

## A simple baseline method for NTCIR-7 MuST T2N task

— Yokohama National University at NTCIR-7 MuST T2N —

Tatsunori MORI Rintaro MIYAZAKI

Graduate School of Environment and Information Sciences

Yokohama National University

79-7 Tokiwadai, Hodogaya, Yokohama 240-8501, Japan

{mori,rintaro}@forest.eis.ynu.ac.jp

### Abstract

*We participated in the free task and the T2N task of NTCIR-7 MuST. In this paper, we will report our participation in the T2N task. The system we prepared was a very simple and straightforward one. It will serve as a baseline for the T2N task. It consists of the following four modules: i) Element expression extractor, ii) Element expression combiner, iii) Date information canonicalizer, and iv) Selector of relevant statistical data. The main part of our system is the module (i) and a kind of chunk recognizer that is implemented in terms of a sequence labeling task for each character in the given text.*

**Keywords:** *T2N task, chunk recognizer, sequence labeling task.*

### 1 Introduction

We participated in the free task and the T2N task of NTCIR-7 MuST. In this paper, we will report our participation in the T2N task. The system we prepared was a very simple and straightforward one. It will serve as a baseline for the T2N task.

With regard to the free task, it was reported by Mori et al.[2]. Please refer to it for further details.

### 2 Related studies

The main part of our system is a kind of chunk recognizer that is implemented in terms of a sequence labeling task for each character in the given text. This scheme has been widely adopted as a method for the named entity recognition (NER)[4, 1]. In order to improve the accuracy of NER, Nakano et al.[3] introduced new features that are related to Bunsetsu seg-

ments<sup>1</sup>. We utilized one of these features, and we also introduced another kind of Bunsetsu feature, namely, the feature of head morpheme of compound noun, which will be introduced in Section 4.2.

### 3 System overview

The system consists of the following modules as shown in Figure 1:

**Module 1** Element expression extractor

**Module 2** Element expression combiner

**Module 3** Date information canonicalizer

**Module 4** Selector of relevant statistical data

Module 1, 2, and 3 process all given documents to extract a set of all possible statistical data, independently of the statistics that are focused in the extraction task. For each given statistic, Module 4 produces a subset of the set by selecting statistical data that are relevant to the statistic. A statistical datum consists of i) *date element*, which is a surface expression of time-point, or date information, and ii) *value element*, which is a surface expression of numerical value with an unit expression.

The element expression extractor extracts all possible date elements and value elements. The extractor is a kind of chunk recognizer that is implemented in terms of a sequence labeling task for each character in the given text. The labeling process is realized based on a machine learning approach.

The element expression combiner finds a date element for each value element in a given text and produces a pair of a date element and a value element. We

<sup>1</sup>A Bunsetsu segment is a Japanese phrasal unit, which consists of at least one content word and zero or more functional words.

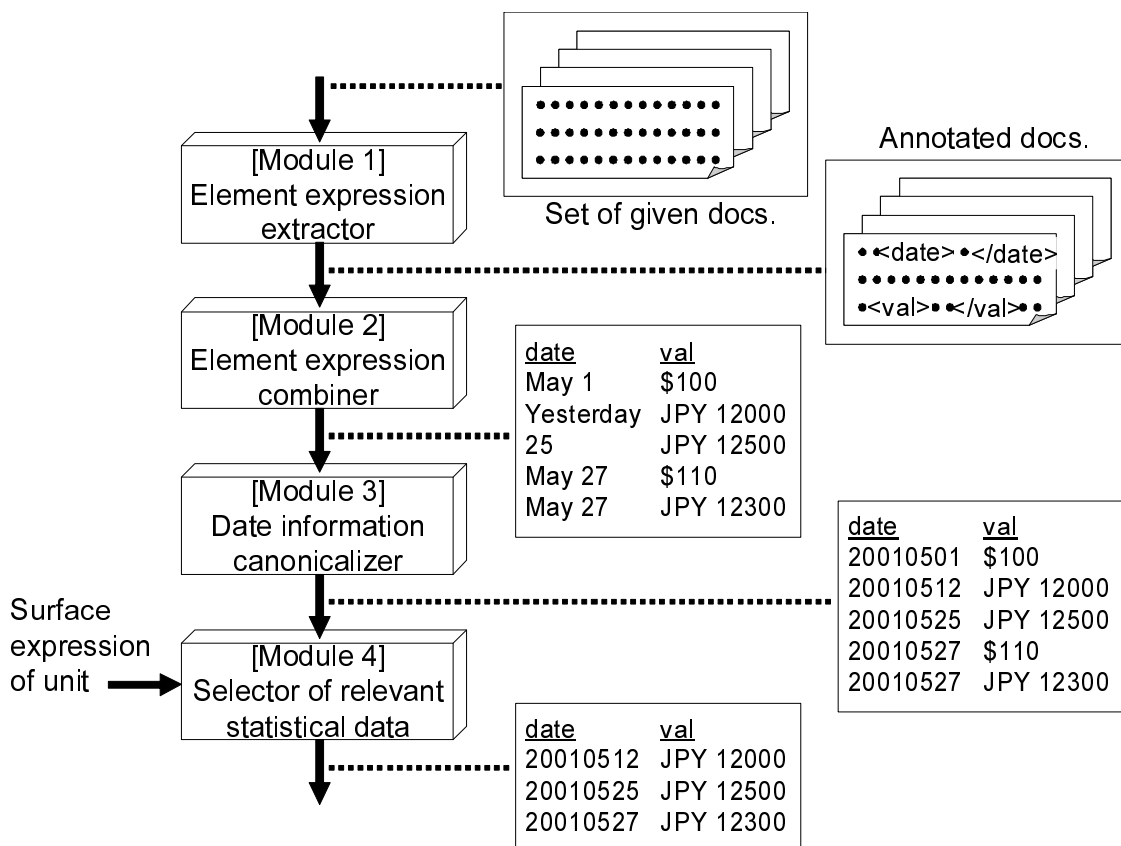


Figure 1. System Overview

introduced a simple heuristic approach to make combinations. It is a heuristic approach based on the distance between two elements.

The date information canonicalizer converts each expression of date information in text into a canonical representation, i.e., the eight digit of ‘YYYYMMDD’ style.

The selector in Module 4 selects statistical data that are relevant to each given statistic. The relevance will be discussed in Section 7.

## 4 Element expression extractor

### 4.1 Element extraction based on the sequence labeling

The element expression extractor recognizes the substrings, or chunks, that correspond to date elements or value elements in given text. Since a text can be regarded as a sequence of some unit expressions, we may implement the process of chunk recognition process as a process of labeling those unit expressions with some appropriate encoding scheme of chunk.

Figure 2 shows a snapshot of the element extraction, in which the unit of expression is a character. The

$i$ -th character along with its context are represented as the tuple of features in the box (A) of Figure 2. According to the feature tuple, the extractor estimates an appropriate label for the  $i$ -th character as shown in the box (B). Each label expresses the state of the corresponding character with some encoding scheme. For example, in Figure 2, the states are encoded with the IOB2 scheme. In the IOB2 scheme, the first character of chunk of the type val is labeled as B-val, and succeeding characters in the chunk is labeled as I-val. Characters that do not belong to any chunks are labeled as O (other).

In order to construct the extractor, we adopt a machine learning approach. In the training phase, a learning algorithm induces a classifier from training data, which contains a set of pairs of (A) a feature tuple and (B) a correct label. In the extraction phase, the induced classifier is applied to a tuple of features obtained from target (unseen) text in order to label each character in the text.

### 4.2 Feature extraction

Several kinds of features are extracted for each character from a given text as shown in the box (A) of

	Character		Morpheme		Canonicalized morpheme		Bunsetsu			Class ID in thesaurus			Chuk label
	Surface expression	Character class	Surface expression	POS	Root form	Representative expression	Left-most NE in BUNSETSU	Head morpheme of compound noun	First level	Second level	Third level		
	ト	KATAK	B-トヨ夕	B-名詞-固有 名詞-組織*	*	ト	名詞-固有 名詞-組織	トヨ夕	*	*	*	O	
	ヨ	KATAK	I-トヨ夕	I-名詞-固有 名詞-組織*	*	ヨ	名詞-固有 名詞-組織	トヨ夕	*	*	*	O	
	夕	KATAK	E-トヨ夕	E-名詞-固有 名詞-組織*	*	夕	名詞-固有 名詞-組織	トヨ夕	*	*	*	O	
	は	HIRAG	S-は	S-助詞-係助詞	*	は		トヨ夕	*	*	*	O	
	9	ZDIGIT	S-9	S-名詞-数	*	9			*	*	*	B-date	
<i>i-2</i>	8	ZDIGIT	S-8	S-名詞-数	*	8			*	*	*	I-date	
<i>i-1</i>	年	OTHER	S-年	S-名詞-接尾-助数詞	*	年		8月	0	1	16	I-date	
<i>i</i>	8	ZDIGIT	B-8月	B-名詞-副詞可能	*	8		8月	*	*	*	I-date	
<i>i+1</i>	月	OTHER	E-8月	E-名詞-副詞可能	*	月		8月	*	*	*	I-date	
<i>i+3</i>	の	HIRAG	S-の	S-助詞-連体化	*	の			*	*	*	O	
	国	OTHER	B-国内	B-名詞-一般	*	国	国内	台数	1	10	103	O	
	内	OTHER	E-国内	E-名詞-一般	*	内	国内	台数	1	10	103	O	
	生	OTHER	B-生産	B-名詞-サ変接続	*	代表表記:生産	国内	台数	3	39	390	O	
	産	OTHER	E-生産	E-名詞-サ変接続	*	代表表記:生産	国内	台数	3	39	390	O	
	台	OTHER	B-台数	B-名詞-一般	*	代表表記:台	国内	台数	*	*	*	O	
	数	OTHER	E-台数	E-名詞-一般	*	数	国内	台数	*	*	*	O	
	が	HIRAG	S-が	S-助詞-格助詞-一般	*	が			*	*	*	O	
	1	ZDIGIT	S-1	S-名詞-数	*	1			*	*	*	B-val	
	8	ZDIGIT	S-8	S-名詞-数	*	8			*	*	*	I-val	
	万	ZDIGIT	S-万	S-名詞-数	*	万			1	12	120	I-val	
	6	ZDIGIT	S-6	S-名詞-数	*	6			*	*	*	I-val	
	3	ZDIGIT	S-3	S-名詞-数	*	3			*	*	*	I-val	
	2	ZDIGIT	S-2	S-名詞-数	*	2			*	*	*	I-val	
	2	ZDIGIT	S-2	S-名詞-数	*	2			*	*	*	I-val	
	台	OTHER	S-台	S-名詞-接尾-助数詞	*	台		台	8	82	828	I-val	
	、	OTHER	S-、	S-記号-読点	*	、			*	*	*	O	

Figure 2. Extraction of element expressions based on the sequence labeling

Figure 2. The feature tuples has a common structure in both the training phase and the extraction phase.

In order to extract features, first, a morphological analyzer and a dependency analyzer are applied to the target text. The dependency analyzer is used not for detecting dependency relation but for finding Bunsetsu boundaries. Then, in a thesaurus, each morpheme is looked up to find the information of its semantic category.

In our experiment, we adopt the following tools: ChaSen as the morphological analyzer, CaboCha as the dependency analyzer, and Kadokawa Ruigo-Shin-Jiten as the thesaurus.

The detailed description of each feature is as follows.

- Features in terms of characters
  - Surface expression** character itself.
  - Character class** the class of character, e.g., Katakana, Hiragana, digit, or others.
- Features in terms of morphemes
  - Surface expression** morpheme itself.
  - POS** the part of speech of morpheme.
- Features in terms of canonicalized morphemes
  - Root form** the root form of morpheme, if the morpheme has the inflection.
  - Representative expression** representative expression of the morpheme that is produced by another morphological analyzer, Juman.
- Features in terms of Bunsetsu segment
  - Left-most NE in Bunsetsu** the POS of the left-most morpheme that is a part of a named entity. If the Bunsetsu has no named entity, the left-most morpheme in the Bunsetsu[3].
  - Head morpheme of compound noun** the right-most morpheme in a series of noun.
- Features in terms of class IDs in the thesaurus that has a three-leveled hierarchy
  - First level** The ID of the class at the first level (more general) to which the morpheme belongs.
  - Second level** The ID of the class at the second level to which the morpheme belongs.
  - Third level** The ID of the class at the third level (more specific) to which the morpheme belongs.

### 4.3 Trancing phase

The annotated corpus for training is constructed from strings in the *KIJI* (article) fields of the file `must_info_abst_0806.xls`. For each string in the *KIJI* field, the substrings that are same as the string in the *JITEN* (time-point, or date) field were annotated with `date`. The substrings that are same as the string in the *val* (value) field in the *type0* field-set were annotated with `val`. Note that we only used above two fields, because we did not treat relative expressions for dates or values in this experiment, at all.

For each character in the annotated corpus, a pair of a feature tuple and a label is obtained. A feature tuple is constructed by extracting features for a character from the (original) text as described in Section 4.2. The label corresponding to the character is derived from the annotated information and the encoding scheme.

By using some learning algorithm, a classifier is automatically so induced as to predict a given correct label from each given feature tuple.

### 4.4 Extraction phase for unseen text

In the extraction phase, a sequence of feature tuples are derived from the sequence of characters in a given text by the feature extraction process described in Section 4.2. The sequence of feature tuples is fed to the classifier described in Section 4.3, and a sequence of labels is predicted as shown in Figure 2. By decoding the sequence of labels, we can find chunks of elements of types `date` or `val`.

## 5 Element expression combiner

After the element expression extractor spots substrings that correspond to date elements or val elements, the element expression combiner finds a plausible date element for each value element in a heuristic manner in order to produces a pair of a date element and a value element. The heuristic is based on the distance between two elements, and is very simple: each value element is combined with the closest date element.

The distance is defined as the number of characters between two elements. In the calculation of distance, an extra distance is added to the total distance as penalty for each time that another sentence boundary is crossed.

If the combiner cannot find any date elements, the default date information of the document is adopted.

The default date information is made of the document ID, which has the day of issue of the document.

## 6 Date information canonicalizer

The date information canonicalizer converts each date element in a given text into a canonical representation, i.e., the eight digit of ‘YYYYMMDD’ style. The canonicalizer consists of a set of rules that are manually constructed from the sample data.

## 7 Selector of relevant statistical data

From the set of all possible pairs of date elements and value elements, the selector of relevant statistical data selects pairs that are relevant to each given statistic. In our implementation, the relevance is judged in a very lenient manner. We do not use the statistic names in name element in the task definition file. The module judges that a pair of data element and value element is relevant, if the suffix of value element is identical to one of unit expressions, which are annotated as `v_unit` in the task definition file. pair of data element and value element Since the selection process is performed independently for each given statistic and the relevance judgment is lenient, a pair may be (wrongly) selected multiple times as parts of different statistics.

## 8 Experiment

### 8.1 Setting of experiment

We employed the following tools, resources, and parameter settings in our experiment in NTCIR-7 MuST T2N.

**Machine learning method** : CRF++<sup>2</sup>, an implementation of the conditional random field.

**Morphological analyzer** : ChaSen<sup>3</sup>.

**Dependency analyzer** : CaboCha<sup>4</sup>.

**Thesaurus** : Kadokawa Rui-go Shin-Jiten (Kadokawa’s new thesaurus).

**distance penalty for crossing sentence boundary** : 5 characters.

<sup>2</sup><http://crfpp.sourceforge.net/>

<sup>3</sup><http://chasen-legacy.sourceforge.jp/>

<sup>4</sup><http://chasen.org/taku/software/cabocho/>

## 8.2 Experimental results

Figure 3 shows the performance of extraction for each set of statistical values in terms of precision, recall and the value of F1 measure. Figure 4 shows the error analysis for each set of statistical values. The extraction result with our system is not good performance. There is a tendency for the precision to be lower than the recall. One of the reasons is the fact that the system does not use the information about the statistic names in name element in the task definition file. The system only deals with unit expressions in order to judge that a pair of data element and value element is relevant to a focused statistic. Therefore, the pairs of data element and value element for different statistics are wrongly confused when their unit expressions are same. As the result, a pair may be judged to be a part of multiple different statistics, and with each statistic the system may tie not only relevant pairs, but also many irrelevant pairs that have the same unit expression. It yields the higher recall and the low precision.

## 9 Conclusion

In this paper, we reported our participation in the T2N task of NTCIR-7 MuST. The system we prepared was a very simple and straightforward one. It consists of the four modules: i) Element expression extractor, ii) Element expression combiner, iii) Date information canonicalizer, and iv) Selector of relevant statistical data. The extraction result with our system was not good performance. There is a tendency for the precision to be lower than the recall. One of the reasons is the fact that the system does not use the information about the statistic names in name element in the task definition file.

On the other hand, we investigated the extraction of statistic names from documents as the free task. Although we could not make use of the extraction method for the T2N task in this participation, we would like to study how does the statistic name extraction contribute to the T2N task in our future work.

## Acknowledgment

We would like to thank people who manage the NTCIR workshops. We are also grateful to Mainichi Shimbun for permitting us to use the documents for research.

This study was partially supported by Grant-in-Aid for Scientific Research (C) (No.19500118) from the

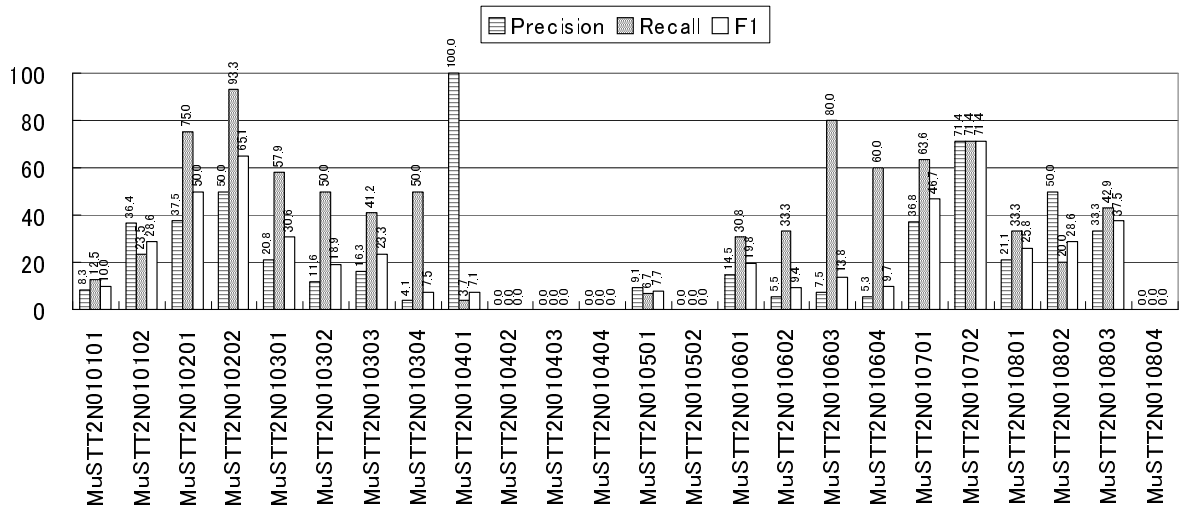


Figure 3. Performance of extraction

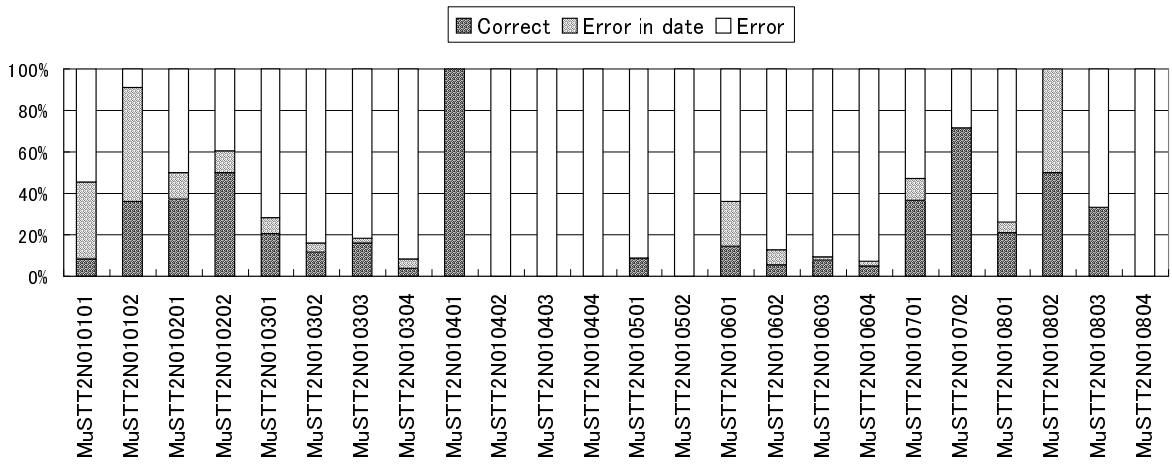


Figure 4. Error analysis

Ministry of Education, Culture, Sports, Science and Technology, Japan.

## References

- [1] M. Asahara and Y. Matsumoto. Japanese named entity extraction with redundant morphological analysis. In *Proceedings of HLT-NAACL 2003*, 2003.
- [2] T. Mori, A. Fujioka, and I. Murata. Automated extraction of statistical expressions from text for information compilation. *Transactions of the Japanese Society for Artificial Intelligence*, 23(5):310–318, 2008. (in Japanese).
- [3] K. Nakano and Y. Hirai. Japanese named entity extraction with Bunsetsu features. *Transactions of Information Processing Society of Japan*, 45(3):934–941, 2004. (in Japanese).
- [4] H. Yamada, T. Kudo, and Y. Matsumoto. Japanese named entity extraction using support vector machine. *IPSJ Journal*, 43(1):44–53, Jan. 2002. (in Japanese).