Pattern Mining Approach to Unsupervised Definition Extraction

Yulan Yan Chikara Hashimoto Kentaro Torisawa
NICT Universal Communication Research Institute, Information Analysis Laboratory

1 Introduction

Definition sentences are useful sources for many NLP tasks, such as taxonomic and non-taxonomic Relation Extraction [7], Question Answering [1] and Paraphrase Acquisition [5]. Definitions of terms are also among the most common types of information users search for on the Web [8].

A great deal of automatic definition extraction methods are supervised, using manually crafted or semi-automatically learned lexico-syntactic patterns. Patterns are either very simple sequences of words (e.g. English pattern "NP is a"[9], Chinese pattern "NP指的是"[10], Japanese pattern "NPとは"[5]) or more complex sequences of words, parts of speech and chunks. However definition sentences occur in highly variable representation styles, and the most frequent definitional pattern "NP is a" is inherently very noisy. Also current approaches using manually created training data to extract definitions suffer from high labor cost.

In this paper, we propose an unsupervised method to extract definitions from the Web by automatically generating highly variable definition patterns. A training data D_{all} which consists of two datasets, D_d and D_{nd}, is constructed automatically. D_d consists of the first sentence of each Wikipedia article. D_{nd} is randomly sampled Web sentences. From D_{all}, a wide range of highly reliable definition patterns are generated automatically. A SVM classifier is trained on D_{all} and then used to automatically extract definitions from a large Web data. The method is applied to English, Chinese and Japanese definition extraction. Experimental results show that our method is effective to extract multi-lingual definition sentences with low costs.

2 Related Work

A great deal of work is concerned with definition extraction. The majority of these approaches are supervised and language dependent, using lexico-syntactic patterns which are manually crafted or semi-automatically learned [7, 9, 4, 5, 10]. Only few papers try to cope with the generality of patterns and domains in real-world large corpora (like the Web). [1] proposed the use of probabilistic lexico-syntactic patterns, called soft patterns, to model definitions. The authors described a soft matching model based on a n-gram language model. [7] proposed a supervised method which learns word lattices to model textual definitions from an annotated dataset with complicated definition structure. Sentences in the training set are generalized to subsequence patterns which are then clustered. For each cluster, a word lattice is created to model a type of definition. Unlike these methods, our proposed method requires only the Wikipedia articles and Web texts of a target language.

3 Proposed Method

Our proposed method starts from an automatic construction of two datasets, a definition dataset D_d and a non-definition dataset D_{nd}. Then from D_d and D_{nd}, n-gram patterns, subsequence patterns and dependency subtrees are automatically generated as definition and non-definition patterns. Finally a SVM classifier is trained and used to extract definitions from Web data.

3.1 Dataset Construction

We build D_d from Wikipedia articles by collecting each articles's first sentence. The title of the article is regarded as the target term and replaced with "<term>" in the definition sentence. We randomly sample Web sentences from a large Web corpus to build D_{nd}. Take the case of English definition extraction. From English Wikipedia, we obtained 2,439,257 definition sentences as D_d after removing first sentences of articles such as category, template, list, and so on. Six million English sentences are randomly sampled from a Web corpus ClueWeb09[1] as D_{nd}. ClueWeb09 is a Web corpus which consists of about 1 billion Web pages in ten languages that were collected in January and February 2009. For D_{nd}, we regard all the noun phrases as defined term candidates. For each non-definition sentence, we iteratively choose a noun phrase and replace it with "<term>" to derive a new sentence.

3.2 Pattern Generation

Given a definition dataset D_d and a non-definition dataset D_{nd}, our method automatically generates definition patterns, such as "<term> is a" and "〈term〉とは*である。", which most of previous methods had to create manually. We assume that definition patterns are frequent in D_d but are infrequent in D_{nd}, and non-definition patterns are frequent in D_{nd} but are infrequent in D_d.

Our method generates three types of definition and non-definition patterns including n-gram patterns, subsequence patterns and dependency subtrees automatically by capturing significant differences between D_d and D_{nd}. To mine definition and non-definition patterns, frequent patterns are generated from each dataset and the support (*supp*) of a pattern ϕ in a dataset D is calculated as follows [3]:

$$supp(\phi, D) := \frac{freg(\phi, D)}{|D|}$$

[1] http://lemurproject.org/clueweb09.php/

Table 1: N-gram pattern examples.

definition pattern		non-definition pattern	
<term> is a	<term>とは	<term> may be	<term>の
<term> ,	<term>は	<term> is not	している
is one of the	は日本の	if you	は<term>
is a species	である	would be	ください
which was	ことである	however ,	<term>や
<term> refers to	の一種	likely to	する<term>

$$freq(\phi, D) := |\{d \in D : \phi \leq d\}|$$

Given a database D, we denote $|D|$ as the number of sentences in D. We write $\phi \leq d$ if sentence d matches pattern ϕ. Then the change in support between D_d and D_{nd} defined as the growth rate is computed as

$$growth_{D_{nd} \to D_d}(\phi) := \frac{supp(\phi, D_d)}{supp(\phi, D_{nd})}, if\ supp(\phi, D_{nd}) \neq 0$$

Patterns whose growth rate ($D_{nd} \to D_d$) is large are identified as definition patterns. Non-definition patterns are identified similarly.

N-gram patterns

We generate definition and non-definition n-grams from D_{all}. Frequent n-grams are collected from each dataset. A support threshold s_n and a minimum growth rate g_n are given to find all definition patterns which satisfy $supp(\phi, D_d) \geq s_n$ and $growth_{D_{nd} \to D_d}(\phi) \geq g_n$, and all non-definition patterns which satisfy $supp(\phi, D_{nd}) \geq s_n$ and $growth_{D_d \to D_{nd}}(\phi) \geq g_n$. Thresholds s_n and g_n are set up for the following subsequence pattern generation and dimensionality reduction of SVM classification. Examples of English and Japanese n-gram patterns are shown in Table 1. We omit Chinese examples here for limited space.

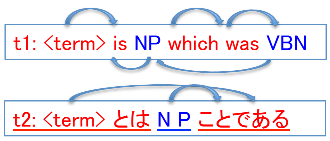
Subsequence Patterns

Subsequence patterns are combinations of ordered n-grams. Generating subsequence patterns using all the n-grams in D_{all} is very time consuming and will generate a huge number of subsequences. Therefore, we generate subsequences that consist of definition and non-definition n-gram patterns obtained from D_d and D_{nd}. We take the sentence "AIG, led by Ken Ham, is one of the largest YEC organizations." as an example. "<term>," and "is one of the" are matched n-gram patterns of the sentence. They are combined with "*" to form a subsequence "<term>, * is one of the *".

Definition subsequence and non-definition subsequence patterns are obtained by giving support threshold s_{sp} and growth rate threshold g_{sp} values. Thresholds s_{sp} and g_{sp} are set up for the following dependency subtree pattern generation and also dimensionality reduction of SVM classification. Table 2 and Table 3 show some English and Japanese definition subsequence pattern examples respectively.

Dependency Subtree Patterns

Dependency subtree patterns are generated based on definition and non-definition subsequence patterns in two steps. In the first step, from each sentence in D_d and D_{nd}, we find all the subsequence patterns that it matches. In the second step, for each matched subsequence pattern, we extract a minimal subtree covering all the words of the subsequence pattern from the dependency tree of the sentence. As shown in the following figure, $t1$ is a dependency subtree generated from a subsequence pattern "<term> is * which was *". For each word in the subsequence pattern (in red font), we label its node as the word itself. For each other word (in blue font) in the subtree, we label the node as its part of speech.



From D_{all}, definition and non-definition dependency subtrees are obtained by giving support threshold s_t and growth rate threshold g_t values in the same way as n-gram patterns and subsequence patterns. The above figure shows two definition dependency subtree examples. $t2$ is a Japanese definition dependency subtree generated from a subsequence pattern "<term>とは*ことである。".

Dependency subtree patterns provide two types of information that n-gram patterns and subsequence patterns do not: dependency between words and part-of-speech of each word. The former is useful for distinguishing two sentences that have different noun phrases as <term> such as s_1 and s_2 below. s_1 and s_2 are derived from the same original sentence "AIG, led by Ken Ham, is one of the largest YEC organizations.". As shown, the dependency subtree of sentence $s1$ and $s2$ are different. Intuitively, the subtree in s_1 is more likely to be a definition dependency subtree.

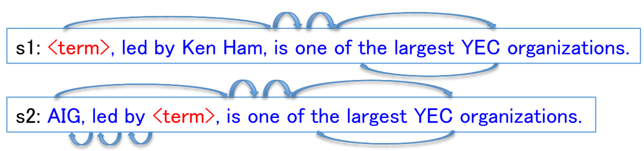


Table 2: English subsequence pattern examples.

definition pattern	non-definition pattern
<term> is * which was *	<term> may be * if they *
<term>, * is one of the *	is <term> * who is *
<term> is * a species *	for <term> * will be *
<term> (born *) is *	<term> in the * however, *

Table 3: Japanese subsequence pattern examples.

definition pattern	non-definition pattern
<term>とは 、*である*	の <term> * を * ます。
<term>は、*にある	<term> の * が * ます。
<term>は*である。	<term >で * して
<term>は*、日本の*	と * <term> を * ます
<term>は、*市にある*	の <term> * ください
『<term>』*の漫画作品*	<term> は * ません

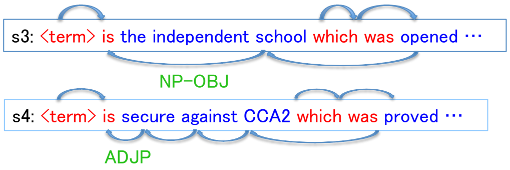


Table 4: Training data statistics.

Language	Positive	Negative
EN	2,439,257	5,000,000
JA	768,429	1,500,000
CH	310,072	600,000

The latter information is useful to preserve important information which is omitted in n-gram patterns and subsequence patterns. For instance, s3 and s4 shown above match the same subsequence pattern "<term> is * which was *". s3 is a definition but s4 is a non-definition. According to their dependency structure, the first "*" matches an objective noun phrase for s3, but an adjective phrase in s4. The dependency subtree in s3 is a definition dependency subtree (t_1 shown above) but the dependency subtree in s4 is obviously not a definition subtree. Dependency subtree patterns can distinguish this difference, which n-gram patterns and subsequence patterns cannot.

3.3 Definition Classifier

A definition classifier is trained on the constructed D_{all} with all the definition and non-definition patterns we generated. The classifier is applied to extract definitions from a Web corpus. For a Web sentence with more than one target term candidates, the classifier assigns a score for each candidate. The one with the highest score is taken as the target term.

4 Experimental Setting

In this paper, our claims are threefold:

- The performance of our unsupervised method is competitive to well-known supervised methods, with much less cost.
- All types of patterns that we propose contribute to the task.
- Our method is language independent.

Besides English, we also apply our method to Japanese and Chinese definition extraction. The training data for Japanese and Chinese are prepared in the same way as we did for English (Section 3.1). As shown in Table 4, the number of non-definition samples is 2 to 3 times the number of definition samples.

Support and growth rate thresholds are turned based on a development dataset which is a subset (1%) of D_{all}. The values we use are shown in Table 5.

The definition classifier evaluation is carried out using a SVM-light classifier with a linear kernel[2].

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

Table 5: Threshold values.

Lang	s_n	g_n	s_{sp}	g_{sp}	s_t	g_t
EN	5.0e-05	5	2.0e-05	2	5.0e-06	2
JA	6.7e-05	2	2.6e-05	2	6.7e-06	2
CH	6.7e-05	2	3.3e-05	2	1.7e-05	2

Table 6: Performance on an English annotated dataset.

Method	P	R	F	A
Proposal	89.16	**93.54**	**91.30**	**91.43**
WCL	**99.88**	42.09	59.22	76.06
Star patterns	86.74	66.14	75.05	81.84
Bigrams	66.70	82.70	73.84	75.80

4.1 Proposed Method vs. Previous Methods

To compare to previous methods, we test our English definition classifier on an existing dataset[3]. It is a corpus of 4,619 Wikipedia sentences, containing 1,908 definition and 2,711 non-definition sentences. The former is a random selection of the first sentences of Wikipedia articles and the latter was obtained by extracting from the same Wikipedia articles sentences in which the page title occurs.

Our English definition classifier trained on D_{all} is used to classify sentences in this dataset. Table 6 shows the results. "WCL" is the method proposed by [7]. [7] built this dataset and used it for both training and testing with 10-fold cross validation. [7] also implemented a baseline method denoted as "Star patterns" and [1]'s method denoted as "Bigrams" on the same datasets. "WCL" showed very high precision (P) (around 99%), higher than our proposed method (89.16%). However, our method achieves a much higher recall (R) (93.54% vs. 42.09%). Our proposed method achieves 91.30% in terms of F-measure (F), and the highest accuracy (A) 91.43%. Thus our method shows the best overall performance. Moreover, the "WCL" method is a supervised method which depends strongly on the annotated definition structure of their training data. Our method is an unsupervised method which uses automatically constructed training data.

4.2 Ablation Test

We conduct ablation tests to evaluate the contributions of different types of patterns. Three English definition classifiers are built on D_{all}:

- #1: uses only n-gram patterns as features.
- #2: uses n-gram patterns and subsequence patterns as features.
- #3: uses all the patterns we proposed.

Table 7 shows the performance of three classifiers on [7]'s dataset. The results show that each type of patterns contributes to the task. After adding dependency subtree patterns, we achieve lower precision but higher recall. The overall performance is the best with all types of patterns.

[3] http://lcl.uniroma1.it/wcl

Table 7: Ablation test.

Classifier	P	R	F	A
#1	87.87	85.64	86.74	89.69
#2	**91.87**	85.46	88.55	90.89
#3	89.16	**93.54**	**91.30**	**93.46**

Table 8: Performance on different languages.

Classifier	P	R	F	A
EN	99.62	99.69	99.65	99.77
JA	98.32	94.83	96.54	98.25
CH	98.30	94.95	96.60	98.27

4.3 Evaluation of Language Independence

We apply our method to English, Japanese, and Chinese definition extraction to examine the language independence of our method. Experiments are performed on the constructed training data D_{all} for each language with 10-fold cross validation.

The results are shown in Table 8. Similar performance is observed for Japanese and Chinese as English, although training samples for Japanese and Chinese are fewer than those for English. All the systems perform well on D_{all}. One may wonder why the performance is much better than on [7]'s dataset. We suspect that one of the main reason is the selection of negative samples. [7] used Wikipedia sentences other than the first sentences of a Wikipedia article as negative examples. On the other hand, we use arbitrary Web sentences as negative examples. Therefore, another observation obtained from the results is that Web sentences can be more easily classified into non-definitions than Wikipedia non-definition sentences. Examples of obtained definitions are shown in Table 9.

5 Conclusion

In this paper, we propose an unsupervised method to extract definitions from the Web. A SVM classifier is trained on two automatically constructed datasets and applied to extract definitions from the Web data. Finally we conclude that:

- From automatically constructed training datasets, D_d and D_{nd}, a wide range of highly reliable definitional patterns can be generated automatically.
- Our method is a language independent method, as our experimental results showed: our method is effective to extract definition sentences of English, Japanese, and Chinese from the Web.
- Our proposed unsupervised method is competitive with the state-of-the-art supervised method.

Table 9: Multi-lingual definition examples.

s1: An acid is a substance which when added to water increases the concentration of hydrogen ions.
s2: The Rocky Mountains, is a mountain range that crosses the western part of the US and Canada.
s3: The original trilogy (often abbreviated OT by fans), is a term used to describe three films.
s4: Tapejara (pronounced : TAP-ah-JAR-ah) is a genus of Brazilian pterosaur from the Cretaceous Period.
s5: The Little Bighorn River is a tributary of the Bighorn River in the United States.
s6: Lynda Barnes (born Lynda Norry) , is one of the world 's leading female Ten-pin bowlers.
s7: Astonishing X-Men is the name of an ongoing comic book series published by Marvel Comics.
s8: 酸とは水素イオンを放出する能力のある物質のこと。
s9: 玉藻とは、讃岐に掛かる枕詩。
s10: 箴言とは：戒め・教訓の意味をもった言葉、格言、警句、金言、 など伝承されたそれらの集成。
s11: 酸的传统定义是当溶解在水中时，溶液中氢离子的浓度增加的化合物。
s12: 放射线是指，伴随着放射性物质的衰变，拥有所释放出能量的微粒或电磁波。
s13: 泪道病主要是指泪道发生阻塞(包括上下泪小管阻塞、泪总管阻塞、 鼻泪管阻塞和慢性泪囊炎等)。

References

[1] Hang Cui, Min-Yen Kan, and Tat-Seng Chua. 2007. *Soft pattern matching models for definitional question answering. ACM Transactions on Information Systems, 25(2):8.*

[2] G. Dong and J. Li. 2005. *Efficient mining of emerging patterns : discovering trends and differences. In Proceedings of KDD, ACM, 1999.*

[3] Fischer, J., Heun, V., Kramer, S. 2005. *Optimal string mining under frequency constraints. In Proc. PKDD 2006. pp. 139–150.*

[4] Atsushi Fujii and Tetsuya Ishikawa. 2002. *Extraction and organization of encyclopedic knowledge information using the World Wide Web. Institute of Electronics, Information, and Communication Engineers, J85-D-II(2): 300–307.*

[5] Chikara Hashimoto, Kentaro Torisawa, Stijn De Sager, Jun'ichi Kazama and Sadao Kurohashi. 2011. *Extracting Paraphrases from Definition Sentences on the Web. In Proc. of the ACL-HLT 2011, pp. 1087-1097*

[6] Eduard Hovy, Andrew Philpot, Judith Klavans, Ulrich Germann, and Peter T. Davis. 2003. *Extending metadata definitions by automatically extracting and organizing glossary definitions. In Proceedings of the 2003 Annual National Conference on Digital Government Research, pages 1–6. Digital Government Society of North America.*

[7] Roberto Navigli and Paola Velardi. 2010. *Learning word-class lattices for definition and hypernym extraction. In Proc. of the ACL 2010, pages 1318–1327.*

[8] Voorhees, E.M. 2001. *Overview of the TREC 2001 Question Answering Track. NIST, USA.*

[9] Jun Xu, Yunbo Cao, Hang Li, and Min Zhao. 2005. *Ranking Definitions with Supervised Learning Methods. In Proc. WWW2005.*

[10] Chunxia Zhang and Peng Jiang. 2009. *Automatic extraction of definitions. In Proceedings of 2nd IEEE International Conference on Computer Science and Information Technology, pages 364–368.*