4C1-IOS-4b-7

# Technical Term Identification for Semantic Analysis of Scientific Papers

Akiko Aizawa[*1*3] Takeshi Sagara[*2] Panot Chimongkol[*3]

[*1]National Institute of Informatics [*2]Info-Proto [*3]The University of Tokyo

In this paper, we first give a brief overview of three well known tasks in natural language processing: term extraction, named entity recognition, and keyphrase extraction. Then, we formalize our technical term identification problem as matching between term instances and term types, which corresponds to surface representation of terms and technical concepts referred to by them. We also present a practical implementation of a term identification system, along with the issues needs to be addressed in future studies.

## 1. Introduction

Extracting terms from text is one of the essential operations in many natural language processing and information retrieval applications. Given natural language text, the operation identifies units in the text that refer to some basic concepts in a target domain. However, regardless of the importance of such terminology extraction [1], only a few established methods are currently available. Most of the methods use naïve measures, such as frequency, C-value/NC-value, or tf-idf [2, 3], that are calculated based on the surface level statistics of the terms in a corpus.

It should be noted here that extracting terms from a corpus is not a simple task but requires many considerations at different levels of natural language understanding. For example, recognizing the spans of the terms is non-trivial since most terms are expressed in multiple words and may even overlap with each other. Also, sense disambiguation or canonical form conversion should be applied for the semantic interpretation of the terms since meanings of terms are not always determined by their surface representations. More importantly, the domain of a target document needs to be identified since terms, as a group, are associated with a specific community that shares a common understanding of the terminology set.

Based on the background, the goal of this paper is to propose a term identification system to combine attributes of terms obtained from heterogeneous linguistic resources. As such, we consider *term identification* as a process of information integration within and across documents. Such a viewpoint is specifically important in semantic analysis of scientific papers whose major goal is identifying relations between scientifically significant concepts. Also, recognizing technical terms enables us to put together diverse pieces of information fragmented in a large corpus.

The rest of the paper is organized as follows. In Section 2, we briefly overview related issues and formulate our term identification problem. Next, in section 3, we describe term features and linguistic resources for term identification. Section 4 briefly introduces our initial attempt to design a prototype term identification system. Section 5 contains discussions and future research directions.

## 2. Term Identification Problem

### 2.1 Related Issues

*Named Entity Recognition* is a task of recognizing named entities of a given class and has been extensively studied in the past [4]. Pattern-based and machine learning-based methods are known to be effective and a number of annotated corpora exist for the training and the evaluation. *Entity resolution* is also tightly connected to named entity recognition, but the task is to extract a set of named entities that refer to the same real-world entities [5].

*Keyphrase extraction* refers to a task of identifying significant terms that characterize a target document [6], and is often used for indexing, similarity calculation, or document labeling. On the other hand, *term extraction* recognizes a terminology set for the entire document collection. In usual cases, no specific class is assumed in keyphrase and term extraction. Unlike named entity recognition, there have not been many annotated corpora for these tasks.

The difference between named entity recognition, keyphrase extraction, and term extraction is illustrated in Figure 1.

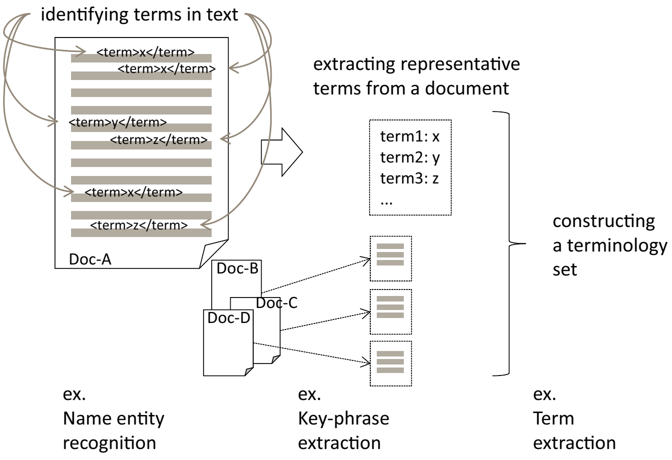

identifying terms in text
<term>x</term>
<term>x</term>
<term>y</term>
<term>z</term>
<term>x</term>
<term>z</term>
Doc-A
extracting representative terms from a document
term1: x
term2: y
term3: z
...
Doc-B
Doc-C
Doc-D
constructing a terminology set
ex. Name entity recognition
ex. Key-phrase extraction
ex. Term extraction

Figure 1: Named entity recognition, keyphrase extraction, and term extraction.

### 2.2 Term type and term instance

First, we introduce two basic notions we use in this paper: *term instance* and *term type.*

- *Term instance* is an individual term that appears in natural language text and therefore is characterized by its context.
- *Term type* represents a distinctive concept with a unique identifier.

They belong to different planes and can be associated with each other by links. Dictionaries can be viewed as authorized nodes on the term type plane. Annotated corpora contain semantic interpretation of the nodes on the term instance plane, or the links between the nodes (Figure 2).

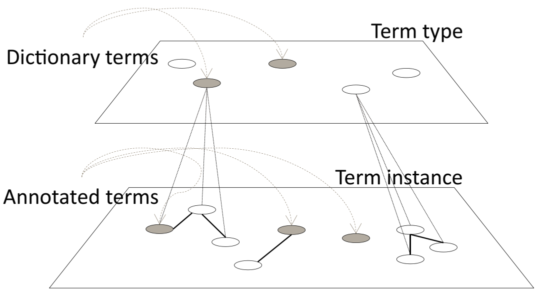


Figure 2: Term type and term instance.

The conventional NLP tasks we have previously seen can be characterized depending on (i) whether the goal is recognizing term instances or term types, and also (ii) whether it assumes existing lexical resources or not. For example, *word sense disambiguation* is a task of finding links between term types and instances given a dictionary; *Entity resolution* is usually defined as a task of finding equivalence links between named entities as term instances; and so on.

With this formulation, (a) identifying nodes on each plane, (b) extracting links between the nodes on the same plane, and (c) recognizing links between the two planes are clearly distinguished. Here, when dealing with named entities, the correspondence between types and instances is obvious and does not need special attention since it is well-supported by the existence of the real world entities. However, this is not the case with technical term identification, which indicates that the task requires different considerations.

In this paper, we propose a framework for simultaneous recognition of term types and instances. For this purpose, the following two issues need to be addressed:

- How to extract context for a given dictionary term?
- How to identify terms in a corpus when they are not registered in a dictionary?

Although this paper does not provide a direct solution to these challenges, some of the key techniques will be explored in the following sections.

## 3. Dictionaries and Corpora

### 3.1 Features for Technical Term Identification

In [7], features for keyphrase extraction are categorized into these three groups: features that represent (i) relationship between document and keyphrases, (ii) relationship among keyphrases, and (iii) relationship among components in a keyphrase. For technical term identification, we define the following four feature classes, differentiated by purpose:

(1) Features for term normalization

The notation for each technical term in a corpus can greatly vary. This is caused by various linguistic phenomena such as capitalization, inflection, hyphenation, compounding, abbreviation, and misspelling. Features for term normalization are used to convert these variations into their canonical forms. The examples include explicit transformation rules (ex. stemming), or a probabilistic model for normalization (ex. learnable string edit distance).

(2) Features for term segmentation

Majority of technical terms are multiwords expressions. And most of the constituent words are also shared by other technical terms. Features for term segmentation contain information on how technical terms are composed and used in a target scientific domain. These features are used for excluding prefix or suffix words that are not considered to be parts of technical terms, or for deciding the boundaries of technical terms in text.

(3) Features for sense disambiguation

Technical terms sometimes contain semantic ambiguity which should be resolved by their context. For this purpose, features for sense disambiguation associate term types with their corresponding context vectors. This includes co-occurring authors' keywords, terms in surrounding text, or terms in definition descriptions.

(4) Features for relation extraction

Features obtained by syntax parsing also contribute to term extraction. These features include lexico-syntactic patterns, dependencies, and semantic role labeling, and are mainly used to identify semantic relations between terms. Together with sense disambiguation features, these features can also be used for extracting synonymous or hypernymous relationship between terms.

### 3.2 Linguistic Resources for Technical Term Identification

Next, we investigated the usability of existing linguistic resources. We chose six widely available resources: (a) human-edited listings of technical terms (*Term list*), (b) controlled keywords manually selected for individual papers (*Controlled keywords*), (c) unrestricted keywords for those papers as assigned by authors (*Authors' keywords*), (d) text body of papers including abstracts and/or full-text content (*Full-text papers*), (e) domain-specific reference books with entry terms and their descriptions (*Reference book*), and (f) Wikipedia articles (*Wikipedia*). We also identified four usability criteria for term identification corresponding to the four types of features described in the previous section (Table 1).

(1) Resources for term normalization
Whether the resources contain notation variations that can be used for analyzing or extracting normalization rules. While *authors' keywords* or *full-text papers* contain rich variation patterns, human edited resources such as *term list*, *controlled keywords*, and *reference books* contain only the canonical forms and therefore unsuitable as resources for term normalization.

(2) Resources for term segmentation
Whether the resources contain technical terms that can be used without any selection nor boundary decision. With *full-text papers*, spans of terms in the text should be firstly identified. Also, with *full-text papers* and *Wikipedia*, technical terms should be distinguished from other general terms.

(3) Resources for sense disambiguation
Whether the resources contain information to construct context vectors of technical terms. Different levels of co-occurrence statistics can be used for the context vectors. In case of *controlled keywords* and *authors' keywords*, terms in the same keyword list can be used. In other cases, the context vectors can be decided based on the co-occurrence of terms in context. In case of *reference books* and *Wikipedia*, context vectors can also be constructed from the descriptions of the entry terms.

(4) Resources for relation extraction
Whether the resources contain natural language sentences with technical terms. Linguistic resources that contain only a collection of terms cannot be used for this purpose.

Table 1: Comparison of different linguistic resources.

| | (1) Term normalization | (2) Term segmentation | (3) Sense disambiguation | (4) Relation extraction |
|---|---|---|---|---|
| Term list | × | ○ | × | × |
| Controlled keywords | × | ○ | ○ | × |
| Authors' keywords | ○ | ○ | ○ | × |
| Full-text papers | ○ | × | ○ | ○ |
| Reference book | × | ○ | ○ | ○ |
| Wikipedia | ○ | × | ○ | ○ |

To summarize, none of the available resources cover all the types of features. On the other hand, the cost for manually constructing a new comprehensive resource is simply infeasible. Therefore, combining different types of linguistic resources becomes crucial. This means that, just like feature engineering is necessary in machine learning practice, a corpus engineering is needed for technical term identification. For example, a statistical model of term construction obtained by *authors' keywords* can be used for out-of-vocabulary term judgment in *full-text article*, and so on.

## 4. Implementation Examples

### 4.1 Combining Sense Disambiguation with Term Normalization

In order to deal with both polysemous and homonymous nature of terms, our first example exploits bilingual translation pairs of technical terms. Assuming that these pairs do not have any semantic ambiguity, we used Japanese-English term translation pairs as basic semantic elements in our implementation. These pairs were obtained from Japanese-English technical term dictionaries (*term list*) and keyword pairs extracted from metadata of scientific paper databases [*1] (*authors' keywords*). By looking at the notation variations contained in these multiple resources, we automatically extracted candidate rules for normalization.

The procedure for rule extraction is as follows: First, pseudo-positive pairs, i.e., pairs that are considered to be variations of the same technical term, were generated. Here, we assumed that any two terms in one language are pseudo-positives if (i) their translations in the other language are the same and also (ii) their edit distance is less than $\alpha$. We used $\alpha = 2$ for Japanese and $\alpha = 4$ for English. Next, pseudo-negative pairs, i.e., pairs that are not considered to be variations of the same technical term, were generated. We chose pairs (i) that appear in the same authors' keyword list, and (ii) with edit distance less than $\beta$. We used $\beta = 6$ for Japanese and $\beta = 8$ for English. For shorter terms, $\beta = 0.7 \times (word\_length)$ was used. Lastly, for both the pseudo-positive and pseudo-negative pairs, transformation rules were extracted using a standard string matching method based on dynamic programming.

Using total 1,304,958 distinctive translation pairs, we identified 110,945 pseudo-positives and 131,461 pseudo-negatives pairs for Japanese. Table 2 shows examples of the extracted transformation rules where the first and the second columns represent the frequency counts of the positive and negative samples in which the rule was applied. For example, the first row shows that the deletion of a character ”法 (method)” at the tail position occurred 1,839 times with positive samples but only once with negative samples. Likewise, the substitution of ”1” or ”2” for ”2”or ”1” never occurred with positive samples but occurred 98 times with negative samples.

We can further assign weights to the extracted translation rules, either based on the frequency counts or by applying machine learning methods. In the latter case, the five-fold cross-validation when using a support vector machine [*2] showed 89.0% accuracy for Japanese and 94.3% accuracy for English term pairs.

### 4.2 Combining Term Segmentation with Relation Extraction

Identifying sentences that contain a specified term type (or vice versa) is a first step for relation extraction. In

[*1] http://ci.nii.ac.jp/

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

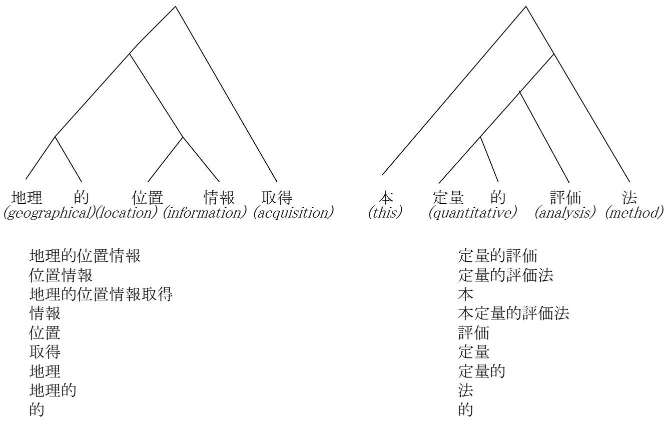
Table 2: Examples of the extracted transformation rules.

| Pos. count | Neg. count | Transformation rules |
|---:|---:|---|
| 1839 | 1 | tail 法 ⟶ (null) |
| 1308 | 0 | body ・ ⟶ (null) |
| 1118 | 1 | body 的 ⟶ (null) |
| 1032 | 11 | body の ⟶ (null) |
| 855 | 0 | body 性 ⟶ (null) |
| 231 | 0 | tail 分析 ⟶ 解析 |
| 130 | 3 | tail 度 ⟶ 性 |
| 105 | 0 | tail 剤 ⟶ 薬 |
| 0 | 98 | body 1 ⟶ 2 |
| 2 | 53 | body 動的 ⟶ 静的 |
| 7 | 47 | body 内 ⟶ 間 |

our second example, we investigated a possibility of utilizing bilingual abstracts of scientific papers for such identification. In our method, we first enumerated all the noun phrases in Japanese and English abstracts of a paper, and then selected terms that appeared in both languages. Note that after applying a procedure described in section 4.1, each term in the dictionary is represented as a translation pair. Based on this, we expect to exclude any semantic ambiguity in a single language by referring to abstracts in both languages.

At this moment, we only detected the longest match spans as terms, but term segmentation can easily be incorporated for better coverage. For this purpose, we implemented a term segmentation system based on word n-gram statistics. Given a list of terms, we first extracted all the word n-grams contained in the list, and for each n-gram, counted the frequencies (i) that the n-grams occurred at the top position of a term, (ii) that the n-gram occurred at the last position of a term, (iii) that the n-gram occurred in the middle of a term, and (iv) that thes n-gram occurred as an independent term. Then, we used the frequency statistics to decide the preferable segmentation of terms. For example, if a n-gram occurs only at the last position of a term, any segmentation starting from the n-gram is considered to be less-preferable.

Examples of the term segmentation are shown in Figure 3. In the figure, two multi-word expressions, "地理 的 位置 情報 取得 (*geographical location information acquisition*)" and "本 定量 的 評価 法 (*this quantitative analysis method*)", are given to the system. The list under the tree gives all possible segmentations sorted by decreasing score. Although they are both composed of five words in Japanese, the system suggests different segmentations and identifies "地理 的 位置 情報 (*geographical location information*)" and "定量 的 評価 (*quantitative analysis*)" as the most significant technical terms. In this example, the term "本" is polysemous (*this* and *book*) and inappropriate as a technical term. Such a case can be excluded using the English abstract counterpart.


地理 的 位置 情報 取得
(geographical)(location) (information) (acquisition)
本 定量 的 評価 法
(this) (quantitative) (analysis) (method)

地理的位置情報
位置情報
地理的位置情報取得
情報
位置
取得
地理
地理的
的

定量的評価
定量的評価法
本
本定量的評価法
評価
定量
定量的
法
的

Figure 3: Examples of term segmentation.

## 5. Discussion and Future Research

In this paper, we revisited a problem of technical term identification and formulated the problem as a complex task that requires diverse considerations at different levels of natural language processing. Future major challenges include inter- and intra-document coreference resolution to identify links between term instances, and a statistical method to automatically annotate technical terms in text using both distributional and pattern-based features.

## References

[1] Kyo Kageura: "The Quantitative Analysis of the Dynamics and Structure of Terminologies," John Benjamins (2012).

[2] Katerina T. Frantzi, Sophia Ananiadou, and Jun-ichi Tsujii: "The C-value/NC-value Method of Automatic Recognition for Multi-Word Terms," Proc. Of ECDL-98, 585-604 (1998).

[3] Ziqi Zhang, Jose Iria, Christopher Brewster, Fabio Ciravegna: "A Comparative Evaluation of Term Recognition Algorithms," Proc. of LREC (2008).

[4] David Nadeau, and Satoshi Sekine: "A survey of named entity recognition and classification," Journal of Linguisticae Investigationes 30(1), 3-26. 2007.

[5] Lise Getoor and Ashwin Machanavajjhala: "Entity Resolution: Theory, Practice and Open Challenges," VLDB-2012 Tutorial (2012).

[6] Richard Hussey, Shirley Williams, and Richard Mitchell: "Automatic Keyphrase Extraction: A Comparison of Methods," Proc. of eKNOW-2012 , 18-23 (2012).

[7] Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin: "SemEval-2010 Task 5: Automatic Keyphrase Extraction from Scientific Articles," Proc. Of the 2009 Workshop on Multiword Expressions, ACL-IJCNLP, 9-16 (2009).