Automatic Generation of the Domain-Specific Dictionary for Text Classification

Yuu Sakurai

School of Information Science,
Japan Advanced Institute of Science and Technology

February 15, 1999

Keywords: text classification, information extraction.

As Internet grows rapidly, the avalanche of information in forms of electronic text perplexes us. Internet users feel difficulties in looking for information that they want from electronic text mass media such as the USENET News (netnews) and WWW.

One of the most effective methods for users who are annoyed when they find certain information is the automated editing; it edits electronic texts in advance. One of the automated editing system is **Sun QA-Pack** that edits articles of **q**uestion-and-**a**nswer (Q and A) newsgroups about Sun workstations as **F**requently **A**sked **Q**uestions (FAQ) for many users. One of the central method of this system is text classification (category assignment). Hierarchical text classification by using their contents is as follows. Using the **domain-specific dictionary** that contains the **domain-specific terms** and their **classification codes**, this method extracts summary and technical terms from question articles, and classifies articles according to the number of assigned codes.

The accuracy of this hierarchical text classification method depends on how many the domain-specific terms in articles that the domain-specific dictionary consists. However, the new domain-specific terms frequently appear in newsgroups that Sun QA-Pack edits. It is difficult to extract these new domain-specific terms from texts, to assign categories to them, and to add them to the domain-specific dictionary all manually.

Hence, I present an idea of extracting terms from texts, and adding them to dictionary all automatically. I achieve this idea by realizing the following.

- the domain-specific term extraction
- the category assignment

First, I describe the domain-specific term extraction. The fundamental structure of Sun QA-Pack is the **classification tree**. In this tree, a node represents a classification concept—the upper nodes represent general concepts, and the lower nodes represent specific concepts. Each node represents category that are assigned articles in correlation with the node's concept. Terms are more effective words to classify articles. They consist of two types: (a) **technical term** that represents the specific classification concepts—classification categories, (b) **proper noun** that belongs to the specific concepts, such as **name-of-programs**, **name-of-systems**, and **name-of-products**. Technical terms can be prepared in advance, because the number of the domain-specific terms that represent the specific categories is limited. However, proper nouns frequently appear in articles. Newsgroups that Sun QA-Pack edits are Q and A newsgroups about Sun workstations. In articles of these newsgroups, proper nouns are in forms of ' **Kata-Kana**', **alphabets**, and **numbers**. The system extracts the strings of 'Kata-Kana', alphabets, and numbers from articles, and makes them the **candidates**.

Second, I describe the category assignment. The system presumes classification codes to be assigned to candidates by using contexts of articles that candidates appear in. This method has two main rules. Each rule uses (i) the relation of verbs and thier 'Kaku-Youso', (ii) the relation of words and the domain-specific terms before and behind them.

The first rule uses that the specific verb takes the domain-specific terms that are assigned to the specific category as its 'Kaku-Youso'. The second rule uses that the specific word takes the domain-specific terms that are assigned to the specific category, before and behind its appearance. This rule has two rules. The first rule uses the expression which indicates the word's category, such as 'X to-i-u Y' (Y of X). It means that category that are assigned to 'X' is 'Y'. The second rule uses that a word that composes the parallel relation takes the domain-specific terms that are assigned the same category before and behind it.

These rules are not perfectly reliable. Therefore, in order to enhance reliability, I set four ranks of reliability, *a*, *b*, *c*, *d*,—reliable to unreliable—to these rules, get reliability by applying rules, count them, translate them to scores, and provide threshold to the scores.

In newsgroups about Sun workstations that Sun QA-Pack edits, the most of the domain-specific terms are assigned to any of four categories: OS, machine, hardware, and software. First, I presume that the domain-specific terms are assigned to any of them. Second, for generation of the domain-specific dictionary, I presume categories of deeper level, that is, the detailed categories.

The target texts for the category assignment method are not only articles in newsgroups. I expect improvement of results by applying to texts from other sources—for example, WWW which has enourmous collections of texts. I expect high accuracy of results by applying this method to them.

I evaluated the performance of these two methods—(1) the domain-specific term extraction, (2) the category assignment. These methods extract the domain-specific terms from various texts at Internet in most cases, and presume appropriate categories to these domain-specific terms.

There are three possible directions for the future category assignment for generation of the domain-specific dictionary: (1) training of the domain-specific term extraction, (2)

training of rules, and resetting their reliability and setting them to rules again according to results, (3) training of the category assignment for texts from WWW.