SYNGRAPH: A Flexible Matching Method based on Synonymous Expression Extraction from an Ordinary Dictionary and a Web Corpus

Tomohide Shibata[†], **Michitaka Odani**[†], **Jun Harashima**[†],
Takashi Oonishi[††], and **Sadao Kurohashi**[†]

[†]Kyoto University, Yoshida-honmachi, Sakyo-ku, Kyoto, 606-8501, Japan
[††]NEC Corporation, 1753, Shimonumabe, Nakahara-Ku, Kawasaki, Kanagawa 211-8666, Japan

{shibata,odani,harashima,kuro}@nlp.kuee.kyoto-u.ac.jp
t-onishi@bq.jp.nec.com

Abstract

This paper proposes a flexible matching method that can assimilate the expressive divergence. First, broad-coverage synonymous expressions are automatically extracted from an ordinary dictionary, and among them, those whose distributional similarity in a Web corpus is high are used for the flexible matching. Then, to overcome the combinatorial explosion problem in the combination of expressive divergence, an ID is assigned to each synonymous group, and SYNGRAPH data structure is introduced to pack the expressive divergence. We confirmed the effectiveness of our method on experiments of machine translation and information retrieval.

1 Introduction

In natural language, many expressions have almost the same meaning, which brings great difficulty to many NLP tasks, such as machine translation (MT), information retrieval (IR), and question answering (QA). For example, suppose an input sentence (1) is given to a Japanese-English example-based machine translation system.

(1) *hotel ni ichiban chikai eki wa doko-desuka*
hotel best near station where is

Even if a very similar translation example (TE) "(2-a) ↔ (2-b)" exists in the TEs, a simple exact matching method cannot utilize this example for the translation.

(2) a. *ryokan no moyori no eki wa*
Japanese hotel nearest station
doko-desuka
where is
b. Where's the nearest station to the hotel?

How to handle these synonymous expressions has become one of the important research topics in NLP.

This paper presents a flexible matching method, which can assimilate the expressive divergence, to solve this problem. This method has the following two features:

1. Synonymy relations and hypernym-hyponym relations are automatically extracted from an ordinary dictionary and a Web corpus.
2. Extracted synonymous expressions are effectively handled by SYNGRAPH data structure, which can pack the expressive divergence.

An ordinary dictionary is a knowledge source to provide synonym and hypernym-hyponym relations (Nakamura and Nagao, 1988; Tsurumaru et al., 1986). A problem in using synonymous expressions extracted from a dictionary is that some of them are not appropriate since they are rarely used. For example, a synonym pair "*suidou*"[1] = "*kaikyou*(strait)" is extracted.

Recently, some work has been done on corpus-based paraphrase extraction (Lin and Pantel, 2001; Barzilay and Lee, 2003). The basic idea of their methods is that two words with similar meanings are used in similar contexts. Although their methods can obtain broad-coverage paraphrases, the obtained paraphrases are not accurate enough to be utilized

[1]This word usually means "water supply".

for achieving precise matching since they contain synonyms, near-synonyms, coordinate terms, hypernyms, and inappropriate synonymous expressions.

Our approach makes the best use of an ordinary dictionary and a Web corpus to extract broad-coverage and precise synonym and hypernym-hyponym expressions. First, synonymous expressions are extracted from a dictionary. Then, the distributional similarity of a pair of them is calculated using a Web corpus. Among extracted synonymous expressions, those whose similarity is high are used for the flexible matching. By utilizing only synonymous expressions extracted from a dictionary whose distributional similarity is high, we can exclude synonymous expressions extracted from a dictionary that are rarely used, and the pair of words whose distributional similarity is high that is not actually a synonymous expression (is not listed in a dictionary).

Another point of our method is to introduce SYNGRAPH data structure. So far, the effectiveness of handling expressive divergence has been shown for IR using a thesaurus-based query expansion (Voorhees, 1994; Jacquemin et al., 1997). However, their methods are based on a bag-of-words approach and thus does not pay attention to sentence-level synonymy with syntactic structure. MT requires such precise handling of synonymy, and advanced IR and QA also need it. To handle sentence-level synonymy precisely, we have to consider the combination of expressive divergence, which may cause combinatorial explosion. To overcome this problem, an ID is assigned to each synonymous group, and then SYNGRAPH data structure is introduced to pack expressive divergence.

2 Synonymy Database

This section describes a method for constructing a synonymy database. First, synonym/hypernym relations are automatically extracted from an ordinary dictionary, and the distributional similarity of a pair of synonymous expressions is calculated using a Web corpus. Then, the extracted synonymous expressions whose similarity is high are used for the flexible matching.

2.1 Synonym/hypernym Extraction from an Ordinary Dictionary

Although there were some attempts to extract synonymous expressions from a dictionary (Nakamura and Nagao, 1988; Tsurumaru et al., 1986), they extracted only hypernym-hyponym relations from the limited entries. In contrast, our method extracts not only hypernym-hyponym relations, but also basic synonym relations, predicate synonyms, adverbial synonyms and synonym relations between a word and a phrase.

The last word of the first definition sentence is usually the hypernym of an entry word. Some definition sentences in a Japanese dictionary are shown below (the left word of ":" is an entry word, the right sentence is a definition, and words in bold font is the extracted words):

yushoku (dinner) : *yugata* (evening) *no* (of) ***shokuji*** **(meal)**.

jushin (barycenter) : *omosa* (weight) *ga* (is) *tsuriatte* (balance) *tyushin* (center) *tonaru* (become) ***ten*** **(spot)**.

For example, the last word *shokuji* (meal) can be extracted as the hypernym of *yushoku* (dinner). In some cases, however, a word other than the last word can be a hypernym or synonym. These cases can be detected by sentence-final patterns as follows (the underlined expressions represent the patterns):

Hypernyms

dosei (Saturn) : ***wakusei*** **(planet)** *no* (of) *hitotsu* (one).

tobi (kite) : ***taka*** **(hawk)** *no* (of) *issyu* (kind).

Synonyms / Synonymous Phrases

ice : ***ice cream*** *no* (of) *ryaku* (abbreviation).

mottomo (most) : ***ichiban*** **(best)**. (∗ one word definition)

moyori (nearest) : ***ichiban*** **(best)** ***chikai*** **(near)** *tokoro* (place)[2]. (∗ less than three phrases)

2.2 Calculating the Distributional Similarity using a Web Corpus

The similarity between a pair of synonymous expressions is calculated based on *distributional similarity* (J.R.Firth, 1957; Harris, 1968) using the Web corpus collected by (Kawahara and Kurohashi, 2006). The similarity between two predicates is defined to be one between the patterns of case examples of each predicate (Kawahara and Kurohashi, 2001). The similarity between two nouns are defined

[2]If the last word of a sentence is a highly general term such as *koto* (thing) and *tokoro* (place), it is removed from the synonymous expression.

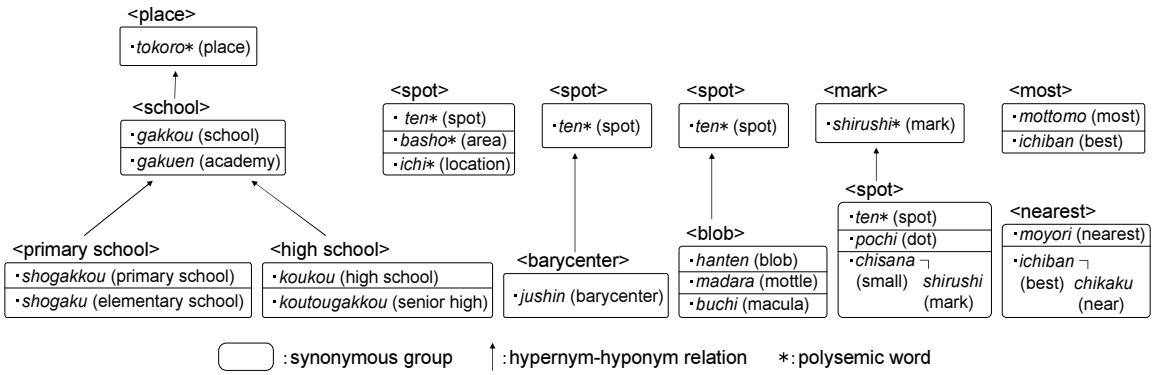

Figure 1: An example of synonymy database.

as the ratio of the overlapped co-occurrence words using the Simpson coefficient. The Simpson coefficient is computed as $\frac{|T(w_1) \wedge T(w_2)|}{min(|T(w_1)|,|T(w_2)|)}$, where $T(w)$ is the set of co-occurrence words of word w.

2.3 Integrating the Distributional Similarity into the Synonymous Expressions

Synonymous expressions can be extracted from a dictionary as described in Section 2.1. However, some extracted synonyms/hypernyms are not appropriate since they are rarely used. Especially, in the case of that a word has multiple senses, the synonym/hypernym extracted from the second or later definition might cause the inappropriate matching. For example, since "*suidou*" has two senses, the two synonym pairs, "*suidou*" = "*jyosuidou*(water supply)" and "*suidou*" = "*kaikyou*(strait)", are extracted. The second sense is rarely used, and thus if the synonymy pair extracted from the second definition is used as a synonym relation, an inappropriate matching through this synonym might be caused. Therefore, only the pairs of synonyms/hypernyms whose distributional similarity calculated in Section 2.2 is high are utilized for the flexible matching.

The similarity threshold is set to 0.4 for synonyms and to 0.3 for hypernyms. For example, since the similarity between "*suidou*" and "*kaikyou*" is 0.298, this synonym is not utilized.

2.4 Synonymy Database Construction

With the extracted binomial relations, a synonymy database can be constructed. Here, polysemic words should be treated carefully[3]. When the relations A=B and B=C are extracted, and B is not polysemic, they can be merged into A=B=C. However, if B is polysemic, the synonym relations are not merged through a polysemic word. In the same way, as for hypernym-hyponym relations, A → B and B → C, and A → B and C → B are not merged if B is polysemic. By merging binomial synonym relations with the exception of polysemic words, synonymous groups are constructed first. They are given IDs, hereafter called SYNID[4]. Then, hypernym-hyponym relations are established between synonymous groups. We call this resulting data as synonymy database. Figure 1 shows examples of synonymous groups in the synonymy database. In this paper, SYNID is denoted by using English gloss word, surrounded by "⟨ ⟩".

3 SYNGRAPH

3.1 SYNGRAPH Data Structure

SYNGRAPH data structure is an acyclic directed graph, and the basis of SYNGRAPH is the dependency structure of an original sentence (in this paper, a robust parser (Kurohashi and Nagao, 1994) is always employed). In the dependency structure, each node consists of one content word and zero or more function words, which is called a *basic node* hereafter. If the content word of a basic node belongs to a synonymous group, a new node with the SYNID is attached to it, and it is called a *SYN node* hereafter. For example, in Figure 2, the shaded nodes are basic nodes and the other nodes are SYN nodes[5].

Then, if the expression conjoining two or more

[3]If a word has two or more definition items in the dictionary, the word can be regarded as polysemic.

[4]Spelling variations such as use of Hiragana, Katakana or Kanji are handled by the morphological analyzer JUMAN (Kurohashi et al., 1994).

[5]The reason why we distinguish basic nodes from SYN nodes is to give priority to exact matching over synonymous matching.

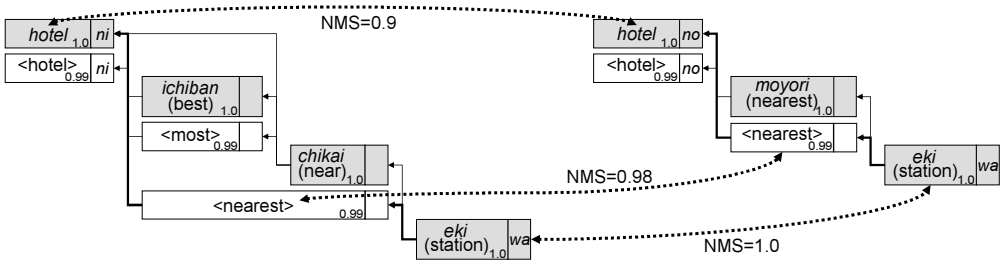

Figure 2: SYNGRAPH matching.

nodes corresponds to one synonymous group, a SYN node is added there. In Figure 2, ⟨nearest⟩ is such a SYN node. Furthermore, if one SYN node has a hyper synonymous group in the synonymy database, the SYN node with the hyper SYNID is also added.

In this SYNGRAPH data structure, each node has a score, NS (Node Score), which reflects how much the expression of the node is shifted from the original expression. We explain how to calculate NSs later.

3.2 SYNGRAPH Matching

Two SYNGRAPHs match if and only if

- all the nodes in one SYNGRAPH can be matched to the nodes in the other one,
- the matched nodes in two SYNGRAPHs have the same dependency structure, and
- the nodes can cover the original sentences.

An example of SYNGRAPH matching is illustrated in Figure 2. When two SYNGRAPHs match each other, their matching score is calculated as follows. First, the matching score of the matching two nodes, NMS (Node Match Score) is calculated with their node scores, NS_1 and NS_2,

$$NMS = NS_1 \times NS_2 \times FI\_Penalty,$$

where we define FI_Penalty (Function word Inconsistency Penalty) is 0.9 when their function words are not the same, and 1.0 otherwise.

Then, the matching score of two SYNGRAPHs, SMS (SYNGRAPH Match Score) is defined as the average of NMSs weighted by the number of basic nodes,

$$SMS = \frac{\sum(\#\text{ of basic nodes} \times NMS)}{\sum \#\text{ of basic nodes}}.$$

In an example shown in Figure 2, the NMS of the left-hand side *hotel* node and the right-hand side *hotel* node is 0.9 (= $1.0 \times 1.0 \times 0.9$). The NMS of the left-hand side ⟨nearest⟩ node and the right-hand side ⟨nearest⟩ node is 0.98 (= $0.99 \times 0.99 \times 1.0$). Then, the SMS becomes $\frac{0.9\times2+0.98\times3+1.0\times2}{2+3+2} = 0.96$.

3.3 SYNGRAPH Transformation of Synonymy Database

The synonymy database is transformed into SYNGRAPHs, where SYNGRAPH matching is iteratively applied to interpret the mutual relationships in the synonymy database, as follows:

Step 1: Each expression in each synonymous group is parsed and transformed into a fundamental SYNGRAPH.

Step 2: SYNGRAPH matching is applied to check whether a sub-tree of one expression is matched with any other whole expressions. If there is a match, a new node with the SYNID of the whole matched expression is assigned to the partially matched nodes group. Furthermore, if the SYNID has a hyper synonymous group, another new node with the hypernym SYNID is also assigned. This checking process starts from small parts to larger parts.

We define the NS of the newly assigned SYN node as the SMS multiplied by a relation penalty. Here, we define the synonymy relation penalty as 0.99 and the hypernym relation penalty as 0.7. For instance, the NS of ⟨underwater⟩ node is 0.99 and that of ⟨inside⟩ node is 0.7.

Step 3: Repeat Step 2, until no more new SYN node can be assigned to any expressions. In the case of Figure 3 example, the new SYN node, ⟨diving⟩ is given to "*suityu* (underwater) *ni* (to) *moguru* (dive)" of ⟨diving(sport)⟩ at the second iteration.

4 Flexible Matching using SYNGRAPH

We use example-based machine translation (EBMT) as an example to explain how our flexible matching method works (Figure 4). EBMT generates a translation by combining partially matching TEs with an input[6]. We use flexible matching to fully exploit the TEs.

[6]How to select the best TEs and combine the selected TEs for generating a translation is omitted in this paper.

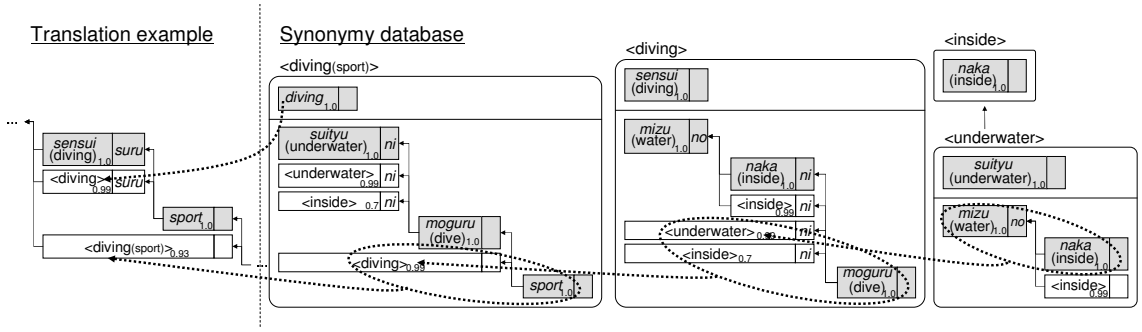

Figure 3: SYNGRAPH transformation of synonymy database.

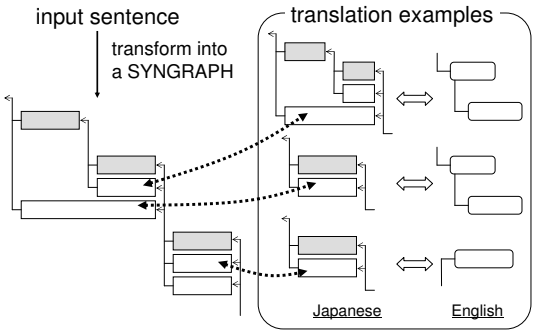

Figure 4: Flexible matching using SYNGRAPH in EBMT.

First, TEs are transformed into SYNGRAPHs by SYNGRAPH matching with SYNGRAPHs of the synonymy database. Since the synonymy database has been transformed into SYNGRAPHs, we do not need to care the combinations of synonymous expressions any more. In the example shown in Figure 3, "*sensui* (diving) *suru* (do) *sport*" in the TE is given ⟨diving(sport)⟩ node just by looking at SYNGRAPHs in ⟨diving(sport)⟩ synonymous group.

Then, an input sentence is transformed into a SYNGRAPH by SYNGRAPH matching, and then the SYNGRAPH matching is applied between all the sub trees of the input SYNGRAPH and SYNGRAPHs of TEs to retrieve the partially matching TEs.

5 Experiments and Discussion

5.1 Evaluation on Machine Translation Task

To see the effectiveness of the our proposed method, we conducted our evaluations on a MT task using Japanese-English translation training corpus (20,000 sentence pairs) and 506 test sentences of IWSLT'05[7]. As an evaluation measure, NIST and BLEU were used based on 16 reference English sentences for each test sentence.

[7]http://www.is.cs.cmu.edu/iwslt2005/.

Table 1: Size of synonymy database.

# of synonymous group	5,046
# of hypernym-hyponym relation	18,590

The synonymy database used in the experiments was automatically extracted from the REIKAI-SHOGAKU dictionary (a dictionary for children), which consists of about 30,000 entries. Table 1 shows the size of the constructed synonymy database.

As a base translation system, we used an EBMT system developed by (Kurohashi et al., 2005). Table 2 shows the experimental results. "None" means the baseline system without using the synonymy database. "Synonym" is the system using only synonymous relations, and it performed best and achieved 1.2% improvement for NIST and 0.8% improvement for BLEU over the baseline. These differences are statistically significant ($p < 0.05$). Some TEs that can be retrieved by our flexible matching are shown below:

- **input:** *fujin* (lady) *you* (for) *toile* (toilet) ↔ **TE:** *josei* (woman) *you* (for) *toile* (toilet)
- **input:** *kantan-ni ieba* (in short) ↔ **TE:** *tsumari* (in other words)

On the other hand, if the system also uses hypernym-hyponym relation ("Synonym Hypernym"), the score goes down. It proves that hypernym examples are not necessarily good for translation. For example, for a translation of *depato* (department store), its hypernym "*mise*(store)" was used, and it lowered the score.

Major errors are caused by the deficiency of word sense disambiguation. When a polysemic word occurs in a sentence, multiple SYNIDs are attached to the word, and thus, the incorrect matching might be occurred. Incorporation of unsupervised word-

Table 2: Evaluation results on MT task.

Synonymy DB	NIST	BLEU
None	8.023	0.375
Synonym	8.121	0.378
Synonym Hypernym	8.010	0.374

Table 3: Evaluation results on IR task.

Method	Synonymy DB	R-prec
Best IREX system	–	0.493
BM25	–	0.474
Our method	None	0.492
	Synonym	0.509
	Synonym Hypernym	0.514

sense-disambiguation of words in dictionary definitions and matching sentences is one of our future research targets.

5.2 Evaluation on Information Retrieval Task

To demonstrate the effectiveness of our method in other NLP tasks, we also evaluated it in IR. More concretely, we extended word-based importance weighting of Okapi BM25 (Robertson et al., 1994) to SYN node-based weighting. We used the data set of IR evaluation workshop IREX, which contains 30 queries and their corresponding relevant documents in 2-year volume of newspaper articles[8]. Table 3 shows the experimental results, which are evaluated with R-precision. The baseline system is our implementation of OKAPI BM25. Differently from the MT task, the system using both synonym and hypernym-hyponym relations performed best, and its improvement over the baseline was 7.8% relative. This difference is statistically significant ($p < 0.05$). This result shows the wide applicability of our flexible matching method for NLP tasks. Some examples that can be retrieved by our flexible matching are shown below:

- **query:** *gakkou-ni* (school) *computer-wo* (computer) *dounyuu* (introduce) ↔ **document:** *shou-gakkou-ni* (elementary school) *pasokon-wo* (personal computer) *dounyuu* (introduce)

6 Conclusion

This paper proposed a flexible matching method by extracting synonymous expressions from an ordinary dictionary and a Web corpus, and introducing SYNGRAPH data structure. We confirmed the effectiveness of our method on experiments of machine translation and information retrieval.

[8]http://nlp.cs.nyu.edu/irex/.

Our future research targets are to incorporate word sense disambiguation to our framework, and to extend SYNGRAPH matching to more structural paraphrases.

References

Regina Barzilay and Lillian Lee. 2003. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *HLT-NAACL 2003*, pages 16–23.

Zellig Harris. 1968. *Mathematical Structures of Language*. Wiley.

Christian Jacquemin, Judith L. Klavans, and Evelyne Tzoukermann. 1997. Expansion of multi-word terms for indexing and retrieval using morphology and syntax. In *35th Annual Meeting of the Association for Computational Linguistics*, pages 24–31.

J.R.Firth. 1957. A synopsis of linguistic theory, 1933-1957. In *Studies in Linguistic Analysis*, pages 1–32. Blackwell.

Daisuke Kawahara and Sadao Kurohashi. 2001. Japanese case frame construction by coupling the verb and its closest case component. In *Proc. of HLT 2001*, pages 204–210.

Daisuke Kawahara and Sadao Kurohashi. 2006. Case frame compilation from the web using high-performance computing. In *Proc. of LREC-06*.

Sadao Kurohashi and Makoto Nagao. 1994. A syntactic analysis method of long japanese sentences based on the detection of conjunctive structures. *Computational Linguistics*, 20(4):507–534.

Sadao Kurohashi, Toshihisa Nakamura, Yuji Matsumoto, and Makoto Nagao. 1994. Improvements of Japanese morphological analyzer JUMAN. In *Proc. of the International Workshop on Sharable Natural Language*, pages 22–28.

Sadao Kurohashi, Toshiaki Nakazawa, Kauffmann Alexis, and Daisuke Kawahara. 2005. Example-based machine translation pursuing fully structural NLP. In *Proc. of IWSLT'05*, pages 207–212.

Dekang Lin and Patrick Pantel. 2001. Discovery of inference rules for question answering. *Natural Language Engineering*, 7(4):343–360.

Junichi Nakamura and Makoto Nagao. 1988. Extraction of semantic information from an ordinary english dictionary and its evaluation. In *Proc. of the 12th COLING*, pages 459–464.

S. E. Robertson, S. Walker, S. Jones, M.M. Hancock-Beaulieu, and M. Gatford. 1994. Okapi at TREC-3. In *the third Text REtrieval Conference (TREC-3)*.

Hiroaki Tsurumaru, Toru Hitaka, and Sho Yoshida. 1986. An attempt to automatic thesaurus construction from an ordinary japanese language dictionary. In *Proc. of the 11th COLING*, pages 445–447.

Ellen M. Voorhees. 1994. Query expansion using lexical-semantic relations. In *SIGIR*, pages 61–69.