Distributed Clustering of Korean Verbs Using Self-Organizing Maps

Seong-Bae Park and **Byoung-Tak Zhang**
Artificial Intelligence Lab. (SCAI)
School of Computer Science and Engineering
Seoul National University
Seoul 151-742, Korea
E-mail : {sbpark,btzhang}@scai.snu.ac.kr

Abstract

In this paper we present a novel method to automatically cluster Korean verbs according to the distribution of object-verb relation. Since SOM (Self-Organizing Maps) shows great performance in analyzing and visualization the input patterns, SOM is adopted as a clustering method. Once the map is created, the verbs unseen during the training phrase can be clustered and the semantic distance among clusters can be calculated with ease using the map. The experiment, which compares the proposed method with the other clustering method based on the relative entropy of probability distribution, shows that the clusters created by SOM reflects those made by relative entropy completely.

Keywords : Korean verbs, word clustering, Self-Organizing Maps, relative entropy

1 Introduction

In statistical approach to natural language processing, it is natural to supplement the language model using information on word classes rather than on words. Since the word classes express the meaning of words naturally, they are helpful in generalizing the language model and solving data sparseness problem in statistical language modeling.

The general method to resolve data sparseness in the statistical language model is to approximate the probability of an unseen word to that of similar words. Hindle measured the similarity between words with *mutual information* [2], but did not suggest how to construct word classes. Yang presented *co-occurrence similarity* to identify Korean noun phrase coordination [8]. He proposed a method to construct word classes, but considered only the overlap of verbs which the nouns can take as its constituents. Thus, he ignored the practical distribution of nouns.

In this paper we propose a novel method to cluster Korean verbs according to the distribution of nouns which are used as an object of each verb. SOM (Self-Organizing Maps) is adopted for this purpose, since SOM shows great performance in analyzing and visualizing the input patterns. Once the map is created, the verbs unseen during the training phrase can be clustered and the semantic distance among clusters can be calculated with ease using the map. We verify the effectiveness of the proposed method by comparing with the model using relative entropy.

Given n : the number of clusters we want.
[step 1] Make a fully-connected weighted graph, where nodes are words and the weight of the edge is the augmented relative entropy between words connected by the edge.
[step 2] $< i, j >$:= two nodes with a minimum augmented relative entropy
[step 3] k := a new node resulting from merging node i and node j
[step 4] For all nodes l such that $l \neq i$ and $l \neq j$ and $l \neq k$,

$$weight(l,k) = \frac{weight(l,i) + weight(l,j)}{2}.$$

[step 5] Remove node i and j from the graph.
[step 6] **if** (# of nodes in graph $> n$) **then goto** [step 2]
else print words in each node.

Figure 1: The greedy algorithm to cluster words according to the augmented relative entropy.

2 Word-Clustering Model

2.1 Clustering by Relative Entropy

Though a number of measures to calculate similarity between words has been proposed [2, 4, 5, 6, 8], we use *relative entropy* or *Kullback-Leibler distance* as a distance among words. For given two probability density functions $p(x)$ and $q(x)$, the relative entropy between $p(x)$ and $q(x)$ is defined as

$$D(p||q) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)},$$

where $0 \log \frac{0}{p} = 0$ and $p \log \frac{p}{0} = \infty$.

Since $D(p||q) \geq 0$ and $D(p||q) = 0$ if and only if $p = q$, it is naturally considered to be a distance between two probability distributions. However, the relative entropy, in general, does not satisfy the communicative law and the triangular inequality. Thus, in this paper, we extended the relative entropy into the *augmented relative entropy* defined as

$$D'(p||q) = \frac{D(p||q) + D(q||p)}{2}.$$

Cluster Number	Verbs
1	완화하다 (*wanwha-hada* : relax) 철폐하다 (*cholpye-hada* : abolish)
2	해독하다 (*haedok-hada* : decode) 암호화하다 (*amhowha-hada* : encode)
3	팔다 (*palda* : sell) 판매하다 (*panmae-hada* : sale) 거듭하다 (*gudeup-hada* : do again) 되풀이하다 (*dwipooli-hada* : repeat)
4	극대화하다 (*geukdaewha-hada* : maximize) 향상시키다 (*hyangsang-sikida* : improve) 약화시키다 (*yakwha-sikida* :weaken) 회복시키다 (*hwoibok-sikida* : recover)
5	거론하다 (*guron-hada* : mention) 언급하다 (*ongeup-hada* : refer)
6	조치하다 (*jochi-hada* : dispose) 몰리치다 (*moolichida* : expel)
7	떠넘기다 (*ddunomgida* : impose) 회피하다 (*whoipi-hada* : avoid)
8	머금다 (*meogeumda* : keep in mouth) 삼키다 (*samkida* : swallow) 글썽이다 (*geulssungihda* : filled with tears)
9	끄덕이다 (*ggeudokihda* : nod) 숙이다 (*sookida* : bend) 갸우뚱하다 (*gya-woo-ddoong-hada* : tilt)

Table 1: The word classes constructed using relative entropy.

For a verb set $V = \{v_1, v_2, \ldots, v_n\}$ and a noun set $N = \{n_1, n_2, \ldots, n_k\}$, the probability of a verb v_i is given as

$$p(v_i) =< p(n_1, v_i), p(n_2, v_i), \ldots, p(n_k, v_i) >$$

and

$$P(n_i|v) = \frac{C(n_i, v)}{\sum_{n_j \in N} C(n_j, v)},$$

where $C(n_i, v)$ is the frequency that n_i and v co-occurred in the corpus.

Figure 1 describes how to cluster words into word classes using augmented relative entropy. After making a fully-connected graph where the nodes represent the words to be clustered and the weight of edges is the augmented relative entropy between two words, we construct word classes by merging nodes. Because two words connected by the edge whose weight is lowest are the most similar words, a new node merged from these two nodes is created in step 3 and the weight of each edge between this new node and existing nodes is updated in step 4. After that, the two nodes who are used in making the new node are removed from the graph in step 5. Since each iteration from step 1 to step 5 removes two nodes and creates one node, the number of nodes is reduced by one in every iteration. Thus, we can create word classes by iterating this process until the number of nodes reaches the desired number. Table 1 shows nine word classes of 24 Korean verbs.

2.2 Clustering by SOM

Since the method to cluster words using relative entropy must calculate the relative entropy for all possible word pairs, it gets more and more complex as the number of words gets large. To solve this limit, we use SOM (Self-Organizing Maps) [3] in this paper. The SOM can be seen as a plane neural array whose cells are regulated variously according to input patterns.

In SOM, every training example is represented as a real vector $\mathbf{x}(t) \in R^n$, where t is the variable for training time. At first, the weight vector of each cell i, which is initialized at random, is trained by the following rule.

$$\mathbf{w}_i(t+1) = \begin{cases} \mathbf{w}_i(t) + \alpha(t)(\mathbf{x}(t) - \mathbf{w}_i(t)) & \text{if } i \in N_c(t) \\ \mathbf{w}_i(t) & \text{otherwise,} \end{cases}$$

where $\alpha(t) \in (0, 1]$ is the learning ratio and N_c represents the neighbor of the winner c in the map. The winner c is determined by the similarity between the input vector $\mathbf{x}$ and the weight vector $\mathbf{w}$. That is, for the given input vector x, the winner c is determined by the following equation,

$$||\mathbf{x} - \mathbf{w}_c|| = \min_i \{||\mathbf{x} - \mathbf{w}_i||\}.$$

As for the neighborhood function used to update the neighboring cells of the winner, a number of functions has been proposed, including Mexican-hat function, bubble function, Gaussian function, etc. Since Gaussian function is easy to implement and stable [3], it is adopted as neighborhood function. Figure 2 shows the clustering result constructed using SOM for the verbs

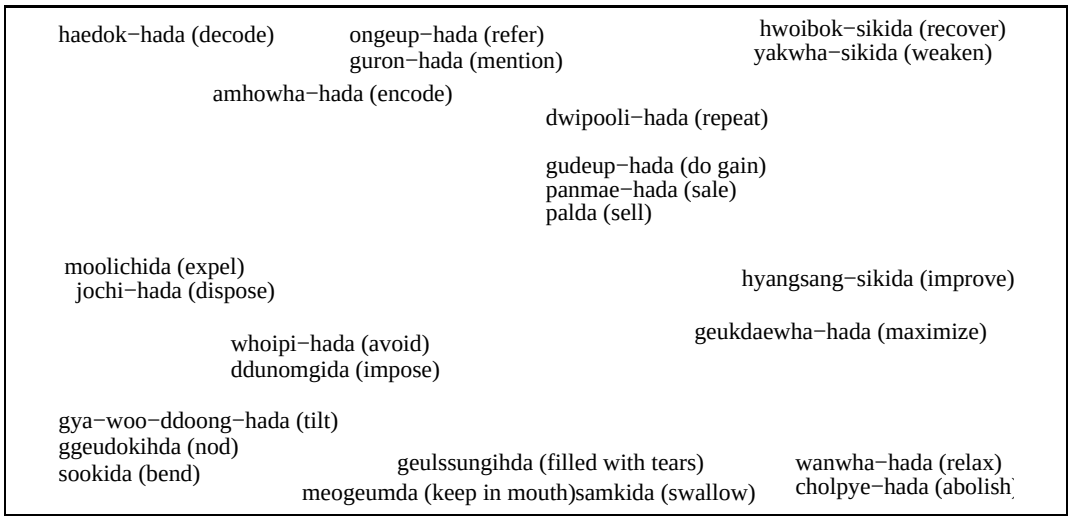


Figure 2: Clustering result trained by SOM. This SOM has 10 × 10 nodes.

in Table 1. It can be verified that the figure reflects Table 1 relatively substantially.

3 Information for Verb Clustering

The n-gram statistical model is one of the most widely used statistical language models and not so complex. This model assumes that only the precedent $n-1$ words can make influence on the probability of a word. Thus, it has some problems. First, the n-gram model does not give any information about the area beyond the boundary. The more severe problem is that a number of data collected based on this model do not reflect well the real context, since it is defined by word sequence not by grammatical role. For example, two sentences "I met him *at*" and "I met him yesterday *at*" must have similar information around the preposition 'at', but trigram model generates two distinct information from the sentences.

This paper clusters Korean verbs using object-verb relation, instead of n-gram model. Because Korean is partially free-order language, the model using object-verb relation is more efficient than n-gram model. Though we need a Korean parser to grasp the object-verb relation, there is no reliable and stable Korean parser. The postpositions and endings represents the grammatical relation in Korean. Especially, the object relation is represented by postposition '*eul*' or '*reul*', and the word containing this postposition has tendency to be an object of the nearest verb after the word. Thus, we can extract the object-verb relation using this information without a Korean parser.

4 Experiments

We extracts verb-object pairs from a corpus using a Korean morphological analyzer and a part-of-speech tagger. The Korean morphological analyzer used for the experiment is developed in KORDIC (Korean R & D Information Center) [7]. The POS tagger developed in Seoul National University is based on the Hidden Markov Model, and shows over than 98% of accuracy even though we select the best candidate.

From the corpus composed of newspaper articles and 0.5 million words, 41,285 object-verb pairs are extracted. Since pairs whose frequency is low commit disorder in the probability of them, those which occurred less than 15 times are excluded. After excluding them, 26,047 pairs remain and have 1,531 nouns and 987 verbs.

Using those pairs as training examples, we cluster verbs using SOM. In order to reduce the complexity of the problem, we first cluster 1,531 nouns into 200 classes by using the algorithm in Figure 1. Thus, each input vector $\mathbf{x}(t) \in R^{200}$ consists of 200 conditional probabilities $p(n_i|v)$ given noun classes. SOM is trained with 987 verb vectors.

The word-clustering method based on Shanon's information theory is one of the most widely used clustering methods. This model regards two verbs, v_i and v_j, as similar verbs if their empirical distribution is similar. The extent of the similarity can be gauged using the extended relative entropy $D'(v_i||v_j)$. From the viewpoint of probability, the clusters made by the relative entropy are optimal.

We assume that the model using the relative entropy always gives the correct result. Under this assumption, the proposed model is evaluated by being compared with the model using the relative entropy. For all possible verb pairs, two models answers whether or not two verbs belong to the same cluster. They answer '*yes*' if they belong, '*no*' otherwise.

To evaluate the proposed method, we use the contingency table method which is widely used in information retrieval and psychology. In this method, recall and precision is defined as follows:

$$recall = \frac{a}{a+c} \cdot 100\%$$

	Answer should be *yes*	Answer should be *no*
The model says *yes*	a	b
The model says *no*	c	d

Table 2: The contingency table to evaluate the model.

$$precision = \frac{a}{a+b} \cdot 100\%,$$

where a, b and c is defined in Table 2. The $F_\beta - score$ [1] which combines precision and recall is defined as

$$F_\beta = \frac{(\beta^2+1) \cdot recall \cdot precision}{\beta^2 \cdot recall + precision},$$

where β is the weight of recall relative to precision. We use $\beta = 1.0$, which corresponds to equal weighting of the two measures.

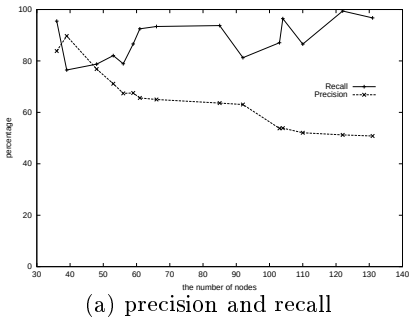
The experiment is performed on various number of SOM cells. Recall and precision is measured for all possible 973,182 ($= 987 \times 986$) verb pairs, and Figure 3 shows the result. According to Figure 3, precision monotonically decreases as the number of SOM cells increases, while recall is preserved uniformly. Thus, F-measure also decreases monotonically. This is because the number of SOM cells is not the correct number of word classes needed. From Figure 3, 100 is considered to be the proper number of SOM cells.

5 Conclusions

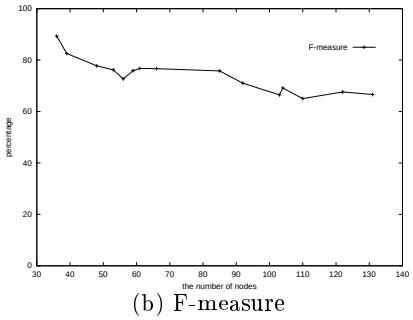
We presented a method to cluster Korean verbs using SOM. Since the clusters created by the proposed method reflects the clusters constructed based on information theory, the effectiveness of the proposed method is proved.

Acknowledgements

This result was supported in part by the Korean Ministry of Education under the BK21-IT Program and by the Korean Ministry of Information and Communication through IITA under grant 00-023.



(a) precision and recall



(b) F-measure

Figure 3: The experimental result of SOM clusters. The graphs show the comparison of SOM clusters with the clusters made by relative entropy on various number of SOM cells. (a) shows recall and precision, and (b) shows F-measure.

References

[1] V. Hatzivassiloglou and N. McKeown, "Towards the Automatic Identification of Adjectival Scales: Clustering Adjectives According to Meaning," In *Proceedings of the 31st Annual Meeting of the ACL*, pp. 172–182, 1993.

[2] D. Hindle, "Noun Classification from Predicate-Argument Structures" In *Proceedings of the 28th Annaul Meeting of the ACL*, pp. 268–275, 1990.

[3] T. Honkela, "Comparisons of Self-Organized Word Category Maps," In *Proceedings of Workshop on Self-Organizing Maps 97*, pp. 298–303, 1997.

[4] J. Hughes and E. Atwell, "The Automated Evlauation of Inferred Word Classifications," In *Proceedings of the 11th European Conference on Artificial Intelligence*, pp. 535–539, 1994.

[5] J. Gao and X. Chen, "Probabilistic Word Classification Based on a Context-Sensitive Binary Tree Method," *Computer Speech and Language*, Vol. 11, No. 4, pp. 307–320, 1997.

[6] F. Pereira, N. Tishby and L. Lee, "Distributional Clustering of English Words," In *Proceedings of the 31st Annual Meeting of the ACL*, pp. 183–190, 1993.

[7] J.-H. Shin, Y.-S. Han and K.-S. Choi, "A HMM Part-of-Speech Tagger for Korean with Word-phrasal Relations," *Current Issues in Linguistic Theory: Recent Advances in NLP*, John Benjamins Publishing Company, pp. 439–449, 1997.

[8] J.-H. Yang, "Conjunct Identification in Korean Noun Phrase Coordination Using Cooccurence Similarity," *Computer Processing of Oriental Language*, Vol. 10, No. 4, pp. 391–408, 1997.