

Precise Information Retrieval Exploiting Predicate-Argument Structures

Daisuke Kawahara[†] Keiji Shinzato[‡] Tomohide Shibata[†] Sadao Kurohashi[†]

[†]Graduate School of Informatics, Kyoto University

[‡]Rakuten Institute of Technology

{dk, shibata, kuro}@i.kyoto-u.ac.jp, keiji.shinzato@mail.rakuten.com

Abstract

A concept can be linguistically expressed in various syntactic constructions. Such syntactic variations spoil the effectiveness of incorporating dependencies between words into information retrieval systems. This paper presents an information retrieval method for normalizing syntactic variations via predicate-argument structures. We conduct experiments on standard test collections and show the effectiveness of our approach. Our proposed method significantly outperforms a baseline method based on word dependencies.

1 Introduction

Most conventional approaches to information retrieval (IR) deal with words as independent terms. In query sentences¹ and documents, however, dependencies exist between words.² To capture these dependencies, some extended IR models have been proposed in the last decade (Jones, 1999; Lee et al., 2006; Song et al., 2008; Shinzato et al., 2008). These models, however, did not achieve consistent significant improvements over models based on independent words.

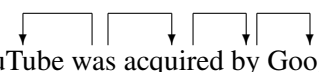
One of the reasons for this is the linguistic variations of syntax, that is, languages are syntactically expressed in various ways. For instance, the same or similar meaning can be expressed using the passive voice or the active voice in a sentence. Previous approaches based on dependencies cannot identify such variations. This is because they use the output of a dependency parser, which generates syntactic (grammatical) dependencies built

¹In this paper, we handle queries written in natural language.

²While dependencies between words are sometimes considered to be the co-occurrence of words in a sentence, in this paper we consider dependencies to be syntactic or semantic dependencies between words.

upon surface word sequences. Consider, for example, the following sentence in a document:

(1) YouTube was acquired by Google.



Dependency parsers based on the Penn Treebank and the head percolation table (Collins, 1999) judge the head of “YouTube” as “was” (“YouTube←was”; hereafter, we denote a dependency by “*modifier←head*”). This dependency, however, cannot be matched with the dependency “YouTube←acquire” in a query like:

(2) I want to know the details of the news that Google acquired YouTube.

Furthermore, even if a dependency link in a query matches that in a document, a mismatch of dependency type can cause another problem. This is because previous models did not distinguish dependency types. For example, the dependency “YouTube←acquire” in query sentence (2) can be found in the following irrelevant document.

(3) Google acquired PushLife for \$25M ...
YouTube acquired Green Parrot Pictures ...

While this document does indeed contain the dependency “YouTube←acquire,” its type is different; specifically, the query dependency is accusative while the document dependency is nominative. That is to say, ignoring differences in dependency types can lead to inaccurate information retrieval.

In this paper, we propose an IR method that does not use syntactic dependencies, but rather predicate-argument structures, which are normalized forms of sentence meanings. For example, query sentence (2) is interpreted as the following predicate-argument structure (hereafter, we denote a predicate-argument structure by $\langle \cdot \cdot \cdot \rangle$):³

³In this paper, we use the following abbreviations:

(4) ⟨NOM:Google acquire ACC:YouTube⟩.

Sentence (1) is also represented as the same predicate-argument structure, and documents including this sentence can be regarded as relevant documents. Conversely, the irrelevant document (3) has different predicate-argument structures from (4), as follows:

(5) a. ⟨NOM:Google acquire ACC:PushLife⟩,

b. ⟨NOM:YouTube acquire ACC:Green Parrot Pictures⟩.

In this way, by considering this kind of predicate-argument structure, more precise information retrieval is possible.

We mainly evaluate our proposed method using the NTCIR test collection, which consists of approximately 11 million Japanese web documents. We also have an experiment on the TREC Robust 2004 test collection, which consists of around half a million English documents, to validate the applicability to other languages than Japanese.

This paper is organized as follows. Section 2 introduces related work, and section 3 describes our proposed method. Section 4 presents the experimental results and discussion. Section 5 describes the conclusions.

2 Related work

There have been two streams of related work that considers dependencies between words in a query sentence.

One stream is based on linguistically-motivated approaches that exploit natural language analysis to identify dependencies between words. For example, Jones proposed an information retrieval method that exploits linguistically-motivated analysis, especially dependency relations (Jones, 1999). However, Jones noted that dependency relations did not contribute to significantly improving performance due to the low accuracy and robustness of syntactic parsers. Subsequently, both the accuracy and robustness of dependency parsers were dramatically improved (Nivre and Scholz, 2004; McDonald et al., 2005), with such parsers being applied more recently to information retrieval (Lee et al., 2006; Song et al., 2008; Shin-

NOM (nominative), ACC (accusative), DAT (dative), ALL (allative), GEN (genitive), CMI (comitative), LOC (locative), ABL (ablative), CMP (comparative), DEL (delimitative) and TOP (topic marker).

zato et al., 2008). For example, Shinzato et al. investigated the use of syntactic dependency output by a dependency parser and reported a slight improvement over a baseline method that used only words. However, the use of dependency parsers still introduces the problems stated in the previous section because of their handling of only syntactic dependencies.

The second stream of research has attempted to integrate dependencies between words into information retrieval models. These models include a dependence language model (Gao et al., 2004), a Markov Random Field model (Metzler and Croft, 2005), and a quasi-synchronous dependence model (Park et al., 2011). However, they focus on integrating term dependencies into their respective models without explicitly considering any syntactic or semantic structures in language. Therefore, the purpose of these studies can be considered different from ours.

Park and Croft (2010) proposed a method for ranking query terms for the selection of those which were most effective by exploiting typed dependencies in the analysis of query sentences. They did not, however, use typed dependencies for indexing documents.

The work that is closest to our present work is that of Miyao et al. (2006), which proposed a method for the semantic retrieval of relational concepts in the domain of biomedicine. They retrieved sentences that match a given query using predicate-argument structures via a framework of region algebra. Thus, they namely approached the task of sentence matching, which is not the same as document retrieval (or ranking). As for the types of queries they used, although their method could handle natural language queries, they used short queries like “TNF activate IL6.” Because of the heavy computational load of region algebra, if a query matches several thousand sentences, for example, then it requires several thousand seconds to return all sentence matches (though it takes on average 0.01 second to return the first matched sentence).

In the area of question answering, predicate-argument structures have been used to precisely match a query with a passage in a document (e.g., (Narayanan and Harabagiu, 2004; Shen and Lapata, 2007; Bilotti et al., 2010)). However, candidate documents to extract an answer are retrieved using conventional search engines without

predicate-argument structures.

3 Information retrieval exploiting predicate-argument structures

3.1 Overview

Our key idea is to exploit the normalization of linguistic expressions based on their predicate-argument structures to improve information retrieval.

The process of information retrieval systems can be decomposed into offline processing and online processing. During offline processing, analysis is first applied to a document collection. For example, typical analyses for English include tokenization and stemming analyses, while those for Japanese include morphological analysis. In addition, previous models using the dependencies between words also used dependency parsing. In this paper, we employ predicate-argument structures analysis, which is detailed in the next subsection.

Following the initial analysis, indexing is performed to produce an inverted index. In most cases, words are indexed as terms, but several previous approaches have also indexed dependencies between words as terms (e.g., (Shinzato et al., 2008)). In our study, however, we do not use syntactic dependencies directly, but rather consider predicate-argument structures. To bring this predicate-argument structure information into the index, we handle predicate-argument structures as a set of typed semantic dependencies. Dependency types are expressed as term features, which are additional information to each term including the list of positions of the term.

As for online processing, we first apply the predicate-argument structure analysis to a query sentence, and then create terms including words and typed semantic dependencies extracted from the predicate-argument structures. Then, we search documents containing these terms from the inverted index, and then finally rank these documents.

In the following subsections, we describe in more detail the procedures of predicate-argument structure analysis, indexing, query processing, and document ranking.

3.2 Analysis of predicate-argument structures

We apply predicate-argument structure analysis to both queries and documents. Predicate-argument

structure analysis normalizes the following linguistic expressions:

- relative clause
- passive voice (the predicate is normalized to active voice)
- causative (the predicate is normalized to normal form)
- intransitive (the predicate is normalized to transitive)
- giving and receiving expressions (the predicate is normalized to a giving expression)

In the case of Japanese, we use the morphological analyzer JUMAN,⁴ and the predicate-argument structure analyzer KNP (Kawahara and Kurohashi, 2006).⁵ The accuracy of syntactic dependencies output by KNP is around 89% and that of predicate-argument relations is around 81% on web sentences. Examples of this predicate-argument structure analysis are shown in Figures 1 and 2. Figure 1 shows an example of relative clause normalization by predicate-argument structure analysis. The syntactic dependencies of the two sentences are different, but this difference is solved by using predicate-argument structures.

Figure 2 shows an example of intransitive verb normalization by predicate-argument structure analysis. In this example, the syntactic dependencies are the same, but different verbs are used.⁶ The analyzer canonicalizes the intransitive verb to its corresponding transitive verb, and also produces the same predicate-argument structure for the two sentences.

If we apply our method to English, deep parsers such as the Stanford Parser⁷ and Enju⁸ can be employed to achieve predicate-argument structure analysis. The Stanford parser can output typed semantic dependencies that conform to the Stanford dependencies (de Marneffe et al., 2006). Enju is an HPSG parser that outputs predicate-argument structures, and arguments are typed as Arg1, Arg2, and so forth. The representation of the dependency types in Enju is the same as that of PropBank (Palmer et al., 2005).

⁴<http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?JUMAN>

⁵<http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?KNP>

⁶In many cases, the lemma of a transitive verb is not the same as that of its corresponding intransitive verb in Japanese.

⁷<http://www-nlp.stanford.edu/software/lex-parser.shtml>

⁸<http://www.nactem.ac.uk/enju/>

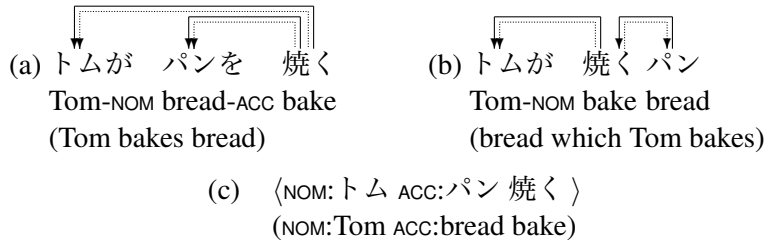


Figure 1: An example of relative clause normalization by predicate-argument structure analysis in Japanese. (a) is a normal-order sentence and (b) is a sentence that contains a relative clause, “トムが焼く” (which Tom bakes). Arrows represent syntactic dependencies. Dotted arrows represent semantic dependencies that constitute predicate-argument structures. Both sentences are normalized to the predicate-argument structure (c).

In this way, though our framework itself is language-independent, our method depends on the availability of a predicate-argument structure analyzer for the target language.

3.3 Indexing

Our method builds an inverted index from the results of the predicate-argument structure analysis. First, word lemmas are registered as terms. We then need to integrate the predicate-argument structure information into the index. One possibility is to represent each predicate-argument structure as a term, but this method leads to a data sparseness problem. This is because the number of arguments in predicate-argument structures varies greatly not only in documents, but also in queries because of information granularity. For example, to express the same event, a predicate-argument structure can omit time or place information.

Instead, we decompose a predicate-argument structure into a set of typed semantic dependencies. A typed semantic dependency is defined as a typed dependency between a predicate and an argument that the predicate governs. For instance, the predicate-argument structure in Figure 2 can be decomposed into the following two typed semantic dependencies:

- (6) a. トム $\xleftarrow{\text{NOM}}$ 上げる
(Tom $\xleftarrow{\text{NOM}}$ raise)
- b. テンション $\xleftarrow{\text{ACC}}$ 上げる
(tension $\xleftarrow{\text{ACC}}$ raise)

These typed semantic dependencies are registered as dependency terms in the index. The type information is encoded as a *term feature*, which is an additional field for each dependency term. This term feature consists of both dependency type information and predicate information. We con-

sider major postpositions in Japanese as dependency types (Table 1). If a dependency type is not listed in this table, then this type is regarded as a special type which we classify as “other.” In addition, a dependency that is not the relation between a predicate and its argument is also classified as “other” (e.g., the dependency between verbs).

The predicate information in the term feature refers to the original predicate type for canonicalized predicates. There are four types: passive, causative, intransitive, and giving expression.

3.4 Query processing

Hereafter, we describe the steps of online processing. When a query sentence is input, both predicate-argument structure analysis and term extraction are applied to the query sentence in the same way indexing is applied. The extracted terms consist of words and typed semantic dependencies and they are used to retrieve documents.

Note that unnecessary expressions like “教えてください” (please tell me) in a query sentence are not used to extract terms.

3.5 Document retrieval and scoring

Using the results of the query processing, documents are then retrieved and ranked. First, documents are retrieved by accessing the inverted index using the terms extracted from the query analysis. Here, we have two options for the logical operator on the terms. If we apply the logical operator AND, we impose a constraint that all the terms must be contained in a retrieved document. Conversely, if we apply the logical operator OR, a retrieved document should have one of the terms. In this study, we use the logical operator OR to retrieve as many documents as possible. This means that we do not apply any methods of selecting or

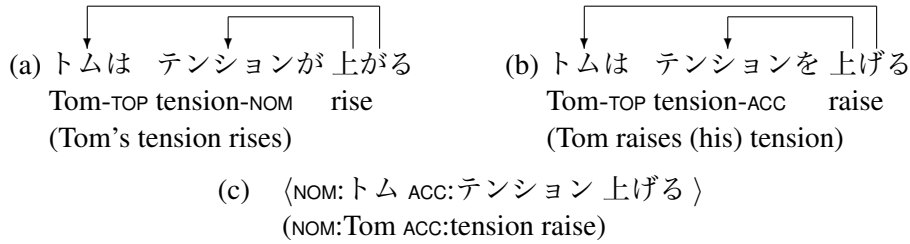


Figure 2: An example of intransitive verb normalization by predicate-argument structure analysis in Japanese. (a) is an intransitive sentence and (b) is a transitive sentence. Arrows represent syntactic dependencies (they are also semantic dependencies in this case). Both sentences are normalized to the predicate-argument structure (c). In particular, the intransitive verb “上がる” (rise) is a different word from the transitive verb “上げる” (raise) but both are canonicalized to the same transitive verb “上げる” (raise) in the predicate-argument structure.

| | | | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|------|------------|-----|-------|-----|
| が | を | に | と | で | から | まで | より | 時間 | 修飾 | の | について | として |
| NOM | ACC | DAT | CNJ | LOC | ABL | DEL | CMP | time | <i>adj</i> | GEN | about | as |

Table 1: Dependency type information in Japanese. The first row is the list of dependency types used in our method. The second row means the translations of the first row, where *adj* means adjuncts such as adverbs.

ranking query terms,⁹ but rely only on document scoring to examine the effectiveness of the use of predicate-argument structures.

Following document retrieval, a relevancy score is assigned to each document, and the documents are ranked according to these relevancy scores. We use Okapi BM25 (Robertson et al., 1992) for estimating the relevancy score between a query and a document. This measure was originally proposed for models based on terms of independent words, but we slightly extend this measure to include estimating relevancy for typed semantic dependencies that are extracted from predicate-argument structures. Our relevancy score is calculated as a weighted sum of the score of words and the score of dependencies. The score of dependencies is further calculated as a weighted sum of the following two scores: the score of dependencies with consistent (matched) type and that with inconsistent (mismatched) type. In particular, the score of dependencies with inconsistent type is reduced compared to the score of dependencies with consistent type.

We denote a set of words in a query q as T_{qw} , and also denote a set of dependencies in q as T_{qd} . This set of dependencies is further divided into two types according to the consistency of dependency features: T_{qd_C} (consistent) and T_{qd_I} (inconsistent). We define the relevancy score between

query q and document d as follows:

$$R(q, d) = \sum_{t \in T_{qw}} BM(t, d) + \beta \left\{ \sum_{t \in T_{qd_C}} BM(t, d) + \gamma \sum_{t \in T_{qd_I}} BM(t, d) \right\}, \quad (1)$$

where β is a parameter for adjusting the ratio of a score calculated from dependency relations to that from words and γ is a parameter for decreasing the weight of inconsistent dependency types. The score $BM(t, d)$ is defined as:

$$BM(t, d) = IDF(t) \times \frac{(k_1+1)F_{dt}}{K+F_{dt}} \times \frac{(k_3+1)F_{qt}}{k_3+F_{qt}}, \quad (2)$$

$$IDF(t) = \log \frac{N-n+0.5}{n+0.5},$$

$$K = k_1 \left\{ (1-b) + b \frac{l_d}{l_{ave}} \right\},$$

where 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, we set these Okapi parameters as $k_1 = 1$, $k_3 = 0$ and $b = 0.6$.

We use the following relevancy score for a baseline method that uses only syntactic dependencies, which is explained in section 4:

$$R(q, d) = \sum_{t \in T_{qw}} BM(t, d) + \beta \sum_{t \in T_{qd}} BM(t, d). \quad (3)$$

⁹We only discard unnecessary expressions in a query as described in subsection 3.4.

This equation is the same as the relevancy score used in Shinzato et al. (2008).

4 Evaluation

In this section, we evaluate and analyze our proposed method on the standard test collections of Japanese and English.

4.1 Evaluation on Japanese Test Collection

4.1.1 Experimental setup

We implemented our proposed method using the open search engine infrastructure TSUBAKI (Shinzato et al., 2008) as a base system. TSUBAKI generates an inverted index from linguistic analyses in an XML format. Note that while TSUBAKI has a facility for using a synonym lexicon, but we did not use it because we performed pure comparisons without referencing synonyms.

We evaluated our proposed method by using the test collection built for the NTCIR-3 (Eguchi et al., 2003) and NTCIR-4 (Eguchi et al., 2004) workshops. These workshops shared a target document set, which consists of 11,038,720 web pages from Japanese domains. We used a high-performance computing environment to perform predicate-argument structure analysis and indexing on these documents. It took three days for analysis and two days for indexing. For the evaluation, we used 127 informational topics (descriptions) defined in the test collections (47 from NTCIR-3 and 80 from NTCIR-4). We also had additional 65 topics that were not used for evaluation in NTCIR-3; we used these 65 topics for parameter tuning. 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.

For each topic, we retrieved 1,000 documents, ranked according to the score $R(q, d)$ in equation (1). We optimized the parameter β as 0.18, and the parameter γ as 0.85 using the additional 65 topics in relation to their mean average precision (MAP) score. We then assessed retrieval performance according to MAP, P@3 (Precision at 3), P@5, P@10 and nDCG@10 (Järvelin and Kekäläinen, 2002). Note that unjudged documents were treated as irrelevant when computing the scores. For the graded relevance of nDCG@10, we mapped highly relevant, relevant, and partially relevant to the values 3, 2, and 1, respectively.

| | MAP | P@3 | P@5 | P@10 | nDCG@10 |
|----------|-----------------|----------------|---------------|----------------|-----------------|
| word | 0.1665 | 0.4233 | 0.4159 | 0.3706 | 0.2323 |
| word+dep | 0.1704 | 0.4233 | 0.4095 | 0.3730 | 0.2313 |
| word+pa | 0.1727** | 0.4418* | 0.4175 | 0.3794* | 0.2370** |

Table 2: Retrieval performance of two baseline methods (“word” and “word+dep”) and our proposed method (“word+pa”). ** and * mean that the differences between “word+dep” and “word+pa” are statistically significant with $p < 0.05$ and $p < 0.10$, respectively.

| | MAP | P@3 | P@5 | P@10 | nDCG@10 |
|----------|-----------------|---------------|---------------|-----------------|-----------------|
| word | 0.2085 | 0.4312 | 0.4302 | 0.3960 | 0.2455 |
| word+dep | 0.2120 | 0.4392 | 0.4286 | 0.3913 | 0.2433 |
| word+pa | 0.2139** | 0.4524 | 0.4333 | 0.3976** | 0.2484** |

Table 3: Retrieval performance without unjudged documents. ** means that the differences between “word+dep” and “word+pa” are statistically significant with $p < 0.05$.

4.1.2 Retrieval performance evaluation

Table 2 lists retrieval performances. In this table, “word” is a baseline method that uses only words as terms, and “word+dep” is another baseline method that uses words and untyped syntactic dependencies as terms. These untyped syntactic dependencies are also available in the results of the predicate-argument structure analyzer KNP. “word+pa” is our proposed model, which considers predicate-argument structures. We also applied the Wilcoxon signed-rank test to the differences between “word+dep” and “word+pa.”

We can see that our proposed method “word+pa” outperformed the baselines “word” and “word+dep” in all the metrics. In particular, the difference between “word+dep” and “word+pa” in MAP was statistically significant with $p = 0.01134$. In addition, P@3 is higher than the baselines by approximately 1.9%. This means that our model can provide more relevant documents on the top of the ranked result. The baseline “word+dep” outperformed the baseline “word” in MAP, which is used as a metric for optimizing the parameters, but did not outperform “word” in P@5 and nDCG@10. That is to say, “word+dep” was not consistently better than “word.”

Generally, relevance judgments on a standard test collection are created using a pooling method, which judges a certain number of documents submitted by every participating system. Systems that are developed after the creation of the test col-

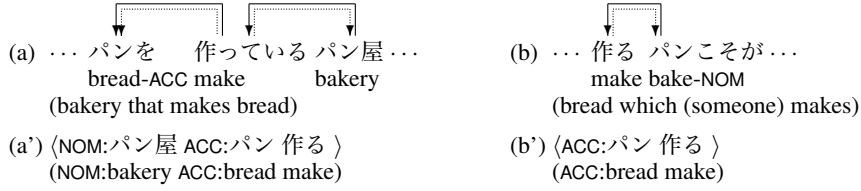


Figure 3: An improved example of relative clause normalization by predicate-argument structure analysis in Japanese. (a) is a part of the query sentence and (b) is a part of a relevant document. Arrows represent syntactic dependencies and dotted arrows represent semantic dependencies. These sentences are normalized to the predicate-argument structures (a') and (b'), respectively.

| | MAP | P@3 | P@5 | P@10 | nDCG@10 |
|----------|-----------------|---------------|---------------|----------------|-----------------|
| word+dep | 0.1769 | 0.4444 | 0.4254 | 0.3921 | 0.2373 |
| word+pa | 0.1790** | 0.4577 | 0.4317 | 0.3984* | 0.2424** |

Table 4: Retrieval performance including additional judgments. The meaning of ** and * is the same as the previous tables.

lection possibly retrieve unjudged documents, but they are usually handled as irrelevant documents, even though they may contain relevant documents. In addition, the number of unjudged documents is likely to increase according to the complexity of systems. To alleviate this bias, we evaluated the three systems without the inclusion of unjudged documents. Table 3 lists the evaluation results. From this table, we can see that “word” was likely to defeat “word+dep,” but “word+pa” consistently outperformed the two baseline methods.

We also evaluated unjudged documents manually. We asked a person who is a certified librarian to judge them. These documents comprise the unjudged documents which appeared in the top 10 results of the two methods (“word+dep” and “word+pa”) for each topic. Table 4 lists the retrieval performances reflecting the inclusion of these additional judgments. From this table, the result of proposed method is consistently better than that of the baseline using syntactic dependencies.

4.1.3 Discussions

By introducing the normalization by predicate-argument structures, our proposed method can retrieve relevant documents that cannot be retrieved or ranked below 1,000 documents by the baseline methods. Figures 3 and 4 show improved examples by the proposed method (“word+pa”) compared to the baseline method (“word+dep”). Figure 3 is an example of the effect of normalizing relative clauses. The following sentences are the original query and a part of relevant document:

- (7) a. 天然酵母のパンを作っているパン屋を見つけない
(I want to find shops that make bread with natural yeast.)
b. ...塩、酵母のみで作るパンこそが、...
(... only the bread that (someone) makes using only salt and yeast ...)

Here, (a) is a query and (b) is a sentence in a relevant document. These sentences have different syntactic dependencies as illustrated in Figure 3, but they are normalized to the predicate-argument structures (a') and (b') in Figure 3. The whole predicate-argument structures are different, but they contain the same typed semantic dependency:

- (8) パン $\xleftarrow{\text{ACC}}$ 作る
(bread $\xleftarrow{\text{ACC}}$ make).

Figure 4 is an example of the effect of normalizing intransitive verbs. The following sentences are the original sentences in a query and a relevant document:

- (9) a. 各地域でお正月に食べる雑煮に入っている具、またはベースとなる味噌などの違いについて調べたい
(I wish to find out about differences in the ingredients and miso stock used to make ozoni soup at New Years in each region.)
b. ... 北海道のお雑煮はシャケやイクラ、じゃがいもを入れるところもある
(in some places, they put salmon, salmon roe and potato in ozoni soup in Hokkaido)

While different verbs are used to express almost the same meaning in these sentences, they are normalized to the predicate-argument structures (a') and (b') in Figure 4. The whole predicate-argument structures are different, but they contain the same typed semantic dependency:

| | |
|---|--|
| <p>(a) ... 雑煮に 入っている 具 ... ozoni-DAT exist ingredients (ingredients that exist in ozoni soup)</p> <p>(a') <ACC:具 DAT:雑煮 入れる > (ACC:ingredients DAT:ozoni soup put)</p> | <p>(b) 雑煮は ... 入れる ... ozoni-TOP put (put in ozoni soup)</p> <p>(b') <DAT:雑煮 入れる > (DAT:ozoni soup put)</p> |
|---|--|

Figure 4: An improved example of intransitive verb normalization by predicate-argument structure analysis in Japanese. (a) is a part of the query sentence and (b) is a part of a relevant document. These sentences are normalized to the predicate-argument structures (a') and (b'), respectively. In particular, the intransitive verb “入る” (exist) is a different word from the transitive verb “入れる” (put) but both are canonicalized to the same transitive verb “入れる” (put) in the predicate-argument structures.

(10) 雑煮 $\xleftarrow{\text{DAT}}$ 入れる
(ozoni soup $\xleftarrow{\text{DAT}}$ put).

Generally speaking, linguistic variations can be roughly divided into two types: syntactic variations and lexical variations. Among syntactic variations, we handled syntactic variations that are related to predicate-argument structures in this study. In our future work, we intend to investigate remaining syntactic variations, such as nominal compounds and paraphrases consisting of larger trees than predicate-argument structures.

The other type is lexical variations, namely synonymous words and phrases. In our approach, they are partially handled in the normalization process to predicate-argument structures. Although handling lexical variations is not the main focus of this paper, we will investigate the effect of incorporating a lexicon of synonymous words and phrases into our model.

4.2 Evaluation on English Test Collection

To validate the effectiveness of the proposed method in other languages than Japanese, we also conducted an experiment on English. We used the TREC Robust 2004 test collection (Voorhees, 2004), which consists of 528,155 English documents and 250 topics (TREC topics 301-450 and 601-700). We used the description queries in these topics, which are written in natural language. Stopwords are removed from the parse of a description and dependencies that contain a stopword in either a modifier or a head are also removed. We used the INQUERY stopword list (Allan et al., 2000). Other experimental settings are the same as the Japanese evaluation.

Table 5 lists retrieval performances. In this table, “word” is a baseline method that uses only lemmatized words as terms, and “word+dep” is another baseline method that uses lemmatized words and syntactic dependencies that are analyzed by the state-of-the-art dependency parser

| | MAP | P@3 | P@5 | P@10 | nDCG@10 |
|----------|----------------|-----------------|-----------------|-----------------|-----------------|
| word | 0.1344 | 0.4498 | 0.4016 | 0.3297 | 0.3527 |
| word+dep | 0.1350 | 0.4337 | 0.4112 | 0.3317 | 0.3517 |
| word+pa | 0.1396* | 0.4618** | 0.4257** | 0.3482** | 0.3659** |

Table 5: Retrieval performance of two baseline methods (“word” and “word+dep”) and our proposed method (“word+pa”) on the TREC test collection. The meaning of ** and * is the same as the previous tables.

MaltParser.¹⁰ “word+pa” is our proposed method, which considers predicate-argument structures converted from the typed semantic dependencies output by the Stanford Parser.¹¹ We can see that our proposed method “word+pa” outperformed the baselines “word” and “word+dep” in all the metrics also on this English test collection.

5 Conclusions

This paper described an information retrieval method that exploits predicate-argument structures to precisely capture the dependencies between words. Experiments on the standard test collections of Japanese and English indicated the effectiveness of our approach. In particular, the proposed method outperformed a baseline method that uses syntactic dependencies output by a dependency parser.

For future work, we plan to optimize ranking by using machine learning techniques such as support vector regression, and to capture any remaining syntactic differences that express similar meanings (i.e., paraphrasing). We used the Okapi BM25 system as our baseline in this study. We will also employ a language model-based information retrieval system as a baseline to confirm the robustness of our approach.

¹⁰<http://www.maltparser.org/>

¹¹To normalize passive constructions, we applied a rule that converts the dependency type “nsubjpass” to “dobj” and “agent” to “nsubj.”

Acknowledgment

This work was supported by JSPS KAKENHI Grant Number 23680015.

References

- James Allan, Margaret E. Connell, W. Bruce Croft, Fangfang Feng, David Fisher, and Xiaoyan Li. 2000. INQUERY and TREC-9. In *Proceedings of the Ninth Text REtrieval Conference*, pages 551–562.
- Matthew W Bilotti, Jonathan Elsas, Jaime Carbonell, and Eric Nyberg. 2010. Rank learning for factoid question answering with linguistic and semantic constraints. In *Proceedings of CIKM2010*, pages 459–468. ACM.
- Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.
- Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *the 5th International Conference on Language Resources and Evaluation*.
- Koji Eguchi, 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 SIGIR2003*.
- Koji Eguchi, 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*.
- Jianfeng Gao, Jian-Yun Nie, Guanyuan Wu, and Guihong Cao. 2004. Dependence language model for information retrieval. In *Proceedings of SIGIR2004*, pages 170–177.
- Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems*, 20(4):422–446.
- Karen Sparck Jones. 1999. What is the role of NLP in text retrieval? In T. Strzalkowski, editor, *Natural language information retrieval*, pages 1–24. Kluwer.
- Daisuke Kawahara and Sadao Kurohashi. 2006. A fully-lexicalized probabilistic model for Japanese syntactic and case structure analysis. In *Proceedings of HLT-NAACL2006*, pages 176–183.
- Changki Lee, Gary Geunbae Lee, and Myung-Gil Jang. 2006. Dependency structure applied to language modeling for information retrieval. *ETRI Journal*, 28(3):337–346.
- Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of ACL2005*, pages 91–98.
- Donald Metzler and W. Bruce Croft. 2005. A markov random field model for term dependencies. In *Proceedings of SIGIR2005*, pages 472–479.
- Yusuke Miyao, Tomoko Ohta, Katsuya Masuda, Yoshimasa Tsuruoka, Kazuhiro Yoshida, Takashi Nomiya, and Jun’ichi Tsujii. 2006. Semantic retrieval for the accurate identification of relational concepts in massive textbases. In *Proceedings of COLING-ACL2006*, pages 1017–1024.
- Srini Narayanan and Sanda Harabagiu. 2004. Question answering based on semantic structures. In *Proceedings of COLING2004*, pages 184–191.
- Joakim Nivre and Mario Scholz. 2004. Deterministic dependency parsing of English text. In *Proceedings of COLING2004*, pages 64–70.
- Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.
- Jae-Hyun Park and W. Bruce Croft. 2010. Query term ranking based on dependency parsing of verbose queries. In *Proceedings of SIGIR2010*, pages 829–830.
- Jae-Hyun Park, W. Bruce Croft, and David A. Smith. 2011. Quasi-synchronous dependence model for information retrieval. In *Proceedings of CIKM2011*, pages 17–26.
- Stephen E. Robertson, Steve Walker, Micheline Hancock-Beaulieu, Aarron Gull, and Marianna Lau. 1992. Okapi at TREC. In *Proceedings of Text REtrieval Conference*, pages 21–30.
- Dan Shen and Mirella Lapata. 2007. Using semantic roles to improve question answering. In *Proceedings of EMNLP-CoNLL2007*, pages 12–21.
- Keiji Shinzato, Tomohide Shibata, Daisuke Kawahara, Chikara Hashimoto, and Sadao Kurohashi. 2008. TSUBAKI: An open search engine infrastructure for developing new information access methodology. In *Proceedings of IJCNLP2008*, pages 189–196.
- Young-In Song, Kyoung-Soo Han, Sang-Bum Kim, So-Young Park, and Hae-Chang Rim. 2008. A novel retrieval approach reflecting variability of syntactic phrase representation. *Journal of Intelligent Information Systems*, 31(3):265–286.
- Ellen M. Voorhees. 2004. Overview of the TREC 2004 robust retrieval track. In *Proceedings of Text REtrieval Conference 2004*.