Classification of Concept Maps Using Bag of Words Model

KRUNOSLAV ZUBRINIC, MARIO MILICEVIC, IVONA ZAKARIJA
University of Dubrovnik
Department of Electrical Engineering and Computing
Cira Carica 4, Dubrovnik, CROATIA
{krunoslav.zubrinic, mario.milicevic, ivona.zakarija}@unidu.hr

Abstract: Concept map is a graphic organizer that depicts relationships among concepts. They are used for organizing and representing information in different areas. In environments where many persons or applications create concept maps, contents and quality of these maps vary. When user wants to find specific information, they have to know to which domain that map belongs. As manual categorization can be a long and demanding procedure, automatic classification of concept maps can help users to identify the relevant map. In this paper we explore the possibility of automatic classification of concept maps using the simple bag of words model. We evaluate performance of two naïve Bayes classifiers against SVM.

Key–Words: Concept map, classification, data mining, text mining, naïve Bayes, SVM

1 Introduction

Concept map (CM) is a diagram which shows various relationships among concepts. As knowledge representation tool they have been successfully used for organizing and representing information in different areas, including education, knowledge management, data modeling, business and intelligence. It includes concepts, usually labeled by nouns or noun phrases, and relationships between them indicated by a line linking two concepts [1, 2].

There has been a remarkable growth in the use of CMs across the world over the past decade. During concept mapping process, the creator constructs a two dimensional representation of concepts and their relationships. That flexibility in constructing of CMs is commonly regarded as an advantage of concept mapping for use in many fields, as the map created reflects what the creator knows of the subject field. Given that each person's understanding of a domain is different, even if people construct CMs on the same topic, the maps constructed by individuals are different [3].

In environments where many persons or different software applications [2] represent information in a form of a CM, contents and the quality of these maps vary. When user wants to access elements of some map, they have to have some sense of the map's scope. It is very difficult and time consuming for them to have insight in all the details of every map, and it would be useful if the system could automatically give information on the scope of every CM. That scope can be recognizable from CM's semantic tags such as title, subject or description. This problem is much bigger in multilingual environments where maps can be created using terms written in different languages.

As manual categorization can be a long and demanding procedure, automatic classification of CMs can help user to select and identify topically relevant CMs. Possible application of that procedure includes assessing CM similarity, structuring and facilitating access to CMs, or automatic proposing and finding additional materials that could be included in existing CM.

The automatic classification of CMs has been studied by several researchers who are focused on development and evaluation of a tool for automatic classification of CMs based on a topological taxonomy [3, 4], similarity of concepts [5] or determining differences between groups of maps based on connections among concepts [6]. In text categorization, several researches treat document as a collection of concepts, rather than independent words [7, 8].

In this research we argue that it is possible to classify CMs if we treat them as a flat, nonhierarchical map of concepts. We use simple bag of words model, which simplifies representation of a CM as an unordered collection of concepts' labels.

2 Problem formulation

Semantic tags, such as title, subject, description and language can help a user understand the scope and structure of the CM. We made a brief analysis of these elements in a set of 600 randomly selected CMs retrieved from public CMAP servers[1].

As correct title we count one that is entered in a CM, and the meaning of which is connected with content of a CM. Titles filled with author's name or text such as "Untitled" or "0" are counted as incorrect. As shown in Fig. 1, creators of a CM frequently fill only title tag, in 92% of observed CMs, while other tags, such as description and keywords, are filled very rarely, in 9% and 15% of maps (respectively). One of the reasons why the title is entered in so many maps is that the title is required in all concept mappings application. We also checked whether the CM is labeled with the correct language, and found rather bad correctness, as language is correct in 76% of the set. The reason for that is that some creators, when using concept mappings applications, leave English as default language, regardless of the language used .

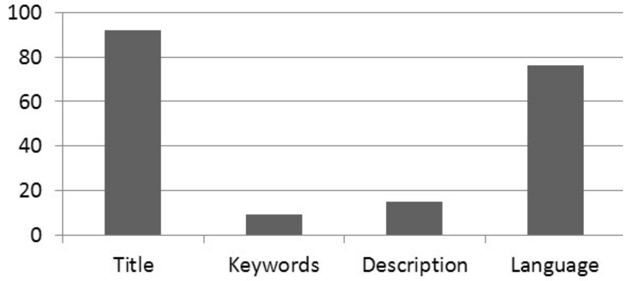

Fig. 1: Percentage of meta tag entry in randomly selected CMs

Although that analysis is not very detailed, on the basis of it we have come to the conclusion that meta tags of CMs in many cases are not detailed enough to identify maps by content.

Automatic classification is a learning process during which a program recognizes the characteristics that distinguish each category from others and constructs a classier when given a set of training examples with class labels. Application of this approach to the CMs can help in automatic categorizing of maps on the basis of similarity of their content. In that way it is possible to reduce the drawbacks of manual tagging.

In this research we classify CMs using simple bag of words approach successfully used in classification of text documents. Using that approach we simplify representation of a CM as an unordered collection of concepts' labels. Algorithms used in classification are naïve Bayes [9] and Support Vector Machine (SVM) [10] that were successfully used in previous researches in text classification.

3 Classification methods

3.1 Naïve Bayes

Naïve Bayes [9] is simple Bayesian supervised classifier that assumes that all attributes are independent of each other given the context of the class. This is the so-called naïve Bayes assumption which is rarely true in the most real-world situations. Despite this, naïve Bayes often performs classification very well [11]. Because of that assumption, the parameters for each attribute can be learned separately, and this greatly simplifies learning, especially when the number of attributes is large.

Documents classification is an example of a domain with a large number of attributes. Those attributes are words, and the number of different words in a document can be large. Naïve Bayes has been successfully applied to document classification in many researches [9, 11–13]. In our research we use multivariate Bernoulli and multinomial model of naïve Bayes.

In both models, a document is shown as an ordered sequence of word events, drawn from the same vocabulary. Assumption is that the probability of each word event in a document is independent of the word's context and position in the document. Thus, each document is represented in the form of bag of words. Classification is based on naïve Bayes assumption that the probability of each word occurring in a document is independent of the occurrence of other words in a document; therefore the probability of a document given its class is product of the probability of the attribute values over all word attributes. Given estimates of parameters calculated from the training documents, classification can be performed on test documents by calculating the posterior probability of each class given the evidence of the test document, and selecting the class with the highest probability. Further description of naïve Bayes models used in this paper is shown in [9].

In the multivariate Bernoulli model, a document is represented with binary vector of words. Each dimension of the space corresponds to word vocabulary. Dimension of the vector for document is either 0 or 1, indicating whether the word occurs at least once in the document. Unlike that, multinomial model of naïve Bayes uses representation of a document as a vector of word occurrences. Information on frequency of each word can help in classification.

3.2 Support Vector Machine

SVM [10] is a supervised learning algorithm for classification problems. It is based on the structural risk

[1] Public CMAP servers used in this study can be accessed at http://cmapspublic.ihmc.us/, http://cmapspublic2.ihmc.us/ and http://cmapspublic3.ihmc.us/

minimization principle from computational learning theory. The idea is to find a hypothesis for which we can guarantee the lowest true error. The true error of a hypothesis is the probability that this hypothesis will make an error on an unseen and randomly selected test example. One important property of SVMs is that their ability to learn can be independent of the dimensionality of the feature space. SVMs measure the complexity of hypotheses based on the margin with which they separate the data, and not the number of features [14].

Topic identication with SVM implies a kind of semantic space in the sense that the learned hyper plane separates documents which belong to different topics in the input space. When learning text classifiers, one has to deal with a great number of features [15]. One way to avoid high dimensional input spaces is to assume that most of the features are irrelevant. Unfortunately, in text categorization there are only few irrelevant features. Results of researches show that even features ranked lowest still contain considerable information and are somewhat relevant [14].

Since SVMs use over fitting protection, which does not necessarily depend on the number of features, they have the potential to handle large feature spaces. Because of that characteristic, this method is suitable for text classification [8, 14, 16].

4 Experiment

Data for classification consists of a set of CMs which were retrieved randomly from public IHMC CMAP servers, using SOAP web services. Fetched documents were in CXL format based on XML [17]. Data preparation was performed using Python scripts and training and classification was performed using WEKA workbench [18]. We evaluated the performance of the naïve Bayes classifiers by comparing them against SVM.

4.1 Retrieving and filtering data by language

CMs retrieved from public servers are created using different languages. As CXL format has attribute ”language“ that should be used for labeling the original language of the map, we hypothesized that value of this element could be used to distinguish maps by language. We encountered problem of many CM creators not using that element, and leaving English as the default language, although they write concept labels in other languages. As seen in the results of our preliminary research shown in Fig. 1, almost 27% of maps have incorrect value of this element. For this reason we have to filter maps, based on language in some other way.

We decided to use very simple solution for language detection based on the list of the most commonly used words in the English language [19]. According to statements by Zip’s law, the frequency of any word in some corpus of natural language is inversely proportional to its rank in the frequency table [20]. As stated in [21], the first 100 of the most frequent words are found in about one-half of all written material, while the first 300 make up about 65% of all written material in English. Our method uses simple binary classifier that has to decide whether a CM should be in the result set of a maps written in English or not, and it is suitable for use on the set of short texts, such as content of a CM.

We hypothesized that a CM is written in English language if, observing a subset of the 50 most common words in the map, at least five of them are found in a set of 500 most common words in English. Test group of CMs included maps written in Croatian, Dutch, German, English, Finnish, French, German, Italian, Polish, Portuguese, Slovak, Spanish and Swedish language. Algorithm was taking full form of individual words, and the data were taken from all elements of the map: title, description, keywords, concepts and relationships. The results of classification are shown in Table 1.

Table 1: Results of classification of CMs based on language

CMs in English language		Correct language	
		YES	NO
Decision of algorithm	YES	92.18%	7.82%
	NO	1.00%	99.00%

Used algorithm allowed 7.82% of the maps written in other languages, while it rejected only 1% of the correct CMs written in English language. These results are slightly lower than the results reported in other researches for English language [19]. The reason for that is that the test set contains some maps where the majority of concepts are written in other languages (e.g. Latin) with few very common English words. The classifier put those maps in a group of maps that have been written using English, although it is not true. Taking into account the simplicity of the used method, we rated these results as satisfactory.

Due to the lack of a larger test set, we believe that actual implementation would benefit from language detection methods that make decision about language based on statistical properties of English language.

The same method can be used for classification of CMs based on any language, provided one has the list of the most frequent words of a specific language.

4.2 Preprocessing

Furthermore, we selected documents so that each document has only one class. In order to assess the classifier's performance, we performed initial manual categorization of the maps to seven different categories: business (a), environment (b), human (c), IT (d), learning (e), society (f) and science & technology (g). All maps that do not fit in those categories were dropped. In the end, we got representation of 524 CMs written in English language.

From the set of CMs in the English language, we extracted labels of all concepts, and represented each map by an array of words. We converted all letters to lower case and removed all words without linguistic meaning using the list of stop words in English language [22].

Since some words carry similar meanings but in different grammatical form, it was necessary to combine them into one attribute. Words in a set were reduced to their basic form using Porter's suffix-stripping algorithm [23]. In this way we reduced a number of attributes, but kept the number of their occurrences. Created sets could show a better representation of these terms and the dataset was reduced for achieving faster processing time.

As a final phase of data preprocessing, we created files in attribute-relation file format (arff) for use in training and classification with WEKA machine learning software.

4.3 Training

Feature selection is classic refinement method in classification. It is an effective dimensionality reduction technique to remove noise feature [15]. In general, the basic idea is to search through all possible combinations of attributes in the data to find which subset of features works best for prediction. Removal is usually based on some statistical measures, such as document frequency, information gain, χ^2 or mutual information [24].

In this research, all training documents were categorized in seven different categories, and the model computes which terms frequently occurred in such category. Using this approach, some useless or irrelevant attributes can be filtered out. In order to achieve better performances, we decided to use χ^2 test as feature selection algorithm. After feature selection we performed classification using set of 8990 attributes.

In a SVM method as a learning algorithm we decided to use Sequential minimal optimization (SMO). That algorithm is conceptually simple, generally faster and has better scaling properties for SVM problems than the standard SVM training algorithm [25]. Observing results of training of SVM classifier we noticed that we could achieve slightly better results if we used binary attributes, and not occurrences of each word.

4.4 Classification

The performance of multivariate Bernoulli model of the naïve Bayes classifier was evaluated by comparing it against multinomial naïve Bayes classifier and SVM trained using SMO algorithm. Results were calculated as average of 10 experiments using 10-fold cross-validation.

We have conducted two experiments. Data set used in the first experiment contained a full set of 8990 attributes and 524 instances. Observing results of the first experiment, we noticed that there are some maps where same words have an unexpectedly large number of occurrences. We checked source documents and found that those CMs deviate from the recommendations to produce quality maps (e.g. map is not created around clear focus question; number of concepts in a map is larger than 30; same concept is used multiple times in one map or concept's labels are very long [1]). For the second experiment we considered these maps as outliers and we removed them from the set.

4.5 Evaluation

Since we constructed data sets so that each CM had single class label, we were able to perform classification experiments where each document is classified in only one of the classes. As the performance measure we used classification effectiveness using F_α measure for $0 \leq \alpha \leq 1$. This measure is defined as:

$$F_\alpha = \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}} \quad (1)$$

where α is a relative degree of importance attached to precision (P) and recall (R). They are common measures in machine learning, and they are defined as:

$$P = \frac{TP}{TP+FP} R = \frac{TP}{TP+FN} \quad (2)$$

where True and False positives (TP/FP) refer to the number of predicted positives that were correct/incorrect, and similarly, True and False Negatives (TN/FN) refer to the number of predicted negatives that were correct/incorrect. These values are taken from contingency table.

In our research, precision and recall are equally important, so we used value $\alpha = 0.5$.

5 Results and discussion

Results of classification are calculated as average of 10 experiments using 10-fold cross-validation. We have conducted two experiments. Data set used in the first experiment contained a full set of attributes and instances. For the second experiment we removed some instances that were recognized as outliers.

The classification effectiveness calculated for each class and weighted averages of all classes calculated in the first experiment are shown in Fig. 2.

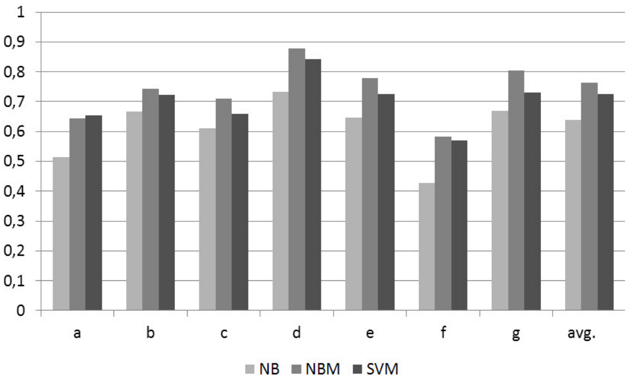

Fig. 2: Effectiveness of classification on full set

In the most classes the best results are achieved using multinomial naïve Bayes (NBM) classifier that correctly classifies 76.72% of all instances. SVM has achieved the best results in only one class (business) with 72.14% of correctly classified instances, while the Bernoulli naïve Bayes classifier (NB) achieved the worst results in all classes with only 63.74% of correctly classified instances.

The second experiment was conducted after removal of outliers on a reduced set of instances. In this experiment the results of all classifiers were slightly better. Those results are shown in Fig. 3.

In all classes the best results were achieved using multinomial naïve Bayes classifier that correctly classifies 79.44% of all instances. Bernoulli naïve Bayes classifier with 67.14% acquired better results than in previous experiment, and in two classes (environment and human) results were even better than results of SVM classifier. SVM correctly classified 72.58% instances, and it achieved only minimum improvement over the previous experiment.

The results show that for classifying of CMs multinomial naïve Bayes classifier that takes into account the number of occurrences of attributes in the set is a good choice. Classification using a reduced set of instances gives better results than the classification with full set of occurrences. Bernoulli naïve Bayes classifier achieved slightly lower results because we used large number of attributes. This algorithm calculates probability counting only appearance of attributes in the document, and not the number of their occurrences. Because of that, it is rather sensitive to the appearance of many attributes that are not important for classification.

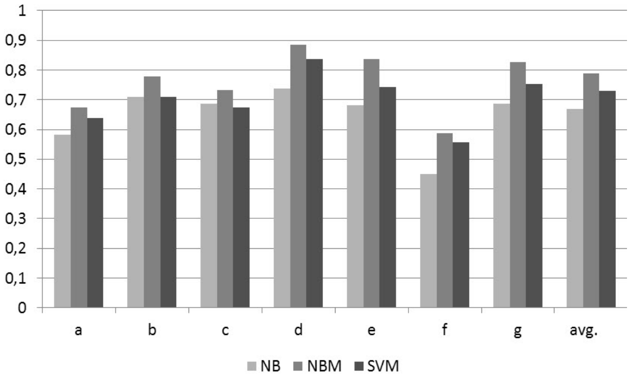

Fig. 3: Effectiveness of classification on reduced set

Correct classification of 79.44% of all instances in the best case can be considered a relatively good result, although further improvements are certainly possible. With bigger number of CMs, algorithms are likely to show better results. Further reduction of attributes using feature selection algorithm through series of testing and evaluation cycles could probably improve results of algorithms that do not use number of attribute's occurrences. As majority of CMs used in this research have a topological organization, we assume that further improvement of the results could be achieved by assigning weighting tags to concepts, depending on their hierarchical level. Another way that could improve the classification results is use of some linguistic tools and techniques such as connecting words with their synonyms. This aims at achieving robustness with respect to linguistic variations such as vocabulary and word choice.

6 Conclusion

In this research we tested the ability of classification of CMs using simple classifiers and bag of words approach that is commonly used in document classification. In two experiments we compared the results of classification randomly selected CMs using three classifiers.

The best results are achieved using multinomial naïve Bayes classifier. On reduced set of attributes and instances that classifier correctly classified 79.44% of instances. We believe that the results are promising, and that with further data preprocessing and adjustment of the classifiers they can be improved.

References:

[1] J. D. Novak and A. J. Cañas, The theory underlying concept maps and how to construct and use them, Tech. Rep., Rev 01-2008, IHMC, 2008.

[2] K. Zubrinic, D. Kalpic, and M. Milicevic, The automatic creation of concept maps from documents written using morphologically rich languages, *Expert Systems with Applications*, vol. 39, no. 16, pp. 12709–12718, 2012.

[3] A. Valerio, D. B. Leake, and A. J. Cañas, Automatic classification of concept maps based on a topological taxonomy and its application to studying features of human-built maps, in *Proceedings of the 3rd International Conference on Concept Mapping*, 2008.

[4] A. A. Kardan, F. Hendijanifard, and S. Abbaspour, Ranking concept maps and tags to differentiate the subject experts in a collaborative e-learning environment, in *Proceedings of the 4th International Conference on Virtual Learning*, pp. 308–315, 2009.

[5] D. B. Leake, A. G. Maguitman, and A. J. Cañas, Assessing conceptual similarity to support concept mapping, in *Proceedings of the 15th International Florida Artificial Intelligence Research Society Conference*, pp. 168–172, 2002.

[6] S. K. Hui, Y. Huangy, and E. I. George, Model-based Analysis of Concept Maps, *Bayesian Analysis*, vol. 3, no. 3, pp. 479–512, 2008.

[7] L. Cai and T. Hofmann, Text categorization by boosting automatically extracted concepts, in *Proceedings of the 26th annual ACM SIGIR*, pp. 182–189, 2003.

[8] M. Sahlgren and R. Cöster, Using bag-of-concepts to improve the performance of support vector machines in text categorization, in *Proceedings of the 20th International Conference on Computational Linguistics*, 2004.

[9] A. McCallum and K. Nigam, A comparison of event models for naïve Bayes text classification, in *Proceedings of AAAI-98 workshop on learning for text categorization*, pp. 41–48, 1998.

[10] C. Cortes and V. Vapnik, Support-vector networks, *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[11] I. Rish, An empirical study of the naïve Bayes classifier, in *IJCAI Workshop on Empirical Methods in Artificial Intelligence*, 2001.

[12] K.-M. Schneider, Techniques for improving the performance of naïve Bayes for text classification, in *Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Text Processing*, pp. 682–693, 2005.

[13] T. M. Mitchell, *Machine Learning*. McGraw Hill, 1997.

[14] T. Joachims, Text categorization with suport vector machines: Learning with many relevant features, in *Proceedings of the 10th European Conference on Machine Learning*, pp. 137–142, 1998.

[15] P. Domingos, A few useful things to know about machine learning, *Communnications of ACM*, vol. 55, no. 10, pp. 78–87, 2012.

[16] E. Leopold and J. Kindermann, Text categorization with support vector machines. How to represent texts in input space?, *Machine Learning*, vol. 46, pp. 423–444, 2002.

[17] A. J. Cañas et al.,KEA: A knowledge exchange architecture based on web services, concept maps and CmapTools, in *Proceedings of the 2nd International Conference on Concept Mapping*, pp. 304–310, 2006.

[18] M. Hall et al., The Weka data mining software: An update, *SIGKDD Explorations*, vol. 11, no. 1, 2009.

[19] W. B. Cavnar and J. M. Trenkle, N-gram-based text categorization, in *Proceedings of 3rd Symposium on Document Analysis and Information Retrieval*, pp. 161–175, 1994.

[20] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*, Cambridge University Press, 1999.

[21] E. B. Fry and J. E. Kress, *The Reading Teacher's Book Of Lists*. John Wiley & Sons, 2006.

[22] List of stop words in english language. Online. ftp://ftp.sunet.se/pub/unix/databases/full-text/smart/english.stop (11th March 2013).

[23] M. F. Porter, An algorithm for suffix stripping, *Program*, vol. 14, no. 3, pp. 130–137, 1980.

[24] Y. Yang and J. O. Pedersen, A comparative study on feature selection in text categorization, in *Proceedings of the 14th International Conference on Machine Learning*, pp. 412–420, 1997.

[25] J. C. Platt, Sequential minimal optimization: A fast algorithm for training support vector machines, Tech. Rep. MSR-TR-98-14, Microsoft, 1998.