji, A. Sun, S. Liao and W. Xu. 2009. Automatic Recognition of Logical Relations for English, Chinese and Japanese in the GLARF Framework. *In SEW-2009 (Semantic Evaluations Workshop) at NAACL HLT-2009*

Viktor Pekar. 2006. *Acquisition of verb entailment from text*. Proc. HLT-NAACL 2006.

Simone Paolo Ponzetto and Michael Strube. 2006 *Exploiting semantic Role Labeling, WordNet and Wikipedia for Coreference Resolution*. Proc. HLT-NAACL 2006.

X. Yang, J. Su, G. Zhou, and C. Tan. 2004. *An NP-cluster approach to coreference resolution*. Proc. 20th International Conference on Computational Linguistics (COLING 2004).

Xiaofeng Yang, Jian Su, Chew Lim Tan.2005. *Improving Pronoun Resolution Using Statistics-Based Semantic Compatibility Information*. Proc. 43rd Annual Meeting of the Assn. for Computational Linguistics.

Xiaofeng Yang and Jian Su. 2007. *Coreference Resolution Using Semantic Relatedness Information from Automatically Discovered Patterns*. Proc. 45th Annual Meeting of the Assn. for Computational Linguistics.

Mining coreference relations between formulas and text using Wikipedia

Minh Nghiem Quoc [1], **Keisuke Yokoi** [2], **Yuichiroh Matsubayashi** [3] **Akiko Aizawa** [1 2 3]
[1] Department of Informatics, The Graduate University for Advanced Studies
[2] Department of Computer Science, University of Tokyo
[3] National Institute of Informatics
{nqminh, kei-yoko, y-matsu, aizawa}@nii.ac.jp

Abstract

In this paper, we address the problem of discovering coreference relations between formulas and the surrounding text. The task is different from traditional coreference resolution because of the unique structure of the formulas. In this paper, we present an approach, which we call *'CDF (Concept Description Formula)'*, for mining coreference relations between formulas and the concepts that refer to them. Using Wikipedia articles as a target corpus, our approach is based on surface level text matching between formulas and text, as well as patterns that represent relationships between them. The results showed the potential of our approach for formulas and text coreference mining.

1 Introduction

1.1 Motivation

Mathematical content is a valuable information source for many users: teachers, students, researchers need access to mathematical resources for teaching, studying, or obtaining updated information for research and development. Although more and more mathematical content is becoming available on the Web nowadays, conventional search engines do not provide direct search of mathematical formulas. As such, retrieving mathematical content remains an open issue.

Some recent studies proposed mathematical retrieval systems that were based on structural similarity of equations (Adeel and Khiyal, 2008; Yokoi and Aizawa, 2009; Nghiem et al., 2009). However, in these studies, the semantics of the equations is still not taken into account. As mathematical equations follow highly abstract and also rewritable representations, structural similarity alone is insufficient as a metric for semantic similarity.

Based on this observation, the primary goal of this paper is to establish a method for extracting implicit connections between mathematical formulas and their names together with the descriptions written in natural language text. This enables keywords to be associated with the formulas and makes mathematical search more powerful. For example, it is easier for people searching and retrieving mathematical concepts if they know the name of the equation *"$a^2 + b^2 = c^2$"* is the *"Pythagorean Theorem"*. It could also make mathematics more understandable and usable for users.

While many studies have presented coreference relations among texts (Ponzetto and Poesio, 2009), no work has ever considered the coreference relations between formulas and texts. In this paper, we use Wikipedia articles as a target corpus. We chose Wikipedia for these reasons: (1) Wikipedia uses a subset of TEX markup for mathematical formulas. That way, we can analyze the content of these formulas using TEX expressions rather than analyzing the images. (2) Wikipedia provides a wealth of knowledge and the content of Wikipedia is much cleaner than typical Web pages, as explained in Giles (2005).

Proceedings of the Second Workshop on NLP Challenges in the Information Explosion Era (NLPIX 2010), pages 69–74,
Beijing, August 2010

1.2 Related Work

Ponzetto and Poesio (2006) attempted to include semantic information extracted from WordNet and Wikipedia into their coreference resolution model. Shnarch et al. (2009) presented the extraction of a large-scale rule base from Wikipedia designed to cover a wide scope of the lexical reference relations. Their rule base has comparable performance with WordNet while providing largely complementary information. Yan et al. (2009) proposed an unsupervised relation extraction method for discovering and enhancing relations associated with a specified concept in Wikipedia. Their work combined deep linguistic patterns extracted from Wikipedia with surface patterns obtained from the Web to generate various relations. The results of these studies showed that Wikipedia is a knowledge-rich and promising resource for extracting relations between representative terms in text. However, these techniques are not directly applicable to the coreference resolution between formulas and texts as we mention in the next section.

1.3 Challenges

There are two key challenges in solving the coreference relations between formulas and texts using Wikipedia articles.

- First, formulas have unique structures such as prior operators and nested functions. In addition, features such as gender, plural, part of speech, and proper name, are unavailable with formulas for coreference resolution. Therefore, we cannot apply standard natural language processing methods to formulas.

- Second, no labeled data are available for the coreference relations between formulas and texts. This means we cannot apply commonly used machine learning-based techniques without expensive human annotations.

1.4 Our Approach and Key Contributions

In this paper, we present an approach, which we call *CDF (Concept Description Formula)*, for mining coreference relations between mathematical Formulas and Concepts using Wikipedia articles. In order to address the previously mentioned challenges, the proposed *CDF* approach is featured as follows:

- First, we consider not only the concept-formula pairs but extend the relation with descriptions of the concept. Note that a "concept" in our study corresponds to a "name" or a "title" of a formula, which is usually quite short. By additionally considering words extracted from the descriptions, we have a better chance of detecting keywords, such as mathematical symbols, and function or variable names, used in the equations.

- Second, we apply an unsupervised framework in our approach. Initially, we extract highly confident coreference pairs using surface level text matching. Next, we collect promising syntactic patterns from the descriptions and then use the patterns to extract coreference pairs. The process enables us to deal with cases where there exist no common words between the concepts and the formulas.

The remainder of this paper is organized as follows: In section 2, we present our method. We then describe the experiments and results in section 3. Section 4 concludes the paper and gives avenues for future work.

2 Method

2.1 Overview of the Method

In this section, we first explain the terms used in our approach. We then provide a framework of our method and the functions of the main modules.

Given a set of Wikipedia articles as input, our system outputs a list of formulas along with their names and descriptions. Herein

- **Concept**: A concept C is a phrase that represents a name of a mathematical formula. In Wikipedia, we extract candidate concepts as noun phrases (NPs) that are either the titles of

Wikipedia articles, section headings, or written in bold or italic. Additional NPs that contain at least one content word are also considered.

- **Description**: A description D is a phrase that describes the concept. In Wikipedia, descriptions often follow a concept after the verb "*be*".

- **Formula**: A formula F is a mathematical formula. In Wikipedia extracted XML files, formulas occur between the $< math >$ and $< /math >$ tags. They are encoded in TEX format.

- **Candidate**: A candidate is a triple of concept, description and formula. Our system will judge if the candidate is qualified, which means the concept is related to the formula.

Figure 1 shows a section of a Wikipedia article and the concepts, descriptions and formulas in this section. Table 1 shows the extracted candidates. Details of how to extract the concepts, descriptions and formulas and how to form candidates are described in the next sections.

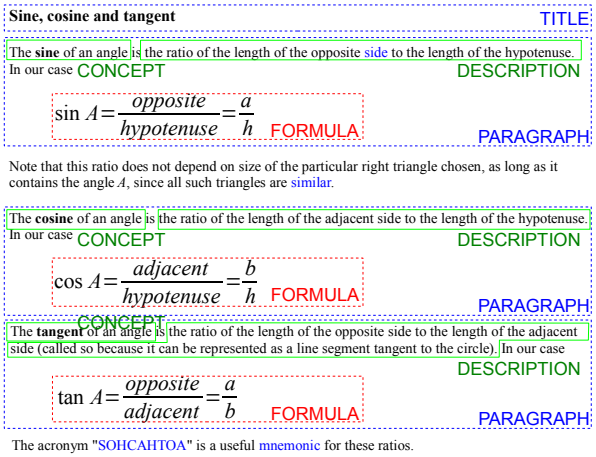

Figure 1: Examples of extracted paragraphs

The framework of the system is shown in Figure 2. The system has four main modules.

- **Text Preprocessor**: processes Wikipedia articles to extract *CDF (Concept Description Formula)* candidates.

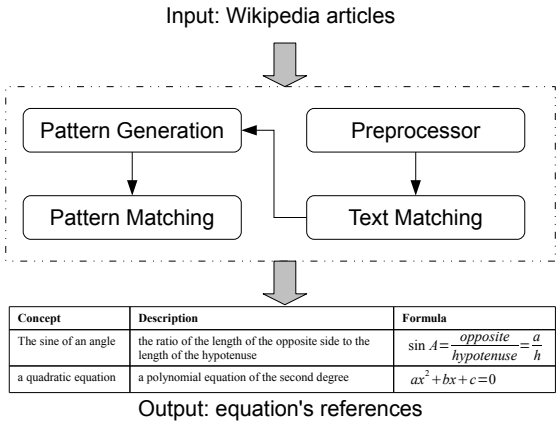

Figure 2: Framework of the proposed approach

- **Text Matching**: extracts reliable and qualified candidates using surface level text matching.

- **Pattern Generation**: generates patterns from qualified candidates.

- **Pattern Matching**: extends the candidate list using the generated patterns.

2.2 Text Preprocessor

This module preprocesses the text of the Wikipedia article to extract *CDF* candidates. Based on the assumption that concepts, their descriptions and formulas are in the same paragraph, we split the text into paragraphs and select paragraphs that contain at least one formula.

On these selected paragraphs, we run Sentence Boundary Detector, Tokenizer and Parser from OpenNLP tools. [1] Based on the parse trees, we extract the noun phrases (NPs) and identify NPs representing concepts or descriptions using the definitions in Section 2.1.

Following the general idea in Shnarch et al. (2009), we use the *"Be-Comp"* rule to identify the description of a concept in the definition sentence. In a sentence, we extract nominal complements of the verb 'to be', assign the NP that occurs after the verb 'to be' as the description of the NP that occurs before the verb. Note that some concepts have descriptions while others do not.

[1] http://opennlp.sourceforge.net/

Table 1: Examples of candidates

Concept	Description	Formula
the sine of an angle	the ratio of the length of the opposite side to the length of the hypotenuse	$\sin A = \frac{\text{opposite}}{\text{hypotenuse}} = \frac{\text{a}}{\text{h}}$
the cosine of an angle	the ratio of the length of the adjacent side to the length of the hypotenuse	$\cos A = \frac{\text{adjacent}}{\text{hypotenuse}} = \frac{b}{h}$
a quadratic equation	a polynomial equation of the second degree	$ax^2 + bx + c = 0$
the quadratic formula		$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$
the complex number i		$i^2 = -1$
the Cahen–Mellin integral		$e^{-y} = \frac{1}{2\pi i} \int_{c-i\infty}^{c+i\infty} \Gamma(s) y^{-s}\, ds$

The *"Be-Comp"* rule can also identify if a formula is related to the concept.

After that, we group each formula F in the same paragraph with concept C and its description D to form a candidate (C, D, F). Table 1 presents candidate examples. Because we only choose paragraphs that contain at least one formula, every concept has a formula attached to it. In order to judge the correctness of candidates, we use the text-matching module, described in the next section.

2.3 Text Matching

In this step, we classify candidates using surface text. Given a list of candidates of the form (C, D, F), this module judges if a candidate is qualified by using the surface text in concept, description and formula. Because many formulas share the same variable names or function names (or part of these names) with their concepts (e.g. the first two candidates in Table 1), we filter these candidates using surface text matching.

We define the similarity between concept C, description D and formula F by the number of overlapped words, as in Eq. 1.

$$sim(\mathrm{F}, \mathrm{CD}) = \frac{|\mathrm{T_F} \cap \mathrm{T_C}|}{\min\{|\mathrm{T_C}|, |\mathrm{T_F}|\}} + \frac{|\mathrm{T_F} \cap \mathrm{T_D}|}{\min\{|\mathrm{T_D}|, |\mathrm{T_F}|\}} \quad (1)$$

T_F, T_C and T_D are sets of words extracted from F, C and D, respectively.

Candidates with $sim(\mathrm{F}, \mathrm{CD})$ no larger than a threshold θ_1 (1/3 in this study) are grouped into the group C_{true}. The rest are filtered and stored in C_0. In this step, function words such as articles, pronouns, conjunctions and so on in concepts and descriptions are ignored. Common operators in formulas are also converted to text, such as *'+' 'plus', '−' 'minus', '\frac' 'divide'*.

Using only concepts for text matching with formulas might leave out various important relations. For example, from the description of the first and second formula in Table 1, we could extract the variable names *"opposite"*, *"adjacent"* and *"hypotenuse"*.

By adding the description, we could get a more accurate judgment of whether the concept and the formula are coreferent. In this case, we can consider the concept, description and the formula form a coreference chain.

After this step, we have two categories, C_{true} and C_0. C_{true} contains qualified candidates while C_0 contains candidates that cannot be determined by text matching. The formulas in C_0 have little or no text relation with their concepts and descriptions. Thus, we can only judge the correctness of these candidates by using the text around the concepts, descriptions and formulas. The surrounding text can be formed into patterns and are generated in the next step.

2.4 Pattern Generation

One difficulty in judging the correctness of a candidate is that the formula does not share any relation with its concept and description. The third candidate in Fig. 1 is an example. It should be classified as a qualified instance but is left behind in C_0 after the "text matching" step.

In this step, we use the qualified instances in C_{true} to generate patterns. These patterns are used in the next step to judge the candidates in C_0. Patterns are generated as follows. First, the concept, description and formula are replaced by CONC, DESC and FORM, respectively. We then simply take the entire string between the first and the last appearance of CONC, DESC and FORM.

Table 2 presents examples of patterns extracted from group C_{true}.

Table 2: Examples of extracted patterns

Pattern
CONC is DESC: FORM
CONC is DESC. In our case FORM
CONC is DESC. So, ..., FORM
CONC FORM
CONC is denoted by FORM
CONC is given by ... FORM
CONC can be written as ... : FORM
FORM where CONC is DESC
FORM satisfies CONC

Using a window surrounding the concepts and formulas often leads to exponential growth in patterns, so we limit our patterns to those between any concept C, description D or formula F.

The patterns we obtained above are exactly the shortest paths from the C nodes to their F node in the parse tree. Figure 3 presents examples of these patterns in parse trees.

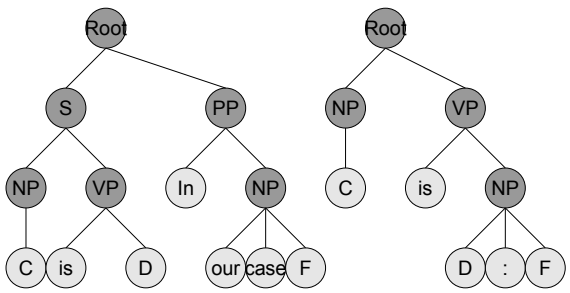

Figure 3: Examples of extracted patterns

2.5 Pattern Matching

In this step, we use patterns obtained from the previous step to classify more candidates in C_0. We use the string distance between the patterns, where candidates' patterns having a string distance to any of the patterns extracted in the previous step no larger than the threshold θ_2 are added into C_{true}.

3 Experiments

3.1 Data

We collected a total of 16,406 mathematical documents from the Wikipedia Mathematics Portal. After the preprocessing step, we selected 72,084 paragraphs that contain at least one formula. From these paragraphs, we extracted 931,716 candidates.

Because no labeled data are available for use in this task, we randomly chose 100 candidates: 60 candidates from C_{true} after the text matching step, 20 candidates added to C_{true} after pattern matching with $\theta_2 = 0$, and 20 candidates added to C_{true} after pattern matching with $\theta_2 = 0.25$ for our evaluation. These candidates were annotated manually. The sizes of the sample sets for human judgment (60, 20 and 20) were selected approximately proportional to the sizes of the obtained candidate sets.

3.2 Results

After the text matching step, we obtained 138,285 qualified candidates in the C_{true} group and 793,431 candidates in C_0. In C_{true}, we had 6,129 different patterns. Applying these patterns to C_0 by exact pattern matching ($\theta_2 = 0$), we obtained a further 34,148 qualified candidates. We obtained an additional 30,337 qualified candidates when we increased the threshold θ_2 to 0.25.

For comparison, we built a baseline system. The baseline automatically groups nearest formula and concept. It had 51 correctly qualified candidates. The results—displayed in Table 3 and depicted in Figure 4—show that our proposed method is significantly better than the baseline in terms of accuracy.

As we can see from the results, when we lower the threshold, more candidates are added to C_{true}, which means we get more formulas and formula names; but it also lowers the accuracy. Although the performance is not as high as other existing coreference resolution techniques, the proposed

Table 3: Results of the system

Module	No. correct/ total	No. of CDF found
Text Matching	41 / 60	138,285
Pattern Matching $\theta_2 = 0$	52 / 80	172,433
Pattern Matching $\theta_2 = 0.25$	56 / 100	202,270

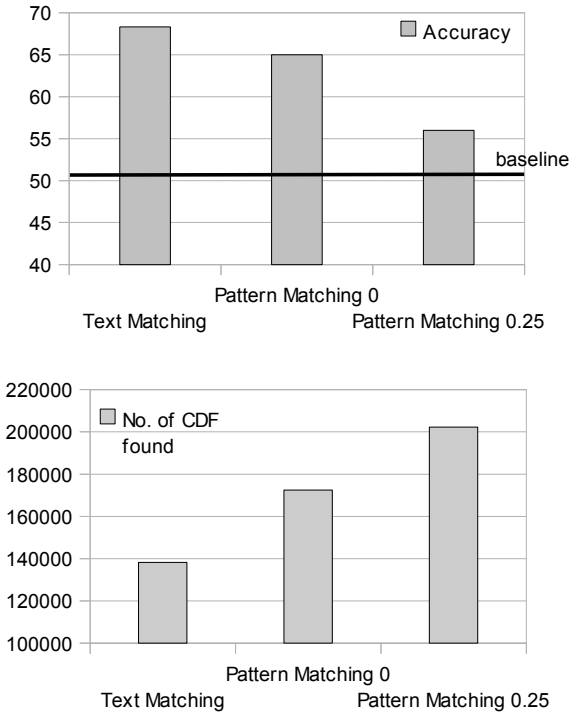

Figure 4: Results of the system

method is a promising starting point for solving coreference relations between formulas and surrounding text.

4 Conclusions

In this paper, we discuss the problem of discovering coreference relations between formulas and the surrounding texts. Although we could only use a small number of annotated data for the evaluation in this paper, our preliminary experimental results showed that our approach based on surface text-based matching between formulas and text, as well as patterns representing relationships between them showed promise for mining mathematical knowledge from Wikipedia. Since this is the first attempt to extract coreference relations between formulas and texts, there is room for further improvement. Possible improvements include: (1) using advanced technology for pattern matching to improve the coverage of the result and (2) expanding the work by mining knowledge from the Web.

References

Eyal Shnarch, Libby Barak and Ido Dagan. 2009. *Extracting Lexical Reference Rules from Wikipedia* Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, pages 450–458

Yulan Yan, Naoaki Okazaki, Yutaka Matsuo, Zhenglu Yang and Mitsuru Ishizuka. 2009. *Unsupervised Relation Extraction by Mining Wikipedia Texts Using Information from the Web* Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, pages 1021–1029

Simone Paolo Ponzetto and Massimo Poesio. 2009. *State-of-the-art NLP Approaches to Coreference Resolution: Theory and Practical Recipes* Tutorial Abstracts of ACL-IJCNLP 2009, page 6

Minh Nghiem, Keisuke Yokoi and Akiko Aizawa. 2009. *Enhancing Mathematical Search with Names of Formulas* The Workshop on E-Inclusion in Mathematics and Science 2009, pages 22–25

Keisuke Yokoi and Akiko Aizawa. 2009. *An Approach to Similarity Search for Mathematical Expressions using MathML* 2nd workshop Towards a Digital Mathematics Library, pages 27–35

Hui Siu Cheung Muhammad Adeel and Sikandar Hayat Khiyal. 2008. *Math Go! Prototype of a Content Based Mathematical Formula Search Engine* Journal of Theoretical and Applied Information Technology, Vol. 4, No. 10, pages 1002–1012

Simone Paolo Ponzetto and Michael Strube. 2006. *Exploiting Semantic Role Labeling, WordNet and Wikipedia for Coreference Resolution* In Proceedings of HLT-NAACL-06, pages 192–199

Jim Giles. 2005. *Internet Encyclopaedias Go Head to Head* Nature Volume: 438, Issue: 7070, pages 900–901

World Wide Web Consortium. *Mathematical Markup Language (MathML) version 2.0 (second edition)* http://www.w3.org/TR/MathML2/

Adverse–Effect Relations Extraction from Massive Clinical Records

Yasuhide Miura [a], Eiji Aramaki [b], Tomoko Ohkuma [a], Masatsugu Tonoike [a], Daigo Sugihara [a], Hiroshi Masuichi [a] and Kazuhiko Ohe [c]

[a] Fuji Xerox Co., Ltd.

[b] Center for Knowledge Structuring, University of Tokyo

[c] University of Tokyo Hospital

yasuhide.miura@fujixerox.co.jp, eiji.aramaki@gmail.com,
{ohkuma.tomoko,masatsugu.tonoike,daigo.sugihara,
hiroshi.masuichi}@fujixerox.co.jp,
kohe@hcc.h.u-tokyo.ac.jp

Abstract

The rapid spread of electronic health records raised an interest to large-scale information extraction from clinical texts. Considering such a background, we are developing a method that can extract adverse drug event and effect (adverse–effect) relations from massive clinical records. Adverse–effect relations share some features with relations proposed in previous relation extraction studies, but they also have unique characteristics. Adverse–effect relations are usually uncertain. Not even medical experts can usually determine whether a symptom that arises after a medication represents an adverse–effect relation or not. We propose a method to extract adverse–effect relations using a machine-learning technique with dependency features. We performed experiments to extract adverse–effect relations from 2,577 clinical texts, and obtained F_1-score of 37.54 with an optimal parameters and F_1-score of 34.90 with automatically tuned parameters. The results also show that dependency features increase the extraction F_1-score by 3.59.

1 Introduction

The widespread use of electronic health records (EHR) made clinical texts to be stored as computer processable data. EHRs contain important information about patients' health. However, extracting clinical information from EHRs is not easy because they are likely to be written in a natural language.

We are working on a task to extract adverse drug event and effect relations from clinical records. Usually, the association between a drug and its adverse–effect relation is investigated using numerous human resources, costing much time and money. The motivation of our task comes from this situation. An example of the task is presented in Figure 1. We defined an adverse–effect relation as a relation that holds between a drug entity and a symptom entity. The sentence illustrates the occurrence of the adverse–effect *hepatic disorder* by the *Singulair* medication.

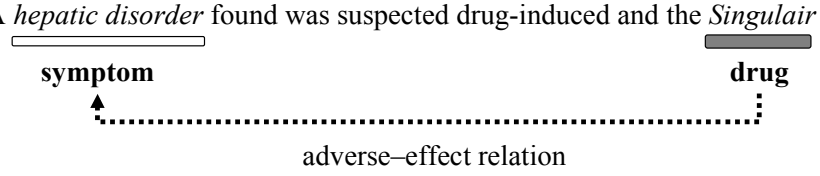

Figure 1. Example of an adverse–effect relation.

Proceedings of the Second Workshop on NLP Challenges in the Information Explosion Era (NLPIX 2010), pages 75–83,
Beijing, August 2010

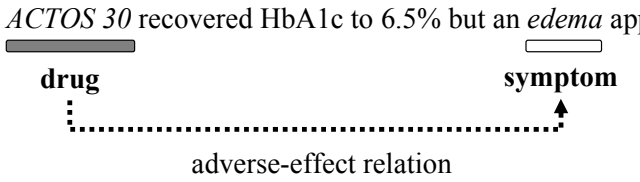

Figure 2. The example of an adverse-effect relation where the suspicion is not stated.

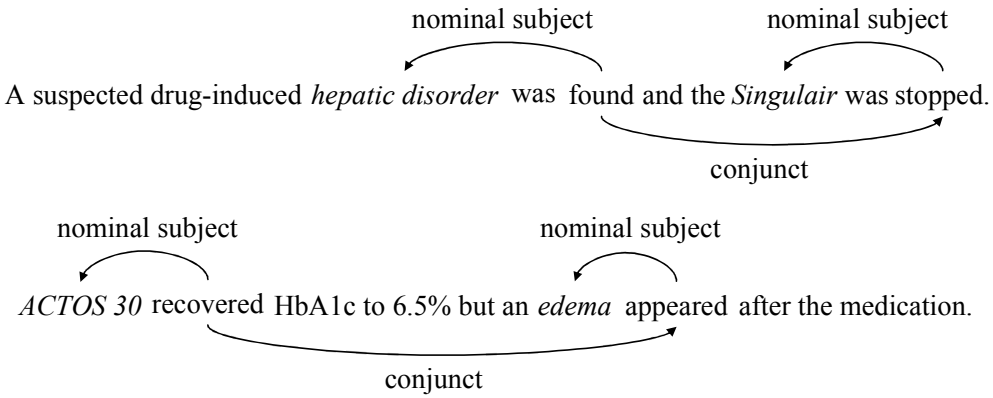

Figure 3. The example of a similarity within dependency structures.

A salient characteristic of adverse–effect relations is that they are usually uncertain. The sentence in the example states that the hepatic disorder is *suspected drug-induced*, which means the hepatic disorder is likely to present an adverse–effect relation. Figure 2 presents an example in which an adverse–effect relation is suspected, but words to indicate the suspicion are not stated. The two effects of the drug—the recovery of HbA1c and the appearance of the edema—are expressed merely as observation results in this sentence. The recovery of HbA1c is an expected effect of the drug and the appearance of the edema probably represents an adverse–effect case. The uncertain nature of adverse–effect relations often engenders **the statement of an adverse–effect relation as an observed fact**. A sentence including an adverse–effect relation occasionally becomes long to list all observations that appeared after administration of a medication. Whether an interpretation that expresses an adverse–effect relation, such as *drug-induced* or *suspected to be an adverse–effect*, exists in a clinical record or not depends on a person who writes it. However, an adverse–effect relation is associated with an undesired effect of a medication. Its appearance would engender an extra action (e.g. *stopped in* the first example) or lead to an extra indication (e.g. *but ... appeared in* the second example). Proper handling of this extra information is likely to boost the extraction accuracy.

The challenge of this study is to capture relations with various certainties. To establish this goal, we used a dependency structure for the adverse–effect relation extraction method. **Adverse–effect statements are assumed to share a dependency structure to a certain degree**. For example, if we obtain the dependency structures as shown in Figure 3, then we can easily determine that the structures are similar. Of course, obtaining such perfect parsing results is not always possible. A statistical syntactic parser is known to perform badly if a text to be parsed belongs to a domain which differs from a domain on which the parser is trained (Gildea, 2001). A statistical parser will likely output incomplete results in these texts and will likely have a negative effect on relation extraction methods which depend on it. The specified research topic of this study is to investigate whether incomplete dependency structures are effective and how they behave in the extraction of uncertain relations.

2 Related Works

Various studies have been done to extract semantic information from texts. SemEval-2007 Task:04 (Girju et al., 2007) is a task to extract semantic relations between nominals. The task includes "Cause–Effect" relation extraction, which shares some similarity with a task that will be presented herein. Saeger et al. (2008) presented a method to extract potential troubles or obstacles related to the use of a given object. This relation can be interpreted as a more general relation of the adverse–effect relation. The protein–protein interaction (PPI) annotation extraction task of BioCreative II (Krallinger et al., 2008) is a task to extract PPI from PubMed abstracts. BioNLP'09 Shared Task on Event Extraction (Kim et al., 2009) is a task to extract bio-molecular events (bio-events) from the GENIA event corpus.

Similar characteristics to those of the adverse–effect relation are described in previous reports in the bio-medical domain. Friedman et al. (1994) describes the certainty in findings of clinical radiology. Certainty is also known in scientific papers of biomedical domains as *speculation* (Light et al., 2004). Vincze et al. (2008) are producing a freely available corpus including annotations of uncertainty along with its scope.

Dependency structure feature which we utilized to extract adverse–effect relations are widely used in relation extraction tasks. We present previous works which used syntactic/dependency information as a feature of a statistical method. Beamer et al. (2007), Giuliano et al. (2007), and Hendrickx et al. (2007) all used syntactic information with machine learning techniques in SemEval-2007 Task:04 and achieved good performance. Riedel et al. (2009) used dependency path features with a statistical relational learning method in BioNLP'09 Shared Task on Event Extraction and achieved the best performance in the event enrichment subtask. Miyao et al. (2008) compared syntactic information of various statistical parsers on PPI.

3 Corpus

We produced an annotated corpus of adverse–effect relations to develop and test an adverse–effect relation extraction method. This section presents a description of details of the corpus.

3.1 Texts Comprising the Corpus

We used a discharge summary among various documents in a hospital as the source data of the task. The discharge summary is a document created by a doctor or another medical expert at the conclusion of a hospital stay. Medications performed during a stay are written in discharge summaries. If adverse–effect relations were observed during the stay, they are likely to be expressed in free text. Texts written in discharge summaries tend to be written more roughly than texts in newspaper articles or scientific papers. For example, the amounts of medications are often written in a name-value list as shown below:
"When admitted to the hospital, Artist 6 mg1x, Diovan 70 mg1x, Norvasac 5 mg1x and BP was 145/83, but after dialysis, BP showed a decreasing tendency and in 5/14 Norvasac was reduced to 2.5 mg1x."

3.2 Why Adverse–Effect Relation Extraction from Discharge Summaries is Important

In many countries, adverse–effects are investigated through multiple phases of clinical trials, but unexpected adverse–effects occur in actual medications. One reason why this occurs is that drugs are often used in combination with others in actual medications. Clinical trials usually target single drug use. For that reason, the combinatory uses of drugs occasionally engender unknown effects. This situation naturally motivates automatic adverse–effect relation extraction from actual patient records.

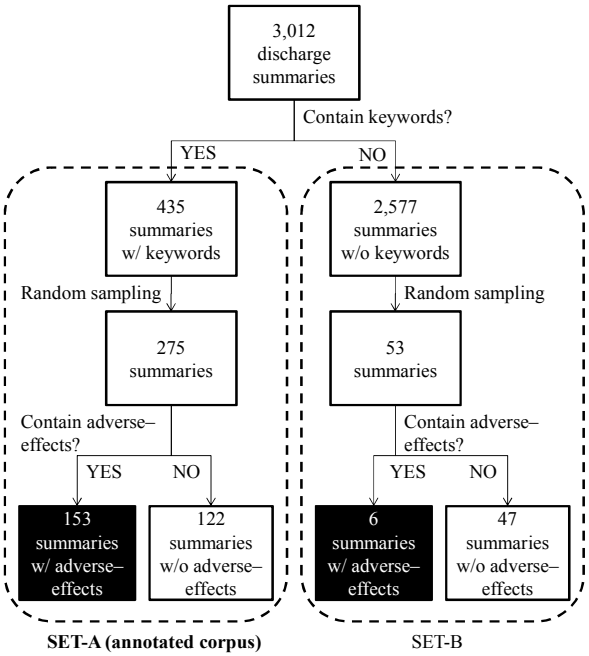

Figure 4. The overview of the summary selection.

3.3 Corpus Size

We collected 3,012 discharge summaries[1] written in Japanese from all departments of a hospital. To reduce a cost to survey the occurrence of adverse–effects in the summaries, we first split the summaries into two sets: SET-A, which contains keywords related to adverse–effects and SET-B, which do not contain the keywords. The keywords we used were "stop, change, adverse effect", and they were chosen based on a heuristic. The keyword filtering resulted to SET-A with 435 summaries and SET-B with 2,577 summaries. Regarding SET-A, we randomly sampled 275 summaries and four annotators annotated adverse–effect information to these summaries to create the adverse–effect relation corpus. For SET-B, the four annotators checked the small portion of the summaries. Cases of ambiguity were resolved through discussion, and even suspicious adverse–effect relations were annotated in the corpus as positive data. The overview of the summary selection is presented in Figure 4.

[1] All private information was removed from them. The definition of private information was referred from the HIPAA guidelines.

Table 1. Markup scheme.

tag	Definition and Examples
drug	The expression of an administrated drug: e.g. *Levofloxacin, Flexeril.*
symptom	The expression of a disease or symptom: e.g. *endometrial cancer, headache*. This tag covers not only a noun phrase but also a verb phrase such as "<symptom>*feels a pain in front of the head*</symptom>".

Table 2. Annotation examples.

<drug relation="1">*Ridora*</drug> resumed because it is associated with an <symptom relation="1">*eczematous rash*</symptom>.
<drug relation="1">ACTOS(30)</drug> brought both <symptom relation="1">*headache*<symptom> and <symptom relation="1">*insomnia*</symptom>.

* If a drug has two or more adverse-effects, symptoms take a same relation ID.

3.4 Quantities of Adverse–Effects in Clinical Texts

55.6% (=158/275) of the summaries in SET-A contained adverse–effects. 11.3% (=6/53) of the summaries in SET-B contained adverse–effects. Since the ratio of SET-A:SET-B is 14.4:85.6, we estimated that about 17.7% (=0.556×0.144+0.113×0.856) of the summaries contain adverse–effects. Even considering that a summary may only include suspected adverse–effects, we think that discharge summaries are a valuable resource to explore adverse–effects.

3.5 Annotated Information

We annotated information of two kinds to the corpus: term information and relation information.

(1) Term Annotation

Term annotation includes two tags: a tag to express a drug and a tag to express a drug effect. Table 1 presents the definition. In the corpus, 2,739 drugs and 12,391 effects were annotated.

(2) Relation Annotation

Adverse–effect relations are annotated as the "relation" attribute of the term tags. We represent the effect of a drug as a relation between a drug tag and a symptom tag. Table 2 presents

Table 3. Features used in adverse-effect extraction.

ID	Feature	Definition and Examples
1	Character Distance	The number of characters between members of a pair.
2	Morpheme Distance	The number of morpheme between members of a pair.
3	Pair Order	Order in which a drug and a symptom appear in a text; "drug–symptom" or "symptom–drug".
4	Symptom Type	The type of symptom: "disease name", "medical test name", or "medical test value".
5	Morpheme Chain	Base–forms of morphemes that appear between a pair.
6	Dependency Chain	Base–forms of morphemes included in the minimal dependency path of a pair.
7	Case Frame Chain	Verb, case frame, and object triples that appear between a pair: e.g. "examine" –"*de*"(case particle) – "inhalation", "begin" –"*wo*"(case particle) –"medication".
8	Case Frame Dependency Chain	Verb, case frame, and object triples included in the minimal dependency path of a pair.

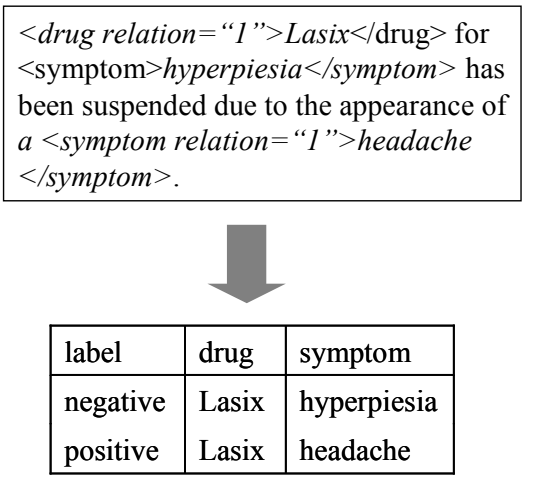

Figure 5. Pair extraction example.

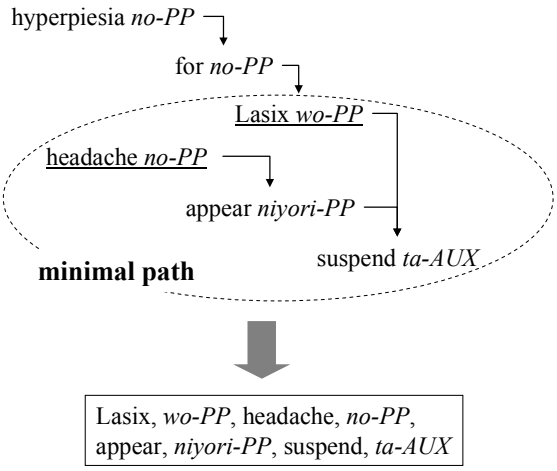

Figure 6. Dependency chain example.

several examples, wherein "relation=1" denotes the ID of a adverse–effect relation. In the corpus, 236 relations were annotated.

4 Extraction Method

We present a simple adverse–effect relation extraction method. We extract drug–symptom pairs from the corpus and discriminate them using a machine-learning technique. Features based on morphological analysis and dependency analysis are used in discrimination. This approach is similar to the PPI extraction approach of Miyao et al. (2008), in which we binary classify pairs whether they are in adverse–effect relations or not. A pattern-based semi-supervised approach like Saeger et al. (2008), or more generally Espresso (Pantel and Pennacchiotti, 2006), can also be taken, but we chose a pair classification approach to avoid the effect of seed patterns. To capture a view of an adverseness of a drug, a statistic of adverse–effect relations is important. We do not want to favor certain patterns and chose a pair classification approach to equally treat every relation. Extraction steps of our method are as presented below.

STEP 1: Pair Extraction

All combinations of drug–symptom pairs that appear in a same sentence are extracted. Pairs

Table 4. Best F_1-scores and their parameters.

ID	Feature Combination	Parameters	Precision	Recall	F_1-score
A	1,2,3,4,5	log(c)=3.0, log(g)=-5.0, p=0.10	26.72	46.21	33.05
B	1,2,3,4,5,6	log(c)=1.0, log(g)=-5.0, p=0.10	33.30	42.43	36.64
C	**1,2,3,4,5,6,7**	**log(c)=1.0, log(g)=-5.0, p=0.10**	**34.39**	**43.06**	**37.54**
D	1,2,3,4,5,6,8	log(c)=1.0, log(g)=-5.0, p=0.10	35.01	40.67	36.78
E	1,2,3,4,5,6,7,8	log(c)=1.0, log(g)=-5.0, p=0.10	35.45	41.05	37.18

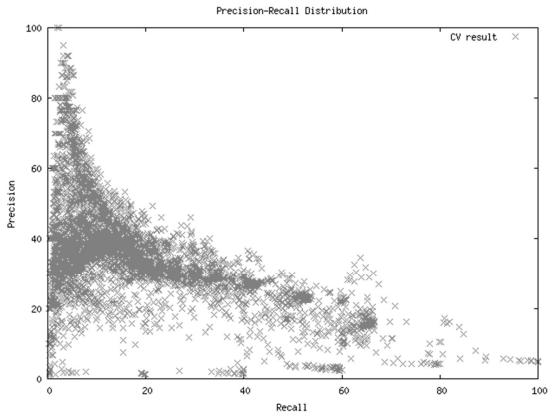

Figure 7. Precision–recall distribution.

with the same relation ID become positive samples; pairs with different relation IDs become negative samples. Figure 5 shows examples of positive and negative samples.

STEP 2: Feature Extraction

Features presented in Table 3 are extracted. The text in the corpus is in Japanese. Some features assume widely known characteristics of Japanese. For example, the dependency feature allows a phrase to depend on only one phrase that appears after a dependent phrase. Figure 6 portrays an example of a dependency chain feature. In the example, most terms were translated into English, excluding postpositions (PP) and auxiliaries (AUX), which are expressed in italic. To reduce the negative effect of feature sparsity, features which appeared in more than three summaries are used for features with respective IDs 5–8.

STEP 3: Machine Learning

The support vector machine (SVM) (Vapnik, 1995) is trained using positive/negative labels and features extracted in prior steps. In testing, an unlabeled pair is given a positive or negative label with the trained SVM.

5 Experiment

We performed two experiments to evaluate the extraction method.

5.1 Experiment 1

Experiment 1 aimed to observe the effects of the presented features. Five combinations of the features were evaluated with a five-fold cross validation assuming that an optimal parameter combination was obtained. The experiment conditions are described below:

A. Data

7,690 drug–symptom pairs were extracted from the corpus. Manually annotated information was used to identify drugs and symptoms. Within 7,690 pairs, 149 pairs failed to extract the dependency chain feature. We removed these 149 pairs and used the remaining 7,541 pairs in the experiment. The 7,541 pairs consisted of 367 positive samples and 7,174 negative samples.

B. Feature Combinations

We tested the five combinations of features in the experiment. Manually annotated information was used for the symptom type feature. Features related to morphemes are obtained by processing sentences with a Japanese morphology analyzer (JUMAN[2] ver. 6.0). Features related to dependency and case are obtained by processing sentences using a Japanese dependency parser (KNP ver. 3.0; Kurohashi and Nagao, 1994).

C. Evaluations

We evaluated the extraction method with all combinations of SVM parameters in certain

[2] http://www-lab25.kuee.kyoto-u.ac.jp/nl-resource/juman-e.html

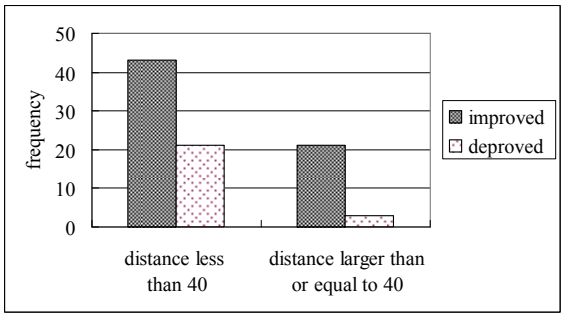

Figure 8. Relation between the number of pairs and the morpheme distance.

ranges. We used LIBSVM[3] ver. 2.89 as an implementation of SVM. The radial basis function (RBF) was used as the kernel function of SVM. The probability estimates option of LIBSVM was used to obtain the confidence value of discrimination**.**

The *gamma* parameter of the RBF kernel was chosen from the range of $[2^{-20}, 2^{0}]$. The *C* parameter of SVM was chosen from the range of $[2^{-10}, 2^{10}]$. The SVM was trained and tested on 441 combinations of *gamma* and *C*. In testing, the probability threshold parameter *p* between [0.05, 0.95] was also chosen, and the F_1-scores of all combination of *gamma*, *C*, and *p* were calculated with five-fold cross validation. The best F_1-scores and their parameter values for each combination of features (optimal F_1-scores in this setting) are portrayed in Table 4. The precision–recall distribution of F_1-scores with feature combination C is presented in Figure 7.

5.2 Experiment 2

Experiment 2 aimed to observe the performance of our extraction method when SVM parameters were automatically tuned. In this experiment, we performed two cross validations: a cross validation to tune SVM parameters and another cross validation to evaluate the extraction method. The experiment conditions are described below:

A. Data

The same data as Experiment 1 were used.

B. Feature Combination

Feature combination C, which performed best in Experiment 1, was used.

C. Evaluation

[3] http://www.csie.ntu.edu.tw/~cjlin/libsvm

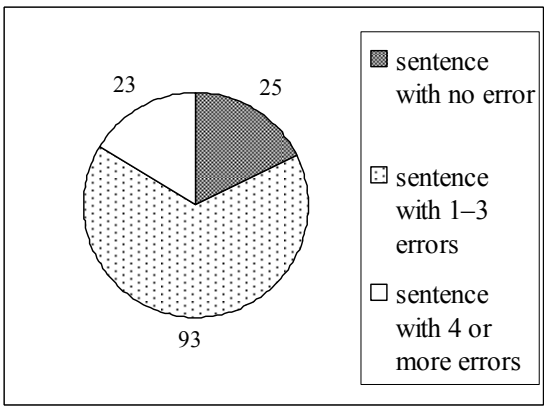

Figure 9. Number of dependency errors in the improved pairs sentences.

Two five-fold cross validations were performed. The first cross validation divided the data to 5 sets (A, B, C, D, and E) each consisting of development set and test set with the ratio of 4:1. The second cross validation train and test all combination of SVM parameters (*C*, *gamma*, and *p*) in certain ranges and decide the optimal parameter combination(s) for the development sets of A, B, C, D, and E. The second cross validation denotes the execution of Experiment 1 for each development set. For each optimal parameter combination of A, B, C, D, and E, the corresponding development set was trained and the trained model was tested on the corresponding test set. The average F_1-score on five test sets marked 34.90, which is 2.64 lower than the F_1-score of Experiment 1 with the same feature combination.

6 Discussion

The result of the experiment reveals the effectiveness of the dependency chain feature and the case-frame chain feature. This section presents a description of the effects of several features in detail. The section also mentions remaining problems in our extraction method.

6.1 Effects of the Dependency Chain Feature and Case-frame Features

A. Dependency Chain Feature

The dependency chain features improved the F_1-score by 3.59 (the F_1-score difference between feature combination A and B). This increase was obtained using 260 improved pairs and 127 deproved pairs. Improved pairs con-

tribute to the increase of a F_1-score. Deproved pairs have the opposite effect.

We observed that improved pairs tend to have longer morpheme distance compared to deproved pairs. Figure 8 shows the relation between the number of pairs and the morpheme distance of improved pairs and deproved pairs. The ratio between the improved pairs and the deproved pairs is 11:1 when the distance is greater than 40. In contrast, the ratio is 2:1 when the distance is smaller than 40. This observation suggests that adverse–effect relations share dependency structures to a certain degree.

We also observed that in improved pairs, dependency errors tended to be low. Figure 9 presents the manually counted number of dependency errors in the 141 sentences in which the 260 improved pairs exist: 65.96 % of the sentences included 1–3 errors. The result suggests that the dependency structure is effective even if it includes small errors.

B. Case-frame Features

The effect of the case-frame dependency chain feature differed with the effect of the dependency chain feature. The case-frame chain feature improved the F_1-score by 0.90 (the F_1-score difference between feature combination B and C), but the case-frame dependency chain feature decreased the F_1-score by 0.36 (the F_1-score difference between feature combination C and E). One reason for the negative effect of the case-frame dependency feature might be feature sparsity, but no clear evidence of it has been found.

6.2 Remaining Problems

A. Imbalanced Data

The adverse–effect relation pairs we used in the experiment were not balanced. Low values of optimal probability threshold parameter *p* suggest the degree of imbalance. We are considering introduction of some kind of methodology to reduce negative samples or to use a machine learning method that can accommodate imbalanced data well.

B. Use of Medical Resources

The extraction method we propose uses no medical resources. Girju et al. (2007) indicate the effect of WordNet senses in the classification of a semantic relation between nominals. Krallinger et al. (2008) report that top scoring teams in the interaction pair subtask used sophisticated interactor protein normalization strategies. If medical terms in texts can be mapped to a medical terminology or ontology, it would likely improve the extraction accuracy.

C. Fully Automated Extraction

In the experiments, we used the manually annotated information to extract pairs and features. This setting is, of course, not real if we consider a situation to extract adverse–effect relations from massive clinical records, but we chose it to focus on the relation extraction problem. We performed an event recognition experiment (Aramaki et al., 2009) and achieved F_1-score of about 80. We assume that drug expressions and symptom expressions to be automatically recognized in a similar accuracy.

We are planning to perform a fully automated adverse–effect relations extraction from a larger set of clinical texts to see the performance of our method on a raw corpus. The extraction F_1-score will likely to decrease, but we intend to observe the other aspect of the extraction, like the overall tendency of extracted relations.

7 Conclusion

We presented a method to extract adverse–effect relations from texts. One important characteristic of adverse–effect relations is that they are uncertain in most cases. We performed experiments to extract adverse–effect relations from 2,577 clinical texts, and obtained F_1-score of 37.54 with optimal SVM parameters and F_1-score of 34.90 with automatically tuned SVM parameters. Results also show that dependency features increase the extraction F_1-score by 3.59. We observed that an increased F_1-score was obtained using the improvement of adverse–effects with long morpheme distance, which suggests that adverse–effect relations share dependency structures to a certain degree. We also observed that the increase of the F_1-score was obtained with dependency structures that include small errors, which suggests that the dependency structure is effective even if it includes small errors.

References

Aramaki, Eiji, Yasuhide Miura, Masatsugu Tonoike, Tomoko Ohkuma, Hiroshi Masuichi, and Kazuhiko Ohe. 2009. TEXT2TABLE: Medical Text Summarization System Based on Named Entity Recognition and Modality Identification. In *Proceedings of the BioNLP 2009 Workshop*, pages 185-192.

Beamer, Brandon, Suma Bhat, Brant Chee, Andrew Fister, Alla Rozovskaya, and Roxana Girju. 2007. UIUC: A Knowledge-rich Approach to Identifying Semantic Relations between Nominals. In *Proceedings of Fourth International Workshop on Semantic Evaluations*, pages 386-389.

Friedman, Carol, Philip O. Alderson, John H. M. Austin, James J. Cimino, and Stephen B. Johnson. 1994. A General Natural-language Text Processor for Clinical Radiology. Journal of the American Medical Informatics Association, 1(2), pages 161-174.

Gildea, Daniel. 2001. Corpus Variation and Parser Performance. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, pages 1-9.

Girju, Roxana, Preslav Nakov, Vivi Nastase, Stan Szpakowicz, Peter Turney, and Deniz Yuret. 2007. SemEval-2007 task 04: Classification of Semantic Relations between Nominals. In *Proceedings of Fourth International Workshop on Semantic Evaluations*, pages 13-18.

Giuliano, Claudio, Alberto Lavelli, Daniele Pighin, and Lorenza Romano. 2007. FBK-IRST: Kernel Methods for Semantic Relation Extraction. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 141-144.

Hendrickx , Iris, Roser Morante, Caroline Sporleder, and Antal van den Bosch. 2007. ILK: Machine learning of semantic relations with shallow features and almost no data. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, 187-190.

Kim, Jin-Dong, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun'ichi Tsujii. 2009. Overview of BioNLP'09 Shared Task on Event Extraction. In *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, pages 1-9.

Krallinger, Martin, Florian Leitner, Carlos Rodriguez-Penagos, and Alfonso Valencia. 2008. Overview of the protein-protein interaction annotation extraction task of BioCreative II. Genome Biology 2008, 9(Suppl 2):S4.

Kurohashi, Sadao and Makoto Nagao. 1994. KN Parser : Japanese Dependency/Case Structure Analyzer. In *Proceedings of The International Workshop on Sharable Natural Language Resources*, pages 22-28. Software available at http://www-lab25.kuee.kyoto-u.ac.jp/nl-resource/knp-e.html.

Light, Marc, Xin Ying Qiu, and Padmini Srinivasan. 2004. The Language of Bioscience: Facts, Speculations, and Statements in Between. In *Proceedings of HLT/NAACL 2004 Workshop: BioLINK 2004, Linking Biological Literature, Ontologies and Databases*, pages 17-24.

Miyao, Yusuke, Rune Sætre, Kenji Sagae, Takuya Matsuzaki, and Jun'ichi Tsujii. 2008. Task-oriented Evaluation of Syntactic Parsers and Their Representations. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 46-54.

Pantel, Patrick and Marco Pennacchiotti. 2006. Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 113-120.

Riedel, Sebastian, Hong-Woo Chun, Toshihisa Takagi, and Jun'ichi Tsujii. 2009. A Markov Logic Approach to Bio-Molecular Event Extraction. In *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, pages 41-49.

Saeger, Stijn De, Kentaro Torisawa, and Jun'ichi Kazama. 2008. Looking for Trouble. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 185-192.

Vapnik, Vladimir N.. 1995. The Nature of Statistical Learning Theory. Springer-Verlag New York, Inc..

Vincze, Veronika, György Szarvas, Richárd Farkas, György Móra, and János Csirik. 2008. The BioScope corpus: biomedical texts annotated for uncertainty, negation and their scopes. BMC Bioinformatics 2008, 9(Suppl 11):S9.

Author Index