Term Recognition Using Technical Dictionary Hierarchy

Jong-Hoon Oh, KyungSoon Lee, and Key-Sun Choi
Computer Science Dept., Advanced Information TechnologyResearch Center (AITrc), and
Korea Terminology Research Center for Language and Knowledge Engineering (KORTERM)
Korea Advanced Institute of Science & Technology (KAIST)
Kusong-Dong, Yusong-Gu Taejon, 305-701 Republic of Korea
{rovellia,kslee,kschoi}@world.kaist.ac.kr

Abstract

In recent years, statistical approaches on ATR (Automatic Term Recognition) have achieved good results. However, there are scopes to improve the performance in extracting terms still further. For example, domain dictionaries can improve the performance in ATR. This paper focuses on a method for extracting terms using a dictionary hierarchy. Our method produces relatively good results for this task.

Introduction

In recent years, statistical approaches on ATR (Automatic Term Recognition) (Bourigault, 1992; Dagan et al, 1994; Justeson and Katz, 1995; Frantzi, 1999) have achieved good results. However, there are scopes to improve the performance in extracting terms still further. For example, the additional technical dictionaries can be used for improving the accuracy in extracting terms. Although, the hardship on constructing an electronic dictionary was major obstacles for using an electronic technical dictionary in term recognition, the increasing development of tools for building electronic lexical resources makes a new chance to use them in the field of terminology. From these endeavour, a number of electronic technical dictionaries (domain dictionaries) have been acquired.

Since newly produced terms are usually made out of existing terms, dictionaries can be used as a source of them. For example, 'distributed database' is composed of 'distributed' and 'database' that are terms in a computer science domain. Further, concepts and terms of a domain are frequently imported from related domains. For example, the term 'Geographical Information System (GIS)' is used not only in a computer science domain, but also in an electronic domain. To use these properties, it is necessary to build relationships between domains. The hierarchical clustering method used in the information retrieval offers a good means for this purpose. A dictionary hierarchy can be constructed by the hierarchical clustering method. The hierarchy helps to estimate the relationships between domains. Moreover the estimated relationships between domains can be used for weighting terms in the corpus. For example, a domain of electronics may have a deep relationship to that of computer science. As a result, terms in the dictionary of electronics domain have a higher probability to be terms of computer science domain than terms in the dictionary of others do (Felber, 1984).

The recent works on ATR identify the candidate terms using shallow syntactic information and score the terms using statistical measure such as frequency. The candidate terms are ranked by the score and are truncated by the thresholds. However, the statistical method solely may not give accurate performance in case of small sized corpora or very specialized domains, where the terms may not appear repeatedly in the corpora.

In our approach, a dictionary hierarchy is used to avoid these limitations. In the next section, we describe the overall method description. In section 2, section 3, and section 4, we describe primary methods and its details. In section 5, we describe experiments and results

1 Method Description

The description of the proposed method is shown in figure 1. There are three main steps in our method. In the first stage, candidate terms that are complex nominal are extracted by a linguistic filter and a dictionary hierarchy is constructed. In the second stage, candidate terms are scored by each weighting scheme. In dictionary weighing scheme, candidate terms are scored based on the kind of domain dictionary where terms appear. In statistical weighting scheme, terms are scored by their frequency in the given corpus. In transliterated word weighting scheme, terms are scored by the number of transliterated foreign words in the terms. In the third stage, each weight is normalized and combined to Term weight (W_{term}), and terms are extracted by Term weight.

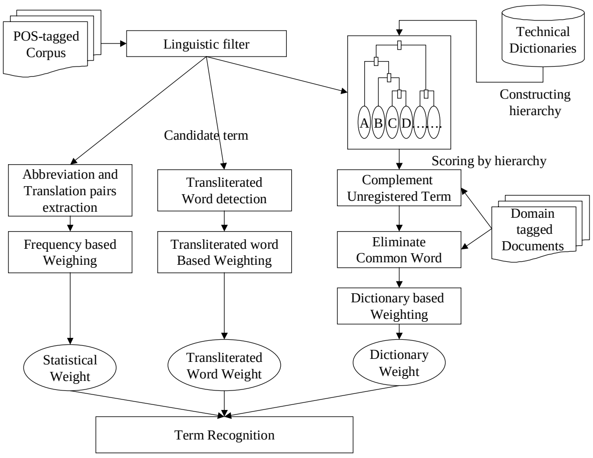

Figure 1. The method description

2 Dictionary Hierarchy

2.1 Resource

Field
Agrochemical, Aerology, Physics, Biology, Mathematics, Nutrition, Casting, Welding, Dentistry, Medical, Electronical engineering, Computer science, Electronics, Chemical engineering, Chemistry.... and so on.

Table 1. The fragment of a list: dictionaries of domains used for constructing the hierarchy.

A dictionary hierarchy is constructed using bi-lingual dictionaries (English to Korean) of the fifty-seven domains. Table 1 lists the domains that are used for constructing the dictionary hierarchy. The dictionaries belong to domains of science and technology. Moreover, terms that do not appear in any dictionary (henceforth we call them unregistered terms) are complemented by a domain tagged corpus. We use a corpus, called ETRI-KEMONG test collection, with the documents of seventy-six domains to complement unregistered terms and to eliminate common term.

2.2 Constructing Dictionary Hierarchy

The clustering method is used for constructing a dictionary hierarchy. The clustering is a statistical technique to generate a category structure using the similarity between documents (Anderberg, 1973). Among the clustering methods, a reciprocal nearest neighbor (RNN) algorithm (Murtaugh, 1983) based on a hierarchical clustering model is used, since it joins the cluster minimizing the increase in the total within-group error sum of squares at each stage and tends to make a symmetric hierarchy (Lorr, 1983). The algorithm to form a cluster can be described as follows:

1. Determine all inter-object (or inter-dictionary) dissimilarity.
2. Form cluster from two closest objects (dictionaries) or clusters.
3. Recalculate dissimilarities between new cluster created in the step2 and other object (dictionary) or cluster already made. (all other inter-point dissimilarities are unchanged).
4. Return to Step2, until all objects (including cluster) are in the one cluster.

In the algorithm, all objects are treated as a vector such as $D_i = (x_{i1}, x_{i2}, \ldots , x_{iL})$. In the step 1, inter-object dissimilarity is calculated based on the Euclidian distance. In the step2, the closest object is determined by a RNN. For given object i and object j, we can define that there is a RNN relationship between i and j when the closest object of i is object j and the closest object of j is object i. This is the reason why the algorithm is called a RNN algorithm. A dictionary hierarchy is constructed by the algorithm, as shown in figure 2. There are ten domains in the hierarchy – this is a fragment of whole hierarchy.

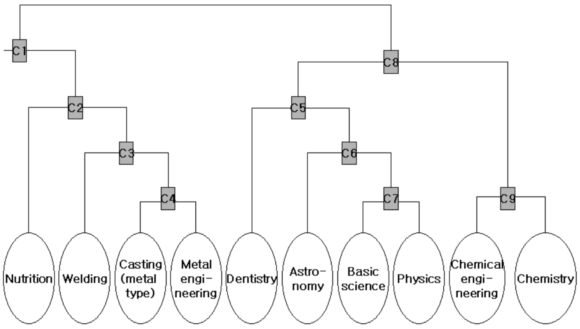

Figure 2. The fragment of whole dictionary hierarchy : The hierarchy shows that domains clustered in the terminal node such as chemical engineering and chemistry are highly related.

2.3 Scoring Terms Using Dictionary Hierarchy

The main idea for scoring terms using the hierarchy is based on the premise that terms in the dictionaries of the target domain and terms in the dictionary of the domain related to the target domain act as a positive indicator for recognizing terms. Terms in the dictionaries of the domains that are not related to the target domain act as a negative indicator for recognizing terms. We apply the premise for scoring terms using the hierarchy. There are three steps to calculate the score.

1. Calculating the similarity between the domains using the formula (2.1) (Maynard and Ananiadou, 1998)

$$similarity_{ij} = \frac{2 \times Common_{ij}}{depth_i + depth_j} \qquad (2.1)$$

where

Depth$_i$: the depth of the domain$_i$ node in the hierarchy

Common$_{ij}$: the depth of the deepest node sharing between the domain$_i$ and the domain$_j$ in the path from the root.

In the formula (2.1), the depth of the node is defined as a distance from the root – the depth of a root is 1. For example, let the parent node of C1 and C8 be the root of hierarchy in figure 2. The similarity between "Chemistry" and "Chemical engineering" is calculated as shown below in table 2:

Domain	Chemistry	Chemical Engineering
Path from the root	Root->C8-> C9->**Chemistry**	Root->C8->C9-> **Chemical Engineering**
Depth$_i$	4	4
Common $_{ij}$	3	3
Similarity $_{ij}$	2*3/(4+4) =0.75	2*3/(4+4) =0.75

Table 2. Similarity$_{ij}$ calculation: The table shows an example in caculating similarity using formula (2.1). In the example, Chemical engineering domain and Chemistry domain are used. Path, Depth, and Common are calculated according to figure 1. Then similarity between domains are determined to 0.75.

2.Term scoring by distance between a target domain and domains where terms appear:

$$Score(term) = \frac{1}{N}\sum_{i=1}^{N} similarity_{ti} \qquad (2.2)$$

where

N: the number of dictionaries where a term appear

Similarity$_{ti}$: the similarity between the target domain and the domain dictionary where a term appears

For example, in figure 2, let the target domain be physics and a term 'radioactive' appear in physics, chemistry and astronomy domain dictionaries. Then similarity between physics and the domains where the term 'radioactive' appears can be estimated by formula (2.1) as shown below. Finally, Score(radioactive) is calculated by formula (2.2) – score is (0.4+1+0.7)/3.:

N	3
similarity $_{\text{physics-chemistry}}$	0.4
similarity $_{\text{physics-physics}}$	1
similarity $_{\text{physics-astronomy}}$	0.7
Score(radioactive)	2.1*1/3 = 0.7

Table 3. Scoring terms based on similarity between domains

3. Complementing unregistered terms and common terms by domain tagged corpora.

$$W_{Dic}(\alpha) = (Score(\alpha)+1) * \sqrt{\frac{\sum_{i=1}^{W} dof_i}{W}} \qquad (2.3)$$

where

W: the number of words in the term 'α'

dof_i: the number of domain that words in the term appear in the domain tagged corpus.

Consider two exceptional possible cases. First, there are unregistered terms that are not contained in any dictionaries. Second, some commonly used terms can be used to describe a special concept in a specific domain dictionary. Since an unregistered term may be a newly created term of domains, it should be considered as a candidate term. In contrast with an unregistered term, common terms should be eliminated from candidate terms. Therefore, the score calculated in the step 2 should be complemented for these purposes. In our method, the domain tagged corpus (ETRI 1997) is used. Each word in the candidate terms – they are composed of more than one word – can appear in the domain tagged corpus. We can count the number of domains where the word appears. If the number is large, we can determine that the word have a tendency to be a common word. If the number is small, we can determine that the word have a high probability to be a valid term. In this paper, the score calculated by the dictionary hierarchy is called Dictionary Weight (W_{Dic}).

3. Statistical Method

The statistical method is divided into two elements. The first element, the Statistical Weight, is based on the frequencies of terms. The second element, the Transliterated word Weight, which is based on the number of transliterated foreign word in the candidate term. This section describes the above two elements.

3.1. Statistical Weight: Frequency Based Weight

In the Statistical Weight, not only abbreviation pairs and translation pairs in a parenthetical expression but also frequencies of terms are considered. Abbreviation pairs and translation pairs are detected using the following simple heuristics:

For a given parenthetical expression A(B),

1. Check on a fact that A and B are abbreviation pairs. The capital letter of A is compared with that of B. If the half of the capital letter are matched for each other sequentially, A and B are determined to abbreviation pairs (Hisamitsu *et. al*, 1998). For example, 'ISO' and 'International Standardization Organization' is detected as an abbreviation in a parenthetical expression 'ISO (International Standardization Organization)'.

2. Check on a fact that A and B are translation pairs. Using the bi-lingual dictionary, it is determined.

After detecting abbreviation pairs and translation pairs, the Statistical Weight (W_{Stat}) of the terms is calculated by the formula (3.1).

$$W_{Stat}(\beta) = \begin{cases} \sum_{\alpha \in S(\beta) \cup \{\beta\}} \left(\sqrt{|\alpha|} \times f(\alpha)\right) & \textit{if} \quad \alpha \quad \textit{is} \quad \textit{nested} \\ \sum_{\alpha \in S(\beta) \cup \{\beta\}} \left[\sqrt{|\alpha|} \times \left(f(\alpha) + \frac{\sum_{\gamma \in T(\alpha)} f(\gamma)}{C(T(\alpha))} \right)\right] & \textit{otherwise} \end{cases} \qquad (3.1)$$

where

α**:** *a candidate term*

$|\alpha|$: *the length of a term'α'*

S (α): abbreviation and translation pairs of 'α'

T(α): The set of candidate terms that nest 'α'

f(α): the frequency of 'α'

C(T(α)): The number of elements in T(α)

In the formula (3.1), the nested relation is defined as follows: let A and B be a candidate term. If A contains B, we define that A nests B.

The formula implies that abbreviation pairs and translation pairs related to 'α' is counted as well as 'α' itself and productivity of words in the nested expression containing 'α' gives more weight, when the generated expression contains 'α'. Moreover, formula (1) deals with a single-word term, since an abbreviation such as GUI (Graphical User Interface) is single word term and English multi-word term usually translated to Korean single-word term – (e.g. distributed database => *bunsan deitabeisu*)

3.2 Transliterated word Weight: By Automatic Extraction of Transliterated words

Technical terms and concepts are created in the world that must be translated or transliterated. Transliterated terms are one of important clues to identify the terms in the given domain. We observe dictionaries of computer science and chemistry domains to investigate the transliterated foreign words. In the result of observation, about 53% of whole entries in a dictionary of a computer science domain are transliterated foreign words and about 48% of whole entries in a dictionary of a chemistry domain are transliterated foreign words. Because there are many possible transliterated forms and they are usually unregistered terms, it is difficult to detect them automatically.

In our method, we use HMM (Hidden Markov Model) for this task (Oh, *et al.*, 1999). The main idea for extracting a foreign word is that the composition of foreign words would be different from that of pure Korean words, since the phonetic system for the Korean language is different from that of the foreign language. Especially, several English consonants that occur frequently in English words, such as ‘p’, ’t’, ’c’, and ‘f’, are transliterated into Korean consonants ‘p’, ‘t’, ‘k’, and ‘p’ respectively. Since these consonants of Korean are not used in pure Korean words frequently, this property can be used as an important clue for extracting a foreign word from Korean. For example, in a word, ‘si-seu-tem’ (system), the syllable ‘tem’ have a high probability to be a syllable of transliterated foreign word, since the consonant of ‘t’ in the syllable ‘tem’ is usually not used in a pure Korean word. Therefore, the consonant information which is acquired from a corpus can be used to determine whether a syllable in the given term is likely to be the part of a foreign word or not.

Using HMM, a syllable is tagged with ‘K’ or ‘F’. A syllable tagged with ‘K’ means that it is part of a pure Korean word. A syllable tagged with ‘F’ means that it is part of a transliterated word. For example, ‘si-seu-tem-eun (system is)’ is tagged with ‘si/F + seu/F + tem/F + eun/K’. We use consonant information to detect a transliterated word like lexical information in part-of-speech-tagging. The formula (3.2) is used for extracting a transliterated word and the formula (3.3) is used for calculating the Transliterated Word Weight (W_{Trl}). The formula (3.3) implies that terms have more transliterated foreign words than common words do.

$$P(T \mid S)P(S) = p(t_1)p(t_2 \mid t_1) \left[\prod_{i=3}^{n} p(t_i \mid t_{i-1}, t_{i-2})\right]\left[\prod_{i=1}^{n} p(s_i \mid t_i)\right] \quad (3.2)$$

where

s_i: i-th consonant in the given word.

t_i: i-th tag (‘F’ or ‘K’) of the syllable in the given word.

$$W_{Trl}(\alpha) = \frac{trans(\alpha)}{|\alpha|} \quad (3.3)$$

where

$|\alpha|$ is the number of words in the term α

***trans(α)** is the number of transliterated words in the term α*

4.Term Weighting

The three individual weights described above are combined according to the following formula (4.1) called Term Weight (W_{Term}) for identifying the relevant terms.

$$W_{term}(\varphi) = \alpha \times f(W_{Dic}(\varphi)) + \beta \times g(W_{Trl}(\varphi)) + \gamma \times h(W_{Stat}(\varphi)) \quad (4.1)$$

Where

φ: a candidate term ‘φ’

f,g,h : normalization function

$\alpha+\beta+\gamma = 1$

In the formula (4.1), the three individual weights are normalized by the function ***f, g,*** and ***h*** respectively and weighted parameter $\boldsymbol{\alpha}$,$\boldsymbol{\beta}$, and $\boldsymbol{\gamma}$. The parameter $\boldsymbol{\alpha}$,$\boldsymbol{\beta}$, and $\boldsymbol{\gamma}$ are determined by experiment with the condition $\boldsymbol{\alpha+\beta+\gamma = 1}$. Each value which is used in this paper is $\boldsymbol{\alpha=0.6}$, $\boldsymbol{\beta=0.1}$, and $\boldsymbol{\gamma=0.3}$ respectively.

5. Experiment

The proposed method is tested on a corpus of computer science domains, called the KT test collection. The collection contains 4,434 documents and 67,253 words and contains documents about the abstract of the paper (Park. *et al.*, 1996). It was tagged with a part-of-speech tagger for evaluation. We examined the performance of the Dictionary Weight (W_{Dic}) to show its usefulness. Moreover, we examined both the performance of the C-value that is based on the statistical method (Frantzi. *et al.*, 1999) and the performance of the proposed method.

5.1 Evaluation Criteria

Two domain experts manually carry out the assessment of the list of terms extracted by the proposed method. The results are accepted as the valid term when both of the two experts agree on them. This prevents the evaluation from being carried out subjectively, when one expert assesses the results. The results are evaluated by a precision rate. A precision rate means that the proportion of correct answers to the extracted results by the system.

5.2 Evaluation by Dictionary Weight (W_{Dic})

In this section, the evaluation is performed using only W_{Dic} to show the usefulness of a dictionary hierarchy to recognize the relevant terms The Dictionary Weight is based on the premise that the information of the target domain is a good indicator for identifying terms. The term in the dictionaries of the target domain and the domain related to the target domain acts as a positive indicator for recognizing terms. The term in the dictionaries of the domains, which are not related to the target domain acts as a negative indicator for recognizing terms. The dictionary hierarchy is constructed to estimate the similarity between one domain and another.

	Top 10%	Bottom 10%
The Valid Term	94%	54.8%
Non-Term	6%	45.2%

Table 4. terms and non-terms by Dictionary Weight

The result, depicted in table 4, can be interpreted as follows: In the top 10% of the extracted terms, 94% of them are the valid terms and 6% of them are non-terms. In the bottom 10% of the extracted terms, 54.8% of them are the valid terms and 45.2% of them are non-terms. This means that the relevant terms are much more than non-terms in the top 10% of the result, while non-terms are much more than the relevant terms in the bottom 10% of the result.

The results are summarized as follow:

- According as a term has a high Dictionary Weight (W_{Dic}), it is apt to be valid.
- More valid terms have a high Dictionary Weight (W_{Dic}) than non-terms do

5.3 Overall Performance

Table 5 and figure 3 show the performance of the proposed method and of the C-value method. By dividing the ranked lists into 10 equal sections, the results are compared. Each section contains the 1291 terms and is evaluated independently.

Section	C-value		The proposed method	
	# of term	Precision	# of term	Precision
1	1181	91.48%	1241	96.13%
2	1159	89.78%	1237	95.82%
3	1207	93.49%	1213	93.96%
4	1192	92.33%	1174	90.94%
5	1206	93.42%	1154	89.39%
6	981	75.99%	1114	86.29%
7	934	72.35%	1044	80.87%
8	895	69.33%	896	69.40%
9	896	69.40%	780	60.42%
10	578	44.77%	379	29.36%

Table 5. Precision rates of C-value and the proposed method : Section contain 1291 terms and precision is evaluated independently. For example, in section 1, since there are 1291 candidate terms and 1241 relevant terms by the proposed method, the precision rate in section 1 is 96.13% .

The result can be interpreted as follows. In the top sections, the proposed method shows the higher precision rate than the C-value does. The distribution of valid terms is also better for the proposed method, since there is a downward tendency from section 1 to section 10. This implies that the terms with higher weight scored by our method have a higher probability to be valid terms. Moreover, the precision rate of our method shows the rapid decrease from section 6 to section 10. This indicates that most of valid terms are located in the top sections.

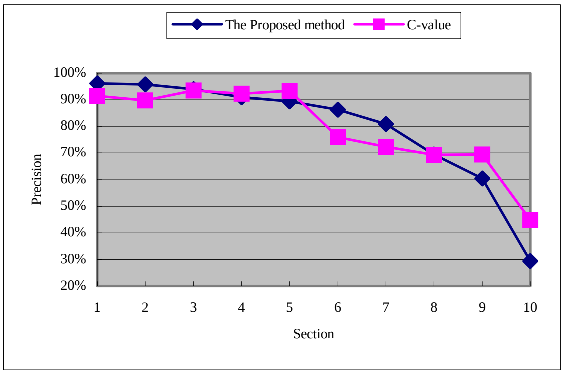

Figure 2. The performance of C-value and the proposed method in each section

The results can be summarized as follow :

- The proposed method extracts a valid term more accurate than C-value does.
- Most of the valid terms are in the top section extracted by the proposed method.

Conclusion

In this paper, we have described a method for term extraction using a dictionary hierarchy. It is constructed by clustering method and is used for estimating the relationships between domains. Evaluation shows improvement over the C-value. Especially, our approach can distinguish the valid terms efficiently – there are more valid terms in the top sections and less valid terms in the bottom sections. Although the method targets Korean, it can be applicable to English by slight change on the Tweight (W_{Trl}).

However, there are many scopes for further extensions of this research. The problems of non-nominal terms (Klavans and Kan, 1998), term variation (Jacquemin *et al*., 1997), and relevant contexts (Maynard and Ananiadou, 1998), can be considered for improving the performance. Moreover, it is necessary to apply our method to practical NLP systems, such as an information retrieval system and a morphological analyser.

Acknowledgements

KORTERM is sponsored by the Ministry of Culture and Tourism under the program of King Sejong Project. Many fundamental researches are supported by the fund of Ministry of Science and Technology under a project of plan STEP2000. And this work was partially supported by the KOSEF through the "Multilingual Information Retrieval" project at the AITrc.

References

Anderberg, M.R. (1973) *Cluster Analysis for Applications.* New York: Academic

Bourigault, D. (1992) *Surface grammatical analysis for the extraction of terminological noun phrases.* In Proceedings of the 14th International Conference on Computational Linguistics, COLING'92 pp. 977-981.

Dagan, I. and K. Church. (1994) *Termight: Identifying and terminology* In Proceedings of the 4th Conference on Applied Natural Language Processing, Stuttgart/Germany, 1994. Association for Computational Linguistics.

ETRI (1997) Etri-Kemong set

Felber Helmut (1984) *Terminology Manual*, International Information Centre for Terminology (Infoterm)

Frantzi, K.T. and S.Ananiadou (1999) *The C-value/NC-value domain independent method for multi-word term extraction.* Journal of Natural Language Processing, 6(3) pp. 145-180

Hisamitsu, Toru and Yoshiki Niwa (1998) *Extraction of useful terms from parenthetical expressions by using simple rules and statistical measures.* In First Workshop on Computational Terminology Computerm'98, pp 36-42

Jacquemin, C., Judith L.K. and Evelyne, T. (1997) *Expansion of Muti-word Terms for indexing and Retrieval Using Morphology and Syntax*, 35th Annual Meeting of the Association for Computational Linguistics, pp 24-30

Justeson, J.S. and S.M. Katz (1995) *Technical terminology : some linguistic properties and an algorithm for identification in text.* Natural Language Engineering, 1(1) pp. 9-27

Klavans, J. and Kan M.Y (1998) *Role of Verbs in Document Analysis*, In Proceedings of the 17th International Conference on Computational Linguistics, COLING'98 pp. 680-686.

Lauriston, A. (1996) *Automatic Term Recognition : performance of Linguistic and Statistical Techniques.* Ph.D. thesis, University of Manchester Institute of Science and Technology.

Lorr, M. (1983) *Cluster Analysis and Its Application*, Advances in Information System Science,8 , pp.169-192

Murtagh, F. (1983) *A Survey of Recent Advances in Hierarchical Clustering Algorithms*, Computer Journal, 26, 354-359

Maynard, D. and Ananiadou, S. (1998) *Acquiring Context Information for Term Disambiguation* In First Workshop on Computational Terminology Computerm'98, pp 86-90

Oh, J.H. and K.S. Choi (1999) *Automatic extraction of a transliterated foreign word using hidden markov model* , In Proceedings of the 11th Korean and Processing of Korean Conference pp. 137-141 (In Korean).

Park, Y.C., K.S. Choi, J.K.Kim and Y.H. Kim (1996). *Development of the KT test collection for researchers in information retrieval*. In the 23th KISS Spring Conference (in Korean)