An Automatic Tagging System and Environments for Construction of Korean Text Database

Woon-Jae Lee*, Key-Sun Choi*, Yun-Ja Lim**, Yong-Ju Lee**, Oh-Woog Kwon*, Hiong-Geun Kim*, Young-Chan Park*

Department of Computer Science, KAIST*
Automatic Interpretation Section, ETRI**

ABSTRACT A set of text database is indispensable to the probabilistic models for speech recognition, linguistic model, and machine translation. We introduce an environment to construct text databases: an automatic tagging system and a set of tools for lexical knowledge acquisition, which provides the facilities of automatic part of speech recognition and guessing.

1. Introduction

Ambiguity is one of the critical problems in natural language processing (NLP). Although the concept itself is not ambiguous, the corresponding surface representation can be ambiguous. For example, a surface form '나는[nanŭn]' can have several alternative morphological interpretations, such as '나[na]'{I}(pronoun) + '는[nŭn]'(postposition)', '나다[nada]'{be born}(verb) + '는[nŭn]'(ending of a word), and '날다[nalda]'{fly} + '는[nŭn]' (ending of a word). We can find syntactic ambiguity in such a sentence like "I saw a man on the hill with a telescope."

One of the ambiguity resolution methods suggested, is the probabilistic model, which is quite simple and well applicable to all the levels of language processing.[3][1]

Probabilistic information of a language is acquired from a large set of texts. The precision of the information depends on the type and the quantity of the texts. In this paper, we elaborate how to construct and how to use the large set of texts.

2. Text Database

The large amount of texts for acquisition of probabilistic information is so called 'corpora'. Issues on corpora are to determine how to construct them and which information to acquire from them.

First we have to choose which texts to be in the corpora. It depends on the processing domain, because the probabilistic information may vary on the processings, or whether we are to recognize speech or to make machine translation system. In order for general use, we have to collect the texts from various domains and the corpora gathered from every domains unbiased is called "balanced". The texts are classified by the domain and we can construct "balanced corpora" from the classified texts.

After balancing the texts, we have another problem: the quantity of texts. The precision of the probablistic information grows higher as we gather more and more texts, but in the view point of time-effectiveness over the cost, about a million to 10 million word phrases are thought to be acceptable. [1]

3. Korean Concordance Programming(KOCP)

KOCP is a tool for text analysis, which shows the concordant context of a word in a text. KOCP can be used to construct a dictionary, to extract index terms for information retrieval, to developing grammars, and as a tool for all the other linguistic researches as well as in language education.

KOCP consists of a dictionary, a morphological analyzer, and a concordance generator.

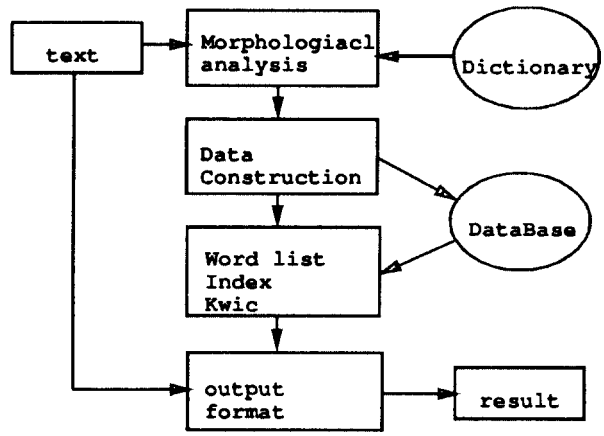

Figure 1 : System Diagram of KOCP

<Morphological analysis> transfers the result of analysis to the <Data Construction> step. In the morphological analysis, it does the recovery of the root, a simple disambiguation and unknow word handling, in addition to the morphological analysis. The result is managed in the unit of word phrase and the distinction between failures, ambiguities, and success of the analysis eases the next step.

In the <Data construction> step, if a word is successfully analyzed then the word and its position with other informations are stored in a database. Each word is orderly stored and a set of easy access methods are provided.

<Output format> generates a visible format of each concordance of words and calculates statistic data and their index. If the morphological analysis fails for a word, then failure result is used to extend the morphology dictionary.

KOCP alone can be used to acquire linguistic information such as morphological, syntactic, and semantic information. When we integrate the KOCP system with a tagging system,

[1] Over 50,000 nouns can be extracted from corpora of 1 million word phrases, and 50,000 is quite acceptable number of nouns for a typical NLP. [ETRI,KAIST,Ulsan Univ.]

a syntactic analyzer or a system of mutual information, we may improve the accuracy and efficiency.

4. Automatic Tagging(ATAG)

Automatic tagging consists of two parts: the part to give initial tag using morphological analysis, and the part to eliminate ambiguity using probabilistic information of the part of speech and the context of word. The accuracy of automatic tagging is determined by tagging algorithm and probabilistic information. Probabilistic information is gathered from the corpora ready-tagged.

□ Basic Assumption of HMM[2][4]

HMM Tagging is fundamentally based on the locality. The locality is sufficient to determine the probabilities. This assumption says that the tag for the current word is only dependent on a few tags previously came and the current word.

- $p(t_{n+1}|t_1 t_2 ... t_n) = p(t_{n+1}|t_{n-k} t_{n-k+1} ... t_n)$
  A few tags in adjacent to the current word affects the determination of the current tag.

- $p(t_k|w_1/t_1 w_2/t_2 .. w_k .. w_n/t_n) = p(t_k|w_k t_{k-1} t_{k-2})$
  If we use the tri-gram, then the current tag is determined by the current word and the two previous tags only.

Note that the words and the categories are separated, hence forth, the dependency between words is not allowed. But the dependencies between a word and its tag, and between tags are allowed.

□ Construction of HMM/3[2]

First we collect a set of manually tagged texts and we extract the tri-gram from these texts for automatic tagging, and then calculate the probabilities. The state transition graph produced by this process is made up of a set of states and a set of arcs that links between the states. A state in a state transition graph reflects the bi-gram and each arc a tri-gram, and each arc has the frequency of the tri-gram. An HMM/3 is produced through following tag tri-grams. Figure 2 depicts the HMM/3 produced by these tag tri-grams.

Pj Nj Va 8
Pj Nj Nj 4
Pj Nj Nia 10
...

In figure 2, each symbol represents the tag and the arc the transition of states. Probability of state transition is attached to each arc.

[2]We implemented a Hidden Markov Model using the tri-gram.

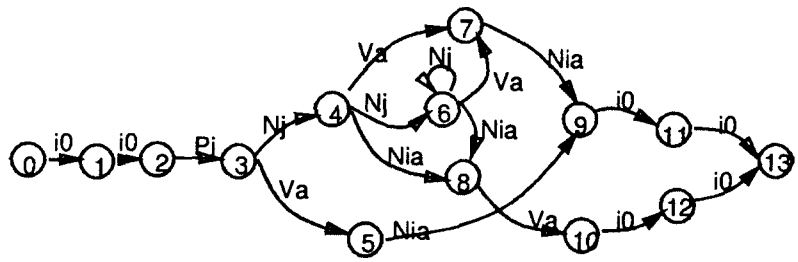

Figure 2: Hidden Markov Model in a tagging system : HMM/3

4.1 Tagging Model of the Korean Language

In a model for Korean tagging, the problem is to choose a T that maximizes the conditional probability for a certain sentence;

$$p(D,T|W) = \frac{p(W|D,T)p(D,T)}{p(W)}$$

Variable W is for a sentence, D for the word in the dictionary for each word phrase, and T for the tag of each word phrase.

According to the basic assumptions of HMM, we can reduce the above expression into the following;

$$\cong \prod \frac{p(w_i|d_it_i)p(d_i|t_i)p(t_i|t_{i-1}t_{i-2})}{p(w_i)}$$

In this expression, $p(w_i|d_it_i) = 1$, and $p(w_i)$ does not affect the maximum value of the probability, hence this formula can be reduced into following;

$$= \prod p(d_i|t_i)p(t_i|t_{i-1}t_{i-2})$$

$$= \prod \prod_{k=1}^{N_i} [p(d_{ik}|t_{ik})]p(t_i|t_{i-1}t_{i-2})$$

$$= \prod \prod_{k=1}^{N_i} \frac{p(d_{ik}t_{ik})}{p(t_{ik})} p(t_i|t_{i-1}t_{i-2})$$

$$d_i = d_{i1}d_{i2}...d_{iN_i}$$

$$t_i = t_{i1}t_{i2}...t_{iN_i}$$

w_i, d_i, t_i represent the i'th word phrase, i'th word in the dictionary, and i'th tag respectively. The number of words in i'th word phrase, or N_i, can be multiple, so the formula is normalized as follows;

$$= \prod \sqrt[N_i]{\prod_{k=1}^{N_i} \frac{p(d_{ik}t_{ik})}{p(t_{ik})}} p(t_i|t_{i-1}t_{i-2})$$

$\prod_{k=1}^{N_i} \frac{p(d_{ik}t_{ik})}{p(t_{ik})}$ of this formula is calculated in the morphological analyzer and $p(t_i|t_{i-1}t_{i-2})$ is the probability stored in HMM/3.

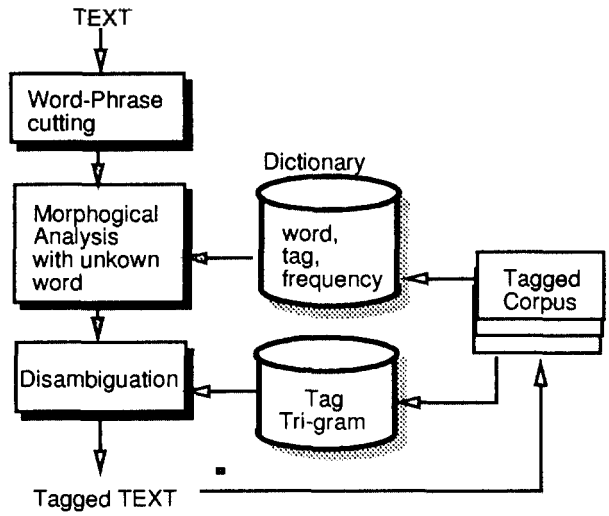

Figure 3 : Block diagram for the automatic tagging system

4.2 Construction of the Tagging System

The automatic tagging process uses the probability of each word and the context probability stored in HMM/3. Figure 3 is the diagram for the system.

An automatic tagger consists of a morphological analyzer, an unknown word processor, and the module for disambiguation. A word phrase is divided by the morphological analyzer and the unknown word processor. The categories have probabilities respectively for each cases. In the disambiguation module, the ambiguity occurred in a word phrase is resolved using the probabilities of the categories stored in the words near by.

4.3 Experiments and Results

The unit of disambiguation in this system (ATAG) is a word phrase. The tagging is done for each word phrase. A tag for a word phrase is represented as the compound of tags for words. The number of tags for word phrases is about 400, and the number was changed during the tagging experiments.

	All Word Phrase	Error of analysis	Hit Ratio
	T	E	(T - E)/T
The Charater of National Education	144	14	0.903
Information Retrieval	1803	103	0.905
Total	1227	117	0.905

Table 1 : Experiment Result

We have manually tagged the Korean Language Text book of Elementary School, which contains about 26,000 word phrases and the result was used to extract the tri-grams. Extracted tri-grams were amount to 2,320 and the bi-grams, 653. The number of dictionary

entries is 11,637 and we have handled unknown words for nouns, predicates, Roman words, numerics. The texts used in the experiments were "The Charter of National Education" and a part of "Information Retrieval".

You may find a tagging system based on the tri-gram can be quite safely used to Korean text tagging, even if we use the basic 14 tag set. Most of the errors were related to unknown words.

Previous systems were very precise up to about 96 $\sim$ 97%, while this system only shows a low precision, 90%. The result of previous systems, however, were achieved through a set of 90 tags, a sufficient dictionary and the context probability gathered from million word phrases. In this system, we only used 14 basic tags, a dictionary of 10 thousand entries and a small amount of texts of 26 thousand word phrases. So the result of this system is not so depressing and if we expand the tag set, the dictionary, and if we make a sufficiently enough training, then this system can have a high degree of precision.

5. Conclusion

We gathered a set of corpora of 3 million word phrases as a text database which can be used for speech recognition and machine translation, in ETRI/KAIST. We implemented a tagging system and a concordance indexer over the corpora. We've done it somewhat incrementally: first, we constructed a tagged corpora through manual- and automatic-tagging, and the resultant tagged corpora and the result of concordance indexing were used for extending the dictionary, which was used again for tagging and indexing. The tagged corpora can be used as a basis of those probabilistic, stochastic approaches to various NLPs, such as speech recognition and machine translation. In order to reduce the ambiguities, the tagging system itself can be used as a preprocessor of a speech recognition system or a machine translation system.

References

1. KAIST, *"A Ressearch on The Korean Text DB Construction for Language Modeling"*, ETRI, 1992, 1993.

2. Kenneth W. Church, *"A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text"*, Proceedings on 2nd Conf. Applied NLP, ACL, 1988, pp. 136-143

3. Steven J. DeRose, *"Grammatical Category Disambiguation by Statistical Optimization"*, Computational Linguistics, Vol. 14, No. 1, Winter 1988, pp. 31-39

4. David M. Magerman, *"Everything You Almost Wanted to Know About Probability Theory, But Were Afraid to Ask"*, U. Penn Tech. Report, 1991