Word Extraction from Corpora and Its Part-of-Speech Estimation Using Distributional Analysis

Shinsuke Mori and Makoto Nagao
Dept. of Electrical Engineering Kyoto University
Yoshida-honmachi, Sakyo, Kyoto, 606-01 Japan
{mori,nagao}@kuee.kyoto-u.ac.jp

Abstract

Unknown words are inevitable at any step of analysis in natural language processing. We propose a method to extract words from a corpus and estimate the probability that each word belongs to given parts of speech (POSs), using a distributional analysis. Our experiments have shown that this method is effective for inferring the POS of unknown words.

1 Introduction

Dictionaries are indispensable in NLP in order to determine the grammatical functions and meanings of words, but the continuous increase of new words and technical terms make unknown words an ongoing problem. A good deal of research has been directed to finding efficient and effective ways of expanding the lexicon. With agglutinative languages like Japanese, the problem is even greater, since even word boundaries are ambiguous. To solve these problems, we propose a method that uses distributional analysis to extract words from a corpus and estimate the probability distribution of their use as different parts of speech.

Distributional analysis was originally proposed by Harris (1951), a structural linguist, as a technique to uncover the structure of a language. Harris intended it as a substitute for what he perceived as unscientific information-gathering by linguists doing field work at that time. Thus, linguists determine whether two words belong to the same class by observing the environments in which the words occur. Recently, this technique has been mathematically refined and used to discover phrase structure from a corpus annotated with POS tags (Brill and Marcus, 1992; Mori and Nagao, 1995). Schütze (1995) used the technique to induce POSs. However, in these researches, the problem of categorial ambiguity (the fact that some words or POS sequences can belong to more than one category), has been ignored.

In this paper, we propose a method that assumes that a word may belong to more than one POS, and provides estimates of the relative probability that it may belong to each of a number of POSs. Our method decomposes an observed probability distribution into a particular linear summation of a given set of model probability distributions. The resulting set of coefficients represents the probability that the observed event belongs to each model event. The application discussed here is word extraction from a Japanese corpus. First we calculate the model probability distribution of each POS by observing the context of each occurrence in a tagged corpus. Then, for each unknown word, we similarly calculate its environment by collecting all occurrences from a raw corpus. Finally, we compute the probability distribution of POSs for a word by comparing its observed environment with the model environments represented by the set of POS distributions.

In subsequent sections, first we discuss the hypothesis, secondly describe the algorithm, thirdly present results of the experiments on the EDR corpus and journal articles, and finally conclude this research.

2 Hypothesis

In this section, first we define environment of a string occurring in a corpus. Next, we propose a hypothesis which gives foundation to our word extraction method.

2.1 Environment of a String in a Corpus

We define the "environment" of a type (character string, group of morphemes, or as the probability distribution of the elements preceding and following occurrences of that type in a corpus. The elements which precede the type are described by the left probability distribution, and those which follow it, by the right probability distribution. For instance, Table 1 shows the one-character environment of the string "楽し" in the EDR corpus (Jap, 1993). This string occurs 181 times, with 12 different characters appearing to its left and 10 to its right.

In general, a probability distribution can be regarded as a vector, so the concatenation of two vectors is also a vector. Thus, the concatenation of the left and right probability distributions for a type is what we call the "environment" of that type, and we represent this by $\boldsymbol{D}$ in the subsequent part of this paper.

Table 1: Environment of the string "楽し"

freq.	prob.	str.		str.	freq.	prob.
13	7.2%	、	楽し	い	16	8.9%
6	3.3%	。		く	3	1.6%
13	7.2%	が		さ	8	4.4%
10	5.6%	て		そ	10	5.6%
8	4.4%	で		ま	7	3.8%
14	7.8%	に		み	41	22.6%
19	10.4%	の		む	38	21.0%
4	2.2%	は		め	16	8.9%
7	3.8%	も		も	4	2.2%
4	2.2%	ら		ん	38	21.0%
83	45.9%	を				
181	100.0%		total		181	100.0%

2.2 Hypothesis Concerning Environment

In general, if a string $\boldsymbol{\alpha}$ is a word which belongs to a POS, it is expected that the environment $\boldsymbol{D}(\boldsymbol{\alpha})$ of the string in a particular corpus will be similar to the environment $\boldsymbol{D}(pos)$ of that POS. Since a word can belong to more than one POS, it is expected that the environment of the string will be similar to the summation across all POSs of the environment of each POS multiplied by the probability that the string occurs as that POS. Therefore, we obtain the following formula:

$$\boldsymbol{D}(\boldsymbol{\alpha}) \approx \sum_k p(pos_k|\boldsymbol{\alpha})\boldsymbol{D}(pos_k) \qquad (1)$$

where $p(pos_k|\boldsymbol{\alpha})$ is the probability that the string $\boldsymbol{\alpha}$ belongs to pos_k, and $\boldsymbol{D}(pos_k)$ is the environment of pos_k. In this formula, summation is calculated for the set of POSs in consideration. As an example, let us take the string "楽し", which is used in the corpus only as a verb and an adjective. If $p(Adj|\text{楽し})$ and $p(Verb|\text{楽し})$ are the probabilities that a particular instance of the string is used as an adjective and a verb respectively, then the environment of the string "楽し" is described by the following formula: $\boldsymbol{D}(\text{楽し}) \approx p(Adj|\text{楽し})\boldsymbol{D}(Adj) + p(Verb|\text{楽し})\boldsymbol{D}(Verb)$.

In most cases, however, formula (1) cannot be solved as a linear equation, since the dimension of probability distribution vector $\boldsymbol{D}$ is greater than that of the independent variables. In addition, we need to minimize the effects of sample bias inherent in statistical estimates of this sort. We therefore reason that the question is to find the set of $p(pos_k|\boldsymbol{\alpha})$ which minimizes the difference between both sides of formula (1) in terms of some measure. We use, as this measure, the square of Euclidean distance betwen vectors. Then it follows that the problem is formalized as an optimization problem (minimize). The decision variables are the elements of the probability distribution vector $\boldsymbol{p}$ which expresses the likelihood that the string is used as each POS:

$$F(\boldsymbol{p}) = |\boldsymbol{D}(\boldsymbol{\alpha}) - \sum_k p_k \boldsymbol{D}(pos_k)|^2 \qquad (2)$$

where $\boldsymbol{p} = (p_1, p_2, \ldots, p_n)$, $p_k = p(pos_k|\boldsymbol{\alpha})$ and n is the number of POSs in consideration. Since each element of $\boldsymbol{p}$ represents a probability, the feasible region V is given as follows:

$$V = \{\boldsymbol{p} \mid 0 \le p_k \le 1, \sum_k p_k = 1\} \qquad (3)$$

The minimum value of $F(\boldsymbol{p})$ will be relatively small when the environment of the string can be decomposed into a linear summation of some POS environments, while it will be relatively large when such a decomposition does not exist. Since all true words must belong to one or more POSs, the minimum value of $F(\boldsymbol{p})$ can be used to decide whether a string is a word or not. We call this value the "word measure," and accept as words all strings with word measure less than a certain threshold.

3 Algorithm

In this section we describe the algorithm used to calculate the word measure of an arbitrary string and the probabilities that the string belongs to each of a set of POSs. We used observations from the EDR corpus, which is divided into words and tagged as to POS, to calculate the POS environments, and then used a raw corpus (no indication of word or morpheme boundaries, and no POS tags) for calculating the string environments.

3.1 Calculating POS Environments

The environment of each POS is obtained by calculating statistics on all contexts that precede and follow the POS in a tagged corpus, as follows:

1. Let all elements of left and right probability vectors be 0.
2. For each occurrence of the POS in the corpus, increment the left vector element corresponding to the context preceding this occurrence of the POS, and increment the right vector element corresponding to the context following the POS.
3. Divide each vector element by the total number of occurrences of the POS.

Figure 1 shows a sample sentence from the EDR corpus, and Table 2 shows the computation of the one-character environment of Noun in the tiny corpus consisting of this single sentence.

In practice, instead of a single character, we used as contexts the preceding or following POS-tagged string (a morpheme or word). Thus the

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
しかし	、	元日	の	紙面	は	新し	い	年	へ	の	姿勢	を	示	す	年賀状	だ	。
conj.	sign	noun	pp	noun	pp	adj.	infl.	noun	pp	pp	noun	pp	verb	infl.	noun	aux.	sign

Figure 1: An Example of EDR Corpus

probability vectors, which consisted of all the contexts found for any POS, were so sparse that we used a hash algorithm.

Table 2: Environment of the Noun

freq.	prob.	str.		str.	freq.	prob.
1	20%	、	noun	だ	1	20%
1	20%	い		の	1	20%
1	20%	す		は	1	20%
2	40%	の		へ	1	20%
				を	1	20%
5	100%		total		5	100%

3.2 Calculating String Environments

The calculation of the environment of an arbitrary string (possible word) in a corpus is basically identical to the POS algorithm above, except that because Japanese has no blank space between words and a raw (unsegmented) corpus is used, the extent of the environment is ambiguous. There are two ways to determine the extent of the left and right environment: one is to specify a fixed number of characters, and the other is to use a look-up-and-match procedure to identify specific morphemes. We adopted the second method, and used as a morpheme lexicon the set of hash keys representing the POS environments. Where there was a conflict between two or more possible matches of a string context with the POS hash keys, the longest match was selected. For instance, although a right context から 'kara' could match either the postposition 'ka' or the postposition 'kara', the longer match 'kara' would always be chosen.

3.3 Optimization

The environments for a string and for each POS which it represents become the parameters of the objective function defined by formula (2), and the optimization of this function then yields the probabilities that the string belongs to each POS. The problem can be solved easily by the optimal gradient method because both the objective function and the feasible region are convex.

4 Results

We conducted two experiments, in each using a range of different thresholds for word measure. One experiment used the EDR corpus as a raw corpus (ignoring the POS tags) in order to calculate recall and precision. The other experiment used articles from one year of the Japanese version of *Scientific American* in order to test whether we could increase the accuracy of the morphological analyzer (tagger) by this method.

Table 3: Recall and precision on EDR corpus

threshold	recall		precision	
of F_{min}	tokens	types	tokens	types
0.10	26.2%	12.6%	96.8%	86.2%
0.15	46.4%	28.3%	94.0%	80.4%
0.20	61.8%	44.0%	90.7%	74.9%
0.25	73.2%	57.1%	87.4%	69.1%
0.30	79.8%	66.7%	—	—
0.35	84.4%	73.7%	—	—

4.1 Conditions of the Experiments

For both experiments, we considered the five POSs to which almost all unknown words in Japanese belong:

1. verbal noun, e.g. 勉強(する) 'benkyou(suru)' "to study"
2. nouns, e.g. 学校 'gakkou' "school"
3. ra-type verb, e.g. 食べ(る) 'tabe(ru)' "to eat"
4. i-type adjective, e.g. 寒(い) 'samu(i)' "cold"
5. na-type adjective, e.g. きれい(な) 'kirei(na)' "clean"

POS environments were defined as one POS-tagged string (assumed to be one morpheme), and were limited to strings made up only of *hiragana* characters plus comma and period. The aim of this limitation was to reduce computational time during matching, and it was felt that morphemes using *kanji* and *katakana* characters are too infrequent as contexts to exert much influence on the results.

Candidate for unknown words were limited to strings of two or more characters appearing in the corpus at least ten times and not containing any symbols such as parentheses. Since there are very few unknown words which consist of only one character, this limitation will not have much effect on the recall.

4.2 Experiment 1: Word Extraction

For evaluation purposes, we conducted a word extraction experiment using the EDR corpus as a raw corpus, and calculated recall and precision for each threshold value (see Table 3). First, we calculated $F_{min}$and $\boldsymbol{p}$ for all character n-grams,

Table 4: Examples of extracted words from "Science"

F_{min}	freq.	string	Action noun	Noun	Ra-type verb	I-type adj.	Na-type adj.
0.04	115	自然	0.00	0.31	0.00	0.00	0.69
0.05	179	十分	0.00	0.00	0.00	0.00	1.00
0.08	103	タンパク質 *	0.00	1.00	0.00	0.00	0.00
0.11	63	MHC分子 *	0.00	1.00	0.00	0.00	0.00

$*$ means unknown word.

$2 \leq n \leq 20$, excluding strings which consisted only of *hiragana* characters. Then, for each threshold level, our algorithm decided which of the candidate strings were words, and assigned a POS to each instance of the word-strings.

Recall was computed as the percent of all POS-tagged strings in the EDR corpus that were successfully identified by our algorithm as words and as belonging to the correct POS. In calculation of the recalls and the precisions, both POS and string is distinguished. Precision was calculated using the estimated frequency $f(\boldsymbol{\alpha}, pos) = p(pos|\boldsymbol{\alpha}) \cdot f(\boldsymbol{\alpha})$ where $f(\boldsymbol{\alpha})$ is the frequency of the string $\boldsymbol{\alpha}$ in the corpus, and $p(pos|\boldsymbol{\alpha})$ is the estimated probability that $\boldsymbol{\alpha}$ belongs to the *pos*.

Judgement whether the string $\boldsymbol{\alpha}$ belongs to *pos* or not was made by hand. The recalls are calculated for ones with the estimated probability more than or equal to 0.1. The reason for this is that the amount of the output is too enormous to check by hand. For the same reason we did not calculate the precisions for thresholds more than 0.25 in Table 3. This table tells us that the lower the threshold is, the higher the precision is. This result is consistent with the result derived from the hypothesis that we described in section 2.2. Besides, there is a tendency that in proportion as the frequency increases the precision rises.

4.3 Experiment 2: Improvement of Stochastic Tagging

In order to test how much the accuracy of a tagger could be improved by adding extracted words to its dictionary, we developed a tagger based on a simple Markov model and analyzed one journal article[1]. Using statistical parameters estimated from the EDR corpus, and an unknown word model based on character set heuristics (any *kanji* sequence is a noun, etc.), tagging accuracy was 95.9% (the percent of output morphemes which were correctly segmented and tagged).

Next, we extracted words from the Japanese version of *Scientific American* (1990; 617,837 characters) using a threshold of 0.25. Unknown words were considered to be those which could not be divided into morphemes appearing in the learning corpus of the Markov model. Table 4 shows examples of extracted words, with unknown words starred. Notice that some extracted words consist of more than one type of character, such as "タンパク質 (protein)." This is one of the advantages of our method over heuristics based on character type, which can never recognize mixed-character words. Another advantage is that our method is applicable to words belonging to more than one POS. For example, in Table 4 "自然 (nature)" is both a noun and the stem of a na-type adjective.

We added the extracted unknown words to the dictionary of the stochastic tagger, where they are recorded with a frequency calculated by the following formula: $(size_e/size_s)f(\boldsymbol{\alpha}, pos)$, where $size_e$ and $size_s$ are the size of the EDR corpus and the size of the *Scientific American* corpus respectively. Using this expanded dictionary, the tagger's accuracy improved to 98.2%. This result tells us that our method is useful as a preprocessor for a tagger.

5 Conclusion

We have described a new method to extract words from a corpus and estimate their POSs using distributional analysis. Our method is based on the hypothesis that sets of strings preceding or following two arbitrary words belonging to the same POS are similar to each other. We have proposed a mathematically well-founded method to compute probability distribution in which a string belongs to given POSs. The results of word extraction experiments attested the correctness of our hypothesis. Adding extracted words to the dictionary, the accuracy of a morphological analyzer augmented considerably.

References

Eric Brill and Mitchell Marcus. 1992. Automatically acquiring phrase structure using distributional analysis. In *Proc. of the DARPA Speech and Natural Language Workshop*

Zellig Harris. 1951. *Structural Linguistics.* University of Chicago Press.

Japan Electronic Dictionary Research Institute, Ltd., 1993. *EDR Electronic Dictionary Technical Guide.*

Shinsuke Mori and Makoto Nagao. 1995. Parsing without grammar. In *Proc. of the IWPT95*

Hinrich Schütze. 1995. Distributional part-of-speech tagging. In *Proc. of the EACL95.*

[1] "Progress in Gallium Arsenide Semiconductors" (*Scientific American*; February, 1990)