Comparison of text classification methods supporting journalistic data organization

Barbara Sikora
University of Applied Sciences
Upper Austria, Campus
Hagenberg
Softwarepark 11, 4232
Hagenberg, Austria
S1510629017@students.fh-hagenberg.at

CHI'14, April 26–May 1, 2014, Toronto, Canada.

DOI string from ACM form confirmation

Abstract
UPDATED—January 12, 2017. In times of information overload, it is particularly important to have access to structured information and the possibility to get an quick overview about a topic. To reach these goals, techniques of information extraction and text classification are needed. This survey gives an short overview about the topic of text classification, already existing projects and the different approaches of algorithms used for data organization. It also provides an short introduction to the main challenges of the related project.

Author Keywords
Classification Problem; KNN; Naive Baiyes; Semantic Fingerprinting; SVM; Rocchio; Semantic Text Analysis; Cortical.io

ACM Classification Keywords
I.2.7 [Natural Language Processing]: Text Classification

Introduction
These days there are multiple forms of language circulating in the world. In recent years a rapid increase of free text data was witnessd like online news, governement documents,court rulings or social media communication. All these types have different structures or are completely unstructured and this complicates searching in it or getting

quick information. Thus, the need for efficient methods for analyzing unstructured or semi-structured texts is growing [7]. For these methods it is important to know what kind of data they should work on. This survey and the related project concentrates on processing journalistic data like articles and reports and its classification problems.

The problem of text classification represents an important part of the fields data mining, machine learning, database and information retrieval. In particular in domains of text mining there are multiple tasks dealing with this problem [13]. Some examples of applications are

- news filtering and organization,
- document organization and retrieval,
- opinion mining and
- email classification and spam filtering [13].

In the related project the focus will be on the first section, news filtering and organization.

The definition of the text classification problem, can be described as follows: At the beginning, we are given a set of data, in this case journalistic articles, where each article is related to a specific topic or so called class value. This data is the training set and it describes the features of the articles belonging to a relative topic. When searching the class for a new, unknown test data, this trainings set is used to compare it with the features of the unknown data and subsequently, to predict the class value [13].

For this prediction the presence or absence of words in the texts plays an important role. The algorithms often need this information to deal with the documents. Additionally useful is the frequency of words [13].

There are different approaches of text classification algorithms. Most of them belong to the statistical approach like the k-nearest-neighbour, the Naive Bayes, decision trees or the Rocchio [12]. One contrarian approach is the neuroscientific approach, which includes the semantic folding or fingerprinting method. This algorithm is based on Hierarchical Temporal Memory Theory of Jeff Hawkins and its proceeding is leaned towards the human brain[11]. More details are following.

Algorithms

When starting to process natural language, some important steps have to be taken to prepare the data for the algorithms and to separate the relevant content from the data. This pre-processing stage contains the tokenization (splitting string to tokens), the stop words removal (removing frequently occuring words like "'a"') and the stemming (transforming words back to their root form) [5, 1]. After preprocessing, the prepared data can be moved forward to the particular algorithm. An overview of the whole process of text classification can be seen in Figure 1.

k-nearest-neighbours

The k-nearest-neighbour algorithm needs the different documents in form of vector space models, where each document forms a vector of words or features. The main part of this alogorithm is a similarity function, which computes the distance between the test document vector and the train vectors. Frequently used functions are the Euclidean Distance, the Manhatten Distance or the Cosine Similarity. After detecting the similarity, the k closest documents are selected and the test instance will be assigned to the class which contains most of the neighbours. This method is effective, but it uses all features of the vectors for the similarity computation, what causes an increase of computation time [5].

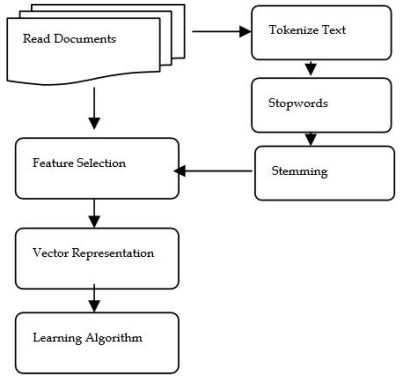

Figure 1: Text Classification Process [5]

Naive Bayes
The Naive Bayes algorithm is a probalistic classifier and it uses the joint probability of words and the related class to predict the category for test instances [12]. The basic assumption of this method is the independence of words in the documents, which makes the combination of features irrelevant as well as the presence or absence of them [12, 5]. These assumption improves the efficiency of the algorithm, because it only needs a small amount of training data to estimate the parameters for classifying the test documents. The main drawback of this method is the relatively low classfication performance [5].

Support Vector Machines
The SVM algorithm is based on the Structural Risk Minimization theory and its idea is to find a hypothesis for which the lowest true error can be guaranteed. This true error is the probability that the hypthesis causes an error [3, 10]. For the SVM postive and negative training sets are needed to find the so called hyperplane, which separates the positive and the negative data sets in the n-dimensional space. The texts which are closest to this plane are the support vectors [6]. This method is very powerful, but its main throwback might be the complexity of the algorithms [5, 6].

Rocchio
The Rocchio method starts the same way like the knn algorithm. It needs the vector space models of the documents and addtionally a prototype vector for each category with all terms contained in the training set [12, 13]. After determining the most similar vector of these prototype vectors to the test document, a category is defined and all documents belonging to this class are given a positive weight, and the documents belonging to the remaining classes a negative one. These weighted vectors form the new protoype vector for the category [5, 13]. This method is often used for filtering in information retrieval, is efficient in computation, but it has a low classification accuracy [12].

Semantic Folding
In comparison to the previous mentioned methods which are using word statistics, this one uses a neuroscience rooted mechanism to detect similarity between natural linguistic documents. The main goal of this theory is to convert the given input into the Sparse Distributed Representation (SDR) form which is manageable in an Hierarchical Temporal Memory Network. This HTM theory has the approach to understand how neo-cortical information processing works and to use this knowledge for natural language processing tasks. The SDR formats are the results of semantic folding and can be easily compared. They consist of large binary vectors, where every single bit in this vectors has a specific meaning. Single word-SDRs can be combined to bigger text-SDRs and still contain all relevant information. This large vectors are called semantic fingerprints

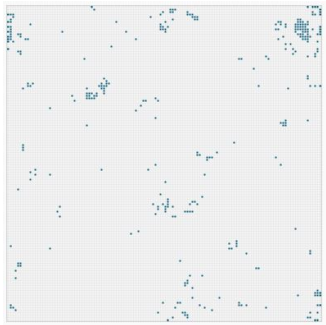

Figure 2: SDR of word "'java"' [2]

and can be also visualized as 2D matrix maps [11]. One example of a semantic fingerprint can be seen in Figure 2.

Related Work

There are several possible tools for classifying natural linguistic texts. Some of them are whole APIs like uClassify[1] or MeaningCloud[2]. And some are Open Source like Weka[3], GATE[4] and NLTK[5]; some are partly Open Source like cortical.io[6]. Further, they differ in what languages they are able to analyze and what methods they provide. In the related project some small PHP libraries, like the NlpTools[7], will be used for the tokenizing, stop word removal and stemming.

Several projects and papers also dealing with the comparison of text classification algorithms are

- the article of M.Trivedi et al. about SVM and Naive Bayes[9],
- the paper of C.Felden et al. about classifying online sources[1],
- the paper of Zach Chase et al. about topic classification of news articles[4] and
- the article of David B. Bracewell about *Category Classification and Topic Discovery of Japanese and English News Articles*[8].

Conclusion

Text classification or in general text mining, is gaining more and more importance in recent decades; especially because of the increase of electronic documents originating from different sources. These resources are mostly unstructured or semi-structured and this handicappes the quick and thoroughly gathering of information. The main goal of text mining is to enable and facilitate users to extract information from textual resources and to handle them fast and simple [5].

This need for a quick understanding of textual data und for extracting only relevant parts of information from it, was the motivation for the related project and the thesis. The project consists roughly of two main tasks: implementing different text classification algorithms and comparing them to find the best solution for organizing english and german articles. These algorithms are mostly the described above. Summarizing, it is a comparison between algorithms with a statistical approach versus the algorithm with the neuroscientific approach, with the goal to find the best solution for journalistic data.

References

[1] Carsten Felden, Heiko Bock, Andre Gräning et al. *Evaluation von Algorithmen zur Textklassifikation*.

[1] https://www.uclassify.com/
[2] https://www.meaningcloud.com/developer/text-classification
[3] https://www.meaningcloud.com/developer/text-classification
[4] https://gate.ac.uk/sale/tao/splitch19.html
[5] http://www.nltk.org/
[6] http://www.cortical.io/
[7] http://php-nlp-tools.com/

Tech. rep. Technische Universität Bergakademie Freiberg, Oct. 2006.

[2] cortical.io. *Demo - Topic Explorer*. 2015. URL: http://www.cortical.io/topic-explorer.html (visited on 01/10/2017).

[3] Thorsten Joachims. "Text Categorization with Support Vector Machines: Learning with Many Relevant Features". In: *10th European Conference on Machine Learning*. ECML-98. 1998, pp. 137–142.

[4] Zach Chase , Nicolas Genain, Orren Karniol-Tambour. *Learning Multi-Label Topic Classification of News Articles*. Tech. rep. stanford.edu, 2013.

[5] Aurangzeb Kahn, Baharum Baharudin, Lam Hong Lee and Khairullah Khan. "A Review of Machine Learning Algorithms for Text-Documents Classification". English. In: *Journal Of Advances In Information Technology* 1.1 (2010), pp. 4–20. DOI: 10.4304/jait.1.1.4-20. URL: http://www.jait.us/uploadfile/2014/1223/20141223050800532.pdf.

[6] Heide Brücher, Gerhard Knolmayer, Marc-Andre Mittermayer. *Document Classification Methods for Organizing Explicit Knowledge*. Tech. rep. Engehaldenstrasse 8, CH - 3012 Bern, Switzerland: Institute of Information Systems, University of Bern, 2002.

[7] Jakub Piskorski and Roman Yangarber. "Multi-source, multilingual information extraction and summarization". In: Berlin Heidelberg: Springer Berlin Heidelberg, 2013. Chap. Information extraction: Past, present and future, pp. 23–49.

[8] David B. Bracewell, Jiajun Yan, Fuij Ren and Shingo Kuroiwa. "Category Classification and Topic Discovery of Japanese and English News Articles". English. In: *Electronic Notes in Theoretical Computer Science* 225.1 (2009), pp. 51–65. DOI: 10.1016/j.entcs.2008.12.066. URL: http://www.sciencedirect.com/science/article/pii/S157106610800529X.

[9] Manali Trivedi, Samrudhi Sharma, Naitik Soni and Sindhu Nair. "Comparison of Text Classification Algorithms". English. In: *International Journal of Engineering Research and Technology* 4.2 (2015), pp. 334–336. ISSN: 2278-0181. URL: http://www.ijert.org/view-pdf/12364/comparison-of-text-classification-algorithms.

[10] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. New York: Springer, 1995.

[11] Francisco E. De Sousa Webber. *Semantic Folding Theroy*. Tech. rep. Cortical.io, Nov. 2015.

[12] Yiming Yang. "An Evaluation of Statistical Approaches to Text Categorization". English. In: *Information Retrieval* 1.1 (1999), pp. 69–90. DOI: 10.1023/A:1009982220290. URL: http://link.springer.com/article/10.1023/A:1009982220290.

[13] Charu C. Aggarwal, ChengXiang Zhai. "Mining Text Data". In: 233 Spring Street, New York, USA: Springer Science and Business Media, 2012. Chap. A SURVEY OF TEXT CLASSIFICATION ALGORITHMS, pp. 163–222.