Automatic extraction of lexical functions from corpora

楊 宇軒 瀋 禎楠 叢 童顔 ルパージュ イヴ
早稲田大学 大学院情報生産システム研究科
yang98@asagi.waseda.jp, shenzhennan0918@fuji.waseda.jp,
tongyan.cong@ruri.waseda.jp, yves.lepage@waseda.jp

Abstract

Lexical functions (LF), developed within Meaning-Text Theory (MTT), systematically represent conventional lexical relations. Due to the high cost of manual construction, large-scale LF resources remain scarce. This paper proposes an automatic pipeline for extracting LFs from large-scale corpora by integrating annotation methods and manually designed extraction rules to identify word pairs corresponding to LFs. Experimental results demonstrate the feasibility of corpus-based automatic extraction for building LF resources and support future lexical-semantic research.

1 Introduction

Lexical knowledge extends beyond individual word meanings and plays a crucial role in natural language processing. Many linguistic phenomena involve conventionalized word combinations, such as collocations, light verb constructions, and functional predicates, which cannot be fully captured by general grammatical rules. Modeling these patterns is essential for applications including lexical choice, paraphrase generation, and multilingual processing.

Meaning-Text Theory (MTT) [1] provides a principled framework for representing such regularities through lexical functions. By abstracting over surface realizations, lexical functions capture systematic semantic relations between lexical units and their typical combinational partners, offering a compact and interpretable representation of lexical co-occurrence patterns [2].

Despite their theoretical importance and practical value, large-scale lexical function resources remain limited. Most existing resources are manually constructed, requiring substantial expert effort and making it difficult to achieve broad coverage or multilingual scalability. These limitations motivate the development of automatic methods for extracting lexical functions directly from corpora.

In this paper, we propose an automatic pipeline for extracting lexical functions from a large-scale corpus. The method combines semantic role labeling [3], dependency parsing [4], and manually designed rules to identify specific lexical functions' data, with a focus on lexical functions **Oper**$_i$ and **Func**$_i$.

2 Background

2.1 Meaning-Text Theory and lexical functions

Lexical functions were developed within the framework of Meaning-Text Theory to describe and systematize semantic relations between lexical units, particularly collocations and lexical derivations. They have been widely used in the construction of explanatory combinatorial dictionaries and function as abstract nodes in certain grammatical representations. Formally, a lexical function f applied to a lexical item L is denoted as $f(L) = \{L_i\}$, where L denotes the keyword and L_i represents the set of lexical units associated with L under a specific semantic relationship. Here, f represents a particular semantic relation, and L_1 ... L_n are the acceptable lexical realizations of that relation with respect to L.

2.2 Syntagmatic lexical functions

The framework of lexical functions consists of approximately sixty standard functions, which are broadly classified into paradigmatic and syntagmatic types. Paradigmatic lexical functions primarily describe categorical relations between lexical units, such as synonymy, antonymy, and derivation. These functions help capture the semantic substitution and hierarchical structures within the lexical system. Syntagmatic lexical functions mainly describe how lexical units are combined in actual language use, especially in collocations and light verb constructions.

In this study, we focus on two light verb lexical functions: **Oper** and **Func**, which are particularly relevant to predicate–argument structures. **Oper**$_i$ describes light verb constructions in which a semantically weak verb combines with a noun to express an action or event, where the index indicates which deep-syntactic argument of the keyword functions as the syntactic subject. **Func**$_i$ describes functional realizations in which a predicate expresses the manifestation or occurrence of a property or state associated with the base word, where the index indicates which deep-syntactic argument of the keyword functions as its first syntactic object. Examples are shown in Table 1.

Table 1 Lexical function **Oper** and **Func** examples

	ja		zh	
LFs	Input	Output	Input	Output
Oper$_1$	影響	与える	報告	作
Oper$_2$	危険	直面する	危險	面對
Oper$_3$	注文	出す	訂單	下
Func$_0$	雨	降る	風險	存在
Func$_1$	傷	つく	援助	來自
Func$_2$	危険	脅かす	談話	關於

3 Methodology

3.1 Overview

Our experiments adopt a pipeline-based framework, as shown in Figure 1. Starting from a large Japanese-Chinese parallel corpus, the data are first tokenized and then annotated using semantic role labeling and dependency parsing to obtain candidate predicate–argument structures together with their syntactic dependency representations. These candidates are subsequently refined to produce normalized predicate– argument structures. Based on the resulting predicate– argument structures, lexical functions are extracted using manually designed linguistic rules that make explicit reference to dependency relations, yielding a corpus-based lexical function dataset.

3.2 Semantic and syntactic annotation

We apply neural network models to do tokenization (see A.2.1), semantic role labeling (see A.2.2), and dependency parsing (see A.2.3). Semantic role labeling is used to identify candidate predicate–argument structures [1] by associating predicates with their arguments. However, SRL outputs often consist of phrase-level argument spans rather than normalized word-level predicate–argument structures. As a result, these outputs cannot be directly used for lexical function extraction and require further refinement.

Dependency parsing is used to analyze grammatical relations between words and the internal syntactic structure of phrases. At the sentence level, dependency relations provide syntactic constraints for validating predicate–argument structures. At the argument level, multi-word argument spans produced by SRL are reduced to their syntactic head words, yielding words compatible with the definition of lexical functions, for lexical functions focus on relations between lemmas.

Together, semantic role labeling and dependency parsing provide normalized predicate– argument structures. These candidates serve as input to the subsequent rule-based extraction stage.

3.3 Rule-based LF data extraction

Lexical functions are extracted using manually designed rules derived from the definitions of **Oper**$_i$ and **Func**$_i$ in Meaning-Text Theory. By relating these theoretical definitions to the outputs of semantic role labeling and dependency parsing, we establish explicit correspondences between lexical function properties and automatically annotated predicate–argument structures.

For each candidate predicate– argument structure, we examine the relation between the predicate and its arguments and determine whether the resulting predicate–argument pair satisfies the defining conditions of **Oper**$_i$ and **Func**$_i$, based on semantic role labels and dependency relations. Dependency labels such as *subj*, *obj*, and *obl* denote the syntactic subject, core object, and oblique argument of the predicate, respectively, and are used as structural constraints in the extraction rules.

Chinese SRL follows a PropBank-style argument-numbering scheme (ARG0, ARG1, ARG2), whereas Japanese SRL adopts a case-marker-based labeling scheme. Although the two schemes are not strictly aligned, approximate correspondences can be established: ARG0 generally corresponds to Japanese ガ (and ノ in

[1] The semantic role labels used in Chinese and Japanese differ in form: Chinese SRL adopts PropBank-style argument indices (e.g., ARG0, ARG1, ARG2), while Japanese SRL relies on case-marker-based labels (e.g., ガ, ヲ, ニ).

Table 2 Extraction rules for Japanese and Chinese

LFs	SRL	DEP	LFs	SRL	DEP
Oper$_1$	ヲ	obj	**Oper**$_1$	ARG1	obj
Oper$_2$	ニ, ヘ $\wedge$ first-obl	obl	**Oper**$_2$	ARG2	obj
Oper$_3$	デ, カラ, マデ $\vee$ second-obl	obl	**Oper**$_3$	ARG3	obj
Func$_0$	ガ, ノ $\wedge$ $\neg$obl	subj	**Func**$_0$	ARG0 $\wedge$ $\neg$ARG1	subj
Func$_1$	ガ, ノ $\wedge$ count(obl) = 1	subj	**Func**$_1$	ARG0 $\wedge$ ARG1 $\wedge$ $\neg$ARG2	subj
Func$_2$	ガ, ノ $\wedge$ count(obl) = 2	subj	**Func**$_2$	ARG0 $\wedge$ ARG1 $\wedge$ ARG2	subj

some constructions), ARG1 to ヲ, and ARG2 / ARG3 to case markers such as ニ, デ, ヘ, カラ, and マデ.

Based on these correspondences, we define extraction rules that map combinations of SRL labels and dependency relations to specific lexical functions. Table 2 summarizes the rules for Chinese and Japanese.

Since both **Oper**$_i$ and **Func**$_i$ require the predicate to be a light verb, an additional filtering step is required after rule-based extraction. Predicate–argument structure that does not correspond to light verb constructions is removed using a predefined list of light verbs, ensuring consistency with the theoretical definition of lexical functions. The implementation of light verb filtering is discussed in 4.1.3.

4 Experiment and result analysis

4.1 Experiment setup

4.1.1 Dataset

We conduct experiments on the JParaCrawl Japanese-Chinese parallel corpus [5], which contains ~4.6M sentence pairs collected from web data and crowdsourcing. Such a corpus provides sufficient scale and linguistic diversity for statistical analysis.

For both Japanese and Chinese, sentences are processed independently using the same preprocessing pipeline to ensure consistency, but individual models are used for annotation. The large scale of the corpus allows statistically meaningful extraction of predicate–argument structures associated with lexical functions.

4.1.2 Annotation models

We implemented the above experimental design using the HanLP [6] framework. For sentence annotation, we employed a BERT-based [7] model to annotate Japanese sentences and the ELECTRA [8] model to annotate Chinese sentences. After corpus deduplication and annotation, we obtained ~ 49.7M predicate–argument structure candidates and ~ 312.3M dependency structures (both Japanese and Chinese results).

4.1.3 Supplementary methods

In the experiments, to improve dataset quality, we jointly considered verb frequency and the number of distinct direct object types co-occurring with each verb. Specifically, we set thresholds on verb frequency and on the number of distinct direct objects in verb–object collocations to capture two key characteristics of light verbs, namely their high frequency and semantic flexibility. We regard the combination of these two criteria as a filtering strategy for light verbs, and applied it to further refine both Japanese and Chinese lexical function datasets.

On the other hand, since the output associated with a given lexical function input effectively constitutes a set of lexical items, we computed the frequency of each possible output for every fixed input and stored the outputs in descending order of frequency. This strategy facilitates efficient access to the most representative realizations.

4.2 Result analysis

4.2.1 Light verb-based filtering problem

Table 3 Statistics on light verb filtering

	ja		zh	
LFs	Original	Filtered	Original	Filtered
Oper$_1$	14,883	372	56,266	54,047
Oper$_2$	6,511	164	6,903	6,903
Func$_0$	60,751	7,763	85,450	71,410
Func$_1$	530	2	87,498	69,913
Func$_2$	20	0	4,076	3,346

Because different models were used for initial annotation, the Japanese and Chinese datasets differ in size, and this discrepancy is further amplified by light-verb filtering. As shown in Table 3, the Japanese data is reduced to about 1/40 of its original size, while a much larger portion of the Chinese data is preserved. This difference likely reflects cross-linguistic variation in light-verb constructions rather than data quality. We attribute the severe reduction in Japanese primarily to the annotation model's performance, which leads to fewer extracted instances; consequently, we do not further analyze Japanese lexical functions **Func**$_1$ and **Func**$_2$, although this issue may be mitigated by increasing the number of sentences.

4.2.2 Mean relational distance

Let $D = \{(x_i, Y_i)\}_{i=1}^{N}$ denote the dataset of a single lexical function. For each instance, the frequency-weighted average output embedding is computed as

$$\bar{\mathbf{y}}_i = \frac{\sum_{y \in Y_i} \mathrm{freq}(y)\, \mathbf{v}(y)}{\sum_{y \in Y_i} \mathrm{freq}(y)}, \qquad (1)$$

and the corresponding relational vector is defined as

$$\mathbf{r}_i = \bar{\mathbf{y}}_i - \mathbf{v}(x_i). \qquad (2)$$

The mean relational distance is then given by

$$\frac{1}{N} \sum_{i=1}^{N} \left\| \mathbf{r}_i - \frac{1}{N} \sum_{j=1}^{N} \mathbf{r}_j \right\|_2 . \qquad (3)$$

Here, x_i and Y_i denote the input word and its associated output word set, respectively; $\mathbf{v}(\cdot)$ is a static word embedding, $\mathrm{freq}(\cdot)$ denotes corpus frequency, and $\|\cdot\|_2$ is the Euclidean norm. The resulting value characterizes the consistency of the relational representation of the lexical function in the embedding space.

Table 4 Mean relational distance comparison. Note: "—" denotes unavailable data due to insufficient data.

	ja		zh	
LFs	Random	Rule-induced	Random	Rule-induced
Oper$_1$	0.332	0.527	0.037	0.481
Oper$_2$	0.110	0.358	0.116	0.550
Func$_0$	0.692	0.866	0.044	0.542
Func$_1$	—	—	0.076	0.563
Func$_2$	—	—	0.065	0.536

Table 4 reports the mean relational distance for different lexical functions in Japanese and Chinese.

The random experiment serves as a baseline in which relation vectors are computed from randomly paired words and therefore lack linguistic constraints, yielding mean relational distances close to zero in both languages. In contrast, the rule-induced condition, which derives relation vectors from lexical function input–output pairs extracted by rules (**Oper**$_i$ and **Func**$_i$), consistently produces substantially larger mean relational distances across all lexical functions with sufficient data in both Japanese and Chinese. This clear and stable gap between the two conditions suggests that the proposed rules induce coherent and shared relational directions in the embedding space.

5 Limitations and future work

Despite the effectiveness of the proposed approach, several limitations remain. Light verb filtering leads to a noticeable reduction of certain lexical function types in the Japanese dataset, with some categories becoming sparse or missing, likely due to limitations in automatic annotation quality. In addition, the extracted dataset lacks a clear hierarchical structure, limiting interpretability. The framework also relies on manually designed rules for a limited set of lexical functions, resulting in restricted coverage.

Future work will focus on applying the dataset to downstream tasks, such as evaluating and fine-tuning large language models to enhance their sensitivity to deep semantic relations. Clustering-based methods will be explored to introduce a more hierarchical structure, and additional linguistic annotations may be incorporated to improve extraction accuracy, albeit at the cost of increased rule complexity.

6 Conclusion

In this paper, we presented a method for the automatic extraction of a lexical functions dataset. By combining semantic role labeling, dependency parsing, and manually created rules, we constructed a lexical function dataset focusing on **Oper**$_i$ and **Func**$_i$ relations.

On the other hand, the lexical function dataset for Japanese and Chinese fills a gap in Meaning-Text Theory resources for Asian languages. This work aims to lower the cost of lexical function extraction, extend lexical function resources to more languages, and support the improvement of LLMs' linguistic capabilities.

References

[1] Igor Mel' čuk and Alain Polguère. A formal lexicon in meaning-text theory (or how to do lexica with words). **Computational Linguistics**, Vol. 13, pp. 261–275, 1987.

[2] Igor Mel' čuk. Collocations and lexical functions. **Phraseology. Theory, analysis, and applications**, pp. 23–53, 1998.

[3] Cynthia A. Thompson, Roger Levy, and Christopher D. Manning. A generative model for semantic role labeling. In Nada Lavrač, Dragan Gamberger, Hendrik Blockeel, and Ljupčo Todorovski, editors, **Machine Learning: ECML 2003**, pp. 397–408, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[4] Joakim Nivre and Mario Scholz. Deterministic dependency parsing of English text. In **COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics**, pp. 64–70, Geneva, Switzerland, aug 23–aug 27 2004. COLING.

[5] Masaaki Nagata, Makoto Morishita, Katsuki Chousa, and Norihito Yasuda. A Japanese-Chinese parallel corpus using crowdsourcing for web mining, 2024.

[6] Han He and Jinho D. Choi. The stem cell hypothesis: Dilemma behind multi-task learning with transformer encoders. In **Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing**, pp. 5555–5577, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)**, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[8] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. Electra: Pre-training text encoders as discriminators rather than generators. In **International Conference on Learning Representations**, 2020.

A Appendix

A.1 Experiment methodology framework

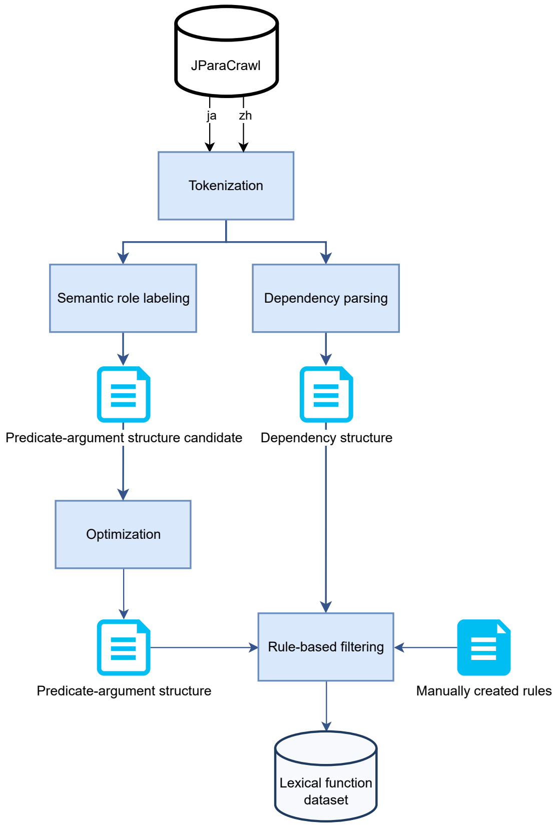

Figure 1 Overview of the lexical function extraction method

A.2 Annotating result example (Japanese)

The following numbers inside the ‘‘srl” part and ‘‘dep” part represent the token index according to the result of tokenization.

A.2.1 Tokenization

```
”tok/fine”: [”この”, ”うち”, ”、”, ”大学”, ”進
学”, ”や”,”就職”, ”で”, ”日本”, ”在住”, ”
中”, ”の”, ”卒業”, ”生”, ”は”, ”160”, ”
名”, ”、”, ”半数”, ”近く”, ”が”, ”再び”, ”
日本”, ”で”, ”暮らして”, ”いる”, ”。”]
```

A.2.2 Semantic role labeling

```
”srl”: [
[[”大学”, ”ニ”, 3, 4],
    [”進学”, ”PRED”, 4, 5],
    [”日本”, ”ニ”, 8, 9]],
[[”日本”, ”デ”, 8, 9],
    [”在住中の卒業生は”, ”ガ”, 9, 15],
    [”在住”, ”PRED”, 9, 10]],
[[”日本在住中の卒業生は”, ”ガ”, 8, 15],
    [”卒業”, ”PRED”, 12, 13]],
[[”日本在住中の”, ”修飾”, 8, 12],
    [”卒業”, ”ノ”, 12, 13],
    [”生”, ”PRED”, 13, 14]],
[[”大学進学や就職で日本在住中の卒業生は”, ”
ガ”, 3, 15],
    [”160”, ”PRED”, 15, 16]],
[[”大学進学や就職で日本在住中の卒業生は”, ”
ノ”, 3, 15],
    [”半数”, ”PRED”, 18, 19]],
[[”半数近くが”, ”ガ”, 18, 21],
    [”日本で”, ”デ”, 22, 24],
    [”暮らして”, ”PRED”, 24, 25]]
]
```

A.2.3 Dependency parsing

The following list shows the result of dependency parsing, where each word is linked to its head by an index and labeled using the universal dependencies annotation style.

```
”dep”: [
  [2,  ”det”],       [17, ”nmod”],     [2,  ”punct”],
  [5,  ”compound”],  [7,  ”nmod”],     [5,  ”case”],
  [11, ”nmod”],      [7,  ”case”],     [11, ”compound”],
  [11, ”compound”],  [14, ”nmod”],     [11, ”case”],
  [14, ”compound”],  [17, ”nsubj”],    [14, ”case”],
  [17, ”nummod”],    [25, ”obl”],      [17, ”punct”],
  [20, ”compound”],  [25, ”nsubj”],    [20, ”case”],
  [25, ”advmod”],    [25, ”obl”],      [23, ”case”],
  [0,  ”root”],      [25, ”aux”],      [25, ”punct”]
]
```