Blog Categorization Exploiting Domain Dictionary and Dynamically Estimated Domains of Unknown Words

Chikara Hashimoto
Graduate School of Science and Engineering
Yamagata University
Yonezawa-shi, Yamagata, 992-8510, Japan
ch@yz.yamagata-u.ac.jp

Sadao Kurohashi
Graduate School of Informatics
Kyoto University
Sakyo-ku, Kyoto, 606-8501, Japan
kuro@i.kyoto-u.ac.jp

Abstract

This paper presents an approach to text categorization that **i)** uses no machine learning and **ii)** reacts on-the-fly to unknown words. These features are important for categorizing Blog articles, which are updated on a daily basis and filled with newly coined words. We categorize 600 Blog articles into 12 domains. As a result, our categorization method achieved an accuracy of 94.0% (564/600).

1 Introduction

This paper presents a simple but high-performance method for text categorization. The method assigns domain tags to words in an article, and categorizes the article as the most dominant domain. In this study, the 12 domains in Table 1 are used following (Hashimoto and Kurohashi, 2007) (H&K hereafter)[1]. Fundamental words are assigned with a domain tag by H&K's domain dictionary, while the domains of non-fundamental words (i.e. unknown words) are dynamically estimated, which makes the method different from previous ones. Another hallmark of the method is that it requires no machine learning. All you need is the domain dictionary and the access to the Web.

Table 1: Domains Assumed in H&K

CULTURE	LIVING	SCIENCE
RECREATION	DIET	BUSINESS
SPORTS	TRANSPORTATION	MEDIA
HEALTH	EDUCATION	GOVERNMENT

[1] In addition, NODOMAIN is prepared for words belonging to no particular domain like *blue* or *people*.

2 The Domain Dictionary

H&K constructed a domain dictionary, where about 30,000 Japanese fundamental content words (JFWs) are associated with appropriate domains. For example, *homer* is associated with SPORTS.

2.1 Construction Process

① Preparing Keywords for each Domain About 20 keywords for each domain were collected manually from words that appear frequently in the Web. They represent the contents of domains.

② Associating JFWs with Domains A JFW is associated with a domain of the highest A_d score. An A_d score of domain is calculated by summing up the top five A_k scores of the domain. Then, an A_k score, which is defined between a JFW and a keyword of a domain, is a measure that shows how strongly the JFW and the keyword are related. H&K adopt the χ^2 statistics to calculate an A_k score and use web pages as a corpus. The number of co-occurrences is approximated by the number of search engine hits when the two words are used as queries. A_k score between a JFW (jw) and a keyword (kw) is given as below.

$$A_k(jw, kw) = \frac{n(ad - bc)^2}{(a+b)(c+d)(a+c)(b+d)} \quad (1)$$

where n is the total number of Japanese web pages,

$$a = hits(jw \ \& \ kw), \quad b = hits(jw) - a,$$
$$c = hits(kw) - a, \quad d = n - (a + b + c).$$

Proceedings of ACL-08: HLT, Short Papers (Companion Volume), pages 69–72,
Columbus, Ohio, USA, June 2008.

Note that $hits(q)$ represents the number of search engine hits when q is used as a query.

③ **Manual Correction** Manual correction of the automatic association[2] is done to complete the dictionary. Since the accuracy of ② is 81.3%, manual correction is not time-consuming.

2.2 Distinctive Features

H&K's method is independent of what domains to assume. You can create your own dictionary. All you need is prepare keywords of your own domains. After that, the same construction process is applied.

Also note that H&K's method requires no text collection that is typically used for machine learning techniques. All you need is the access to the Web.

3 Blog Categorization

The categorization proceeds as follows: 1 Extract words from an article, 2 Assign domains and IDFs to the words, 3 Sum up IDFs for each domain, 4 Categorize the article as the domain of the highest IDF.[3] As for 2, the IDF is calculated as follows:[4]

$$\text{IDF}(w) = log\frac{\text{Total \# of Japanese web pages}}{\text{\# of hits of } w} \quad (2)$$

Fundamental words are assigned with their domains and IDFs by the domain dictionary, while those for unknown words are dynamically estimated by the method described in §4.

4 Domain Estimation of Unknown Words

The domain (and IDF) of unknown word is dynamically estimated exploiting the Web. More specifically, we use Wikipedia and Snippets of Web search, in addition to the domain dictionary. The estimation proceeds as follows (Figure 1): ❶ Search the Web with an unknown word, acquire the top 100 records, and calculate the IDF. ❷ Get the Wikipedia article about the word from the search result if any, estimate the domain of the word with the Wikipedia-strict module (§4.1), and exit. ❸ When no Wikipedia article about the word is found, then get any Wikipedia article in the top 30 of the search result if any, estimate the domain with the Wikipedia-loose module (§4.1), and exit. ❹ If no Wikipedia article is found in the top 30 of the search result, then remove all corporate snippets. ❺ Estimate the domain with the Snippets module (§4.2) if any snippet is left in the search result, and exit. ❻ If no snippet is left but the unknown word is a compound word containing fundamental words, then estimate the domain with the Components module (§4.3), and exit. ❼ If no snippet is left and the word does not contain fundamental words, then the estimation is a failure.

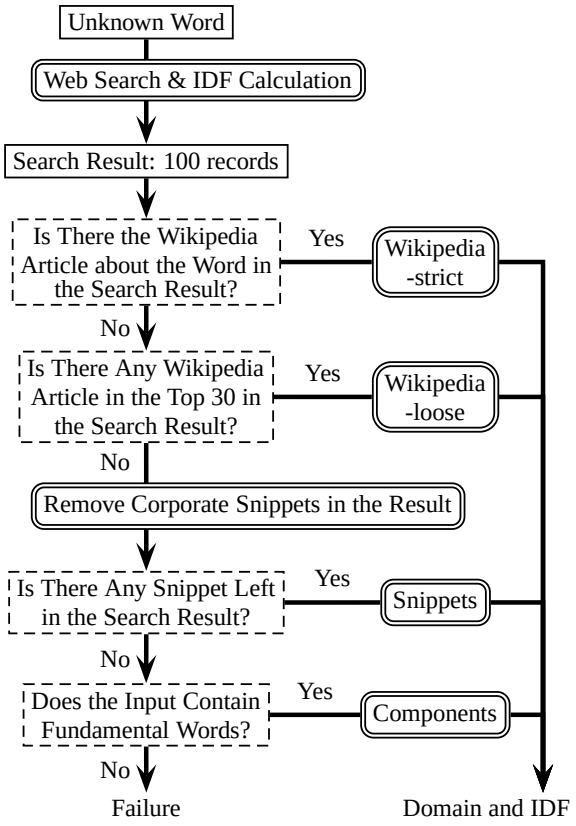

Figure 1: Domain Estimation Process

4.1 Wikipedia(-strict|-loose) Module

The two Wikipedia modules take the following procedure: 1 Extract only fundamental words from the Wikipedia article. 2 Assign domains and IDFs to the words using the domain dictionary. 3 Sum up IDFs for each domain. 4 Assign the domain of the highest IDF to the unknown word. If the domain is NODOMAIN, the second highest domain is chosen for the unknown word under the condition below:

[2] In H&K's method, reassociating JFWs with NODOMAIN is required before ③. We omit that due to the space limitation.

[3] If the domain of the highest IDF is NODOMAIN, the article is categorized as the second highest domain.

[4] We used 10,000,000,000 as the total number.

Second-highest-IDF/ NODOMAIN*'s-IDF*$>$0.15

4.2 Snippets Module

The Snippets module takes as input the snippets that are left in the search result after removing those of corporate web sites. We remove snippets in which corporate keywords like *sales* appear more than once. The keywords were collected from the analysis of our preliminary experiments. Removing corporate snippets is indispensable because they bias the estimation toward BUSINESS. This module is the same as the Wikipedia modules except that it extracts fundamental words from residual snippets.

4.3 Components Module

This is basically the same as the others except that it extracts fundamental words from the unknown word itself. For example, the domain of *finance market* is estimated from the domains of *finance* and *market*.

5 Evaluation

5.1 Experimental Condition

Data We categorized 600 Blog articles from Yahoo! Blog (`blogs.yahoo.co.jp`) into the 12 domains (50 articles for each domain). In Yahoo! Blog, articles are manually classified into Yahoo! Blog categories ($\simeq$ domains) by authors of the articles.

Evaluation Method We measured the accuracy of categorization and the domain estimation. In categorization, we tried three kinds of words to be extracted from articles: fundamental words (**F only** in Table 3), fundamental and simplex unknown words (i.e. no compound word) (**F+SU**), and fundamental and all unknown words (both simplex and compound, **F+AU**). Also, we measured the accuracy of N best outputs (**Top N**). During the categorization, about 12,000 unknown words were found in the 600 articles. Then, we sampled 500 estimation results from them. Table 2 shows the breakdown of the 500 unknown words in terms of their correct domains. The other 167 words belong to NODOMAIN.

5.2 Result of Blog Categorization

Table 3 shows the accuracy of categorization. The **F only** column indicates that a rather simple method like the one in §3 works well, if fundamental words are given good clues for categorization: the domain in our case. This is consistent with Kornai et al. (2003), who claim that only positive evidence matter in categorization. Also, **F+SU** slightly outperformed **F only**, and **F+AU** outperformed the others. This shows that the domain estimation of unknown words moderately improves Blog categorization.

Errors are mostly due to the system's incorrect focus on topics of secondary importance. For example, in an article on a sightseeing trip, which should be RECREATION, the author frequently mentions the means of transportation. As a result, the article was wrongly categorized as TRAFFIC.

Table 2: Breakdown of Unknown Words

CULT	42	LIVI	19	SCIE	38
RECR	15	DIET	19	BUSI	32
SPOR	27	TRAN	28	MEDI	23
HEAL	22	EDUC	24	GOVE	44

Table 3: Accuracy of Blog Categorization

Top N	F only	F+SU	F+AU
1.	0.89	0.91	0.94
2.	0.96	0.97	0.98
3.	0.98	0.98	0.99

5.3 Result of Domain Estimation

The accuracy of the domain estimation of unknown words was 77.2% (386/500). Table 4 shows the frequency in use and accuracy for each domain estimation module.[5] The Snippets module was used most frequently and achieved the reasonably good accuracy of 76%. Though the Wikipedia-strict module showed the best performance, it was used not

Table 4: Frequency and Accuracy for each Module

	Frequency	Accuracy
Wiki-s	0.146 (73/500)	0.85 (62/73)
Wiki-l	0.208 (104/500)	0.70 (73/104)
Snippt	0.614 (307/500)	0.76 (238/307)
Cmpnt	0.028 (14/500)	0.64 (9/14)
Failure	0.004 (2/500)	——

[5] Wiki-s, Wiki-l, Snippt and Cmpnt stand for Wikipedia-strict, Wikipedia-loose, Snippets and Components, respectively.

so often. However, we expect that as the number of Wikipedia articles increases, the best performing module will be used more frequently.

An example of newly coined words whose domains were estimated correctly is デイトレ, which is the abbreviation of デイトレード *day-trade*. It was correctly assigned with BUSINESS by the Wikipedia-loose module.

Errors were mostly due to the subtle boundary between NODOMAIN and the other particular domains. For instance, person's names that are common and popular should be NODOMAIN. But in most cases they were associated with some particular domain. This is due to the fact that virtually any person's name is linked to some particular domain in the Web.

6 Related Work

Previous text categorization methods like Joachims (1999) and Schapire and Singer (2000) are mostly based on machine learning. Those methods need huge quantities of training data, which is hard to obtain. Though there has been a growing interest in semi-supervised learning (Abney, 2007), it is in an early phase of development.

In contrast, our method requires no training data. All you need is a manageable amount of fundamental words with domains. Also note that our method is NOT tailored to the 12 domains. If you want your own domains to categorize, it is only necessary to construct your own dictionary, which is also domain-independent and not time-consuming.

In fact, there have been other proposals without the burden of preparing training data. Liu et al. (2004) prepare representative words for each class, by which they collect initial training data to build classifier. Ko and Seo (2004) automatically collect training data using a large amount of unlabeled data and a small amount of seed information. However, the novelty of this study is the on-the-fly estimation of unknown words' domains. This feature is very useful for categorizing Blog articles that are updated on a daily basis and filled with newly coined words.

Domain information has been used for many NLP tasks. Magnini et al. (2002) show the effectiveness of domain information for WSD. Piao et al. (2003) use domain tags to extract MWEs.

Previous domain resources include WordNet (Fellbaum, 1998) and HowNet (Dong and Dong, 2006), among others. H&K's dictionary is the first fully available domain resource for Japanese.

7 Conclusion

This paper presented a text categorization method that exploits H&K's domain dictionary and the dynamic domain estimation of unknown words. In the Blog categorization, the method achieved the accuracy of 94%, and the domain estimation of unknown words achieved the accuracy of 77%.

References

Steven Abney. 2007. *Semisupervised Learning for Computational Linguistics*. Chapman & Hall.

Zhendong Dong and Qiang Dong. 2006. *HowNet and the Computation of Meaning*. World Scientific Pub Co Inc.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Chikara Hashimoto and Sadao Kurohashi. 2007. Construction of Domain Dictionary for Fundamental Vocabulary. In *ACL '07 Poster*, pages 137–140.

Thorsten Joachims. 1999. Transductive Inference for Text Classification using Support Vector Machines. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pages 200–209.

Youngjoong Ko and Jungyun Seo. 2004. Learning with Unlabeled Data for Text Categorization Using Bootstrapping and Feature Projection Techniques. In *ACL '04*, pages 255–262.

András Kornai, Marc Krellenstein, Michael Mulligan, David Twomey, Fruzsina Veress, and Alec Wysoker. 2003. Classifying the Hungarian web. In *EACL '03*, pages 203–210.

Bing Liu, Xiaoli Li, Wee Sun Lee, , and Philip Yu. 2004. Text Classification by Labeling Words. In *AAAI-2004*, pages 425–430.

Bernardo Magnini, Carlo Strapparava, Giovanni Pezzulo, and Alfio Gliozzo. 2002. The Role of Domain Information in Word Sense Disambiguation. *Natural Language Engineering, special issue on Word Sense Disambiguation*, 8(3):359–373.

Scott S. L. Piao, Paul Rayson, Dawn Archer, Andrew Wilson, and Tony McEnery. 2003. Extracting multiword expressions with a semantic tagger. In *Proceedings of the ACL 2003 workshop on Multiword expressions*, pages 49–56.

Robert E. Schapire and Yoram Singer. 2000. BoosTexter: A Boosting-based System for Text Categorization. *Machine Learning*, 39(2/3):135–168.