Computational Linguistics and Intelligent Text Processing Lecture Notes in Computer Science N 2588, Springer-Verlag, 2003 (Proceedings of CICLing-2003) pp.564-574

Probabilistic Word Vector and Similarity based on Dictionaries

Satoshi Suzuki

NTT Communication Science Laboratories
NTT, Japan
satoshi@cslab.kecl.ntt.co.jp

Abstract. We propose a new method for computing the probabilistic vector expression of words based on dictionaries. This method provides a well-founded procedure based on stochastic process whose applicability is clear. The proposed method exploits the relationship between headwords and their explanatory notes in dictionaries. An explanatory note is a set of other words, each of which is expanded by its own explanatory note. This expansion is repeatedly applied, but even explanatory notes expanded infinitely can be computed under a simple assumption. The vector expression we obtain is a semantic expansion of the explanatory notes of words. We explain how to acquire the vector expression from these expanded explanatory notes. We also demonstrate a word similarity computation based on a Japanese dictionary and evaluate it in comparison with a known system based on $TF \cdot IDF$. The results show the effectiveness and applicability of this probabilistic vector expression.

1 Introduction

Word frequency vectors for information retrieval (IR) are generally calculated with *Term Frequency* $\cdot$ *Inverse Document Frequency* ($TF \cdot IDF$) or simple normalization. While these methods are certainly useful and effective in some applications, they are heuristic and do not seem to be firmly grounded in a principle that explains why these methods are selected or why they work well. Papineni, for example, showed that IDF is optimal for document self-retrieval with respect to a generalized Kullback-Leibler distance [1]. However, this argument does not take into account the co-occurrence of words and, therefore, cannot be applied to $TF \cdot IDF$. Such uncertainty regarding $TF \cdot IDF$ may often cause confusion when it is applied to particular applications, for example, not knowing whether these word frequency vectors can be reasonably added up or multiplied. To avoid such confusion, we investigate a new well-grounded method for computing word frequency vectors that can be used instead of $TF \cdot IDF$ or simple normalization.

Recently, learning methods based on stochastic processes have become popular in the field of computational learning theories because of their simple descriptions and logically founded procedures. As for IR, some probabilistic methods

have also been proposed lately. Hofmann, for example, suggested Probabilistic Latent Semantic Indexing (PLSI), which provides an alternative method that can be written as a matrix product resembling the singular-value decomposition underlying Latent Semantic Indexing (LSI) [2]. Using a probabilistic description makes it easy to understand what each process does and how the processes are applied. Hence, we also try to apply a stochastic process to the computation of word frequency vectors from dictionaries to establish a well-grounded method.

The method we propose constructs probabilistic vectors by expanding the semantics of words that are given as explanatory notes in dictionaries. The explanatory notes may not sufficiently describe the general meaning of the words, but each explanatory note consists of words that are further explained by their own explanatory notes. Such semantic expansion can be repeatedly applied to assemble many large explanatory notes. We can therefore expect them to provide a more general description of word semantics.

A way of dealing with these large explanatory notes expanded infinitely will be described in the next section. We explain how to deal with headwords and their explanatory notes in dictionaries and produce a word frequency vector based on a stochastic process.

To check the effectiveness of the proposed vector expression, we examined an application for measuring word similarity that is also based on a stochastic process. Our definition of word similarity and our computational method is detailed in Section 3. Results of computational experiments with a Japanese dictionary are also reported in that section.

2 Probabilistic Word Vector

2.1 Basic Idea

Dictionaries are composed of sets consisting of a headword and a related explanatory note. However, the explanatory note does not always explain the headword sufficiently. Therefore, we investigated a method of realizing ideal explanatory notes from the original notes. This approach is based on the following assumption (see Figure 1).

A headword is explained by its explanatory note, and the words in the explanatory note are also explained by their own explanatory notes. Consequently, hierarchical explanations may continue infinitely. As a result, a headword obtains many large explanatory notes, each of which has a different depth of hierarchy. Here, we assume that the ideal explanatory note is a probabilistic combination of these large explanatory notes, whose ratios become smaller according to the hierarchical depth. This assumption makes it possible to calculate the ideal explanatory note even if the hierarchical explanatory note at infinity cannot be computed.

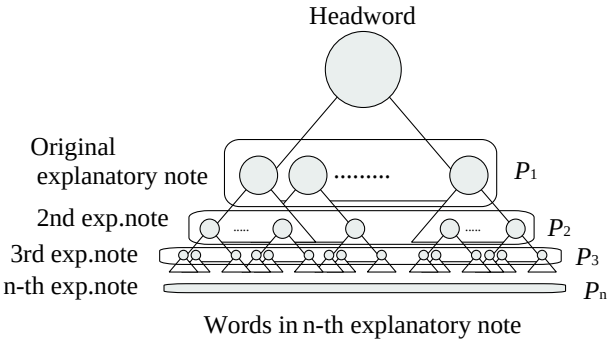


Fig. 1. Model of semantic expansion

2.2 Methods

Here, we describe how to compute ideal explanatory notes from dictionaries. First, we explain the notation of word frequency in explanatory notes. Explanatory notes are expressed as a set of probabilistic word frequencies.

We write the relationship between headword w_i and word w_j in the form $P(w_j^{(1)}|w_i)$, where $w_j^{(1)}$ means word w_j in the original (first) explanatory note. This means that $P(w_j^{(1)}|w_i)$ is the probability that word w_j appears in the explanatory note of headword w_i. The probabilities over all headwords can be formulated as a square matrix:

$$A = \begin{bmatrix} P\left(w_1^{(1)}|w_1\right) & P\left(w_1^{(1)}|w_2\right) & \cdots & P\left(w_1^{(1)}|w_m\right) \\ P\left(w_2^{(1)}|w_1\right) & \ddots & & \\ \vdots & & \ddots & \\ P\left(w_m^{(1)}|w_1\right) & & & P\left(w_m^{(1)}|w_m\right) \end{bmatrix}, \tag{1}$$

where m is the number of headwords in the dictionaries. Each element $P(w_j^{(1)}|w_i)$ is equal to the probabilistic frequency of w_j in the explanatory note of w_i, i.e.,

$$P(w_j^{(1)}|w_i) = \frac{N(w_j^{(1)})}{\sum_{all\ k} N(w_k^{(1)})}, \tag{2}$$

where $N(w_j^{(1)}), N(w_k^{(1)})$ is the frequency of the word in the explanatory note of w_i. Column vectors of probability matrix A are the original word frequency vectors.

Next, we try to obtain a secondary explanatory note that is a probabilistic combination of the original explanatory notes. All words in the original explanatory note are regarded as headwords, and their explanatory notes are probabilistically combined into a secondary explanatory note. The probability of word w_j

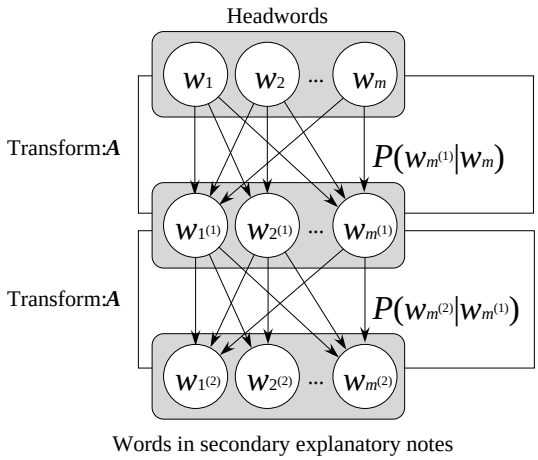


Fig. 2. Model of secondary explanatory notes

in the secondary explanatory note of headword w_i is expressed in a formula:

$$P(w_j^{(2)}|w_i) = \sum_{all\ k} P(w_j^{(2)}|w_k^{(1)})P(w_k^{(1)}|w_i), \tag{3}$$

where $w_k^{(1)}$ is a word in the explanatory note of w_i. Formula (3) over all words can be written as a formula of matrix A: A^2.

Figure 2 shows a model of secondary explanatory notes. All paths from a headword on the top layer to a word on the bottom layer pass through one of the words on the second layer. Formula (3) shows all of these paths, and matrix A expresses the relationship between the neighboring two layers.

Generally, we can formulate probability $P(w_j^{(n)}|w_i)$ as follows, where $w_j^{(n)}$ is a word in the nth explanatory note of headword w_i:

$$P(w_j^{(n)}|w_i) = \tag{4}$$
$$\sum_{all\ k_{n-1}} \sum_{all\ k_{n-2}} \cdots \sum_{all\ k_1} P(w_j^{(n)}|w_{k_{n-1}}^{(n-1)})P(w_{k_{n-1}}^{(n-1)}|w_{k_{n-2}}^{(n-2)}) \cdots P(w_{k_1}^{(1)}|w_i).$$

Formula (4) over all words can also be written as a formula of matrix A: A^n.

Now, we probabilistically combine all the explanatory notes from the first to infinity. That is, we compute the following formula:

$$C = P_1 A + P_2 A^2 + \cdots + P_n A^n + \cdots, \tag{5}$$

where $P_1, P_2, \cdots, P_n, \cdots$ are probabilities of selecting the models of the hierarchical explanatory note.

Figure 3 shows a model of the ideal explanatory notes. This model illustrates the probabilistic combinations of all hierarchical explanatory notes as expressed by formula (5).

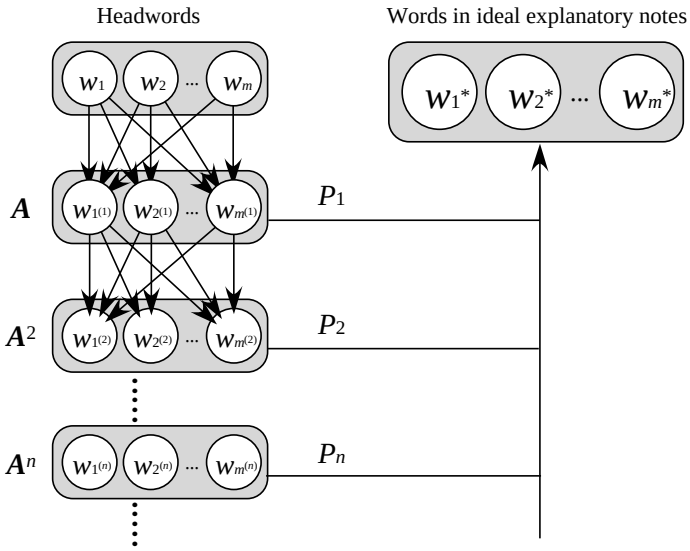


Fig. 3. Model of ideal explanatory notes

Generally, it is extremely difficult to calculate C exactly. However, when the probability P_n becomes smaller at a certain rate according to n, C can be formulated as

$$C = b(aA + a^2A^2 + \cdots + a^nA^n + \cdots), \tag{6}$$

where a, b are parameters that satisfy the following:

$$0 < a < 1, \tag{7}$$

$$P_n = ba^n, \tag{8}$$

$$\sum_{n=1}^{\infty} P_n = 1. \tag{9}$$

We can obtain b as a formula of a from the infinite series given by these equations:

$$b = \frac{1-a}{a}. \tag{10}$$

Consequently, we can transform formula (6) into an equation:

$$(I - aA)C = (1 - a)A. \tag{11}$$

If matrix $(I - aA)$ is non-singular, we can directly compute matrix C by the following formula:

$$C = (1 - a)A(I - aA)^{-1}. \tag{12}$$

Alternatively, we could use a numerical solution of linear equations of the ith column vector v_i of C:

$$(I - aA)v_i = (1 - a)A_i, \tag{13}$$

where A_i is the ith column vector of A. Otherwise, we could estimate v_i with some learning methods in formula (13). In any case, vector v_i can be computed.

The (j, i) element of matrix C is $P(w_j^*|w_i)$, which indicates the probability that word w_j appears in the ideal explanatory note of headword w_i. We can therefore regard the ith column vector of matrix C as a probabilistic frequency vector of word w_i.

2.3 Computation of word vectors

We next describe simulation results based on the method presented above. We used a Japanese dictionary in the simulation [3]. As preprocessing, general nouns and verbal nouns[1] were listed as headwords, and example sentences in their explanatory notes were as far as possible excluded. ChaSen [4] was used as a morphological analyzer. The total number of headwords was 44,050, and the average number of words in an original explanatory note was about seven.

First, probability matrix A was calculated with formula (2), which is a 44,050-dimensional square matrix.

Second, all column vectors of matrix C were estimated by a learning method that minimizes squared errors to solve equation (13), where parameter a was set at 0.9. After the learning, we excluded words where the learning did not converge or where the learning error was bigger than a certain threshold. The result provided us with 43,616 headwords and their probabilistic frequency vectors. The average number of non-zero elements of the column vectors was around 25,000. This means that more than half of all the headwords are used in each ideal explanatory note.

Table 1 shows examples of the probabilistic frequency vectors.[2] Probabilistic frequencies in the original and ideal explanatory notes are listed in the table with regard to two headwords. These values are elements of probabilistic frequency vectors, and all of the elements of each column vector naturally add up to 1. We can roughly say that the probabilistic frequency in an ideal explanatory note is large according to the probabilistic frequency in the original explanatory note, aside from the headword itself.

3 Word Similarity

To evaluate the probabilistic word vector, we tried to compute word similarity. First, we define the similarity of words and explore a method for computing it, which is based on a stochastic process.

[1] Some nouns work as verbs with a post-positional auxiliary verb "suru" in Japanese. For example, "denwa"(telephone) + "suru" means 'make a phone call'.

[2] See the next section for a detailed explanation of word similarity.

3.1 Definition and Method

We define the similarity of words as the probability that a headword is estimated from the ideal explanatory note of another headword. This similarity expresses how closely a headword represents the ideal explanatory note of another headword. Therefore, the similarity of all headwords to a certain headword can be described as a probability vector.

The probability that headword w_i represents a word w_j in an ideal explanatory note is formulated as follows:

$$P(w_i|w_j^*) = \frac{P(w_j^*|w_i)P(w_i)}{\sum_{all\ k} P(w_j^*|w_k)P(w_k)}, \tag{14}$$

where $P(w_i)$ is the *a priori* probability of w_i. Note that $P(w_i|w_j^*)$ is the probability of a headword estimated from an ideal explanatory note, not of a word in the next hierarchy of the explanatory note. We cannot calculate $P(w_i|w_j^*)$ directly but can use the (i,j) element of the probabilistic frequency matrix C as $P(w_i^*|w_j)$ in formula (14).

The similarity of headword w_i from headword w_j is obtained by processing all the words in the ideal explanatory note of w_j, i.e.,

$$\begin{aligned} P(w_i|w_j) &= \sum_{all\ k} P(w_i|w_k^*)P(w_k^*|w_j) \\ &= \sum_{all\ k} \frac{P(w_k^*|w_i)P(w_i)P(w_k^*|w_j)}{\sum_{all\ l} P(w_k^*|w_l)P(w_l)}. \end{aligned} \tag{15}$$

3.2 Simulation

To compute formula (15), we applied the results of the probabilistic frequency vector obtained in Section 2.3. $P(w_k^*|w_i)$ in the formula is the kth element of the probabilistic word vector of w_i. By contrast, the values of another unknown parameter, *a priori* probability $P(w_i)$, are not yet given. $P(w_i)$ means the probability of word w_i appearing without a precondition. Here, it should be remembered that all headwords appear once in a dictionary. Hence, we can assume the *a priori* probabilities of all headwords to be equal, giving the following formula:

$$P(w_i|w_j) = \sum_{all\ k} \frac{P(w_k^*|w_i)P(w_k^*|w_j)}{\sum_{all\ l} P(w_k^*|w_l)}. \tag{16}$$

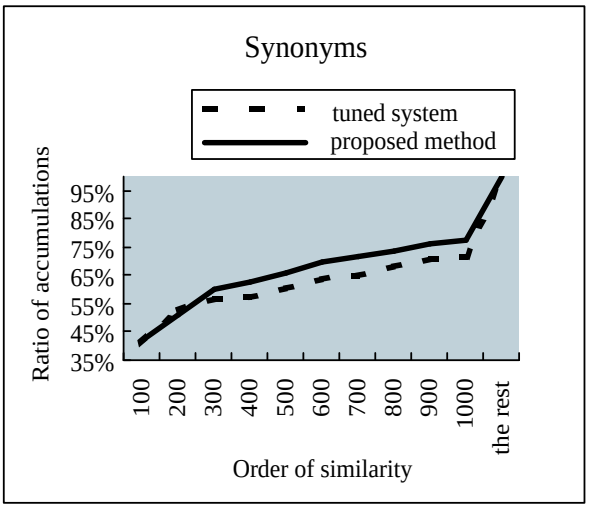
The results of the computation with formula (16) are also shown in Table 1. Compared with the probabilistic frequency result, shown on the left hand in the table, common words such as "標準 (standard)" or "人物 (person)" have relatively low values. By contrast, words that are semantically close but do not appear in the explanatory notes have high values, e.g., "レベルアップ (raise level)" or "ガード (guard)".

レベル (level)				
Words	Probabilistic frequency		Word similarity	
in explanatory note	Original exp.	Ideal exp.	Similarity	Order
水準儀 (leveling instrument)	0.1	0.012765	0.005534	1
レベル (level)	0.1	0.012487	0.005037	2
レベルアップ (raise level)	0	0	0.003480	3
平準 (make level)	0	0	0.001155	4
水準 (level)	0.2	0.025732	0.000914	5
精度 (precision)	0	0.002002	0.000791	6
儀 (affair)	0	0.000002	0.000754	7
準 (semi-)	0	0.000001	0.000684	8
准 (semi-)	0	0.000001	0.000684	8
別儀 (distinguish)	0	0	0.000617	10
級 (degree)	0.1	0.011949	0.000273	25
トップ (top)	0.1	0.011806	0.000197	47
標準 (standard)	0.2	0.026058	0.000092	166
程度 (grade)	0.1	0.019590	0.000085	187
段階 (echelon)	0.1	0.014523	0.000070	277
⋮				

ボディーガード (bodyguard)				
Words	Probabilistic frequency		Word similarity	
in explanatory note	Original exp.	Ideal exp.	Similarity	Order
ボディーガード (bodyguard)	0	0.024390	0.025666	1
ガード (guard)	0	0	0.004786	2
身辺 (one's affair)	0.25	0.047686	0.003050	3
警衛 (guard)	0	0	0.002760	4
親衛 (guard the king)	0	0.000010	0.002464	5
用心棒 (bodyguard)	0.25	0.026339	0.002308	6
エスコート (escort)	0	0	0.002254	7
座右の銘 (familiar proverb)	0	0	0.001947	8
警護 (guard)	0	0.000192	0.001810	9
護衛 (guard)	0.25	0.033887	0.001569	10
人物 (person)	0.25	0.026765	0.000090	224
⋮				

Table 1. Examples of probabilistic word vector and similarity.

(A)



(B)

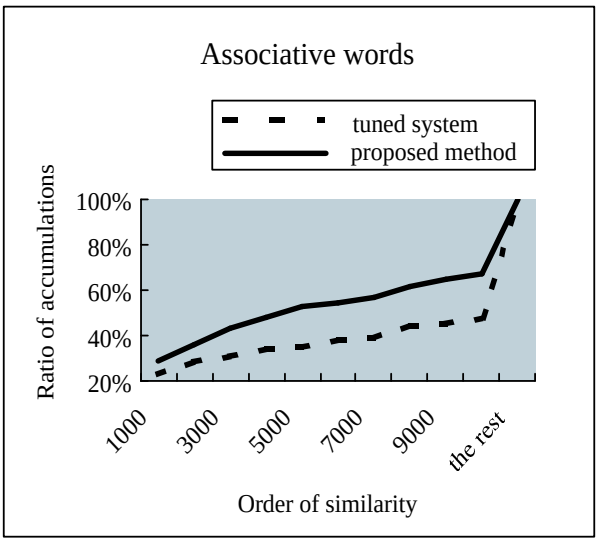


Fig. 4. Comparison of proposed method and tuned system.

3.3 Evaluation

We evaluated our results by comparing them with those of the system proposed by Kasahara et al. [5]. This system vectorizes original explanatory notes in a dictionary using $TF \cdot IDF$ and measures word similarity in terms of the inner product of those vectors. A feature of this system is that the vectors produced by $TF \cdot IDF$ are manually tuned to accord with human feelings.

For comparison with our results, we used psychological data from questionnaires on word similarities [6]. In the psychological experiment, 77 subjects were asked if the stimulus words were synonyms of the headwords or not. Stimulus words that gained more than 50% agreement from the other subjects were regarded as *true* synonyms.

We analyzed the population of the order of each *true* synonym in the computed similarity. Associative words were also examined in the same manner. Figure 4 shows the difference between our method and the tuned system. It illustrates the ratio of accumulated *true* synonyms (A) and *true* associative words (B) plotted over the order of similarity. In both cases, the plotted values of the proposed method are almost always larger than those of the tuned system, which means that these *true* words appear earlier in the order of similarity in our results than in that of the tuned system. As for associative words, the effectiveness is more significant. However, these are not accurate comparisons because the tuned system contains words other than nouns, e.g., verbs and adjectives. Nevertheless, we can easily expect our method to have almost the same results as the simulation results, even with words other than nouns, because common or frequently used words such as verbs have a low similarity in our method as shown in Table 1.

4 Discussion

As mentioned at the beginning of this paper, we know that the $TF \cdot IDF$ method is useful and works well. However, it does not provide us with a well-grounded comprehension. By contrast, due to the stochastic process, it is quite clear what the proposed method computes, and why the procedure is necessary. This clarity is required when we are confused as to how to apply frequency vectors. For example, if we assume that an application contains the process $(A + A^T)$, is it possible to compute this reasonably? Here, A is a word-by-word square matrix such as that used in our simulation. $TF \cdot IDF$ gives us no idea whether it is possible to add A and A^T. However, in terms of the stochastic process, it is clear that adding $P(w_i|w_j)$ and $P(w_j|w_i)$ does not make sense. Of course, a matrix based on $TF \cdot IDF$ need not abide by the rules of a stochastic process. However, the meanings of the matrix elements are still the same. It is easy to understand this idea if we assume a document-by-word matrix instead of a word-by-word matrix.

In our simulation of word similarity, the probability $P(w_i|w_j)$ was given by formula (16). This process resembles a calculation of the inner product of

$TF \cdot IDF$ vectors when headwords are regarded as document indices. This is because, in this case, the denominator adds up word frequencies over all documents, and the numerator is the word frequency in a document. A widely used document similarity method computes the inner product of $TF \cdot IDF$ vectors, i.e., $\sum (TF)^2 \cdot (IDF)^2$. On the other hand, our method nearly computes $\sum (TF)^2 \cdot IDF$ as follows:

$$\begin{aligned} P(w_i|w_j) &= \sum_{all\ k} \frac{P(w_k^*|w_i)P(w_k^*|w_j)}{\sum_{all\ l} P(w_k^*|w_l)} \\ &= \sum_{all\ k} \frac{P(w_k^*|w_i)}{\sqrt{\sum_{all\ l} P(w_k^*|w_l)}} \frac{P(w_k^*|w_j)}{\sqrt{\sum_{all\ l} P(w_k^*|w_l)}} \\ &\simeq \frac{TF}{\sqrt{DF}} \frac{TF}{\sqrt{DF}}. \end{aligned}$$

As described above, our method clarifies how we can use the method for other applications. From this point of view, the proposed method is significantly different from $TF \cdot IDF$, although these two processes work similarly in some ways. As for the word similarity, we may be able to undertake some further work to evaluate its accuracy, but the simulation results clearly show the effectiveness of the probabilistic frequency vectors.

5 Conclusions

We proposed a probabilistic method for computing word frequency vectors based on dictionaries. This method is significant in its well-founded procedure. A stochastic process clearly shows how to employ this method for certain applications. As an example of such applications, we demonstrated the computation of word similarity. The results show the effectiveness of our approach.

The key feature of our method is the semantic expansion of dictionaries. However, the dictionaries themselves may influence this expansion. To avoid such an influence, we may need to use as many dictionaries as possible or investigate a way of applying corpora to our procedure.

References

1. Papineni, K.: Why Inverse Document Frequency? NAACL, Pittsburg, (2001)
2. Hofmann, T.: Probabilistic Latent Semantic Indexing. 22nd Intl. Conf. on Research and Development in Information Retrieval (SIGIR). (1999) 50-57
3. Kindaichi, H., Ikeda, Y.: Gakken Japanese Dictionary, 2nd Ed. Gakushu-kenkyusha. (1988)
4. Matsumoto, Y. et al.: Morphological Analysis System ChaSen version 2.2.1 Manual. http://chasen.aist-nara.ac.jp/. (2000)
5. Kasahara, K., Matsuzawa, K. Ishikawa, T.: A Method for Judgement of Semantic Similarity between Daily-used Words by Using Machine Readable Dictionaries. Information Processing Society of Japan. *38* (1997) 1272-1283

6. Kasahara, K., Inago, N., Kanasugi, Y., Nagamori, C., Kato, T.: Analysis of Word Relationship. 9th Workshop on Linguistic Engineering, Japanese Society for Artificial Intelligence (JSAI). (2001) 17-27