Automatic Semantic Sequence Extraction from Unrestricted Non-Tagged Texts

Shiho Nobesawa and **Hiroaki Saito** and **Masakazu Nakanishi**
Dept. of Computer Science
Keio University
3-14-1 Hiyoshi Kohoku, Yokohama 223-8522, Japan
{shiho, hxs, czl}@nak.ics.keio.ac.jp

Abstract

Mophological processing, syntactic parsing and other useful tools have been proposed in the field of natural language processing(NLP). Many of those NLP tools take dictionary-based approaches. Thus these tools are often not very efficient with texts written in casual wordings or texts which contain many domain-specific terms, because of the lack of vocabulary.

In this paper we propose a simple method to obtain domain-specific sequences from unrestricted texts using statistical information only. This method is language-independent.

We had experiments on sequence extraction on email texts in Japanese, and succeeded in extracting significant semantic sequences in the test corpus. We tried morphological parsing on the test corpus with ChaSen, a Japanese dictionary-based morphological parser, and examined our system's efficiency in extraction of semantic sequences which were not recognized with ChaSen. Our system detected 69.06% of the unknown words correctly.

1 Introduction

Recognition of contained words is an important preprocessing for syntactic parsing. Word recognition is mostly done based on dictionary lookup, and unknown words often cause parse errors. Thus most of the researches have been done on fixed corpora with special dictionaries for the domain.

Part-of-speech(POS) tags are often used for term recognition. This kind of preprocessing is often time-consuming and causes ambiguity. When it comes to the corpus with high rate of unknown words it is not easy to do a fair parsing with dictionaries and rules.

Obtaining the contained terms and phrases correctly can be an efficient preprocessing. In this paper we propose a method to recognize domain-specific sequences with simple and non-costy processing, which enables the use of unrestricted corpora for NLP tools.

We concentrate on building a tool for extracting meaningful sequences automatically with less preparation. Our system only needs a fair size of non-tagged training corpus of the target language. No restriction is required for the training corpus. We do not need any preprocessing for the training corpus.

We had experiments on email messages in Japanese and our system could recognize 69.06% of the undefined sequences of the test corpus.

2 Japanese Characters and Terms

Taking a word as a basic semantic unit simplifies the confusing tasks of processing real languages. However single words are often not a good unit regarding the meaning of the context, because of the polysemy of the words(Fung, 1998). Instead a phrase or a term can be taken as smallest semantic units.

Most of the phrase/term extraction systems are based on recognizing noun phrases, or domain-specific terms, from large corpora. Argamon et al.(1998) proposed a memory-based approach for noun phrase, which was to learn patterns with several sub-patterns. Ananiadou(1994) proposed a methodology based on term recognition using morphological rules.

2.1 Term Extraction in Japanese

Japanese has no separator between words. On noun phrase extraction many researches have been done in Japanese as well, both stochastic and grammatical ways. In stochastic approaches n-gram is one of the most fascinating model. Noun phrase extraction(Nagao and Mori, 1994), word segmentation(Oda and Kita,

1999) and diction extraction are the major issues. There also are many researches on segmentation according to the entropy. Since Japanese has a great number of characters use of the information of letters is also a very interesting issue.

2.2 Characters in Japanese

Unlike English, Japanese has great amount of characters for daily use. Japanese is special not only for its huge set of characters but its containing of three character types. *Hiragana* is a set of 71 phonetic characters, which are mostly used for function words, inflections and adverbs. *Katakana* is also a set of phonetic characters, each related to a hiragana character. The use is mainly restricted to the representation of foreign words. It's also used to represent pronunciations. *Kanji* is a set of Chinese-origin characters. There are thousands of kanji characters, and each kanji holds its own meaning. They are used to represent content words. We also use alphabetical characters and Arabic numerals.

3 Overview

This system takes Japanese sentences as input. It processes sentences one by one, and we obtain segments of the sentences which are recognized as meaningful sequences as output. The flow of this system is as follows(Figure 1): Our system takes one sentence as an input at one time, and calculates the scores between two neighboring letters according to the statistical data driven from the training corpus. After scoring the system decides which sequences to extract.

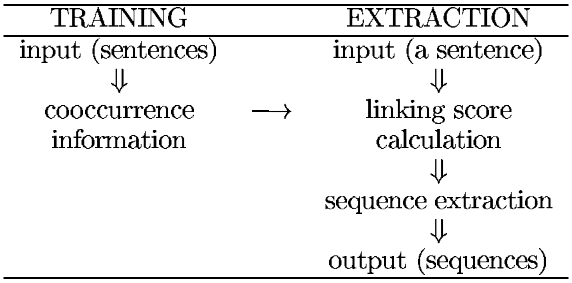

Figure 1: The Flow of the System

3.1 Automatic Sequence Extraction

Nobesawa et al.(1996; 1999) proposed a system which estimates the likelihood of a string of letters be a meaningful block in a sentence. This method does not need any knowledge of lexicon, and they showed that it was possible to segment sentences in meaningful way only with statistical information between letters. The experiment was in Japanese, and they also showed that the cooccurrence information between Japanese letters had enough information for estimating the connection of letters.

We use this point in this paper and had experiments on extracting meaningful sequences in email message texts to make up the lack of vocabulary of dictionaries.

3.2 Scoring

Our system introduces the linking score, which indicates the likelihood that two letters are neighboring as a (part of) meaningful string(Nobesawa et al., 1996).

Only with neighboring bigrams it is impossible to distinguish the events '*XY*' in '*AXYB*' from '*CXYD*'. Thus we introduce d-bigram which is a bigram cooccurrence information concerning the distance(Tsutsumi et al., 1993).

Expression (1) calculates the score between two neighboring letters;

$$UK(i) = \sum_{d=1}^{d_{max}} \sum_{j=i-(d-1)}^{i} MI_d(w_j, w_{j+d}; d) \times g(d) \qquad (1)$$

where w_i as an event , d as the distance between two events, d_{max} as the maximum distance used in the processing (we set $d_{max} = 5$), and $g(d)$ as the weight function on distance (for this system $g(d) = d^{-2}$(Sano et al., 1996), to decrease the influence of the d-bigrams when the distance get longer (Church and Hanks, 1989)). When calculating the linking score between the letters w_i and w_{i+1}, the d-bigram information of the letter pairs around the target two (such as (w_{i-1}, w_{i+2}; 3)) are added.

Expression (2) calculates the mutual information between two events with d-bigram data;

$$MI_d(x, y; d) = log_2 \frac{P(x, y; d)}{P(x)P(y)} \qquad (2)$$

where x, y as events, d for the distance between two events, and $P(x)$ as the probability.

3.3 Sequence Extraction

Using the linking score calculated according to the statistical information, our system searches for the sequences to extract (thus we call our system LSE(linky sequence extraction) system).

Figure 2 shows an example graph of the linking scores for a sentence. Each alphabet letter on the x-axis stands for a letter in a sentence.

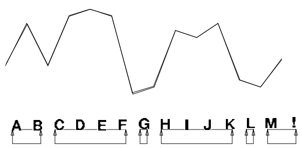

Figure 2: The Score Graph

The linking scores between two neighboring letters are dotted on the graph on the y-axis. Since the linking score gets higher when the pair has stronger connection, the mountain-shaped lines may get considered as unsegmentable blocks of letters. The linking scores of the pairs in longer words/phrases can be higher with the influence of the statistical information of other letter pairs around them. On the other hand, the linking score between two letters which are accidentally neighboring gets lower, and it makes valley-shaped point on the score graph. Our system extracts the mountain-shaped parts of the sentence as the 'linky sequences', which is considered to be meaningful according to the statistical information. In example Figure 2, strings *AB*, *CDEF* and *HIJK* might be extracted.

The height of mountains are not fixed, according to the likelihood of the letters blocked as a string. Thus we need a threshold to decide strings to extract according to the required size and the strength of connection. With higher threshold the strings gets shorter, since the higher linking score means that the neighboring letters can be connected only when they have stronger connection between them.

3.4 How the System Uses the Statistical Information

Figure 3 shows the example graph on a sentence "お元気ですか？[o-gen-ki-de-su-ka-?]"(: How are you?)(Sano, 1997). Each graph line indicates the linking score of the sentence after learning some thousands of sentences of the target domain (for this graph we used a postcard corpus as the target domain, and for the base domain we took a newspaper corpus). When the system have no information on the postcard domain, the system could indicate that only the pair of letters "元気 (gen-ki)" is connectable (there is a mountain-shaped line for this pair). Obtaining the information of postcard corpus, the linking scores of every pair in this sentence get bigger, to make higher mountain. And the shape of the mountain also changes to a flat one mountain which contains whole sentence from a steep mountain with deep valleys.

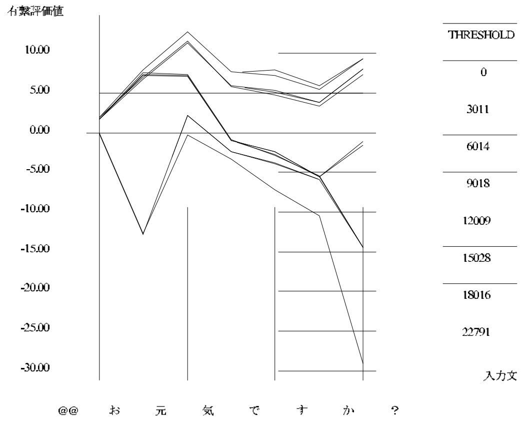

Figure 3: Score Graph for "@@お元気ですか？(@@-o-gen-ki-de-su-ka-?: How are you?)"

4 Experiments

We had experiments on extracting semantic sequences based only on letter cooccurrence information.

We tried a dictionary-based Japanese morphological parser ChaSen ver. 1.51(1997) on the test corpus as well to check sequences which a dictionary-based parser can not recognize.

4.1 Corpus

We chose email messages as the corpora for experiments of our system. Email messages are mostly written in colloquialism, especially when they are written by younger people to send to their friends. In Japanese colloquialism has casual wording which differs from literary style. Casual wording contains emphasizing and terms not in dictionary such as slangs. In English an emphasized word may be written in capital letters, such as in "it SURE is not true", which is easily connected to the basic word "sure". We do the same kind of letter type changes in Japanese for emphasizing, however, since the relationship between letter types is not the same as English, it is not easy to connect the emphasized terms and the basic terms.

4.1.1 Training Corpus

The training corpus we used to extract statistical information is a set of email messages sent between young female friends during 1998 to 1999. This corpus does not cover the one used as the test corpus. All the messages were sent to one receiver, and the number of senders is 17. The email corpus contains 351 email messages, which has 7,865 sentences(176,380 letters, i.e. 22.4 letters per sentence on average).

We did not include quotations of other emails in the training corpus to avoid double-counting of same sentences, though email messages often contain quotations.

4.1.2 Test Corpus

The test corpus is a set of email messages sent between young female friends during 1999. This corpus is not a part of the training corpus. All the messages were sent to one receiver, and the number of senders is 3. This corpus contains 1,118 sentences(24,160 letters, i.e. 21.6 letters per sentence on average).

4.2 Preliminary Results

Figure 4 shows the distribution of the linking scores. The average of the scores is 0.34. The pairs of letters with higher linking scores are treated as highly 'linkable' pairs, that is, pairs with strong connection according to the statistical information of the domain (actually of the training corpus).

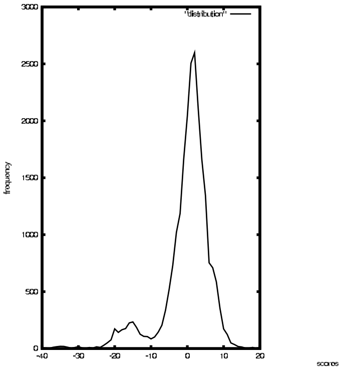

Figure 4: Score Distribution

Pairs of letters with high scores are mainly found in high-scored sequences (Table 1).

Table 1 shows a part of the sequences extracted with our system using letter cooccurrence information. The threshold of extraction for Table 1 is 5.0.

Table 1: Sequences Extracted Based on Letter Cooccurrence

sequence	meaning	frequency
		(with scores over 5.0)
···[a]	......	72
そう	so	52
けど[a]	but	48
から[a]	therefore	43
メール	mail	39
（笑）	(laugh)	36
ワタシ[b]	I	29
それ	it	26
····[a]	......	25
自分	myself	20
ネット[a]	net/Internet	20
!![a]	!!	16
リンク	link	15
友達	friend	13

[a] casual wording

[b] representation change (written in katakana)

These sequences which extracted frequently are the ones often use in the target domain.

4.3 Undefined Words with ChaSen

Since ChaSen is a dictionary-based system, it outputs unknown strings of letters as they are, with a tag 'undefined word'.

Table 2 shows the number of sequences which ChaSen resulted as "undefined words". The row 'undefined words' indicates the sequences which were labeled as 'undefined word' with ChaSen, and the row 'parsing errors' stands for the sequences which were not undefined words with ChaSen but not segmented correctly[1]. The extraction threshold is 0.5.

ChaSen had 627 undefined words as its output. Since the test corpus contains 1,118 sentences, 56.08% of the sentences had an undefined word on average. As it is impossible to divide an undefined sequence into two undefined words, when two or more undefined sequences are neighboring they are often connected into one undefined word[2] Thus the real number of undefined sequences can be more than counted. Table 2 shows that our system on statistical information can be a help to recover 69.06% of the undefined sequences detected by ChaSen.

[1] Since our system is not to put POS tags, we do not count tagging errors with ChaSen (i.e., we do not contain tagging errors in the 'parsing errors').

[2] ChaSen can divide two neighboring undifined sequences when the letter types of the sequences differs.

Table 2: Undefined Words with ChaSen

undefined words		w/ LSE system		
frequency	#total	suc.[a]	part.[b]	failed
over 10	281	230	7	44
3 – 9	143	100	13	30
2	56	43	4	9
1	147	60	44	43
total	627	433	68	126
		69.06%	10.85%	20.10%

[a] suc.: succeeded to extract

[b] part.: partially extracted

Table 2 also shows that this system has better precision with the sequences with larger frequency. For the sequences with frequency over 10 times (in the test corpus), 81.85% of the sequences have extracted correctly. Ignoring sequences which appeared in the test corpus once, the rate of correct extraction rose up to 77.71%.

Table 3 shows how our system worked with the sequences which are found as undefined words with ChaSen parsing system. The threshold for extraction is 0.5. Table 3 shows that the biggest reason for the undefined words are the problem of the representation. As described in Section 4.3.2, we change the way of description when we want to emphasize the sequence. The pronunciation extension with adding extra vowels or extension marks is also for the same reason. Adding these two categories, 356 sequences out of 627 undefined words(56.78%) are caused in this emphasizing.

Table 3: Categories of undefined Words

undefined words		w/ LSE system		
category	#total	suc.[a]	part.[b]	failed
proper nouns	60	39	17	4
new words	70	48	12	10
letter additions	119	89	4	26
changes[c]	276	194	28	54
term. marks[d]	58	43	0	15
smileys	15	9	6	0
etc.	29	12	1	16
total	627	433	68	126

[a] suc.: succeeded to extract

[b] part.: partially extracted

[c] changes: representation changes

[d] term. marks: termination marks

Termination marks as undefined words contain sequences such as “……” and “！！”. The termination marks not in dictionary often indicate the impression, such as surprise, happiness, considering and so on.

New words including proper nouns are the actual ‘undefined words’. ChaSen had 130 of them as its output, that is 20.73% of the undefined words.

4.3.1 Letter Types in Undefined Words

Table 4 shows the types of letters included in the ‘undefined words’ with ChaSen. The figures indicate the numbers of letters.

We had 627 undefined words in the test corpus with ChaSen (Table 2), which contain 1,493 letters totally. The average length of the undefined words is thus 2.38. 70.40% of the letters in undefined words were katakana letters(Table 4), which are phonetic and often used for describing new words. Katakana letters are also often used for emphasizing sequences.

Table 4: Letter Types of Undefined Words

undefined words			w/ LSE system		
type	variety	#total	suc.[a]	part.[b]	failed
kanji	1	19	19	0	0
hiragana	12	200	155	7	38
katakana	73	1051	712	188	151
numeral	1	1	0	1	0
alphabet	23	122	43	72	7
symbol	22	100	39	37	24
	total	1493	968	305	220

[a] suc.: succeeded to extract

[b] part.: partially extracted

On the other hand, there was only one letter each for kanji and numeral figure. That is because each kanji letter and numeral figure has its own meaning, and those letters are mostly found in the dictionary, even though the tags are not semantically correct. Or, as for kanji letters, it sometimes can be tagged with incorrect segmentation[3]. Thus undefined words in kanji letters are not counted as ‘undefined words’ mostly, and instead they cause segmentation failure(Section 4.4).

4.3.2 Representation Changes

Since Japanese have two phonetic character sets, we have several ways to represent one term; in kanji (if there is any for the term), in hiragana, in katakana, or several character type mixed. It is also possible to use Romanization to represent a term.

[3] “この [ko-no](:this)/世界 [se-kai](:the world)” is incorrectly segmented as “この世 [ko-no-yo](:the present life)/界 [kai](:world)”; “kono yo” is a fixed phrase, and “kai” is a suffix for a noun to put the meaning of “the world of”, e.g. “文学界 (:the literary world)”

Table 5 shows the numbers of ChaSen errors according to the representation change. Most of the dictionaries have only one basic representation for one term as its entry[4]. However, in casual writing we sometimes do not use the basic representation, to emphasize the term, or just to simplify the writing.

Table 5: Undefined Words because of Representation Changes

undefined words		w/ LSE system		
subcategory	#total	suc.[a]	part.[b]	failed
term changes	40	33	3	4
katakana	137	102	12	23
change & katakana	55	34	10	11
etc.	44	25	3	16
total	276	194	28	54

[a] suc.: succeeded to extract

[b] part.: partially extracted

4.3.3 Pronunciation Extension

In Japanese language we have many kinds of function words to put at the end of sentences (or sometimes ‘bunsetsu’ blocks). The function words for sentence ends are to change the sound of the sentences, to represent friendliness, ordering, and other emotions. These function words are not basically used in written texts, but in colloquial sentences.

In Japanese language we put extra letters to represent the lengthening of a phone. Since almost all Japanese phones have vowels, to lengthen a phone for emphasizing we put extra vowels or extension marks after the letter. Table 6 shows that 74.79% of the small letters which resulted as undefined words with ChaSen could be salvaged as parts of semantic sequences with our system.

Table 6: Extra Letters output as Undefined Words

letter	ぁ	ぃ	ぅ	ぇ	ぉ	っ	ッ	ン	total
	a	i	u	e	o	*t*	*t*	n	
suc.[a]	39	2	5	32	7	3	1	0	89
part.[b]	0	0	0	0	0	4	0	0	4
failed	5	1	4	2	1	7	5	1	26
total	44	3	9	34	8	14	6	1	119

[a] suc.: succeeded to extract

[b] part.: partially extracted.

These small letters in this table are extra letters to change the pronunciation; i.e. they are mostly not included in the dictionary. However they are actually a part of the word, since they could not be separated from the previous sequences.

4.4 Segmentation Failure with ChaSen

Table 7 shows the result of the extraction of sequences which ChaSen made parsing errors. It indicates that our system could recognize 70.88% of the sequences which ChaSen made parsing errors.

Table 7: Segmentation Failure with ChaSen

undefined words		w/ LSE system		
category	#total	suc.[a]	part.[b]	failed
A	42	41	1	0
B	60	35	10	15
C	92	81	5	6
D	11	2	5	4
E	8	4	3	1
F	176	106	37	33
G	19	10	5	4
H	257	154	73	30
I	253	233	6	14
J	115	82	19	14
total	941	667	159	115
		70.88%	16.90%	12.22%

A: sequences incl. alphabetical characters

B: sequences incl. numeral figures

C: proper nouns

D: new words excl. proper nouns

E: fixed locutions

F: sequences with representation changes

G: sequences in other character types

H: emphasized expressions

I: termination marks

J: parsing errors

[a] suc.: succeeded to extract

[b] part.: partially extracted

Category F is for the sequences which changed their representations according to the terms’ pronunciation changes for casual use. For example, “やっぱ [ya-*p*-pa]” is a casual form of “やはり [ya-ha-ri](: as I thought)”. In casual talking using original term “yahari” sounds a little too polite. Some common casual forms are in dictionaries, but not all.

For the category B, our system failed to extract 25 sequences. All the sequences in B are with counting suffixes. 12 sequences out of the

[4] Dictionaries may have phonetic representations for the entries, not as headings.

25 could not be connected with the counting suffixes, e.g. “３０日 [3-0-nichi](: 30 days, or, the 30th day)” got over-segmented between zero and the suffix. We have a big variety of counting suffixes in Japanese and since our system is only on letter cooccurrence information we could not avoid the over-segmentation.

Category G indicates the sequences which are written in other character types for emphasizing. The major changes are: (1) to write in hiragana characters instead of kanji characters, and (2) to write in katakana characters to emphasize the term.

5 Conclusion

Dictionary-based NLP tools often have worse precision with texts written in casual wordings and texts which contain many domain-specific terms. Term recognition system available for any corpora as a preprocessing enables the use of NLP tools on many kinds of texts.

In this paper we proposed a simple method for term recognition based on statistical information. We had experiments on extracting semantically meaningful sequences according to the statistical information drawn from the training corpus, and our system recognized 69.06% of the sequences which were tagged as undefined words with a conventional morphological parser.

Our system was efficient in recognizing different representations of terms, proper nouns, and other casual wording phrases. This helps to salvage semantically meaningful sequences not in dictionaries and this can be an efficient preprocessing.

6 Future Work

In this paper we proposed a simple term recognition method based only on statistical information. There may be several ways to combine the extracted sequences with the dictionaries. We may need to put POS tags to the sequences for the use with other NLP tools. We expect that we can use tagging tools for this.

This system we propsed is language-independent. For example, we can use this system on English to extract English sequences which appeared frequently in the training corpus, such as proper nouns.

References

Sophia Ananiadou. 1994. A Methodology for Automatic Term Recognition. *Coling-94*, pages 1034–1038.

Shlomo Argamon, Ido Dagan, and Yuval Krymolowski. 1998. A Memory-Based Approach to Learning Shallow Natural Language Patterns. *Coling-ACL'98*, pages 67–73, August.

Kenneth W. Church and Patrick Hanks. 1989. Word Association Norms, Mutual Information, and Lexicography. *The 27th Annual Conference of the Association of Computational Linguistics.*

Pascale Fung. 1998. Extracting Key Terms from Chinese and Japanese texts. *The International Journal on Computer Processing of Oriental Language, Special Issue on Information Retrieval on Oriental Languages.*

Yuji Matsumoto, Akira Kitauchi, Tatsuo Yamashita, Yoshitaka Hirano, Osamu Imaichi, and Tomoaki Imamura. 1997. Japanese Morpholofical Analysis System ChaSen 1.51 Manual. Technical report, Nara Institute of Science and Technology. http://cactus.aist-nara.ac.jp/lab/nlt/chasen.html.

Makoto Nagao and Shinsuke Mori. 1994. A New Method of N-gram Statistics for Large Number of n and Automatic Extraction of Words and Phrases from Large Text Data of Japanese. *Coling-94*, pages 611–615, August.

Shiho Nobesawa, Junya Tsutsumi, Da Jiang Sun, Tomohisa Sano, Kengo Sato, and Masakazu Nakanishi. 1996. Segmenting Sentences into Linky Strings Using D-bigram Statistics. *Coling-96*, pages 586–591, August.

Shiho Nobesawa, Hiroaki Saito, and Masakazu Nakanishi. 1999. String Extraction Based Only on Statistic Linkability. *ICCPOL'99*, pages 23–28, March.

Hiroki Oda and Kenji Kita. 1999. A Character-Based Japanese Word Segmenter Using a PPM*-Based Language Model. *ICCPOL'99*, pages 527–532, March.

Tomohisa Sano, Junya Tsutsumi, Da Jiang Sun, Shiho Nobesawa, Kengo Sato, Kumiko Omori, and Masakazu Nakanishi. 1996. An Experiment on Good Usages of D-bigram Statistics in Natural Language Evaluation. *2nd Annual Meeting of the ANLP (NLP96)*, pages 185–188. Written in Japanese.

Tomohisa Sano. 1997. Natural Language Processing Using Dynamic Statistical Information. Master's thesis, Keio University. Written in Japanese.

Junya Tsutsumi, Tomoaki Nitta, Kotaro Ono, and Shiho Nobesawa. 1993. A Multi-Lingual Translation System Based on A Statistical Model. *JSAI Technical report, SIG-PPAI-9302-2*, pages 7–12. Written in Japanese.