Proceedings of the Sixth International Workshop on Information
Retrieval with Asian Languages, July 2003, pp. 153-160.

Cross-Language Information Retrieval Based on Category Matching Between Language Versions of a Web Directory

Fuminori Kimura
Graduate School of
Information Science,
Nara Institute of
Science and Technology
8916-5 Takayama,
Ikoma, Nara, Japan

Akira Maeda
Department of
Computer Science,
Ritsumeikan University
1-1-1 Noji-Higashi,
Kusatsu, Shiga, Japan

Masatoshi Yoshikawa
Information Technology
Center, Nagoya University
Furo-cho, Chigusa-ku,
Nagoya, Aichi, Japan

Shunsuke Uemura
Graduate School of
Information Science,
Nara Institute of
Science and Technology
8916-5 Takayama,
Ikoma, Nara, Japan

Abstract

Since the Web consists of documents in various domains or genres, the method for Cross-Language Information Retrieval (CLIR) of Web documents should be independent of a particular domain. In this paper, we propose a CLIR method which employs a Web directory provided in multiple language versions (such as Yahoo!). In the proposed method, feature terms are first extracted from Web documents for each category in the source and the target languages. Then, one or more corresponding categories in another language are determined beforehand by comparing similarities between categories across languages. Using these category pairs, we intend to resolve ambiguities of simple dictionary translation by narrowing the categories to be retrieved in the target language.

1 Introduction

With the popularity of the Internet, more and more languages are becoming to be used for Web documents, and it is now much easier to access documents written in foreign languages. However, existing Web search engines only support the retrieval of documents which are written in the same language as the query, so the monolingual users are not able to retrieve documents written in non-native languages efficiently. Also, there might be cases, depending on the user's demand, where information written in a language other than the user's native language is rich. Needs for retrieving such information must be large. In order to satisfy such needs on a usual monolingual retrieval system, the user him-/herself has to manually translate the query by using a dictionary, etc. This process not only imposes a burden to the user but also might choose incorrect translations for the query, especially for languages that are unfamiliar to the user.

To fulfill such needs, researches on Cross-Language Information Retrieval (CLIR), a technique to retrieve documents written in a certain language using a query written in another language, have been active in recent years. A variety of methods, including employing corpus statistics for the translation of terms and the disambiguation of translated terms, are studied and a certain results has been obtained. However, corpus-based disambiguation methods are heavily affected by the domain of the training corpus, so the retrieval effectiveness for other domains might drop significantly. Besides, since the Web consists of documents in various domains or genres, the method for CLIR of Web documents should be independent of a particular domain.

In this paper, we propose a CLIR method which employs Web directories provided in multiple language versions (such as Yahoo!). Our system uses two or more language versions of a Web directory. One version is the query language, and others are the target languages. From these language versions, category correspondences between languages are estimated in advance. First, feature terms are extracted from Web documents for each category in the source and the target languages. Then, one or more corresponding categories in another language are determined beforehand by comparing similarities between categories across languages. Using these category pairs, we intend to resolve ambiguities of simple dictionary translation by narrowing the categories to be retrieved in the target language.

2 Related Work

Approaches to CLIR can be classified into three categories; document translation, query translation, and the use of inter-lingual representation. The approach based on translation of target documents has the advantage of utilizing existing machine translation systems, in which more content information can be used for disambiguation. Thus, in general, it achieves a better retrieval effectiveness than those based on query translation(Sakai, 2000). However, since it is impractical to translate a huge document collection beforehand and it is difficult to extend this method to new languages, this approach is not suitable for multilingual, large-scale, and frequently-updated collection of the Web. The second approach transfers both documents and queries into an inter-lingual representation, such as bilingual thesaurus classes or a language-independent vector space. The latter approach requires a training phase using a bilingual (parallel or comparable) corpus as a training data.

The major problem in the approach based on the translation and disambiguation of queries is that the queries submitted from ordinary users of Web search engines tend to be very short (approximately two words on average (Jansen et al., 2000)) and usually consist of just an enumeration of keywords (i.e. no context). However, this approach has an advantage that the translated queries can simply be fed into existing monolingual search engines. In this approach, a source language query is first translated into target language using a bilingual dictionary, and translated query is disambiguated. Our method falls into this category.

It is pointed out that corpus-based disambiguation methods are heavily affected by the difference in domain between query and corpus. Hull suggests that the difference between query and corpus may cause bad influence on retrieval effectiveness in the methods that use parallel or comparable corpora (Hull, 1997). Lin et al. conducted comparative experiments among three monolingual corpora that have different domains and sizes, and has concluded that large-scale and domain-consistent corpus is needed for obtaining useful co-occurrence data (Lin et al., 1999).

On the Web retrieval, which is the target of our research, the system has to cope with queries in many different kinds of topics. However, it is impractical to prepare corpora that cover any possible domains. In our previous paper(Kimura et al., 2003), we proposed a CLIR method which uses documents in a Web directory that has several language versions (such as Yahoo!), instead of using existing corpora, in order to improve the retrieval effectiveness. In this paper, we propose an extension of our method which takes account of the hierarchical structure of Web directories. Dumais et al.(Dumais and Chen, 2000) suggests that the precision of Web document classification could be improved to a certain extent by limiting the target categories to compare by using the hierarchical structure of a Web directory. In this paper, we try to improve our proposed method by incorporating the hierarchical structure of a Web directory for merging categories.

3 Proposed System

3.1 Outline of the System

Our system uses two or more language versions of a Web directory. One version is the query language (language A in Figure 1), others are the target languages to be retrieved (language B in Figure 1). From these language versions, category correspondences between languages are estimated in advance.

The preprocessing consists of the following four steps: 1) term extraction from Web documents in each category, 2) feature term extraction, 3) translation of feature terms, and 4) estimation of category correspondences between different languages. Figure 1 illustrates the flow of the preprocessing. This example shows a case that category a in language A corresponds to a category in language B. First, the system extracts terms from Web documents which belong to category a (1). Secondly, the system calculates the weights of the extracted terms. Then higher-weighted terms are extracted as the feature term set f_a of category a (2). Thirdly, the system translates the feature term set f_a into language B (3). Lastly, the system compares the translated feature term set of category *a* with feature term sets of all categories in language B, and estimates the corresponding category of category a from language B (4).

These category pairs are used on retrieval. First, the system estimates appropriate category for the query in the query language. Next, the system selects the corresponding category in the target language using the pre-estimated category pairs. Finally, the system retrieves Web documents in the selected corresponding category.

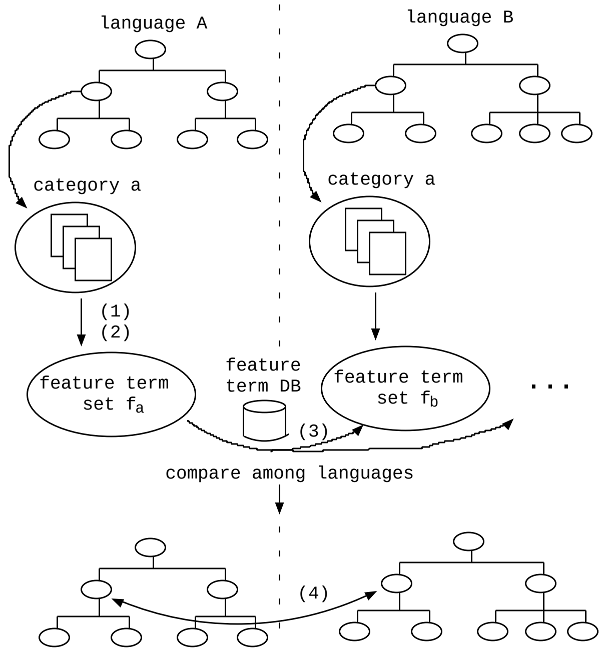

Figure 1: Preprocessing.

3.2 Preprocessing

3.2.1 Feature Term Extraction

The feature of each category is represented by its feature term set. Feature term set is a set of terms that seem to distinguish the category. The feature term set of each category is extracted in the following steps: First, the system extracts terms from Web documents that belong to a given category. In this time, system also collect term frequency of each word in each category and normalize these frequency for each category. Second, the system calculates the weights of the extracted terms using TF·ICF (term frequency · inverse category frequency). Lastly, top n ranked terms are extracted as the feature term set of the category.

Weights of feature terms are calculated by TF·ICF. TF·ICF is a variation of TF·IDF (term frequency · inverse document frequency). Instead of using a document as the unit, TF·ICF calculates weights by category. TF·ICF is calculated as follows:

$$tf \cdot icf(t_i, c) = \frac{f(t_i)}{N_c} \cdot \log \frac{N}{n_i} + 1$$

where t_i is the term appearing in the category c, $f(t_i)$ is the term frequency of term t_i, N_c is the total number of terms in the category c, n_i is number of the categories that contain the term t_i, and N is the number of all categories in the directory.

3.2.2 Category Matching Between Languages

For estimating category correspondences between languages, we compare each feature term set of a category which is extracted in section 3.2.1, and calculates similarities between categories across languages.

In order to compare two categories between languages, feature term set must be translated into the target language. First, for each feature term, the system looks up the term in a bilingual dictionary and extracts all translation candidates for the feature term. Next, the system checks whether each translation candidate exists in the feature term set of the target category. If the translation candidate exists, the system checks the candidate's weight in the target category. Lastly, the highest-weighted translation candidate in the feature term set of the target category is selected as the translation of the feature term. Thus, translation candidates are determined for each category, and translation ambiguity is resolved.

If no translation candidate for a feature term exists in the feature term set of the target category, that term is ignored in the comparison. However, there are some cases that the source language term itself is useful as a feature term in the target language. For example, some English terms (mostly abbreviations) are commonly used in documents written in other languages (e.g. “WWW”, “HTM”, etc.). Therefore, in case that no translation candidate for a feature term exists in the feature term set of the target category, the feature term itself is checked whether it exists in the feature term set of the target category. If it exists, the feature term itself is treated as the translation of the feature term in the target category.

As an example, we consider that an English term “system” is translated into Japanese for the category “コンピュータとインターネット >ソフトウェア >セキュリティ (Computers and Internet >Software >Security)” (hereafter called “セキュリティ” for short). The English term “system” has the following translation candidates in a dictionary; “宇宙 (universe/space)”, “方法 (method)”, “組織 (organization)”, “器官 (organ)”, “システム (system)”, etc. We check each of these translation candidates in the feature term set of the category “セキュリティ.” Then the highest-weighted term of these translation candidates in the category “セキュリティ” is determined as the translation of the English term “system” in this category. If no translation candidate exists in the feature term set of the category “セキュリティ,” the English term “system” itself is treated as the translation.

Once all the feature terms are translated, the system calculates the similarities between categories across languages. The similarity between the source category a and the target category b is calculated as the total of multiplying the weights of each feature term in the category a by the weight of its translation in the category b. The similarity of the category a for the category b is calculated as follows:

$$sim(a,b) = \sum_{f \in f_a} w(f,a) \cdot w(t,b)$$

where f is a feature term, f_a is the feature term set of category a, t is the translation of f in the category b, and $w(f,a)$ is the weight of f in a.

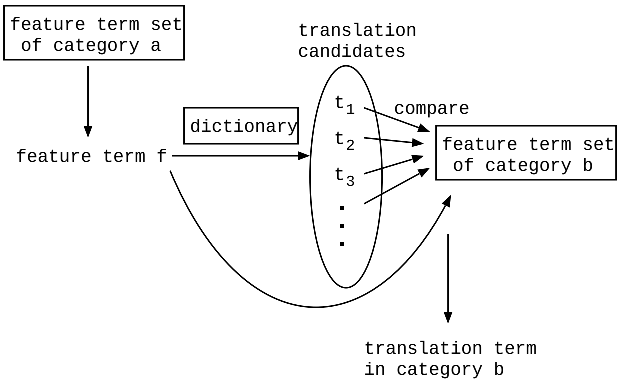
feature term set of category a
translation candidates
dictionary
t1
t2
t3
compare
feature term f
feature term set of category b
translation term in category b

Figure 2: Feature term translation.

The system calculates the similarities of category a for each category in the target language using the above-mentioned method. Then, a category with the highest similarity in the target language is selected as the correspondent of category a.

As an example, we consider an example of calculating the similarity of an English category “Computers and Internet >Security and Encryption” (hereafter called “Encryption” for short) for the category “セキュリティ” which is mentioned above. Suppose that the feature term set of the category “Encryption” has the following feature terms; “privacy”, “system”, etc., and the weights of these terms are 0.007110, 0.006327, $\cdots$. Also suppose that the Japanese translations of these terms are “プライバシー (privacy)”, “システム (system)”, etc., and the weights of these terms are 0.023999, 0.047117, $\cdots$. In this case, the similarity of the category “Encryption” (s_1) for the category “セキュリティ” (s_2) is calculated as follows:

$$\begin{aligned} sim(s_1, s_2) = \quad & 0.007110 \times 0.023999 \\ & +0.006327 \times 0.047117 \\ & +\cdots \end{aligned}$$

3.2.3 Retrieval

Figure 3 illustrates the processing flow of a retrieval. When the user submits a query, the following steps are processed.

In our system, a query consists of some keywords, not of a sentence. We define the query vector $\vec{q}$ as follows:

$$\vec{q} = (q_1, q_2, \ldots, q_n)$$

where q_k is the weight of the k-th keyword in the query. We define the values of all q_k are 1.

First, the system calculates the relevance between the query and each category in the source language, and determines the relevant category of the query in the source language (1). The relevance between the query and each category is calculated by multiplying the inner product between query terms and the feature term set of the target category by the angle of these two vectors. The relevance between query q and category c is calculated as follows:

$$rel(q,c) = \vec{q} \cdot \vec{c} \cdot \frac{\vec{q} \cdot \vec{c}}{|\vec{q}| \cdot |\vec{c}|}$$

where $\vec{c}$ is a vector of category c defined as follows:

$$\vec{c} = (w_1, w_2, \ldots, w_n)$$

where w_k is the weight of the k-th keyword in the feature term set of c.

If there is more than one category whose relevance for the query exceeds a certain threshold, all of them are selected as the relevant categories of the query. It is because there might be some cases that, for example, documents in the same domain belong to different categories, or a query concept belongs to multiple domains.

Second, the corresponding category in the target language is selected by using category correspondences between languages mentioned in section 3.2.2 (2). Third, the query is translated into the target language by using a dictionary and the feature term set of the corresponding category (3). Finally, the system retrieves documents in the corresponding category (4).

4 Category Merging

4.1 Previous Experiments

In our previous paper(Kimura et al., 2003), we conducted experiments of category matching using the subsets of English and Japanese versions of Yahoo!.

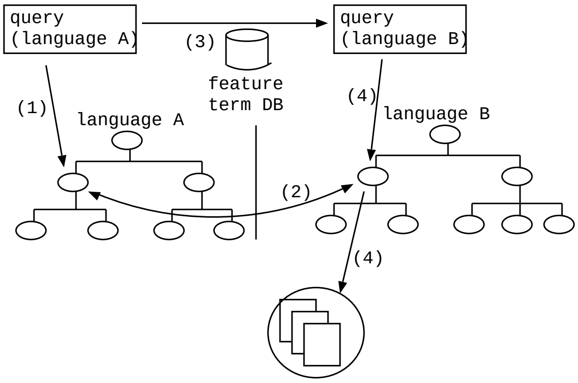

Figure 3: Processing in retrieval.

The English subset is 559 categories under the category "Computers and Internet" and the Japanese subset is 654 categories under the corresponding category "コンピュータとインターネット (Computers and Internet)." Total size of English web pages in each category after eliminating HTML tags are 45,905 bytes on average, ranging from 476 to 1,084,676 bytes. Total size of Japanese web pages are 22,770 bytes on average, ranging from 467 to 409,576 bytes.

In our previous experiments, we could not match categories across languages with adequate accuracy. It may have been caused by the following reasons; one possible reason is that the size of Web documents was not enough for statistics in some categories, and another is that some categories are excessively divided as a distinct domain.

For the former observation, we eliminated the categories whose total bytes of Web documents are less than 30KB, but the results were not improved.

4.2 Method of Category Merging

Considering the result of the above experiments, we need to solve the problem of excessive division of categories in order to accurately match categories between languages.

The problem might be caused by the following reasons; one possible reason is that there are some categories which are too close in topic, and it might cause poor accuracy. Another possible reason is that some categories have insufficient amount of text in order to obtain statistically significant values for feature term extraction. Considering the above observations, we might expect that the accuracy will be improved by merging child categories at some level in the category hierarchy in order to merge some categories similar in topic and to increase the amount of text in a category.

Accordingly, we solve the problem by merging child categories into the parent category at some level using the directory hierarchy. As child categories are specialized ones of the parent category, we can assume that these categories have similar topic. Besides, even if two categories have no direct link from each other, we can assume that categories that have same parent category might also have similar topic.

However, we still need further investigation on at which level categories should be merged.

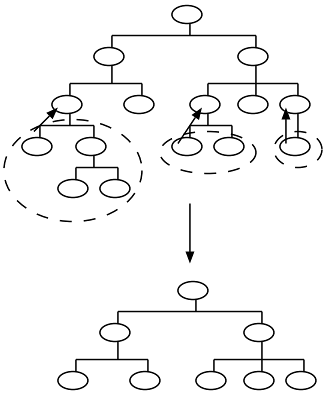

Figure 4: Category merging.

5 Experiments

We are conducting experiments of the proposed method to detect relevance category of a query. In this experiment, we used the same subsets mentioned in section 4.1. We merged the categories three levels below the category "Computers and Internet" into the parent. The number of categories after category merging is 342 in English and 265 in Japanese.

At first, we have done the experiment using the following formula that uses only inner product, before using the calculation mentioned in section 3.2.3.

$$rel_{inner}(q,c) = \vec{q} \cdot \vec{c}$$

In this experiment, the query has three terms: "encryption"(=q_1), "security"(=q_2), and "system"(=q_3).

Table 1 is the list of top 10 relevant categories in first experiment. Almost all the categories in the Table 1 are relevant to the query. Thus, the relevance calculation method by only inner product is regarded as an effective method. However, this method has the following problem. The category that has few query terms might be given high relevance when the category has the only one query term whose weight in the category is extremely high.

In order to reduce this effect, we propose the improved method mentioned in section 3.2.3. The method is revised to take account of the angle between $\vec{q}$ and $\vec{c}$. Ultimately, the most relevant category has the vector whose length is long and whose factors are flat. The length is considered by inner product, on the other hand, flatness is considered by the angle between $\vec{q}$ and $\vec{c}$.

Table 2 is the list of top 10 relevant categories in the second experiment using revised method. Although noticeable improvement does not appear, the relevance of the categories which matches few query terms are ranked lower than the first experiment.

6 Conclusions

In this paper, we proposed a method using a Web directory for CLIR. The proposed method is independent of a particular domain because it uses documents in a Web directory as the corpus. Our method is particularly effective for the case that the document collection covers wide range of domains such as the Web. Besides, our method does not require expensive linguistic resources except for a dictionary. Therefore, our method can easily be extended to other languages as long as the language versions of a Web directory exist and the dictionary can be obtained.

Future work includes improving the category matching method and the evaluation of retrieval effectiveness.

References

Susan Dumais and Hao Chen. 2000. Hierarchical classification of Web content. *Proceedings of the 23rd*

ACM International Conference on Research and Development in Information Retrieval(SIGIR2000).

David A. Hull. 1997. Using structured queries for disambiguation in cross-language information retrieval. *Electronic Working Notes of the AAAI Symposium on Cross-Language Text and Speech Retrieval*.

Bernard J. Jansen, Amanda Spink, and Tefko Saracevic. 2000. Real life, real user queries on the Web. *Information Processing & Management*, 36(2).

Fuminori Kimura, Akira Maeda, Masatoshi Yoshikawa, and Shunsuke Uemura. 2003. Cross-Language Information Retrieval using Web Directory Structure. *The 14th Data Engineering Workshop, (in Japanese)*.

Chuan-Jie Lin, Wen-Cheng Lin, Guo-Wei Bian, and Hsin-Hsi Chen. 1999. Description of the NTU Japanese-English cross-lingual information retrieval system used for NTCIR workshop. *First NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition*.

Tetsuya Sakai. 2000. MT-based Japanese-English cross-language IR experiments using the TREC test collections. *Proceedings of The Fifth International Workshop on Information Retrieval with Asian Languages (IRAL2000)*.

Table 1: The list of top 10 relevant category calculated by inner product.

category name	relevance	weight($q_1/q_2/q_3$)
Computers and Internet/Security and Encryption/Challenges/	0.166845	0.112607/0.054238/0.000000
Computers and Internet/Security and Encryption/Conferences/	0.126984	0.000000/0.126984/0.000000
Computers and Internet/Security and Encryption/Web Directories/	0.106283	0.012577/0.093706/0.000000
Computers and Internet/Security and Encryption/Organizations/	0.089169	0.006647/0.076520/0.006002
Business and Economy/Business to Business/Computers/Security and Encryption/	0.087314	0.006391/0.074656/0.006267
Computers and Internet/Security and Encryption/Encryption Policy/	0.086271	0.075185/0.011086/0.000000
Computers and Internet/Security and Encryption/Mailing Lists/	0.075399	0.017247/0.058152/0.000000
Computers and Internet/Software/Operating Systems/File Systems/	0.075088	0.027648/0.024968/0.022472
Computers and Internet/Internet/World Wide Web/Security and Encryption/	0.073100	0.005671/0.05612/0.011309
Computers and Internet/Software/Operating Systems/Inferno/	0.070922	0.000000/0.000000/0.070922

Table 2: The list of the top 10 relevance category calculated by proposed method in section 3.2.3.

category name	relevance	weight($q_1/q_2/q_3$)
Computers and Internet/Security and Encryption/Challenges/	0.128587	0.112607/0.054238/0.000000
Computers and Internet/Software/Operating Systems/File Systems/	0.074822	0.027648/0.024968/0.022472
Computers and Internet/Security and Encryption/Conferences/	0.073314	0.00000/0.126984/0.000000
Computers and Internet/Security and Encryption/Web Directories/	0.068980	0.012577/0.093706/0.000000
Computers and Internet/Security and Encryption/Organizations/	0.059585	0.006647/0.07652/0.006002
Business and Economy/Business to Business/Computers/Security and Encryption/	0.058539	0.006391/0.074656/0.006267
Computers and Internet/Security and Encryption/Encryption Policy/	0.056542	0.075185/0.011086/0.000000
Computers and Internet/Security and Encryption/Mailing Lists/	0.054113	0.017247/0.058152/0.000000
Computers and Internet/Internet/World Wide Web/Security and Encryption/	0.053628	0.005671/0.05612/0.011309
Computers and Internet/Programming and Development/Languages/Java/Security/	0.046474	0.000000/0.054276/0.01271