Minimally Supervised Method for Multilingual Paraphrase Extraction from Definition Sentences on the Web

Yulan Yan[*] **Chikara Hashimoto**[‡] **Kentaro Torisawa**[§]
Takao Kawai[¶] **Jun'ichi Kazama**[‖] **Stijn De Saeger**[**]
[* ‡ § ¶ ‖ **] Information Analysis Laboratory
Universal Communication Research Institute
National Institute of Information and Communications Technology (NICT)
{[*]yulan, [‡]ch, [§]torisawa, [**]stijn}@nict.go.jp

Abstract

We propose a minimally supervised method for multilingual paraphrase extraction from definition sentences on the Web. Hashimoto et al. (2011) extracted paraphrases from Japanese definition sentences on the Web, assuming that definition sentences defining the same concept tend to contain paraphrases. However, their method requires manually annotated data and is language dependent. We extend their framework and develop a minimally supervised method applicable to multiple languages. Our experiments show that our method is comparable to Hashimoto et al.'s for Japanese and outperforms previous unsupervised methods for English, Japanese, and Chinese, and that our method extracts 10,000 paraphrases with 92% precision for English, 82.5% precision for Japanese, and 82% precision for Chinese.

(1) a. Paraphrasing is the use of your own words to express the author's ideas without changing the meaning.
b. Paraphrasing is defined as a process of transforming an expression into another while keeping its meaning intact.

(2) a. 言い換えとは、ある表現をその意味内容を変えずに別の表現に置き換えることを言います。 (Paraphrasing refers to the replacement of an expression into another without changing the semantic content.)
b. 言い換えとは、ある言語表現をできるだけ意味や内容を保ったまま同一言語の別の表現に変換する処理である。 (Paraphrasing is a process of transforming an expression into another of the same language while preserving the meaning and content as much as possible.)

(3) a. 意译是指译者在不改变原文意思的前提下，完全改变原文的句子结构。 (Paraphrasing refers to the transformation of sentence structure by the translator without changing the meaning of original text.)
b. 意译是指只保持原文内容，不保持原文形式的翻译方法。 (Paraphrasing is a translation method of keeping the content of original text but not keeping the expression.)

Figure 1: Multilingual definition pairs on "paraphrasing."

1 Introduction

Automatic paraphrasing has been recognized as an important component for NLP systems, and many methods have been proposed to acquire paraphrase knowledge (Lin and Pantel, 2001; Barzilay and McKeown, 2001; Shinyama et al., 2002; Barzilay and Lee, 2003; Dolan et al., 2004; Callison-Burch, 2008; Hashimoto et al., 2011; Fujita et al., 2012).

We propose a minimally supervised method for multilingual paraphrase extraction. Hashimoto et al. (2011) developed a method to extract paraphrases from definition sentences on the Web, based on their observation that definition sentences defining the same concept tend to contain many paraphrases. Their method consists of two steps; they extract definition sentences from the Web, and extract phrasal paraphrases from the definition sentences. Both steps require supervised classifiers trained by manually annotated data, and heavily depend on their target language. However, the basic idea is actually language-independent. Figure 1 gives examples of definition sentences on the Web that define the same concept in English, Japanese, and Chinese (with English translation). As indicated by underlines, each definition pair has a phrasal paraphrase.

We aim at extending Hashimoto et al.'s method to a minimally supervised method, thereby enabling acquisition of phrasal paraphrases within one language, but in different languages without manually annotated data. The first contribution of our work is to develop a minimally supervised method for multilingual definition extraction that uses a classifier distinguishing definition from non-definition. The classifier is learnt from the first sentences in

Proceedings of NAACL-HLT 2013, pages 63–73,
Atlanta, Georgia, 9–14 June 2013. ©2013 Association for Computational Linguistics

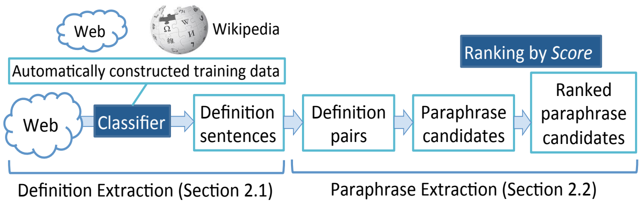

Web
Wikipedia
Automatically constructed training data
Web
Classifier
Definition sentences
Definition pairs
Paraphrase candidates
Ranking by *Score*
Ranked paraphrase candidates
Definition Extraction (Section 2.1)
Paraphrase Extraction (Section 2.2)

Figure 2: Overall picture of our method.

Wikipedia articles, which can be regarded as the definition of the title of Wikipedia article (Kazama and Torisawa, 2007) and hence can be used as positive examples. Our method relies on a POS tagger, a dependency parser, a NER tool, noun phrase chunking rules, and frequency thresholds for each language, in addition to Wikipedia articles, which can be seen as a manually annotated knowledge base. However, our method needs no additional manual annotation particularly for this task and thus we categorize our method as a minimally supervised method. On the other hand, Hashimoto et al.'s method heavily depends on the properties of Japanese like the assumption that characteristic expressions of definition sentences tend to appear at the end of sentence in Japanese. We show that our method is applicable to English, Japanese, and Chinese, and that its performance is comparable to state-of-the-art supervised methods (Navigli and Velardi, 2010). Since the three languages are very different we believe that our definition extraction method is applicable to any language as long as Wikipedia articles of the language exist.

The second contribution of our work is to develop a minimally supervised method for multilingual paraphrase extraction from definition sentences. Again, Hashimoto et al.'s method utilizes a supervised classifier trained with annotated data particularly prepared for this task. We eliminate the need for annotation and instead introduce a method that uses a novel similarity measure considering the occurrence of phrase fragments in global contexts. Our paraphrase extraction method is mostly language-independent and, through experiments for the three languages, we show that it outperforms unsupervised methods (Paşca and Dienes, 2005; Koehn et al., 2007) and is comparable to Hashimoto et al.'s supervised method for Japanese.

Previous methods for paraphrase (and entailment) extraction can be classified into a distributional similarity based approach (Lin and Pantel, 2001; Geffet and Dagan, 2005; Bhagat et al., 2007; Szpektor and Dagan, 2008; Hashimoto et al., 2009) and a parallel corpus based approach (Barzilay and McKeown, 2001; Shinyama et al., 2002; Barzilay and Lee, 2003; Dolan et al., 2004; Callison-Burch, 2008). The former can exploit large scale monolingual corpora, but is known to be unable to distinguish paraphrase pairs from antonymous pairs (Lin et al., 2003). The latter rarely mistakes antonymous pairs for paraphrases, but preparing parallel corpora is expensive. As with Hashimoto et al. (2011), our method is a kind of parallel corpus approach in that it uses definition pairs as a parallel corpus. However, our method does not suffer from a high labor cost of preparing parallel corpora, since it can automatically collect definition pairs from the Web on a large scale. The difference between ours and Hashimoto et al.'s is that our method requires no manual labeling of data and is mostly language-independent.

2 Proposed Method

Our method first extracts definition sentences from the Web, and then extracts paraphrases from the definition sentences, as illustrated in Figure 2.

2.1 Definition Extraction

2.1.1 Automatic Construction of Training Data

Our method learns a classifier that classifies sentences into definition and non-definition using automatically constructed training data, *TrDat*. *TrDat*'s positive examples, *Pos*, are the first sentences of Wikipedia articles and the negative examples, *Neg*, are randomly sampled Web sentences. The former can be seen as definition, while the chance that the sentences in the latter are definition is quite small.

Our definition extraction not only distinguishes definition from non-definition but also identities the defined term of a definition sentence, and in the paraphrase extraction step our method couples two definition sentences if their defined terms are identical. For example, the defined terms of (1a) and (1b) in Figure 1 are both "Paraphrasing" and thus the two definition sentences are coupled. For *Pos*, we mark up the title of Wikipedia article as the defined term. For *Neg*, we randomly select a noun phrase in a sen-

	N-gram definition pattern	N-gram non-definition pattern
(A)	^`[term]` is the `[term]` is a type of	`[term]` may be `[term]` is not

	Subsequence definition pattern	Subsequence non-definition pattern
(B)	`[term]` is * which is located `[term]` is a * in the	you may * `[term]` was `[term]` * , who is

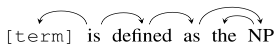
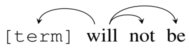
	Subtree definition pattern	Subtree non-definition pattern
(C)	 `[term]` is defined as the NP	 `[term]` will not be

Table 1: Examples of English patterns.

		English		Japanese		Chinese	
Type	Representation	*Pos*	*Neg*	*Pos*	*Neg*	*Pos*	*Neg*
N-gram	Surface	120	400	30	100	20	100
	Base	120	400	30	100	—	—
	POS	2,000	4,000	500	500	100	400
Subsequence	Surface	120	400	30	100	20	40
	Base	120	400	30	100	—	—
	POS	2,000	2,000	500	500	200	400
Subtree	Surface	5	10	5	10	5	5
	Base	5	10	5	10	—	—
	POS	25	50	25	50	25	50

Table 2: Values of frequency threshold.

tence and mark it up as a (false) defined term. Any marked term is uniformly replaced with `[term]`.

2.1.2 Feature Extraction and Learning

As features, we use patterns that are characteristic of definition (definition patterns) and those that are unlikely to be a part of definition (non-definition patterns). Patterns are either *N-grams*, *subsequences*, or *dependency subtrees*, and are mined automatically from *TrDat*. Table 1 shows examples of patterns mined by our method. In (A) of Table 1, "^" is a symbol representing the beginning of a sentence. In (B), "*" represents a wildcard that matches any number of arbitrary words. Patterns are represented by either their words' surface form, base form, or POS. (Chinese words do not inflect and thus we do not use the base form for Chinese.)

We assume that definition patterns are frequent in *Pos* but are infrequent in *Neg*, and non-definition patterns are frequent in *Neg* but are infrequent in *Pos*. To see if a given pattern ϕ is likely to be a definition pattern, we measure ϕ's probability rate $Rate(\phi)$. If the probability rate of ϕ is large, ϕ tends to be a definition pattern. The probability rate of ϕ is:

$$Rate(\phi) = \frac{freq(\phi, Pos)/|Pos|}{freq(\phi, Neg)/|Neg|}, iffreq(\phi, Neg) \neq 0.$$

Here, $freq(\phi, Pos) = |\{s \in Pos : \phi \subseteq s\}|$ and $freq(\phi, Neg) = |\{s \in Neg : \phi \subseteq s\}|$. We write $\phi \subseteq s$ if sentence s contains ϕ. If $freq(\phi, Neg) = 0$, $Rate(\phi)$ is set to the largest value of all the patterns' $Rate$ values. Only patterns whose $Rate$ is more than or equal to a $Rate$ threshold ρ_{pos} and whose $freq(\phi, Pos)$ is more than or equal to a frequency threshold are regarded as definition patterns. Similarly, we check if ϕ is likely to be a non-definition pattern. Only patterns whose $Rate$ is less or equal to a $Rate$ threshold ρ_{neg} and whose $freq(\phi, Neg)$ is more than or equal to a frequency threshold are regarded as non-definition patterns. The probability rate is based on the growth rate (Dong and Li, 1999).

ρ_{pos} and ρ_{neg} are set to 2 and 0.5, while the frequency threshold is set differently according to languages, pattern types (N-gram, subsequence, and subtree), representation (surface, base, and POS), and data (*Pos* and *Neg*), as in Table 2. The thresholds in Table 2 were determined manually, but not really arbitrarily. Basically they were determined according to the frequency of each pattern in our data (e.g. how frequently the surface N-gram of English appears in English positive training samples (*Pos*)).

Below, we detail how patterns are acquired. First, we acquire N-gram patterns. Then, subsequence patterns are acquired using the N-gram patterns as input. Finally, subtree patterns are acquired using the subsequence patterns as input.

N-gram patterns We collect N-gram patterns from *TrDat* with N ranging from 2 to 6. We filter out N-grams using thresholds on the $Rate$ and frequency, and regard those that are kept as definition or non-definition N-grams.

Subsequence patterns We generate subsequence patterns as ordered combinations of N-grams with the wild card "*" inserted between them (we use two or three N-grams for a subsequence). Then, we check each of the generated subsequences and keep it if there exists a sentence in *TrDat* that contains the subsequence and whose root node is contained in the subsequence. For example, subsequence "`[term]` is a * in the" is kept if a term-marked sentence like "`[term]` is a baseball player in the Dominican Republic." exists in *TrDat*. Then, patterns are filtered out using thresholds on the $Rate$ and frequency as we did for N-grams.

Subtree patterns For each definition and non-definition subsequence, we retrieve all the term-marked sentences that contain the subsequence from *TrDat*, and extract a minimal dependency subtree that covers all the words of the subsequence from each retrieved sentence. For example, assume that we retrieve a term-marked sentence "[term] is usually defined as the way of life of a group of people." for subsequence "[term] is * defined as the". Then we extract from the sentence the minimal dependency subtree in the left side of (C) of Table 1. Note that all the words of the subsequence are contained in the subtree, and that in the subtree a node ("way") that is not a part of the subsequence is replaced with its dependency label ("NP") assigned by the dependency parser. The patterns are filtered out using thresholds on the $Rate$ and frequency.

We train a SVM classifier[1] with a linear kernel, using binary features that indicate the occurrence of the patterns described above in a target sentence.

In theory, we could feed all the features to the SVM classifier and let the classifier pick informative features. But we restricted the feature set for practical reasons: the number of features would become tremendously large. There are two reasons for this. First, the number of sentences in our automatically acquired training data is huge (2,439,257 positive sentences plus 5,000,000 negative sentences for English, 703,208 positive sentences plus 1,400,000 negative sentences for Japanese and 310,072 positive sentences plus 600,000 negative sentences for Chinese). Second, since each subsequence pattern is generated as a combination of two or three N-gram patterns and one subsequence pattern can generate one or more subtree patterns, using all possible features leads to a combinatorial explosion of features. Moreover, since the feature vector will be highly sparse with a huge number of infrequent features, SVM learning becomes very time consuming. In preliminary experiments we observed that when using all possible features the learning process took more than one week for each language. We therefore introduced the current feature selection method, in which the learning process finished in one day but still obtains good results.

Original Web sentence: Albert Pujols is a baseball player.
Term-marked sentence 1: [term] is a baseball player.
Term-marked sentence 2: Albert Pujols is a [term].

Figure 3: Term-marked sentences from a Web sentence.

2.1.3 Definition Extraction from the Web

We extract a large amount of definition sentences by applying this classifier to sentences in our Web archive. Because our classifier requires term-marked sentences (sentences in which the term being defined is marked) as input, we first have to identify all such defined term candidates for each sentence. For example, Figure 3 shows a case where a Web sentence has two NPs (two candidates of defined term). Basically we pick up NPs in a sentence by simple heuristic rules. For English, NPs are identified using TreeTagger (Schmid, 1995) and two NPs are merged into one when they are connected by "for" or "of". After applying this procedure recursively, the longest NPs are regarded as candidates of defined terms and term-marked sentences are generated. For Japanese, we first identify nouns that are optionally modified by adjectives as NPs, and allow two NPs connected by "の" (*of*), if any, to form a larger NP. For Chinese, nouns that are optionally modified by adjectives are considered as NPs.

Then, each term-marked sentence is given a feature vector and classified by the classifier. The term-marked sentence whose SVM score (the distance from the hyperplane) is the largest among those from the same original Web sentence is chosen as the final classification result for the original Web sentence.

2.2 Paraphrase Extraction

We use all the Web sentences classified as definition and all the sentences in *Pos* for paraphrase extraction. First, we couple two definition sentences whose defined term is the same. We filter out definition sentence pairs whose cosine similarity of content word vectors is less than or equal to threshold C, which is set to 0.1. Then, we extract phrases from each definition sentence, and generate all possible phrase pairs from the coupled sentences. In this study, phrases are restricted to predicate phrases that consist of at least one dependency relation and in which all the constituents are consecutive in a

[1] http://svmlight.joachims.org.

f_1	The ratio of the number of words shared between two candidate phrases to the number of all of the words in the two phrases. Words are represented by either their surface form ($f_{1,1}$), base form ($f_{1,2}$) or POS ($f_{1,3}$).
f_2	The identity of the leftmost word (surface form ($f_{2,1}$), base form ($f_{2,2}$) or POS ($f_{2,3}$)) between two candidate phrases.
f_3	The same as f_2 except that we use the rightmost word. There are three corresponding subfunctions ($f_{3,1}$ to $f_{3,3}$).
f_4	The ratio of the number of words that appear in a candidate phrase segment of a definition sentence s_1 and in a segment that is NOT a part of the candidate phrase of another definition sentence s_2 to the number of all the words of s_1's candidate phrase. Words are in their base form ($f_{4,1}$).
f_5	The reversed ($s_1 \leftrightarrow s_2$) version of $f_{4,1}$ ($f_{5,1}$).
f_6	The ratio of the number of words (the surface form) of a shorter candidate phrase to that of a longer one ($f_{6,1}$).
f_7	Cosine similarity between two definition sentences from which two candidate phrases are extracted. Only content words in the base form are used ($f_{7,1}$).
f_8	The ratio of the number of parent dependency subtrees that are shared by two candidate phrases to the number of all the parent dependency subtrees. The parent dependency subtrees are adjacent to the candidate phrases and represented by their surface form ($f_{8,1}$), base form ($f_{8,2}$), or POS ($f_{8,3}$).
f_9	The same as f_8 except that we use child dependency subtrees. There are 3 subfunctions ($f_{9,1}$ to $f_{9,3}$) of f_9 type.
f_{10}	The ratio of the number of context N-grams that are shared by two candidate phrases to the number of all the context N-grams of both candidate phrases. The context N-grams are adjacent to the candidate phrases and represented by either the surface form, the base form, or POS. The N ranges from 1 to 3, and the context is either left-side or right-side. Thus, there are 18 subfunctions (3 × 3 × 2).

Table 3: Local similarity subfunctions, $f_{1,1}$ to $f_{10,18}$.

sentence. Accordingly, if two definition sentences that are coupled have three such predicate phrases respectively, we get nine phrase pairs, for instance. A phrase pair extracted from a definition pair is a paraphrase candidate and is given a score that indicates the likelihood of being a paraphrase, *Score*. It consists of two similarity measures, *local similarity* and *global similarity*, which are detailed below.

Local similarity Following Hashimoto et al., we assume that two candidate phrases (p_1, p_2) tend to be a paraphrase if they are similar enough and/or their surrounding contexts are sufficiently similar. Then, we calculate the local similarity (*localSim*) of (p_1, p_2) as the weighted sum of 37 similarity subfunctions that are grouped into 10 types (Table 3.) For example, the f_1 type consists of three subfunctions, $f_{1,1}$, $f_{1,2}$, and $f_{1,3}$. The 37 subfunctions are inspired by Hashimoto et al.'s features. Then, *localSim* is defined as:

$$\mathit{localSim}(p_1, p_2) = \max_{(d_l, d_m) \in DP(p_1, p_2)} ls(p_1, p_2, d_l, d_m).$$

Here, $ls(p_1, p_2, d_l, d_m) = \sum_{i=1}^{10} \sum_{j=1}^{k_i} \frac{w_{i,j} \times f_{i,j}(p_1, p_2, d_l, d_m)}{k_i}$. $DP(p_1, p_2)$ is the set of all definition sentence pairs that contain (p_1, p_2). (d_l, d_m) is a definition sentence pair containing (p_1, p_2). k_i is the number of subfunctions of f_i type. $w_{i,j}$ is the weight for $f_{i,j}$. $w_{i,j}$ is *uniformly* set to 1 except for $f_{4,1}$ and $f_{5,1}$, whose weight is set to -1 since they indicate the unlikelihood of (p_1, p_2)'s being a paraphrase. As the formula indicates, if there is more than one definition sentence pair that contains (p_1, p_2), *localSim* is calculated from the definition sentence pair that gives the maximum value of $ls(p_1, p_2, d_l, d_m)$. *localSim* is local in the sense that it is calculated based on only one definition pair from which (p_1, p_2) are extracted.

Global similarity The global similarity (*globalSim*) is our novel similarity function. We decompose a candidate phrase pair (p_1, p_2) into *Comm*, the common part between p_1 and p_2, and *Diff*, the difference between the two. For example, *Comm* and *Diff* of ("keep the meaning intact", "preserve the meaning") is ("the meaning") and ("keep, intact", "preserve"). *globalSim* measures the semantic similarity of the *Diff* of a phrase pair. It is proposed based on the following intuition: phrase pair (p_1, p_2) tend to be a paraphrase if their surface difference (i.e. *Diff*) have the same meaning. For example, if "keep, intact" and "preserve" mean the same, then ("keep the meaning intact", "preserve the meaning") is a paraphrase.

globalSim considers the occurrence of *Diff* in global contexts (i.e., all the paraphrase candidates from all the definition pairs). The *globalSim* of a given phrase pair (p_1, p_2) is measured by basically counting how many times the *Diff* of (p_1, p_2) appears in all the candidate phrase pairs from all the definition pairs. The assumption is that *Diff* tends to share the same meaning if it appears repeatedly in paraphrase candidates from all definition sentence pairs, i.e., our parallel corpus. Each occurrence of *Diff* is weighted by the *localSim* of the phrase pair in which *Diff* occurs. Precisely, *globalSim* is defined as:

Threshold	The frequency threshold of Table 2 (Section 2.1.2).
NP rule	Rules for identifying NPs in sentences (Section 2.1.3).
POS list	The list of content words' POS (Section 2.2).
Tagger/parser	POS taggers, dependency parsers and NER tools.

Table 4: Language-dependent components.

$$\mathit{globalSim}(p_1, p_2) = \sum_{(p_i,p_j)\in PP(p_1,p_2)} \frac{\mathit{localSim}(p_i, p_j)}{M}.$$

$PP(p_1, p_2)$ is the set of candidate phrase pairs whose *Diff* is the same as (p_1, p_2).[2] M is the number of similarity subfunction types whose weight is 1, i.e. $M = 8$ (all the subfunction types except f_4 and f_5). It is used to normalize the value of each occurrence of *Diff* to [0, 1].[3] *globalSim* is global in the sense that it considers all the definition pairs that have a phrase pair with the same *Diff* as a target candidate phrase pair (p_1, p_2).

The final score for a candidate phrase pair is:

$$\mathit{Score}(p_1, p_2) = \mathit{localSim}(p_1, p_2) + \ln \mathit{globalSim}(p_1, p_2).$$

The way of combining the two similarity functions has been determined empirically after testing several other ways of combining them. This ranks all the candidate phrase pairs.

Finally, we summarize language-dependent components that we fix manually in Table 4.

3 Experiments

3.1 Experiments of Definition Extraction

We show that our unsupervised definition extraction method is competitive with state-of-the-art supervised methods for English (Navigli and Velardi, 2010), and that it extracts a large number of definitions reasonably accurately for English (3,216,121 definitions with 70% precision), Japanese (651,293 definitions with 62.5% precision), and Chinese (682,661 definitions with 67% precision).

3.1.1 Preparing Corpora

First we describe *Pos*, *Neg*, and the Web corpus from which definition sentences are extracted. As the source of *Pos*, we used the English Wikipedia of April 2011 (3,620,149 articles), the Japanese Wikipedia of October 2011 (830,417 articles), and the Chinese Wikipedia of August 2011 (365,545 articles). We removed category articles, template articles, list articles and so on from them. Then the number of sentences of *Pos* was 2,439,257 for English, 703,208 for Japanese, and 310,072 for Chinese. We verified our assumption that Wikipedia first sentences can mostly be seen as definition by manually checking 200 random samples from *Pos*. 96.5% of English *Pos*, 100% of Japanese *Pos*, and 99.5% of Chinese *Pos* were definitions.

As the source of *Neg*, we used 600 million Japanese Web pages (Akamine et al., 2010) and the ClueWeb09 corpus for English (about 504 million pages) and Chinese (about 177 million pages).[4] From each Web corpus, we collected the sentences satisfying following conditions: 1) they contain 5 to 50 words and at least one verb, 2) less than half of their words are numbers, and 3) they end with a period. Then we randomly sampled sentences from the collected sentences as *Neg* so that $|\mathit{Neg}|$ was about twice as large as $|\mathit{Pos}|$: 5,000,000 for English, 1,400,000 for Japanese, and 600,000 for Chinese.

In Section 3.1.3, we use 10% of the Web corpus as the input to the definition classifier. The number of sentences are 294,844,141 for English, 245,537,860 for Japanese, and 68,653,130 for Chinese.

All the sentences were POS-tagged and parsed. We used TreeTagger and MSTParser (McDonald et al., 2006) for English, JUMAN (Kurohashi and Kawahara, 2009a) and KNP (Kurohashi and Kawahara, 2009b) for Japanese, MMA (Kruengkrai et al., 2009) and CNP (Chen et al., 2009) for Chinese.

3.1.2 Comparison with Previous Methods

We compared our method with the state-of-the-art supervised methods proposed by Navigli and Velardi (2010), using their WCL datasets v1.0 (`http://lcl.uniroma1.it/wcl/`), definition and non-definition datasets for English (Navigli et al., 2010). Specifically, we used its training data ($TrDat_{wcl}$, hereafter), which consisted of 1,908 definition and

[2] If there are more than one (p_i, p_j) in a definition pair, we use only one of them that has the largest *localSim* value.

[3] Although we claim that our idea of using *globalSim* is effective, we do not claim that the above formula for calculating is the optimal way to implement the idea. Currently we are investigating a more mathematically well-motivated model.

[4] `http://lemurproject.org/clueweb09.php/`

Method	Precision	Recall	F1	Accuracy
Proposed$_{def}$	86.79	**86.97**	**86.88**	**89.18**
WCL-1	**99.88**	42.09	59.22	76.06
WCL-3	98.81	60.74	75.23	83.48

Table 5: Definition classification results on $TrDat_{wcl}$.

2,711 non-definition sentences, and compared the following three methods. *WCL-1* and *WCL-3* are methods proposed by Navigli and Velardi (2010). They were trained and tested with 10 fold cross validation using $TrDat_{wcl}$. *Proposed*$_{def}$ is our method, which used *TrDat* for acquiring patterns (Section 2.1.2) and training. We tested *Proposed*$_{def}$ on each of $TrDat_{wcl}$'s 10 folds and averaged the results. Note that, for *Proposed*$_{def}$, we removed sentences in $TrDat_{wcl}$ from *TrDat* in advance for fairness. Table 5 shows the results. The numbers for *WCL-1* and *WCL-3* are taken from Navigli and Velardi (2010). *Proposed*$_{def}$ outperformed both methods in terms of recall, F1, and accuracy. Thus, we conclude that *Proposed*$_{def}$ is comparable to *WCL-1*/*WCL-3*.

We conducted ablation tests of our method to investigate the effectiveness of each type of pattern. When using only N-grams, F1 was 85.41. When using N-grams and subsequences, F1 was 86.61. When using N-grams and subtrees, F1 was 86.85. When using all the features, F1 was 86.88. The results show that each type of patterns contribute to the performance, but the contributions of subsequence patterns and subtree patterns do not seem very significant.

3.1.3 Experiments of Definition Extraction

We extracted definitions from 10% of the Web corpus. We applied *Proposed*$_{def}$ to the corpus of each language, and the state-of-the-art supervised method for Japanese (Hashimoto et al., 2011) (*Hashi*$_{def}$, hereafter) to the Japanese corpus. *Hashi*$_{def}$ was trained on their training data that consisted of 2,911 sentences, 61.1% of which were definitions. Note that we removed sentences in *TrDat* from 10% of the Web corpus in advance, while we did not remove Hashimoto et al.'s training data from the corpus. This means that, for *Hashi*$_{def}$, the training data is included in the test data.

For each method, we filtered out its positive outputs whose defined term appeared more than 1,000 times in 10% of the Web corpus, since those terms tend to be too vague to be a defined term or refer to an entity outside the definition sentence. For example, if "the college" appears more than 1,000 times in 10% of the corpus, we filter out sentences like "The college is one of three colleges in the Coast Community College District and was founded in 1947." For *Proposed*$_{def}$, the number of remaining positive outputs is 3,216,121 for English, 651,293 for Japanese, and 682,661 for Chinese. For *Hashi*$_{def}$, the number of positive outputs is 523,882.

For *Proposed*$_{def}$ of each language, we randomly sampled 200 sentences from the remaining positive outputs. For *Hashi*$_{def}$, we first sorted its output by the SVM score in descending order and then randomly sampled 200 from the top 651,293, i.e., the same number as the remaining positive outputs of *Proposed*$_{def}$ of Japanese, out of all the remaining sentences of *Hashi*$_{def}$.

For each language, after shuffling all the samples, two human annotators evaluated each sample. The annotators for English and Japanese were not the authors, while one of the Chinese annotators was one of the authors. We regarded a sample as a definition if it was regarded as a definition by both annotators. Cohen's kappa (Cohen, 1960) was 0.55 for English (moderate agreement (Landis and Koch, 1977)), 0.73 for Japanese (substantial agreement), and 0.69 for Chinese (substantial agreement).

For English, *Proposed*$_{def}$ achieved 70% precision for the 200 samples. For Japanese, *Proposed*$_{def}$ achieved 62.5% precision for the 200 samples, while *Hashi*$_{def}$ achieved 70% precision for the 200 samples. For Chinese, *Proposed*$_{def}$ achieved 67% precision for the 200 samples. From these results, we conclude that *Proposed*$_{def}$ can extract a large number of definition sentences from the Web moderately well for the three languages.

Although the precision is not very high, our experiments in the next section show that we can still extract a large number of paraphrases with high precision from these definition sentences, due mainly to our similarity measures, *localSim* and *globalSim*.

3.2 Experiments of Paraphrase Extraction

We show (1) that our paraphrase extraction method outperforms unsupervised methods for the three languages, (2) that *globalSim* is effective, and (3) that our method is comparable to the state-of-the-art su-

Proposed$_{Score}$**:** Our method. Outputs are ranked by *Score*.

Proposed$_{local}$**:** This is the same as *Proposed*$_{Score}$ except that it ranks outputs by *localSim*. The performance drop from *Proposed*$_{Score}$ shows *globalSim*'s effectiveness.

Hashi$_{sup}$**:** Hashimoto et al.'s supervised method. Training data is the same as Hashimoto et al. Outputs are ranked by the SVM score (the distance from the hyperplane). This is for Japanese only.

Hashi$_{uns}$**:** The unsupervised version of *Hashi*$_{sup}$. Outputs are ranked by the sum of feature values. Japanese only.

***SMT*:** The phrase table construction method of Moses (Koehn et al., 2007). We assume that Moses should extract a set of two phrases that are paraphrases of each other, if we input monolingual parallel sentence pairs like our definition pairs. We used default values for all the parameters. Outputs are ranked by the product of two phrase translation probabilities of both directions.

***P&D*:** The distributional similarity based method by Paşca and Dienes (2005) (their "N-gram-Only" method). Outputs are ranked by the number of contexts two phrases share. Following Paşca and Dienes (2005), we used the parameters $LC = 3$ and $MaxP = 4$, while $MinP$, which was 1 in Paşca and Dienes (2005), was set to 2 since our target was phrasal paraphrases.

Table 6: Evaluated paraphrase extraction methods.

pervised method for Japanese.

3.2.1 Experimental Setting

We extracted paraphrases from definition sentences in *Pos* and those extracted by *Proposed*$_{def}$ in Section 3.1.3. First we coupled two definition sentences whose defined term was the same. The number of definition pairs was 3,208,086 for English, 742,306 for Japanese, and 457,233 for Chinese.

Then we evaluated six methods in Table 6.[5] All the methods except *P&D* took the same definition pairs as input, while *P&D*'s input was 10% of the Web corpus. The input can be seen as the same for all the methods, since the definition pairs were derived from that 10% of the Web corpus. In our experiments *Exp1* and *Exp2* below, all evaluation samples were shuffled so that human annotators could not know which sample was from which method. Annotators were the same as those who conducted the evaluation in Section 3.1.3. Cohen's kappa (Cohen, 1960) was 0.83 for English, 0.88 for Japanese, and 0.85 for Chinese, all of which indicated reasonably good (Landis and Koch, 1977). We regarded a candidate phrase pair as a paraphrase if both annotators regarded it as a paraphrase.

Exp1 We compared the methods that take definition pairs as input, i.e. *Proposed*$_{Score}$, *Proposed*$_{local}$, *Hashi*$_{sup}$, *Hashi*$_{uns}$, and *SMT*. We randomly sampled 200 phrase pairs from the top 10,000 for each method for evaluation. The evaluation of each candidate phrase pair (p_1, p_2) was based on bidirectional checking of entailment relation, $p_1 \rightarrow p_2$ and $p_2 \rightarrow p_1$, with p_1 and p_2 embedded in contexts, as Hashimoto et al. (2011) did. Entailment relation of both directions hold if (p_1, p_2) is a paraphrase. We used definition pairs from which candidate phrase pairs were extracted as contexts.

Exp2 We compared *Proposed*$_{Score}$ and *P&D*. Since *P&D* restricted its output to phrase pairs in which each phrase consists of two to four words, we restricted the output of *Proposed*$_{Score}$ to 2-to-4-words phrase pairs, too. We randomly sampled 200 from the top 3,000 phrase pairs from each method for evaluation, and the annotators checked entailment relation of both directions between two phrases using Web sentence pairs that contained the two phrases as contexts.

3.2.2 Results

From ***Exp1***, we obtained precision curves in the upper half of Figure 4. The curves were drawn from the 200 samples that were sorted in descending order by their score, and we plotted a dot for every 5 samples. *Proposed*$_{Score}$ outperformed *Proposed*$_{local}$ for the three languages, and thus *globalSim* was effective. *Proposed*$_{Score}$ outperformed *Hashi*$_{sup}$. However, we observed that *Proposed*$_{Score}$ acquired many candidate phrase pairs (p_1, p_2) for which p_1 and p_2 consisted of the same content words like "send a postcard to the author" and "send the author a postcard," while the other methods tended to acquire more content word variations like "have a *chance*" and "have an *opportunity*." Then we evaluated all the methods in terms of how many paraphrases with content word variations were extracted. We extracted from the evaluation samples only candidate phrase pairs whose *Diff* contained a content word (*content word variation pairs*), to see how many

[5] We filtered out phrase pairs in which one phrase contained a named entity but the other did not contain the named entity from the output of *Proposed*$_{Score}$, *Proposed*$_{local}$, *SMT*, and *P&D*, since most of them were not paraphrases. We used Stanford NER (Finkel et al., 2005) for English named entity recognition (NER), KNP for Japanese NER, and BaseNER (Zhao and Kit, 2008) for Chinese NER. *Hashi*$_{sup}$ and *Hashi*$_{uns}$ did the named entity filtering of the same kind (footnote 3 of Hashimoto et al. (2011)), and thus we did not apply the filter to them any further.

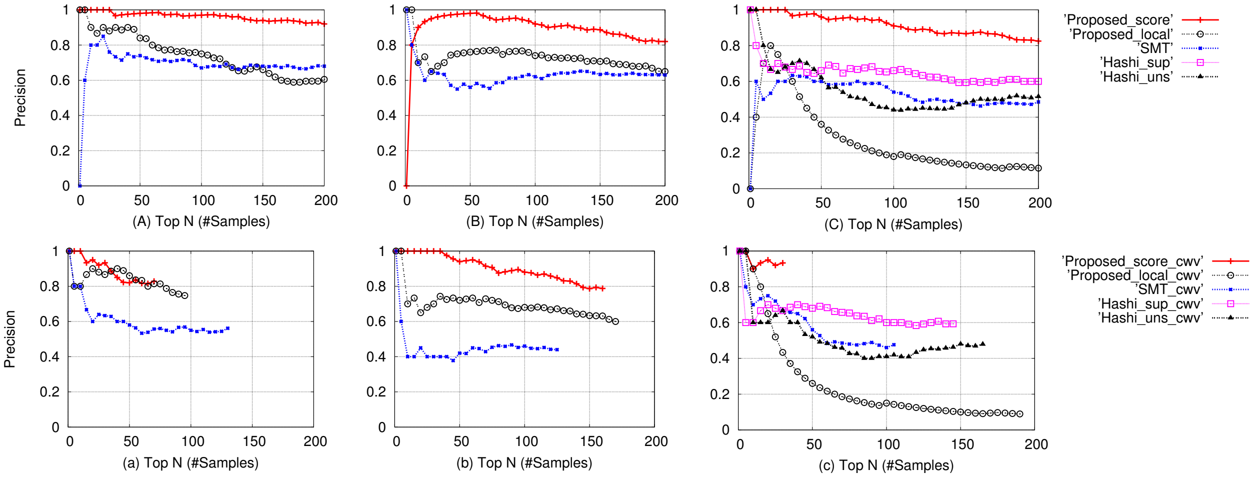


Figure 4: Precision curves of *Exp1*: English (A)(a), Chinese (B)(b), and Japanese (C)(c).

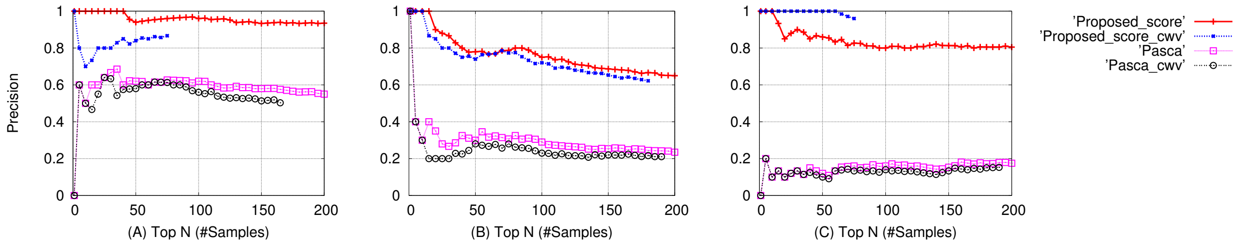


Figure 5: Precision curves of *Exp2*: English (A), Chinese (B), and Japanese (C).

of them were paraphrases. The lower half of Figure 4 shows the results (curves labeled with *_cwv*). The number of samples for $Proposed_{Score}$ reduced drastically compared to the others for English and Japanese, though precision was kept at a high level. It is due mainly to the *globalSim*; the *Diff* of the non-content word variation pairs appears frequently in paraphrase candidates, and thus their *globalSim* scores are high.

From ***Exp2***, precision curves in Figure 5 were obtained. *P&D* acquired more content word variation pairs as the curves labeled by *_cwv* indicates. However, $Proposed_{Score}$'s precision outperformed *P&D*'s by a large margin for the three languages.

From all of these results, we conclude (1) that our paraphrase extraction method outperforms unsupervised methods for the three languages, (2) that *globalSim* is effective, and (3) that our method is comparable to the state-of-the-art supervised method for Japanese, though our method tends to extract fewer content word variation pairs than the others.

Table 7 shows examples of English paraphrases extracted by $Proposed_{Score}$.

is based in Halifax = is headquartered in Halifax
used for treating HIV = used to treat HIV
is a rare form = is an uncommon type
is a set = is an unordered collection
has an important role = plays a key role

Table 7: Examples of extracted English paraphrases.

4 Conclusion

We proposed a minimally supervised method for multilingual paraphrase extraction. Our experiments showed that our paraphrase extraction method outperforms unsupervised methods (Paşca and Dienes, 2005; Koehn et al., 2007; Hashimoto et al., 2011) for English, Japanese, and Chinese, and is comparable to the state-of-the-art language dependent supervised method for Japanese (Hashimoto et al., 2011).

References

Susumu Akamine, Daisuke Kawahara, Yoshikiyo Kato, Tetsuji Nakagawa, Yutaka I. Leon-Suematsu, Takuya Kawada, Kentaro Inui, Sadao Kurohashi, and Yutaka Kidawara. 2010. Organizing information on the web to support user judgments on information credibility. In *Proceedings of 2010 4th International Universal Communication Symposium Proceedings (IUCS 2010)*, pages 122–129.

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

Regina Barzilay and Kathleen R. McKeown. 2001. Extracting paraphrases from a parallel corpus. In *Proceedings of the 39th Annual Meeting of the ACL joint with the 10th Meeting of the European Chapter of the ACL (ACL/EACL 2001)*, pages 50–57.

Rahul Bhagat, Patrick Pantel, and Eduard Hovy. 2007. Ledir: An unsupervised algorithm for learning directionality of inference rules. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP2007)*, pages 161–170.

Chris Callison-Burch. 2008. Syntactic constraints on paraphrases extracted from parallel corpora. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 196–205.

Wenliang Chen, Jun'ichi Kazama, Kiyotaka Uchimoto, and Kentaro Torisawa. 2009. Improving dependency parsing with subtrees from auto-parsed data. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, EMNLP '09, pages 570–579, Singapore. Association for Computational Linguistics.

Jacob Cohen. 1960. Coefficient of agreement for nominal scales. In *Educational and Psychological Measurement*, pages 37–46.

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: exploiting massively parallel news sources. In *Proceedings of the 20th international conference on Computational Linguistics (COLING 2004)*, pages 350–356, Geneva, Switzerland, Aug 23–Aug 27.

Guozhu Dong and Jinyan Li. 1999. Efficient mining of emerging patterns: discovering trends and differences. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '99, pages 43–52, San Diego, California, United States.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pages 363–370.

Atsushi Fujita, Pierre Isabelle, and Roland Kuhn. 2012. Enlarging paraphrase collections through generalization and instantiation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2012)*, pages 631–642.

Maayan Geffet and Ido Dagan. 2005. The distributional inclusion hypotheses and lexical entailment. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pages 107–114.

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

Chikara Hashimoto, Kentaro Torisawa, Stijn De Saeger, Jun'ichi Kazama, and Sadao Kurohashi. 2011. Extracting paraphrases from definition sentences on the web. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1087–1097, Portland, Oregon, USA, June. Association for Computational Linguistics.

Jun'ichi Kazama and Kentaro Torisawa. 2007. Exploiting Wikipedia as external knowledge for named entity recognition. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 698–707, Prague, Czech Republic, June. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June. Association for Computational Linguistics.

Canasai Kruengkrai, Kiyotaka Uchimoto, Jun'ichi Kazama, Yiou Wang, Kentaro Torisawa, and Hitoshi Isahara. 2009. An error-driven word-character hybrid model for joint chinese word segmentation and pos tagging. In *Proceedings of the Joint Conference of the*

47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, pages 513–521, Suntec, Singapore, August. Association for Computational Linguistics.

Sadao Kurohashi and Daisuke Kawahara. 2009a. Japanese morphological analyzer system juman version 6.0 (in japanese). Kyoto University, http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?JUMAN.

Sadao Kurohashi and Daisuke Kawahara. 2009b. Japanese syntax and case analyzer knp version 3.0 (in japanese). Kyoto University, http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?KNP.

J. Richard Landis and Gary G. Koch. 1977. Measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

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

Dekang Lin, Shaojun Zhao Lijuan Qin, and Ming Zhou. 2003. Identifying synonyms among distributionally similar words. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03)*, pages 1492–1493.

Ryan McDonald, Kevin Lerman, and Fernando Pereira. 2006. Multilingual dependency analysis with a two-stage discriminative parser. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, CoNLL-X ’06, pages 216–220, New York City, New York.

Roberto Navigli and Paola Velardi. 2010. Learning word-class lattices for definition and hypernym extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1318–1327, Uppsala, Sweden, July. Association for Computational Linguistics.

Roberto Navigli, Paola Velardi, and Juana María Ruiz-Martínez. 2010. An annotated dataset for extracting definitions and hypernyms from the web. In *Proceedings of LREC 2010*, pages 3716–3722.

Marius Paşca and Péter Dienes. 2005. Aligning needles in a haystack: paraphrase acquisition across the web. In *Proceedings of the Second international joint conference on Natural Language Processing*, IJCNLP’05, pages 119–130, Jeju Island, Korea.

Helmut Schmid. 1995. Improvements in part-of-speech tagging with an application to german. In *Proceedings of the ACL SIGDAT-Workshop*, pages 47–50.

Yusuke Shinyama, Satoshi Sekine, and Kiyoshi Sudo. 2002. Automatic paraphrase acquisition from news articles. In *Proceedings of the 2nd international Conference on Human Language Technology Research (HLT2002)*, pages 313–318.

Idan Szpektor and Ido Dagan. 2008. Learning entailment rules for unary template. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING2008)*, pages 849–856.

Hai Zhao and Chunyu Kit. 2008. Unsupervised segmentation helps supervised learning of character tagging for word segmentation and named entity recognition. In *Proceedings of the Sixth SIGHAN Workshop on Chinese Language Processing*, pages 106–111, Hyderabad, India.