Finding Member Articles for Wikipedia Lists

Shuang Sun, Mizuho Iwaihara

Graduate School of Information, Production and Systems, Waseda University 2-7 Hibikono, Wakamatu-ku, Kitakyushu-shi, Fukuoka-ken, 808-0135 Japan
E-mail: sunshuang@asagi.waseda.jp, iwaihara@waseda.jp

Abstract The largest online encyclopedia Wikipedia holds a large number of lists created by human editors, where each list contains links to its member articles with its member description. Also the topic of the list is described in its list title and leading section. Lists may have very similar topics, and containment relationship exist between certain lists. Our objective is that for a given Wikipedia article and list, we determine whether the article can be added to the list. Its solution can be utilized on automatic generation of lists, as well as generation of categories based on lists, to help self-organization of knowledge structure. In this paper, we discuss building classifiers for judging on whether an article belongs to a list or not, where features are extracted from various components including list titles, leading sections, as well as texts of member articles. We also use articles from similar lists as negative examples. We report our initial evaluation results based on Bayesian and other classifiers, and also discuss feature selection.

Keywords automatic text classification, Bayesian classifier, TF-IDF, feature selection

1 Introduction

With the development of the information technology and the spread of the Internet, electronic information resource is increasing rapidly. By now, the static web page has already been more than 10 billion.

Besides, a number of libraries, publishers and information centers are managing their own document data bases, in order to provide fast and comprehensive access contents, where articles are indexed reflecting document structures such as body texts, abstracts, titles etc. Wikipedia relies on crowd sourcing where numerous contributors help mainlining the encyclopedia, although a number of tasks are automated. Also, Yahoo's well-known website hires more than 100 specialists from various fields to read, label and classify all the updated web pages. Although manual text classification is having a good quality, manual text classification has limits in scalability and cost. Text classification is the process of deciding whether the target article belongs to one certain class or not according to the content of it. Here the "target article" can be news, technology reports, emails, technology patents, web pages, e-books and so on. The word "class" can be referred to the topics (sports, economics, arts etc.), the style of the writing (romantic or reality), and the relationship between the target article and others from the class. Automatic text classification (ATC) is required in various tasks of management of document databases.

In this paper, we discuss maintenance of Wikipedia lists. A Wikipedia list is created with a list topic, and articles of that topic are manually added to the list. Wikipedia lists are kind of Wikipedia articles.

Newly created articles need to be added to appropriate Wikipedia lists to be effectively searched and discovered by users.

In this paper, we discuss construction of Wikipedia lists. When we search "list of " in Wikipedia, we obtain 1,382,427 lists. The core of the problem is finding member articles for given Wikipedia lists.

Table 1.1 Statistics of Wikipedia lists

Item	Articles
Wikipedia Lists	1,383,879

We define the Wikipedia list membership problem is to deciding a given article (target article) should belong to a given Wikipedia list (target list), where member articles of the list is given as positive examples. Member articles of different lists can be utilized as negative examples, but the choice of which lists should be sampled against the target list is open.

Finding member articles for Wikipedia lists is interesting on ATC and having real application. Obviously, identifying membership of Wikipedia articles is a sort of determining membership of a text to a class. Its solution can be utilized on automatic generation of lists, as well as generation of categories based on lists, to help self-organization of knowledge structure.

To describe the list in vector, we introduce four features. These features are keyword, title, metadata and common words. Keyword feature means that all words after pre-processing ranked by TF-IDF Value of which the doc set is the whole documents from positive and negative samples and the target document is the positive list.

In this paper, we choose to apply four classifiers to determine whether given an article belongs to a given Wikipedia list. They are Naïve Bayes, KNN, Decision tree and SVM.

In Section 2, related work on text classification is shown. In Section 3, we explain the technologies for feature selection and Bayes classifiers. Here we utilize the TF-IDF method to solve the problem. Firstly, the text can be represented by a vector consisting of a series of TF-IDF values of all the words from the article. After comparing the rank of the TF-IDF values, we choose top 20 words as the feature words of this article and drop other words. In this way, we effectively improve the correction rate of classification and reduce the training time for classifier.

In Section 4, we present a detailed architecture of our method. In Section 5, we carry out experiments and present evaluation results. In Section 6 and 7, we show a conclusion and future.

2 Related Work

From the viewpoint of text classification, various models have been proposed, such as SVM, Naïve Bayes, KNN, and Decision Tree and so on. In all, SVM shows generally superior performance, while other models have fairly equal performance. However, the result of the classifier significantly depends on document sets. Therefore, we generally agree that there is no unique best classifier for all data, there often exists most suitable classifier.

The Bayes method has been widely used on model recognition and model classification contributing to its probability theory basis and it achieves great success on image processing, decision support, live surveillance, especially in text classification. Ludovic [2] used advanced Bayes model on classifying semi-structured text, and achieved better result than SVM model. Compared with other models, the Bayes method is more simple and intuitive. It combines prior knowledge and posteriori knowledge together, in order to avoid intuition bias brought by prior information and the noise brought by only using sample data. Therefore, researchers have already done deep research on Bayes classification method.

As early as in 1968, Chow and Liu [3] proposed learning method for tree Bayes model; Cooper and Herskovits [4] came up with a K2 algorithm based on hill-climbing search strategy; Remco [5] used MDL instead of evaluation function to create a new K3 algorithm based on K2; Friedman [6] proposed the Structural EM algorithm combining Bayesian parameter learning and structural learning to optimize parameters of the models.

Beside the previous work, the Bayes classification method also affects greatly recently. Xiaoming Liu [7] and other three people proposed an improved FloatBoost algorithm for Naïve Bayes text classification called DifBoost, which combines Divide and Conquer Principal with the FloatBoost algorithm. Karl-Michael Schneider [8] showed that when all word frequency information is eliminated from the document vectors, the multinomial Naive Bayes model performs even better. In 2012, Swezey [9] improved Naive Bayes text classification to address the problem of classification by hierarchically structured topics with a large amount of classes.

Even though researches have been done deeply on Bayes classification method, there are still more work to do. Besides this, we still need to try other classifications to see which on performs better.

3 Basic Theory of feature selection and classifiers

Theoretically, more features mean that we obtain more detailed description of the document, and therefore we are able to provide stronger classification ability. But in the practical application, when using VSM to represent a document, the size of document vectors may reach 10 thousand attributions. For limited training samples, too many features will greatly affect not only the classifier learning efficiency, but also lead to excessive adaptation problems between classifier and training samples. Therefore, to improve the performance of document classification, we must solve feature dimension reduction, in order to meet the needs of the document classification.

Different from feature extraction that creates new features from functions of the original features, feature selection only drop redundant or irrelevant features.

3.1 Feature selection

In the paper, we choose TF-IDF to filter features. TF-IDF, short for term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.

TF means term frequency that is the number of times that term t occurs in document d.

$$\mathrm{tf}(t,d) = 0.5 + \frac{0.5 * f(t,d)}{\max\{f(t,d) : t \in d\}} \quad (1)$$

IDF means inverse document frequency that is the number of documents containing the term, and then taking the logarithm of that quotient.

$$\mathrm{idf}(t,D) = log^{N}_{|\{d \in D : t \in d\}|} \quad (2)$$

$$\text{TF-IDF} = \mathrm{tf}(t,d) * \mathrm{idf}(t,d) \quad (3)$$

3.2 Classifiers

In the paper, we use four different classifiers. They are Naïve Bayes, KNN, Decision tree and SVM.

Firstly, we describe a general algorithm based on Naïve Bayes classifier text classification. Interestingly, such probability method is currently effective and common known text document classification algorithm in all fields. This system was first proposed by Lewis (1991), Lang (1995) and Joachims (1996).

Broadly speaking, Naïve Bayes classifier (v_{NB}) is putting the word in the document into the class that makes the word has the largest probability of being observed here. It follows the usual Naïve Bayes independence assumptions. Independence assumed that.

$$P(a1, \ldots an) = \prod_1^n P(ai|vj) \quad (4)$$

And,

$$P(ai|vj) \approx \frac{nk+1}{n+|Vocabulary|} \quad (5)$$

$$P(vj) = \frac{|docsj|}{|Examples|} \quad (6)$$

In this setting, the probability of a word is independent of the position of another word appears in a location.

In the learning process, the algorithm makes analysis of all training documents, and extracts out all common word mark; then in a different target, calculate the probability of the word in the target class to obtain the necessary estimates. Later on, when given a new instance to be classified, the algorithm calculates the v_{NB} according to the function below.

$$v_{NB} = \underset{vj \in \{v1,v2\}}{\arg max} P(vj) \prod_1^n P(ai|vj) \quad (7)$$

Note that if any word appears in the new document but not appear in the training set, it will be simply ignored.

Naïve Bayesian model uses the most simple network structure, assuming that each of the variables X is conditionally independent from the class variable C, that is, each feature variable sees the class variable C as the sole parent node. Figure 3.1 visually describe the Structural attribution of Bayesian model.

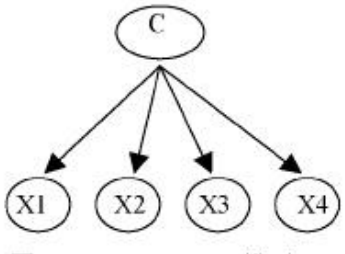

Figure 3.2 Naïve Bayes classifier

Besides Naïve Bayes classifier, here we also use k-nearest neighbor classifier (k-NN). In the example of filtering spams [10], features are extracted from the email social network and then these features are feed to k-NN classifiers to detect the potential spams. Here we just briefly discuss k-NN with an example.

Suppose we have a set of points in a metric space Ω, with each point assigned a label 0 or 1. Let (X_1 , Y_1), (X_2 , Y_2), . . . , (Xn ,

Yn) be a labeled sample and let (X, Y) be the query. In the k-nearest neighbor classifier, we predict the label of the query based on which class is more common among the k closest points to X in the labeled sample.

As for SVM (support vector machines), it is supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked for belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier.

Decision Tree is a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

The decision tree is a simple but effective learning method which takes specified attributes variables and outputs target values. It usually used in this way: the input variables are vectors we extract from each article while the target variable can take a binary value (1 for positive articles and 0 for negative ones) so that the decision tree becomes a classification tree.

4 Feature candidates

As mentioned in Section 1, our objective is to determine membership of a target article to a target list. Its solution is the basis of a broad range of applications such as automatic generation of lists, as well as generation of categories based on lists, and self-organization of knowledge structure. In this paper, to determine appropriate articles for the target list, we introduce four features, consisting of keyword, title, metadata and common words.

1) *Keyword feature* means that all words after pre-processing ranked by TF-IDF values in the corpus consisting of the target list as positive samples and several selected lists as negative samples. In Table 4.1, we list an example of top 15 keywords from our example target list "List of tallest buildings in Albuquerque".

Table 4.1 Top15 keywords for List of tallest buildings in Albuquerque

Keyword	TF-IDF
new mexico	0.03533
Albuquerque	0.02678
Tallest	0.02635
Timeline	0.02421
Stand	0.02371
1990	0.02299
Earli	0.02049
Indic	0.02049
Held	0.02049
Bank	0.01889
Titl	0.01785
Tower	0.01555
Feet	0.01417
Plaza	0.01286
Skyscrap	0.01272

2) *Title feature* captures words occurring in headlines (titles and sub titles) of the lists and those of member articles, where the words are weighted by TFIDF values, similarly to the keyword feature. Figure 4.1 shows an example of title and sub-title of our running example.

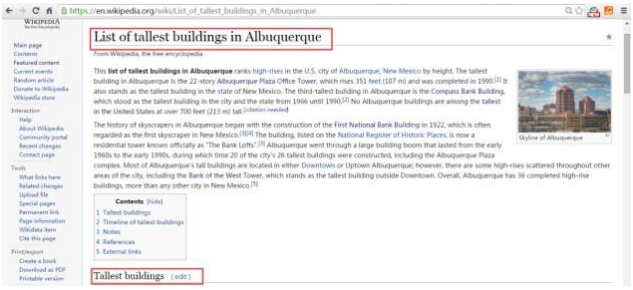

Figure 4.1Title and sub-titles of Wikipedia lists

3) *Metadata feature* captures words from the info-boxes of member articles of a list, weighted by TF-IDF values. An info-box in Wikipedia is a template used to collect and present a subset of information about its subject. It is a structured document containing a set of attribute–value pairs [11], and represents a summary of information about the subject of an article [12]. Info-box here compared with text, is semi-structured, and easy for developers to obtain desired information of subjects than text. Figure 4.2 is the info box of Wikipedia article Albuquerque Plaza.

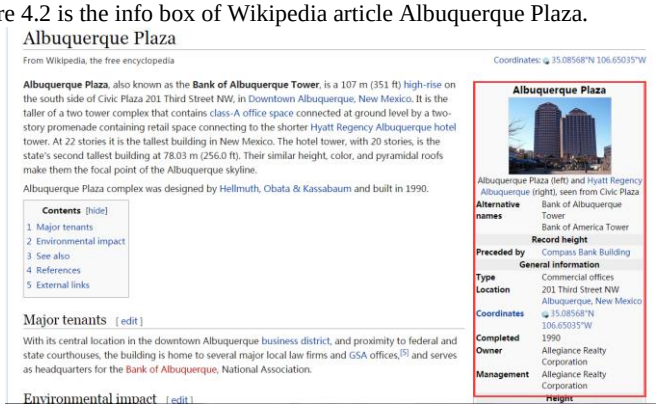

Figure 4.2 Info box of Albuquerque Plaza

4) *Common word feature* captures words occur both in titles and descriptions of the list and its articles. We can show a portion of common words in positive data set after filtering stop words: highrise, new mexico, height, tallest, albuquerque, state, construction and building.

5 Experiments and evaluation

5.1 Experiment settings

Wikipedia has over five million articles in English and is the largest free encyclopedia. Therefore, Wikipedia provide us with the large and accessible corpus in order to testify the feasibility and effectiveness.

We sample articles of training sets and the test sets are all from Wikipedia. Averagely, we choose 50 articles from one Wikipedia list as positive samples and 300 articles as negative samples.

Detailed experimental procedure is as follows:

(1) Data collection: crawl articles from the Wikipedia websites by web spider.

(2) Preparing the data set: after pre-processing articles, parse the text file into the entry vectors.

(3) Evaluating features: calculate value of the four features.

(4) Training classifiers: We utilize the classifiers listed in Table 5.3.

(5) Evaluations: Use the trained classifiers to calculate the error rate.

After selection of Step 3, a part of the word vectors (keyword feature) of the lists are shown in Table 5.1

Table 5.1 part of the result of feature selection

List	Keyword Feature
List of tallest buildings in Albuquerque	High-rise, construct, build, Albuquerque, new, mexico
1st Academy Awards	Film, 1st, award, best, academy, known
List of Bay Area Rapid Transit stations	Bay, street, san, francisco, station, transist

5.2 Datasets and experiments

Experiment 1

In Experiment 1, the positive and negative lists we choose are shown in Tables 5.2 and 5.3, respectively.

The positive list we choose is "List of tallest buildings in Albuquerque". It contains 50 articles including "Albuquerque Plaza Office Tower", "Hyatt Regency Albuquerque", "Compass Bank Building" and so on. From the perspective of words, it is 10,050 words in total.

As for negative data, we choose two similar lists, "List of University of New Mexico buildings" and "List of tallest buildings in Tokyo". At the same time, in order to enlarge the size of negative data, we choose another 227 articles containing word "building" by search function of Wikipedia website.

Table 5.2 Positive list for Experiment 1: "List of tallest buildings in Albuquerque"

Member articles	Amount
Albuquerque Plaza Office Tower	50 articles
Hyatt Regency Albuquerque	
Compass Bank Building	
Albuquerque Petroleum Building	10,050 words
Bank of the West Tower	
……	

Table 5.3 Negative dataset for Experiment 1

Negative data	Amount
Articles from List of University of New Mexico buildings	300 articles
Johnson Gymnasium(in Albuquerque, New Mexico)	72,894 words
Articles from List of tallest buildings in Tokyo	
other 227 articles containing word "building"	

After we evaluate the four features, we carry out classification experiments using the four classifiers. The result of Experiment 1 is shown in Table 5.3.

Table 5.3 Result of Experiment 1

Method	Precision	Recall	F1
KNN	0.94	0.93	0.93
Naïve Bayes	0.87	0.8	0.84
Decision tree	0.8	0.8	0.8
SVM	0.84	0.83	0.84

Table 5.3 shows that in Experiment 1, among the four classifiers, KNN performs better than others on this time's data set, and is followed by both Naïve Bayes and SVM, and Decision tree's performance is the fourth.

Experiment 2

Positive and negative lists for Experiment 2 are shown in Tables 5.4 and 5.5, respectively.

In Experiment 2, articles from "1st academy awards" are chosen as positive data. It contains 72 articles, that is 209,568 words.

Negative data includes articles from "59th academy awards" and articles containing "film" that we collected by search function of Wikipedia website. In total, it's 432 articles, as well as 1,213,489 words.

Table 5.4 Positive dataset for Experiment 2

Positive data	
Articles from 1st academy awards	72 articles 209,568 words

Table 5.5 Negative dataset for Experiment 2

Negative data	
Articles from 59th academy awards (256 articles)	432 articles 1,213,489 words
176 articles containing "film" that we collected by search function of Wikipedia	

After training and test, we obtained the result shown in Table 5.6.

Table 5.6 Result for experiment 2

Method	Precision	Recall	F1
KNN	0.85	0.84	0.84
Naïve Bayes	0.8	0.8	0.8
Decision tree	0.75	0.46	0.57
SVM	0.78	0.81	0.79

We can see from the results that precision is lower than Experiment 1. However, KNN still performs better than other classifiers.

As for reasons, this is a description list. It contains articles about films, directors, actors, musicians and so on. Some articles do contain "academy awards" but not clarify the order of session.

Therefore, many wrong cases are separated into the positive part, causing the precision to reduce.

6 Conclusion

In this paper, we discussed the membership problem of Wikipedia articles to lists. We applied four different classifiers to decide whether the target article belongs to one certain list or not and comparing performance among these classifiers. We performed the hold out cross validation to evaluate the precision and recall of the results. From the results, we can see that our method based on four features has relatively acceptable performance. Also, the KNN classifier performs better than others on our particular data sets. However, we need further evaluations to confirm this tendency. Nevertheless, our method has a major significance because this is the first step to solve this type of Wikipedia knowledge base enrichment. After extensions, it can be applied to various practical problems.

7 Future work

The future objective of our work is to construct Wikipedia featured lists automatically. Here, featured lists means that they are excellent on prose, lead, comprehensive, structure, style and stability [13].

In the near future, we plan to design more sophisticated linguistic and structural features for improving accuracy. Also, we plan to evaluate on different training sets that can lead to different result. For this purpose, choosing appropriate positive and negative samples requires a careful consideration, as well as trying more different types of classifiers are necessary.

Reference

[1]. Wikipedia, https://en.m.wikipedia.org

[2]. Denoyer L, Gallinari P. Bayesian network model for semi-structured document classification. Information Processing and Management, 2004, 40(5):807-827.

[3]. Chow C K, Liu C N. Approximating discrete probability distribution dependence trees. IEEE transactions on Information Theory, IT-14(3): 1968.462-468

[4]. Herskovits E. Computer-based probabilistic-Network Construction. USA: Stanford University, 1991

[5]. Bouckaert R. A Stratified Simulation Scheme for Inference in Bayesian Belief Networks. In Uncertainty in AI, Proceedings of the Tenth Conference, 1994, 110-117

[6]. Friedman N. The Bayesian structural EM algorithm. In Uncertainty in Artificial Intelligence (UAI). 1998 129-138.

[7]. Xiaoming Liu, Jianwei Yin, Jinxiang Dong, Memon Abdul Ghafoor. An Improved FloatBoost Algorithm for Naïve Bayes Text Classification. USA: Lancaster University, 2005

[8]. Karl-Michael Schneider. On Word Frequency Information and Negative Evidence in Naive Bayes Text Classification. 2004

[9]. Wikipedia, https://en.wikipedia.org/wiki/Feature_selection

[10]. Wu, Chih-Hung. "Behavior-based spam detection using a hybrid method of rule-based techniques and neural networks." Expert Systems with Applications 36.3 (2009): 4321-4330.

[11]. Baeza-Yates, Ricardo; King, Irwin, eds. (2009). Weaving services and people on the World Wide Web. Springer. ISBN 9783642005695. LCCN 2009926100.

[12]. Yu, Liyang (2011). A Developer's Guide to the Semantic Web. Springer. doi:10.1007/978-3-642-15970-1. ISBN 9783642159695.

[13]. https://en.wikipedia.org/wiki/Wikipedia:Featured_list_criteria