Enhancing Automatic Blog Classification Using Concept-Category Vectorization

Ramesh Kumar Ayyasamy[1], Saadat M. Alhashmi[1], Siew Eu-Gene[2],
and Bashar Tahayna[1]

[1]School of Information Technology, [2]School of Business
Monash University, Malaysia
{ramesh.kumar,alhashmi,siew.eu-gene,bashar.tahayna}@monash.edu

Abstract. Blogging has gained popularity in recent years. Blog, a user generated content is a rich source of information and many research are conducted in finding ways to classify blogs. In this paper, we present the solution for automatic blog classification through our new framework using Wikipedia's category system. Our framework consists of two stages: The first stage is to find the meaningful terms from blogposts to a unique concept as well as disambiguate the terms belonging to more than one concept. The second stage is to determine the categories to which these found concepts appertain. Our *Wikipedia based blog classification* framework categorizes blog into topic based content for blog directories to perform future browsing and retrieval. Experimental results confirm that proposed framework categorizes blogposts effectively and efficiently.

Keywords: Blog classification, Wikipedia, Weighting Scheme.

1 Introduction

The blogosphere is expanding in an unrivalled speed till this day. As per *BlogPulse*[1] statistics, there are 170 million identified blogs, 954,136 new blogs in last 24 hours and 1,010,813 blogposts are indexed in last 24 hours. Blogs are user-generated content, where blogger writes any topic of his/her interests and topics could be diverse. These diverse and dynamic natures of blogs make blog classification much more challenging task than the traditional text classification. Blog classification could provide a structure for organizing blog pages for efficient indexing and classification. Several text classification approaches have been developed to leverage this rich source of information. Notable approaches are mostly drawn from machine learning techniques such as SVM [1], ANN [2], Naïve-Bayes [3], Bayesian Network [4] and K-NN classifiers [5]. Over the years, the research on text classification has become more mature [18], and these techniques are applied only for webpages and other text documents. Traditional classification algorithms [1, 2, 3, 4, 5] follows *bag-of-Word* approach and accounts only for the term frequency in the documents and ignores semantic relationship between key words. To resolve this problem, is to enrich document representation with background knowledge represented by Ontology.

[1] http://www.blogpulse.com

Y. Wang and T. Li (Eds.): Knowledge Engineering and Management, AISC 123, pp. 487–497.
springerlink.com

Research has been done to exploit general ontologies such as WordNet [6, 7] and domain specific ontology [8, 9] for content based categorization of large corpora of documents. However, they all have limited coverage and are not regularly updated. To solve this, ontology terms could be enriched, and then it has its own drawback. While enriching original content with ontology terms may cause information loss.

To address the above issues, recent research [10, 11, 12, 13, 14] explored the usage of knowledge base derived from the Internet, such as Wikipedia. Wikipedia is based on wiki, where articles are registered and uploaded, and links are built in real time. Each wiki articles describes a single topic. It covers wide concepts and new domains. This wide coverage attracted the researchers' attention to treat Wikipedia as a knowledge base. In Wikipedia, the association relation between a concept and a category is communicated by a link, called *category link*. The category and the category link express its own direction towards its belonging concepts. The category system is edited and maintained by Wikipedia users as well as articles.

In our work, we first build the *concept by category* matrix using Wikipedia, which explicitly derives concept relationships. We then propose a framework which consists of two stages: first stage is to find the meaningful terms from blogposts to a unique concept as well as, disambiguate the terms belonging to more than one concept. The second stage is to determine the categories to which these found concepts appertain.

We summarize our contribution as follows:

- Unlike traditional classification techniques, our framework does automatic blog classification and does not use any manual training data to classify blogs.
- As Wikipedia refines the categories into narrow subcategories, our framework combines the use of vast number of these organized human knowledge.

The reminder of the paper is organized as follows: Section 2 defines the key terms used in our paper. Section 3 reviews the existing works on blog classification and Wikipedia. Section 4 describes our proposed framework, including n-gram based concept extraction, mapping terms to concepts, mapping concepts to categories and automatic blog classification. In Section 5, we evaluate the proposed framework with real data set and discuss experimental results. Finally, we conclude our paper in Section 6.

2 Definition of Terms

In this paper, we use these following terms:

- *"Blogpost (B)"* refers to single blogpost from the collection. Let χ be the blog data set, where $\chi = \{B_1, B_2, ..., B_n\}, n \in \mathbb{Z}^+$
- *"Term(T)"* refers to a meaningful word from a particular blogpost. Let D be the blogpost which consists of set of terms, where $B = \{T_1, T_2, ..., T_m\}, m \in \mathbb{Z}^+$
- *"Concept(C)"* refers to a Wikipedia article title. We treat each article title as a concept.
- *"Subcategories"* refers to Wikipedia's subcategories. Let *SC* consists of set of concepts, where $SC = \{C_1, C_2, ..., C_i\}, i \in \mathbb{Z}^+$
- *"Categories"* refers to Wikipedia's 12 main categories, where each category *Ct* consists of set of subcategories, $Ct = \{SC_1, SC_2, ..., SC_j\}\, j \in \mathbb{Z}^+$ and main parental categories (*Pcat*) where $Pcat = \{Ct_1, Ct_2, ...Ct_{12}\}$

3 Related Work

Recently there has been a huge interest in utilizing Wikipedia to enhance the text mining tasks such as text/web mining. To deal with Ontology coverage, research [10, 11, 12, 13, 14], explored the usage of knowledge base-Wikipedia derived from the Internet. Wikipedia has been used and demonstrated by researchers to improve the performance of text classifiers. Schonhofen et al. [10] exploited the titles and categories of Wikipedia articles to characterize documents. This method does not prove the relation between the input terms. In addition, it requires a lot of pre-processing of the Wikipedia articles themselves. In our work, we conceptualize that using this huge knowledge base-Wikipedia, the accuracy problem deriving from Natural Language Processing can be avoided. Wang et al.[11] and Syed et al.[12] constructed a thesaurus of concepts from Wikipedia and demonstrated its efficacy in enhancing previous approaches for text classification. Gabrilovich et al.[13] applied structural knowledge repository-Wikipedia as feature generation technique. Their work confirmed that background knowledge based features generated from Wikipedia can help text categorization. Shirakawa et al.[14] proved category system in Wikipedia is not in a tree structure but a network structure.

As blog classification is in the early stages of the research, there is no suitable way to identify and categorize blogposts. Hence it is an open problem for research [17]. A very recent work on blog classification is presented in [16, 19, 20, 21]. Qu et al. [19] proposed an approach to the automatic classification of blogs into four genres: personal diary, news, political, and sports. Using *tf.idf* [22] document representation and Naïve Bayes classification, this work [19] achieved the accuracy of 84%. The authors [16] compared the effectiveness of using tags against titles and descriptions. They trained a support vector machine on several categories and implemented a tag expansion algorithm to better classify blogs. Bayoudh et al. [20] have used the K-Nearest Neighbor (K-NN) algorithm for blog classification. This classifier was built by using Part-Of-Speech (POS) tagger knowledge. This work empirically states that, nouns are relevant for the determination of the documents meaning and topics. Elgersma [21] addressed the task of separating personal from non-personal blogs and achieved upto 90% classified scores.

4 Wikipedia Based Blog Classification Framework

This section introduces our framework (Figure 2) leveraging Wikipedia concept and category information to improve blog classification. We explain our proposed framework by defining *n-gram based concept extraction*, mapping terms to concepts, mapping concepts to categories and automatic blog classification respectively.

4.1 n-gram Based Concept Extraction

Our proposed method is a relatedness measurement method that computes the relatedness (among words) based on co-occurrence analysis. As a simple case, we map each word to a *concept*; the combination of these concepts/words can create a new *concept* (compound concept). Let us assume A is a word mapped to a *concept* C_1, B is a word mapped to a *concept* C_2, and ω is the set of extracted terms from blogpost.

Then there is a chance that the combination of *A* & *B* can produce a new concept (see Figure 1). Consider the following example,

$$\$s = \text{"}President\ of\ the\ United\ States\text{"}$$

$$\mathrm{X} = \text{"}President\text{"}\ , \mathrm{Y} = \text{"}United\text{"}, \mathrm{Z} = \text{"}States\text{"}, \omega = \{\mathrm{X},\mathrm{Y},\mathrm{Z}\}$$

$$\omega = \begin{cases} L^{*} - \left(\overline{\beta}\right); \omega \in C \\ else \qquad \phi \end{cases}$$

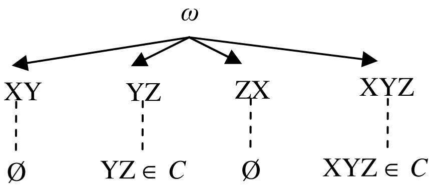


Fig. 1. Related measurement method

Where L^{*} is all possible combination of concepts and $\overline{\beta}$ is set of non-related terms, such as *XY* and *ZX*. *C* is the set of concepts such that *YZ* is mapped to *Country* concept and *XYZ* is mapped to *President* concept. Since the order of *X*, *Y* and *Z* can give different concept or no concept at all, we neglect the reordering. For effective classification, we mine existing knowledge bases, which can provide a conceptual corpus.

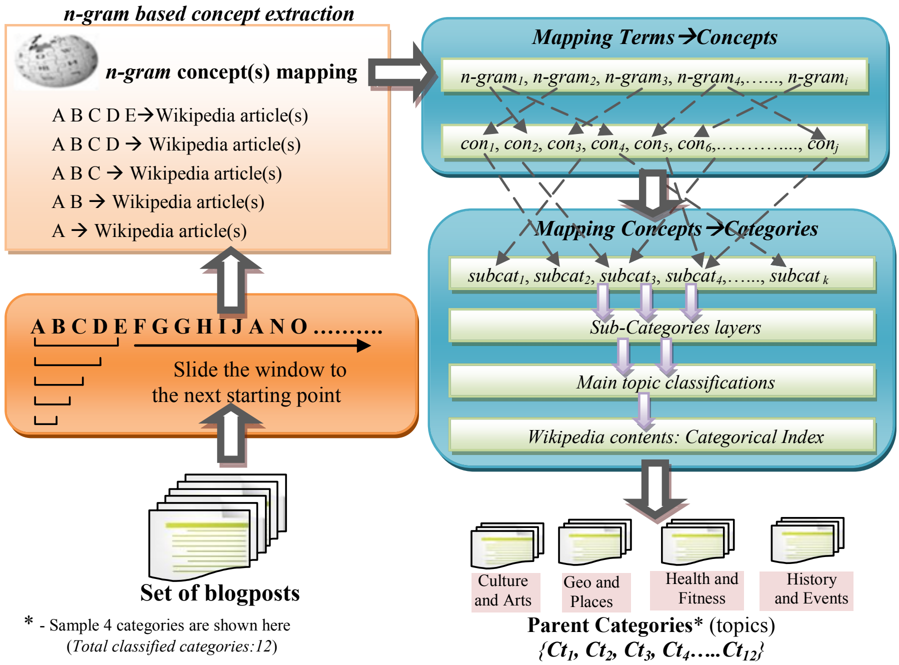

n-gram based concept extraction
n-gram concept(s) mapping
A B C D E→Wikipedia article(s)
A B C D → Wikipedia article(s)
A B C → Wikipedia article(s)
A B → Wikipedia article(s)
A → Wikipedia article(s)
A B C D E F G G H I J A N O ....
Slide the window to the next starting point
Set of blogposts
* - Sample 4 categories are shown here
(Total classified categories:12)
Mapping Terms →Concepts
n-gram1, n-gram2, n-gram3, n-gram4,……, n-grami
con1, con2, con3, con4, con5, con6,…………..., conj
Mapping Concepts →Categories
subcat1, subcat2, subcat3, subcat4,……, subcat k
Sub-Categories layers
Main topic classifications
Wikipedia contents: Categorical Index
Culture and Arts
Geo and Places
Health and Fitness
History and Events
Parent Categories* (topics)
{Ct1, Ct2, Ct3, Ct4…..Ct12}

Fig. 2. Wikipedia based blog classification Framework

Wikipedia is one of the best online knowledge base, to provide an individual English article with more than 3 million concepts. With that said, an important feature of Wikipedia is that on each wiki page, there is a URI, which unambiguously represent the concept. Two words co-occur if they appear in an article within some distance of each other. Typically, the distance is a window of *k* words. The window limits the co-occurrence analysis area of an arbitrary range. The reason of setting a window is due to the number of co-occurring combinations that becomes too huge when the number of words in an article is large. Qualitatively, the fact that two words often occur close to each other is more likely to be significant, compared to the fact that they occur in the same article, even more so when the number of words in an article is huge. As explained above, our concrete process flow algorithm (Algorithm 1) using *n-gram* model (n = 3) and Wikipedia is shown:

Algorithm 1. *n-gram based concept extraction*

```
Input:
  $w$ is the $WindowSize$ , and $max = n = 3$
Output:
  The concepts $W_i$
Description:
1: Set Scanning $WindowSize\ w = n$ , $MaxnGramsize = max$ , $i = 1$
2: Parse a given blogpost text at position $p=i$ until $p=EOF$
3: Extract word-sequence and set $W = w_i w_{i+1} \ldots.. w_{i+MaxnGramsize}$
4: If $W$ exists as a Wikipedia page (concept)
5:    Retrieve $W$ , set $p = i + MaxnGramsize$
6:    Insert $W$ into $W_i$
7:    Repeat from Step 3
8: else
9:    Set $MaxnGramsize = MaxnGramsize - 1$ & Update $W = w_i w_{i+1} \ldots.. w_{i+MaxnGramsize}$
10:If $Size(W) = 0$, then set $p = p + 1$; Goto Step 3
11:Goto Step 4
```

4.2 Mapping Terms→Concepts

To a certain degree, few *n-gram extracted terms* can map more than one *concept* in Wikipedia. This is called *concept disambiguation*. For example,

$$\$s = "CIA"$$

$$S = \{SC_1, SC_2, \ldots..SC_n\}; SC_i \in one\ or\ more\ Ct, i \geq 1$$

$$\$s \leq_m S_i; i \in \mathbb{Z}^+$$

where “CIA” can be mapped to several concepts that correspond to different subcategories like *Educational Institutes*, *Organizations*, *Airports*, and *Technology*. To address this issue, we use the disambiguation based on the context, i.e., each ambiguous concept in the text will be corresponding to a category, based on the scheme described in the following subsection. Then, for each ambiguous concept, we make a voting based on the majority from the “*Category histogram*” (Figure 3) . A *category histogram* is a distribution of the main categories and their number of appearances in a blogpost.

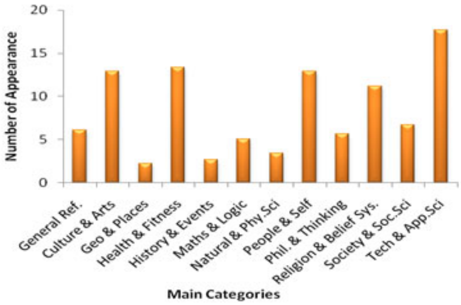


Fig. 3. Category Histogram

For example, if A is an ambiguous concept mapped to a set $SC = \{C_1, C_2, ..., C_i\}, i \in \mathbb{Z}^+$, then, we find the relevant main category (see Table 1 for the list of main categories) of each SC_j. Next we select the major category that has the highest frequency in the given blogpost. By doing this, we insist on selecting the major context category as the most relevant topic. We measure the influence of each individual post on the overall classification. To that end, we use our concept based weighting scheme [15]. In a blog, a post corresponds to concepts and category(s) and a common category clearly represent semantic associations between concepts. Therefore, extracting associations between concepts is achieved by extracting important categories in the blog by using *conf.idf* (equation 1). The importance of each concept in a blogpost (B) can be defined as follows:

$$Conf \cdot idf(C_k, B_j) = Conf(C_k + C_s, B_j).\log \frac{|N|}{N(C_k)} \qquad (1)$$

Where $Conf(C_k, B_j)$ denotes the number of times a concept C_k occurs in a blogpost B_j, and $N(t_k)$ denotes the number of blogposts N in which C_k occurs at least once.

4.3 Mapping Concepts→Categories

Since Wikipedia has a well-structured category system [11], nearly all concepts belong to more than one sub-category, and nearly all sub-categories belong to categories each other form a category mapping. Shirakawa et al. [14] have proposed concept vectorization methods. A vector value between two nodes on a graph, consisting of category network in Wikipedia, is defined based on the number of paths and the length of each path. It is true that the length of the hopcount (Figure 4) gives an intuitive measure on the belonging degree between a concept and a category. However, this is a very bias measure. Without generalization, a concept can be mapped to huge number of categories. In other words, concepts that appear in a blogpost are totally context dependent.

As mentioned in the below example (Figure 4), "Barack Obama" belongs to several categories, namely: *Society and Social Sciences*, *People and Self*, *Geography and Places*, and *General Reference*. For an educational blogpost discussing about "Columbia University", it is possible to find the previous concept several times (since Obama is an alumnus); therefore, it should not be classified into, or tagged as a "political" blogpost. As we can see from the Figure 4, the work proposed by [14] will assign a localized stronger belonging degree ("Barack Obama", "*Society and Social Sciences*") and ignores the underlying context.

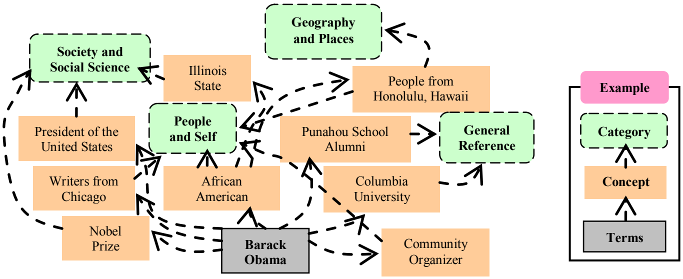


Fig. 4. An example: Wikipedia Sub-Category. A concept can belong to more than one Category.

We constructed *Concept* × *Category* matrix based on frequency of concepts that occur in a collection of categories, such that each row represents a *Category* and each column represents a *Concept.*

4.4 Automatic Blog Classification

Obviously, bloggers are free to post anything on their blogs. For example, a blog that extremely concerned about *entertainment*, and *music* can also have a post on a political issue. However, this blog must not be categorized as *politics*. Based on this observation, we measure the influence of each individual post on the overall classification. The frequency of the category is calculated based on the frequency of the belonging concept. If more than one concept is mapped to a blogpost, then the sum of the frequencies of these concepts is the category frequency.

Algorithm 2. Automatic blog classification

```
Input:
   Blogpost $B$
Output:
   Blogposts to Categories
Description:
1: $W_i$ = {Ct, $Wc$}      // $W_i$ =n-gram based concept extraction($B$), $Wc$ is the word count
2: Selectionsort ($W_i$)
3: post_length : = {$B$} //total number of words in blogpost $B$
4: Threshold = get (Empirical Threshold value) / 100 * post_length //
5: $\forall Ct \in W_i$
6:   If $Wc \geq$ Threshold
7:       $B \rightarrow Ct$
8:   Else
9:       Continue
10: End $\forall$
11:End
```

We explained our automatic blog classification through Algorithm 2 above. For a given blogpost W_i consists *of* the categories of the concept found, and the word count for each category. *Selection sort* is used to sort the W_i on descending order using word count (Wc). Threshold is calculated (Line 4) based on dividing the empirical threshold value by 100 and multiplying the blogpost length (post_length). Empirical threshold value was calculated using the number of lines for each individual document.

The threshold value or empirical value used would change, depending on the size of the document (for example: 100). There is no fixed threshold for the whole corpus, because small blogposts have less critical terms and large blogposts can fit into many categories. If word count is greater than the threshold, then blogpost belongs to the particular category. As blogpost can be of multicategories, this process continues until the word count is less than the threshold.

5 Experiments

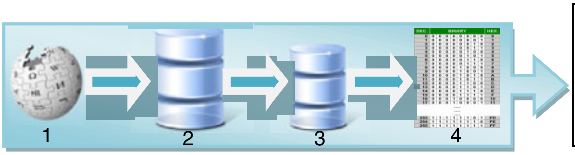
This section describes the experiment that test the efficiency of our proposed framework. We carried out experiments using part of TREC BLOGs08 dataset. We compared our framework that uses *conf.idf* with traditional *tf.idf* weighting scheme.

5.1 TREC Dataset

TREC BLOGs08[2] dataset is a well-known dataset in blog mining research area. This dataset consists of the crawl of Feeds, associated Permalink, blog homepage, and blogposts and blog comments. This is a sample of blogosphere crawled over from 14 January 2008 to 10 February 2009. The crawled feeds are mostly from *BlogSpot, LiveJournal, Xanga,* and *MSN Spaces*. The blog permalinks and homepages are encoded using HTML. This dataset is a combination of both English and Non-English blogs.

5.2 Wikipedia Data

We downloaded the Wikipedia database dumps[3] on 15 February 2010 and extracted 3,207,879 Wikipedia articles. After pre-processing and filtering (as illustrated in Figure 5), we used 3,101,144 article titles and are organized into 145,990 subcategories.



1. Downloading English Wiki dump,
2. Dump imported into My SQL database,
3. Extracting article titles and subcategories,
4. Mapping article titleà subcategories based on wiki's main category

Fig. 5. Stages in creating the knowledge base

5.3 Experimental Dataset Preparation

As our research focus is only on blog classification and not on language identifier, we filtered the blogs by English language. In order to extract the blogposts from blog documents, the HTML source code of blog pages should be parsed (which includes removal of HTML tags, scripts, etc.), and to be converted into plain text. We used blogpost extraction program named *BlogTEX*[4] to extracts blog posts from TREC Blog

[2] http://ir.dcs.gla.ac.uk/test_collections/blogs08info.html
[3] http://download.wikipedia.org
[4] http://sourceforge.net/projects/blogtex

dataset. We utilized the list of stop words in our program to identify the English blogs. Furthermore, as our dataset is huge and our motive is to test our framework, we collected around 41,178 blogposts for our blog classification. We treated blog title and blog content as blogposts in our experiment.

5.4 Experimental Results

Based on our concept weighting scheme (*conf.idf*) total of 155,980 *concepts* are retrieved by using our framework. Table 1 lists the results of our experiment that shows the *Categorical index* of Wikipedia, the number of blogposts classified, and number of concepts retrieved.

Table 1. Experimental Result

Categorical Index	Our framework Classification using *conf.idf*		Categorical Index	Our framework Classification using *conf.idf*	
	# of blogposts	# of concepts		# of blogposts	# of concepts
General Reference	878	3758	*Natural and Physical sciences*	1560	8530
Culture and Arts	8580	18245	*People and Self*	6279	10125
Geography and Places	4558	23220	*Philosophy and Thinking*	2345	7591
Health and Fitness	4082	21409	*Religion and Belief*	2345	8315
History and Events	2471	12746	*Society and Social sciences*	4657	13170
Mathematics and Logic	213	2561	*Tech and Applied sciences*	3210	26310

TREC BLOGs08 dataset was crawled during US election. During classification, we noticed that certain posts are classified on multi categories. Blogposts which is classified under *People and Self* category is also in *Society and Social sciences*. For example, a blogpost which discussed about Obama, (*People and Self*) has also discussed about the Democratic Party (*Society and Social sciences*). Majority of posts are classified under *Culture and Arts*, *Society and Social sciences*, *People and Self*, and *Tech and Applied sciences*. From our dataset very few blogposts are classified under *Mathematics and Logic* category.

5.5 Framework Comparison Based on Weighting Scheme

Feature selection and feature extraction plays an important role in identifying meaningful terms or concepts for blog classification. In our framework, we used *conf.idf* during feature selection and feature extraction process. Traditional text classification uses *tf.idf* weighting scheme and follows *bag-of-word* approach. Majority of research works in the area of text classification used Support Vector Machine (SVM). We compared our framework which uses *conf.idf* with the traditional SVM framework using *tf.idf* weighting scheme. We utilized the same set of 41,178 blogposts and partitioned into two sets: two-third blogs were used for training and the rest one-third for testing. The experiment was conducted based on the same 12 categories (Table 2), using SVM^{light} package[5]. We performed traditional pre-processing methods such as infectional stemming, stop word removal, and conversion to lower case for blogposts.

[5] http://svmlight.joachims.org

5.6 Performance Evaluation

The performance of blog classification can be measured in several ways. We use the standard definition of *precision,* and *recall* as performance measure to evaluate our framework with *tf.idf* based blog classification. *Precision* measures the percentage of *"categories found and correct"* divided by the *"total categories found"*. Recall measures the percantage of *"categories found and correct"* divided by the *"total categories correct"*. We evaluated the classification performance by collecting random 250 blogposts from the classified output respectively (shown in Table 2 below).

Table 2. Performance evaluation based on random 250 classified blogposts

Categorical Index	**Our classification framework using *conf.idf***		**SVM Classification framework using *tf.idf***	
	Precision	**Recall**	**Precision**	**Recall**
General Reference	0.787	0.791	0.719	0.621
Culture and Arts	0.761	0.700	0.655	0.548
Geography and Places	0.778	0.671	0.726	0.615
Health and Fitness	0.744	0.735	0.691	0.723
History and Events	0.732	0.732	0.663	0.640
Mathematics and Logic	0.764	0.738	0.730	0.598
Natural and Physical sciences	0.792	0.653	0.623	0.531
People and Self	0.847	0.729	0.715	0.692
Philosophy and Thinking	0.790	0.611	0.671	0.580
Religion and Belief	0.750	0.708	0.698	0.659
Society and Social sciences	0.806	0.742	0.728	0.620
Tech and Applied sciences	0.828	0.674	0.725	0.653
Average	**0.782**	**0.707**	**0.695**	**0.623**

Table 2 shows that our framework classification using *conf.idf* performs better than the SVM classification framework using *tf.idf* weighting scheme. For example, blogposts which discusses about "Iraq war", as per *conf.idf* was classified under *History and Events* (sub: Modern History) and *Society and Social sciences* (sub: Politics). When traditional classification *tf.idf* scheme is used, "Iraq war" blogpost was wrongly classified under *Geography and places* (sub: places). Our framework produced better precision (78%) and recall (70.7%) than the SVM classification framework.

6 Conclusion

In this paper, we presented an automatic classification of blogs using the *n-gram* technique to extract the possible concepts from the underlying posts. Wikipedia was used as an external knowledge base to map each n-gram concept to its corresponding categories. We conducted extensive experiments to measure the efficiency of our proposed framework. The experimental results proves that proposed system distinguishes blogs that belong to more than one category and has a better performance and success than the traditional SVM classification approaches.

References

1. Joachims, T.: Text categorization with Support Vector Machines: Learning with many relevant features. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 137–142. Springer, Heidelberg (1998)
2. Ng, H.T., Goh, W.B., Low, K.L.: Feature selection, perceptron learning, and a usability case study for text categorization. In: Proc. of ACM SIGIR, pp. 67–73 (1997)
3. McCallum, A., Nigam, K.: A Comparison of Event Models for Naïve Bayes Text Classification. In: AAAI 1998 Workshop on Learning for Text Categorization (1998)
4. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian Network Classifiers. Machine Learning 29, 131–163 (1997)
5. Yang, Y.: Expert network: Effective and efficient learning from human decisions in text categorization and retrieval. In: Proc. of ACM SIGIR, pp. 13–22 (1994)
6. Hotho, A., Staab, S., Stumme, G.: WordNet improves text document clustering. In: Proc. of ACM SIGIR (2003)
7. Budanitsky, A., Hirst, G.: Evaluating wordnet-based measures of lexical semantic relatedness. Computational Linguistics 32(1), 13–47 (2006)
8. Bloehdorn, S., Hotho, A.: Boosting for text classification with semantic features. In: Proc. of the MSW 2004 Workshop at the 10th ACM SIGKDD, pp. 70–87 (2004)
9. Jing, L., Ng, M.K., Huang, J.Z.: Knowledge-based vector space model for text clustering. KAIS 25, 35–55 (2009)
10. Schonhofen, P.: Identifying Document Topics Using the Wikipedia Category Network. In: Proc. of the IEEE/WIC/ACM International Conference on Web Intelligence, pp. 456–462 (2006)
11. Wang, P., Hu, J., Zeng, H.J., Chen, Z.: Using Wikipedia knowledge to improve text classification. KAIS 19(3), 265–281 (2009)
12. Syed, Z., Finin, T., Joshi, A.: Wikipedia as an Ontology for Describing Documents. In: Proc. of the AAAI International Conference on Weblogs and Social Media (2008)
13. Gabrilovich, E., Markovitch, S.: Overcoming the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge. In: AAAI (2006)
14. Shirakawa, M., Nakayama, K., Hara, T., Nishio, S.: Concept vector extraction from Wikipedia category network. In: Proc. of the ICUIMC, pp. 71–79 (2009)
15. Tahayna, B., Ayyasamy, R.K., Alhashmi, S.M., Siew, E.: A Novel Weighting Scheme for Efficient Document Indexing and Classification. In: 4th International Symposium on Information Technology, pp. 783–788 (2010)
16. Sun, A., Suryanto, M.A., Liu, Y.: Blog Classification Using Tags: An Empirical Study. In: Goh, D.H.-L., Cao, T.H., Sølvberg, I.T., Rasmussen, E. (eds.) ICADL 2007. LNCS, vol. 4822, pp. 307–316. Springer, Heidelberg (2007)
17. Ounis, I., Macdonald, C., Soboroff, I.: On the TREC BlogTrack. In: ICWSM (2008)
18. Mahinovs, A., Tiwari, A.: Text classification method review. Decision Engineering Report Series, pp. 1-13 (2007)
19. Qu, H., Pietra, A.L., Poon, S.: Automated Blog Classification: Challenges and Pitfalls. Computational Approaches to Analyzing Weblogs, pp. 184–186 (2006)
20. Bayoudh, I., Béchet, N., Roche, M.: Blog classification: Adding linguistic knowledge to improve the k-nn algorithm. In: Intelligent Information Processing, pp. 68–77 (2008)
21. Elgersma, E.: Personal vs non-personal blogs. In: Proc. of ACM SIGIR, pp. 723–724 (2008)
22. Salton, G., Buckley, C.: Term-weighting Approaches in Automatic Text Retrieval. Information Processing and Management 24(5), 513–523 (1988)