Multilingual Paraphrase Extraction from Definition Sentences on the Web

Yulan Yan  Chikara Hashimoto  Kentaro Torisawa
Takao Kawai  Jun'ichi Kazama  Stijn De Saeger
NICT Universal Communication Research Institute, Information Analysis Laboratory

1. INTRODUCTION
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.

We aim at extending Hashimoto et al.’s method to a minimally supervised method, thereby enabling acquisition of phrasal paraphrases for multiple 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 Wikipedia articles, which can be regarded as the definition of the title of Wikipedia article and hence can be used as positive examples. Our method relies on a POS tagger, a dependency parser, noun phrase chunking rules, and frequency thresholds, 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.

Our definition extraction not only distinguishes definition from non-definition but also identifies the defined term of definition sentence. For Pos, we mark up the title of Wikipedia article as the defined term. For Neg, we randomly select a noun phrase in a sentence and mark it up as a (false) defined term. Any marked term is uniformly replaced with a special symbol [term].

2. PROPOSED METHOD
Our method first extracts definition sentences from the Web, and then extracts paraphrases from the definition sentences, as illustrated in Figure 1.

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.

Our definition extraction not only distinguishes definition from non-definition but also identifies the defined term of definition sentence. For Pos, we mark up the title of Wikipedia article as the defined term. For Neg, we randomly select a noun phrase in a sentence and mark it up as a (false) defined term. Any marked term is uniformly replaced with a special symbol [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 \( \phi \) is likely to be a definition pattern, we measure \( \phi \)'s growth rate [Dong and Li, 1999]. If the growth rate of \( \phi \) is large, \( \phi \) tends to be a definition pattern. The growth rate \( \text{Growth}_\text{pos}(\phi) \) is defined as:
### N-gram definition pattern

- \([\text{term}]\) is the type of \([\text{term}]\) may be
- \([\text{term}]\) is a type of \([\text{term}]\) is not

### Subsequence definition pattern

- \([\text{term}]\) is \(\ast\) which is located \([\text{term}]\) is \(\ast\) in the
  - you may \(\ast\) \([\text{term}]\)
  - was \([\text{term}]\), \(\ast\), who is

### Subtree definition pattern

- \([\text{term}]\) is defined as the NP \([\text{term}]\) will not be

| N-gram non-definition pattern | N-gram non-definition pattern |
|-------------------------------|-------------------------------|
| \([\text{term}]\) | \([\text{term}]\) |

Table 1: Examples of English patterns.

\[
\text{Growth}_{pos}(\phi) = \frac{\text{support}(\phi, \text{Pos})}{\text{support}(\phi, \text{Neg})}, \text{if } \text{support}(\phi, \text{Neg}) \neq 0.
\]

\[
\text{Growth}_{neg}(\phi) = \frac{\text{support}(\phi, \text{Neg})}{\text{support}(\phi, \text{Pos})}, \text{if } \text{support}(\phi, \text{Pos}) \neq 0.
\]

Here, \(\text{support}(s, \text{Pos}) = \frac{\text{freq}(\phi, \text{Pos})}{|\text{Pos}|}\) and \(\text{freq}(\phi, \text{Pos}) = \{s \in \text{Pos} : \phi \subseteq s\}\). \text{support}(\phi, \text{Neg}) and \(\text{freq}(\phi, \text{Neg})\) are defined similarly. We write \(\phi \subseteq s\) if sentence \(s\) contains \(\phi\). If \(\text{support}(\phi, \text{Neg}) = 0\), \(\text{Growth}_{pos}(\phi)\) is set to the largest value of all the patterns’ \(\text{Growth}_{pos}(\phi)\) values. Only patterns whose \(\text{Growth}_{pos}\) is more than or equal to growth rate threshold and whose \(\text{freq}(\phi, \text{Pos})\) is more than or equal to a frequency threshold are regarded as definition patterns. Similarly, to see if \(\phi\) is likely to be a non-definition pattern, we measure the growth rate \(\text{Growth}_{neg}(\phi)\):

\[
\text{Growth}_{neg}(\phi) = \frac{\text{support}(\phi, \text{Neg})}{\text{support}(\phi, \text{Pos})}, \text{if } \text{support}(\phi, \text{Pos}) \neq 0.
\]

Growth rate threshold is uniformly set to 2, 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).

#### 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 growth 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 “\(\ast\)” 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. Then, the patterns are filtered out using thresholds on the growth 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. Note that in the subtree a node that is not a part of the subsequence is replaced with its dependency label. The patterns are filtered out using thresholds on the growth rate and frequency.

We train a SVM classifier with a linear kernel, using binary features that indicate the occurrence of the patterns in a target 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 2 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 and two NPs are merged into one when they are connected by “for” or “of”. 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. For all the languages, among those NPs that overlap, we use only the largest one.

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

### 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 \(< 0.1\). 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 sentence. 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, \(\text{Score}\). 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 \(\text{localSim}\) of \((p_1, p_2)\) as the weighted sum of 37 similarity subfunctions that are grouped into 10 types (Table 2). The 37 subfunctions are inspired by Hashimoto et al. Then, \(\text{localSim}\) is defined as:

\[
\text{localSim}(p_1, p_2) = \max \left( \sum \text{ls}(p_1, p_2, d_l, d_m) \right).
\]

Here,

\[
\text{ls}(p_1, p_2, d_l, d_m) = \sum_{i=1}^{10} \sum_{j=1}^{k_i} w_{i,j} \times f_{i,j}(p_1, p_2, d_l, d_m).
\]

\(DP(p_1, p_2)\) 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 unlikelyhood of \((p_1, p_2)\)’s being a paraphrase. As the formula indicates, if there is more than one definition sentence pair that contains \((p_1, p_2)\), \(\text{localSim}\) is calculated from the definition sentence pair that gives the maximum value of \(\text{ls}(p_1, p_2, d_l, d_m)\).

\(\text{localSim}\) is local in the sense that it is calculated based on only one definition pair from which \((p_1, p_2)\) are extracted.

Original Web sentence: Albert Pujols is a baseball player.
Term-marked sentence 1: \([\text{term}]\) is a baseball player.
Term-marked sentence 2: Albert Pujols is \([\text{term}]\).

Figure 2: Term-marked sentences from a Web sentence.
Global similarity. The global similarity (globalSim) is our novel similarity function that considers the occurrence of phrase fragments (Diff, explained shortly) in global contexts. First, 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 \(\langle\text{keep the meaning intact}\rangle\) are the same, \((p_1, p_2)\) should be a paraphrase. globalSim calculates how likely it is that the meaning of the Diff of a given \((p_1, p_2)\) is the same by basically counting how many times the Diff appears in all the candidate phrase pairs from all the definition pairs, with each occurrence of Diff weighted by the localSim of the phrase pair in which Diff occurs. Precisely, globalSim is defined as:

\[
\text{globalSim}(p_1, p_2) = \sum_{(p_1, p_2) \in P, (p_1, p_2) \neq \text{Comm}} \frac{\text{localSim}(p_1, p_2)}{M}.
\]

\(PP(p_1, p_2)\) is the set of candidate phrase pairs whose Diff is the same as \((p_1, p_2)\). \(M\) is the number of similarity subfunction types whose weight is 1, i.e. \(M = 8\) (all the subfunction types except \(f_1\) and \(f_5\)). It 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:

\[
\text{Score}(p_1, p_2) = \text{localSim}(p_1, p_2) + \ln \text{globalSim}(p_1, p_2).
\]

This ranks all the candidate phrase pairs.

### Table 2: Local similarity subfunctions, \(f_1\) to \(f_{10,18}\)

| \(f_1\) | The ratio of the number of words shared between two candidate phrases to the number of all the words in the two phrases. Words are represented by their surface form \((f_{1,1})\), base form \((f_{1,2})\) or POS \((f_{1,3})\). |
| \(f_2\) | The ratio of the number of words whose weight is \(1\) and in a segment that is NOT a part of the candidate phrase to that of a longer one \(f_{2,1}\). |
| \(f_3\) | The ratio of the number of all the context N-grams of both candidate phrases. Words are in their base form \((f_{3,1})\). |
| \(f_4\) | The ratio of the number of words that appear in a candidate phrase segment of a definition sentence and that is not a part of the candidate phrase of another definition sentence to the number of all the words of the candidate phrase. Words are in their base form \((f_{4,1})\). |
| \(f_5\) | The reversed \((f_1+f_2)\) version of \(f_{5,1}\). |
| \(f_6\) | The ratio of the number of words (the surface form) of a shorter candidate phrase to that of a longer one \(f_{6,1}\). |

### Table 3: Definition classification results on WCL.

| Method         | Precision | Recall | F1   | Accuracy |
|----------------|-----------|--------|------|----------|
| Proposed\(_{df}\) | WCL-1     | 86.79  | 86.97| 86.88    | 89.18    |
| WCL-2          | 99.88     | 42.09  | 59.22| 76.06    |
| WCL-3          | 98.81     | 60.74  | 75.23| 83.48    |

As the source of Neg, we used 600 million Japanese Web pages [Akamine et al., 2010] and the ClueWeb09 corpus for English and Chinese [http://lemurproject.org/clueweb09.php/]. 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 \(|\text{Neg}|\) was about twice as large as \(|\text{Pos}|\): 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.

#### 3.1.2 Comparison with Previous Methods

We compared our method with the state-of-the-art supervised methods proposed byNavigli and Velardi [2010], WCL-1 and WCL-3, using their WCL dataset v1.0 (http://lcl.uniroma1.it/wcl/). They were trained and tested with 10 fold cross validation using WCL. Proposed\(_{df}\) is our method, which used TrDat for acquiring patterns (Section 2.1.2) and training. We tested Proposed\(_{df}\) on each of WCL’s 10 folds and averaged the results. Note that, for Proposed\(_{df}\), we removed sentences in WCL from TrDat in advance for fairness. Table 3 shows the results. The numbers for WCL-1 and WCL-3 are taken fromNavigli and Velardi [2010]. Proposed\(_{df}\) outperformed both methods in terms of recall, F1, and accuracy. Thus, we conclude that Proposed\(_{df}\) is comparable to WCL-1 and WCL-3.

#### 3.1.3 Experiments of Definition Extraction

We extracted definitions from 10% of the Web corpus. We applied Proposed\(_{df}\) to the corpus of each language, and evaluated its positive outputs after filtering out those positive outputs whose defined term appeared more than 1,000 times in 10% of the Web corpus. The number of remaining positive outputs is 3,216,121 for English, 651,293 for Japanese, and 682,661 for Chinese.

For each language, we randomly sampled 200 sentences from the remaining positive outputs, and asked two human annotators to evaluate each sample. We regarded a sample as a definition if it was regarded as a definition by both annotators.

As a result, Proposed\(_{df}\) achieved 70% precision for English, 62.5% for Japanese, and 67% for Chinese. 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

#### 3.2.1 Experimental Setting

Copyright(C) 2013 The Association for Natural Language Processing. All Rights Reserved.
**ProposedScore.** Our method. Outputs are ranked by Score.

**ProposedLocal.** Our method. Outputs are ranked by localSim.

**HashiSup.** Hashimoto et al.’s supervised method. Training data is the same as Hashimoto et al. Outputs are ranked by the SVM score. This is for Japanese only.

**HashiUns.** The unsupervised version of HashiSup. Japanese only.

**SMT.** The phrase table construction method of Moses [Koehn et al., 2007]. We input our definition pairs as monolingual parallel sentence pairs. Outputs are ranked by the product of two phrase translation probabilities of both directions.

Table 4: Evaluated paraphrase extraction methods.

We extracted paraphrases from definition sentences in Pos and those extracted by ProposedLocal 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 five methods in Table 4. We filtered out phrase pairs in which one phrase contained a named entity but the other did not contain the named entity since most of them were not paraphrases.

All the methods took the same definition pairs as input. We regarded a candidate phrase pair as a paraphrase if both annotators regarded it as a paraphrase.

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.

### 3.2.2 Results

We obtained precision curves in the upper half of Figure 3. ProposedScore outperformed ProposedLocal for the three languages, and thus globalSim was effective. ProposedScore outperformed HashiSup. However, we observed that ProposedScore acquired many candidate phrase pairs \((p_1, p_2)\) for which \(p_1\) and \(p_2\) consisted of the same content words like “個人宅まで注文商品を届ける” (deliver ordered products to private home), and “注文商品を個人宅へ届ける” (deliver to private home ordered products), while the other methods tended to acquire more content word variations. 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 of them were paraphrases. The lower half of Figure 3 shows the results (curves labeled with _cwv). The number of samples for ProposedScore reduced drastically compared to the others for English and Japanese, though precision was kept at a high level.

From all of these results, we conclude (1) that our paraphrase extraction method outperforms all the previous 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.

**References**

Akamine, S., Kawahara, D., Kato, Y., Nakagawa, T., Leon-Suematsu, Y. I., Kawada, T., Inui, K., Kurohashi, S., and Kidawara, Y. (2010). Organizing information on the web to support user judgments on information credibility. In IUCS 2010.

Dong, G. and Li, J. (1999). Efficient mining of emerging patterns: discovering trends and differences. In KDD ’99.

Hashimoto, C., Torisawa, K., De Saeger, S., Kazama, J., and Kurohashi, S. (2011). Extracting paraphrases from definition sentences on the web. In ACL2011.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In ACL2007.

Navigli, R. and Velardi, P. (2010). Learning word-class lattices for definition and hyponym extraction. In ACL2010.