Automatic Completion of Korean Words for Open Vocabulary Pen Interface

Sungho Ryu and Jin-Hyung Kim

AI Lab., CS Div., KAIST, Kusung-Dong, Yusung-Gu, Taejon, Korea,
shryu@ai.kaist.ac.kr, jkim@ai.kaist.ac.kr

Abstract. An automatic completion method of general Korean words is proposed. A word model that describes the frequency of usage is used for generation of word candidates from a given prefix. In experiments, several different models for Korean words were tested for prediction performance. The results show that the best model can reduce the number of writing characters about 38% with 5 candidates.

1 Introduction

The automatic completion of a word is predicting the entire word from a prefix. If the accuracy of prediction is high, the overall input efficiency can be greatly improved. However, because the natural language is not a mathematically well-defined entity, the prediction of word is very complicated.

The prediction of word is highly dependent on the modeling of words itself. From a practical point of view, the most important issues in modeling natural language are the size of lexicons and the reliability of estimated likelihood of each lexical entry. The desirable word model should be able to generate reliable estimate of the likelihood for any word, even if it was not observed in the training. In this paper, the automatic completion methods for Korean are tested using various word models.

2 Prediction of Korean Words

A Korean word is usually composed of several morphemes. However, the high irregularity in inflections and derivations makes it difficult to perform morpheme-based prediction. Therefore, the word is chosen as a basic unit of prediction instead of the morpheme in this paper.

The basic rule of prediction is choosing the most probable string based on a given clue. In a pen-based interface, the clue is a partially complete prefix of the word.

The probability of a word can be modeled using the frequency of usage of words. The probability of a prefix is the sum of all probabilities of words that can be derived from the prefix (1).

D. Lopresti, J. Hu, and R. Kashi (Eds.): DAS 2002, LNCS 2423, pp. 111–114, 2002.
© Springer-Verlag Berlin Heidelberg 2002

$$p_S(s) = \sum_{\forall w \text{ that has } s \text{ as a prefix}} p(w) \tag{1}$$

Then, the prediction can be regarded as selecting a longer prefix based on the likelihood.

$$s' = \arg\max_{s'} \frac{p_S(s')}{p_S(s)} \tag{2}$$

where s' is longer string that has s as its prefix.

Therefore, the prediction is primarily dependent on the choice of the word probability model, *p*(W).

In dictionary-based model, the probability of each word is explicitly specified using the trie. The probabilities of words are directly calculated from the occurrences of each word in a large raw text corpus. In order to estimate probability of unseen words, the probabilities are smoothed using Lidstone's method.

In character n-gram based models, a word is represented as a sequence of characters. The probability of the word is the joint probability of the character sequence. To make the computation tractable, the joint probability is usually approximated using n-th order Makov assumption. For example, in trigram-based model (character n-gram with length 3), the probability of a word can be calculated as follows.

$$p(w) = p(c_1)\,p(c_2 \mid c_1)\,p(c_3 \mid c_1, c_2) \cdots p(c_n \mid c_{n-2}, c_{n-1}) \tag{3}$$

In n-gram based models, the equation for prediction becomes much simpler. By ruling out common terms, eq. (2) can be reduced to a conditional probability mass function (4).

$$s' = c_1, \ldots c_l, c_{next}$$
$$\text{where } c_1, c_2, \ldots c_l, \text{ is the given prefix,} \tag{4}$$
$$\text{and } c_{next} = \arg\max_{C} p(C \mid c_{l-n+1} \cdots c_{l-1})$$

Probabilities of character n-grams are calculated from raw text corpus using Katz's back-off method.

3 Prediction Interface

For fair comparison between different models, a pen-based interface with word prediction is used. In this interface, the basic unit of recognition is a character. When each character is recognized, the word model generates candidates from the current prefix. Users can either select a word from candidates to complete the whole word, or write subsequent characters by themselves if it does not exist.

The proposed interface has three primitive operations:

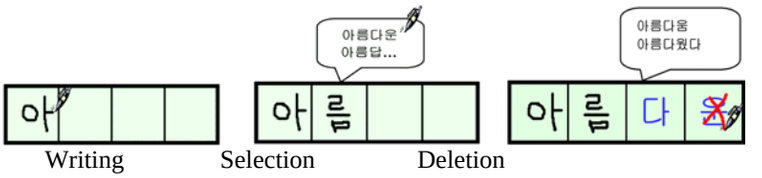

Fig. 1. The primitive operations

- Writing
  Writing an individual character.
- Selection
  Selecting one of the candidates generated by the prediction interface.
- Deletion
  Deleting characters after a certain position. Used to trigger a new prediction at a specific position within the string.

The total cost of input is defined as a weighted sum of the ratio of primitive operations.

$$Cost\ of\ input = \frac{\alpha_{writing} n(Writing) + \alpha_{selection} n(Selection) + \alpha_{Deletion} n(Deletion)}{n(Total\ Chars)} \quad (5)$$

The weights represent the easiness and robustness of each operation. In general, it can be assumed that the weight of writing is much greater than the weights of selection or deletion.

4 Experiments

For training word models, '97 KAIST raw text corpus was used. It has about 1,200,000 distinct words and 15,500,000 words in total. The prediction performance of each model was tested using '96 KAIST raw text corpus. It has about 1,300,000 distinct words and 15,800,000 words in total. The number of total characters used for testing was about 45,600,000.

The table below shows the number of primitive operations for each model. In all models, 5 candidates were generated for the selection.

Primitive Operations	Models			
	Word-trie	bigram	trigram	4gram
n(Writing)/ n(Total char)	0.66	0.72	0.62	0.62
n(Selection)/ n(Total char)	0.32	0.27	0.34	0.34
n(Deletion)/ n(Total char)	0.02	0.005	0.01	0.01

Assuming $\alpha_{writing}$=1.0, $\alpha_{selection}$=0.1, and $\alpha_{deletion}$=0.5, the cost of input for each model can be calculated as below:

	Models			
	Word -trie	bigram	trigram	4gram
Cost of input	0.71	0.75	0.66	0.66

The word trie model performed better than the bigram model. This indicates that a prefix with sufficient length is required in order to get an accurate prediction result. However, if a sufficiently long prefix is given, the n-gram-based models have better performance than the dictionary-based one.

In the trigram model and the 4gram model, about 38% of writing operation was omitted or replaced. This indicates about 1/3 of reduction of input cost. However, the 4gram model requires almost twice size than the trigram model while presenting almost identical performance. Therefore, the trigram model is more desirable in terms of the performance and the size requirement.

5 Summary

In this paper, the prediction methods for general Korean word were tested using various word models. The prediction method is basically selecting the most likely word that can be generated from a given prefix. The candidates of words are generated by the word model, which describes the likelihood of words based on the frequency of usage.

The tested word models are a word-dictionary model and several n-gram based models. Experiments show that the trigram model performs best. In trigram model, about 38% of actual writing could be replaced by other simpler operations. Using the assumed weights for each operation, this is a rough equivalent of reducing total cost of input by 1/3.

References

1. C. D. Manning, Foundations of statistical natural language processing, MIT Press, 1999
2. D. Jurafsky and J. H. Martin,. Speech and language recognition, Prentice-Hall, 2000
3. I. Bazzi et.al., An omnifont open-vocabulary OCR system for English and Arabic, IEEE Trans. on PAMI., vol.21, no.6, pp.495-504, 1999
4. J.K.Park and J.Kim, A study on error correction in Hangul text recognition system, Master Thesis, KAIST, 1989