International Journal of Computer (IJC)

ISSN 2307-4523 (Print & Online)

© Global Society of Scientific Research and Researchers

http://ijcjournal.org/

Exploiting Class Label Frequencies for Text Classification

Fragos Kostas*

Technological Educational institute of Athens (TEIA), Athens 12210, Greece

Email: kfragos@sch.gr

Abstract

Document classification is an example of Machine Learning (ML) in the form of Natural Language Processing (NLP). By classifying text, we are aiming to assign one or more classes or categories to a document, making it easier to manage and sort. In the vast majority of document classification techniques a document is represented as a bag of words consisting of all the individual terms making up the document together with the number of times each term appears in the document. The number of term occurrences is known as local term frequencies and it is very common to make use of the local term frequencies at the price of some added information in the classification model. In this work, we extend our previous work on medical article classification [1,2] by simplifying the weighting scheme in the ranking process using class label frequencies to device a simple weighting formula inspired from traditional information retrieval task. We also evaluate the proposed approach using more research experimental data. The method we propose here, called CLF KNN first, it uses a lexical approach to identify frequency terms in the document texts and then, it uses this information coupled with class label information in corpus in a sophisticated way to devise a weighting ranking scheme in classification decision process. The evaluation experiments on two collections: The Ohsumed collection of medical documents and the 20 Newsgroup messages collection, show that the proposed method significantly outperforms traditional KNN classification.

Keywords: Text Categorization; Term Frequency; Class Label Frequency; Document Text Classification.

1. Introduction

Several classification schemes have been successfully employed by classification literature in the recent years [3,4,5,6,7]. Authors have proposed different methods for categorizing the documents. The work in [3], proposes a novel text classifier using DNN, in an effort to improve the computational performance of addressing big text data with hybrid outliers. Specifically, through the use of denoising autoencoder (DAE) and restricted Boltzmann machine (RBM).

----

* Corresponding author.

In another work [4], a new method using rule-based approach was proposed for text categorization. In that method, authors introduced the idea of lexical syntactic patterns as classification features. A novel framework ROLEX-SP was proposed concentrating on capturing the correct classes of text as well as reducing classification errors. Machine learning techniques have extensively used to conduct the classification task. Principal Component Analysis (PCA) is a dimensionality reduction method in which a covariance analysis between factors takes place [8]. PCA method was used in the field of text mining, especially two of its variants namely the neural PCA and kernel PCA for categorization of text documents by extracting semantic concepts. KNN (k-nearest neighbor) was intensively studied as an effective classification model in decades [9]. In [10], authors evaluated and studied some machine learning methods: k nearest neighbor (KNN), support vector machines (SVM) and naive Bayes (NB) classifiers and consider that KNN and naive Bayes are simple and robust machine leaning classification techniques. More concretely, paper reports a comparative study for medical text categorizations on four machine learning methods: k Nearest Neighbor (kNN), Support Vector Machines (SVM), Naïve Bayes (NB) and Clonal Selection Algorithm Based on Antibody Density (CSABAD).

In traditional machine learning algorithms, every instance in any dataset is represented using the same set of features. The features can be continuous, categorical or binary. Work on [11] considers that K-Nearest Neighbor (KNN) classification is one of the most fundamental and simple classification methods and when there is little or no prior knowledge about the distribution of the data, the KNN method should be one of the first choices for classification. The kNN rule classifies each unlabeled example by the majority label of its k-nearest neighbors in the training set. Despite its simplicity, the kNN rule often yields competitive results and in certain domains, when cleverly combined with prior knowledge, it has significantly advanced the state-of-the-art [13,14]. Work on [15] proposes "KNN classification was developed from the need to perform discriminant analysis when reliable parametric estimates of probability densities are unknown or difficult to determine. In spite of these good advantages, it has some disadvantages such as: high computation cost in a test query, the large memory to implement, low accuracy rate in multidimensional data sets, parameter K, unclearness of distance type [11].

In this paper we have used a novel lexical approach to classify real world text classification articles. It is based on identifying tokens or lexemes in the document. The KNN algorithm is used for text categorization as it is an efficient and very popular algorithm. To overcome the disadvantages of KNN algorithm, a new interesting algorithm is proposed here which partially overcomes the low accuracy rate of KNN. It preprocesses the train set, computing the ranking score of any train samples. Then the final classification is executed using weighted KNN which is employed the ranking score as the multiplication factor.

The rest of the paper is organized as follows: Section 2 describes KNN text classification algorithm. Section 3 explains the proposed classification algorithm. Section 4 describes the experimental data. Section 5 shows the evaluation results on experimental data. Finally, sections 6 and 7 conclude the contributions of the research.

2. KNN Classification

In text classification field, KNN is one of the most important non-parameter algorithms [16] and it is a supervised learning algorithm. The classification rules are generated by the training samples themselves without any additional data. The KNN classification algorithm predicts the test sample’s category according to the K training samples which are the nearest neighbors to the test sample, and judge it to that category which has the largest category probability. The process of KNN algorithm to classify sample X is given below [16]:

```
Algorithm

Input Parameters: Data set consisting of document vectors of terms, k

Output: Classified document vectors

Step 1: Store all the training document vectors.

Step 2:    For each unseen vector which is to be classified

A. Compute distance of it with all the training vectors.

B. Find the k nearest training vectors to the unseen tuple.

C. Assign the class which is most common in the k nearest training vectors to the unseen vector.

End for
```

3. The Proposed Classification Algorithm

Before explaining our algorithm we will give a quick informal explanation of TF-IDF weighting scheme in text classification [17,18]. Essentially, TF-IDF works by determining the relative frequency of words in a specific document compared to the inverse proportion of that word over the entire document corpus. Intuitively, this calculation determines how relevant a given word is in a particular document. Words that are common in a single or a small group of documents tend to have higher TFIDF numbers than common words such as articles and prepositions. The formal procedure for implementing TF-IDF has some minor differences over all its applications, but the overall approach works as follows. Given a document collection *D*, a word *w*, and an individual document $d \epsilon D$, we calculate:

$$w_d = f_{w,d} * \log(\frac{|D|}{f_{w,D}}) \qquad (1)$$

Where $f_{w,d}$ equals the number of times *w* appears in *d* (local frequency), $|D|$ is the size of the corpus, and $f_{w,D}$ equals the number of documents in which *w* appears in *D* (global frequency).

In text classification the main idea is to assign to a document the class label of the data according to *K* high ranked data points of the train set. Thinking in a similar way, like in TF-IDF weighting scheme, we device a new weighting scheme for class label prediction deriving information from class label frequencies in the training corpus. We propose an algorithm, called Class Label Frequency KNN (CLF KNN), to classify text articles using local and global class label frequencies as a weighting scheme to rank all data points in the training set. In the first step of this procedure, the ranking score of each data sample in the train set is computed, and then, a weighted KNN is performed on any test samples. In the ranking process, for each class *x* in the training set a local class label frequency *Lx* is computed according to *A* nearest neighbors of the point. Among the *A* nearest neighbors of a train sample *x*, *Lx* is computed counting the number of points with the same label to the label of *x*. We define the *Lx* as follows:

$$Lx = \Sigma i\ I(\ lbl(x),\ lbl(Ni(x)),\ i=1..A \qquad (2)$$

Where *A* is the number of considered neighbors and *lbl(x)* returns the true class label of the sample *x*. *Ni(x)* stands for the *ith* nearest neighbor of the point *x*. The function *I* returns 1 if *lbl(x), lbl(Ni(x))* are equal.

It seems unlikely that a large number of occurrences of a class in *A* nearest neighbors truly carry many times the significance of a single occurrence. Accordingly, there has been considerable research into variants of term frequency in document classification that go beyond simply counting the number of occurrences of a term [18,19]. So, in first stage we modify frequency numbers using a simple logarithmic normalization factor frequencies instead of using simple class label frequencies. This assign a new normalized weight from local neighborhood frequencies given by:

$$NLx = 1+\log\ (lf(x)),\ \text{if}\ lf(x)>0,\ 0\ \text{otherwise} \qquad (3)$$

In the second step, we compute the global class label frequency *Gx* of the point *x* defined as follows:

$$Gx = \log(S/M) \qquad (4)$$

Where *M* is the counting of *lbl(x)* in the training corpus and *S* is the total size of the training corpus.

Finally, we combine the above two frequencies to compute a composite ranking score *Wx* for train sample *x* given by:

$$Wx = NLx * Gx \qquad (5)$$

Weighted KNN is one of the variations of KNN method which uses the K nearest neighbors of the unknown test sample, regardless of their classes, but then uses weighted votes from each sample rather than a simple majority or plurality voting rule. Each of the K neighbors is given a weighted vote that is usually equal to some decreasing function of its distance from the unknown test sample. In our CLF KNN method, we compute the weighted vote for each neighbor in the training sample using the formula of equation 5. These weights are then summed for each class, and the class with the largest total sum is chosen. The above technique is very simple and it has the effect of giving greater importance to the reference samples that have greater rankness to the test sample. So, the decision is less affected by reference samples which are not very stable in the feature space in comparison with other samples

4. Evaluation Data and Preparation

The databases we studied are:

Ohsumed test collection: This collection is compiled by William Hersh [20]. Ohsumed collection includes medical abstracts from the MeSH categories of the year 1991. In this work we used the first 20,000 documents divided in 10,000 for training and 10,000 for testing. The specific task was to categorize the 23 cardiovascular diseases categories. After selecting the subset of this category, the number of the unique abstracts is equal to 13,929 (6,286 for training and 7,643 for testing). The above collection is a part of the whole collection containing a larger number of documents (34,389 cardiovascular diseases abstracts out of 50,216 medical abstracts of the year 1991). The purpose of the text categorization is to “allocate” the documents to one or multiple categories of the total 23 Cardio vascular diseases. A document belongs to a category if it contains at least one indexing term of the category. The documents are first scanned and preprocessed to identify tokens from the abstracts of journal articles. Stop words and special characters are removed from the abstracts of the articles. Then the articles are sent to Lexical Analyzer. The lexical analyzer scans the characters in the given input and group them into tokens (terms). Tokens in our research are keywords and their related or synonymous words. Finally, each journal article in this medical collection can be expressed as a vector of tokens: $x=<k_1,k_2,..k_n>$. The journal article thus is considered as a data point in the vector space of tokens.

20 Newsgroups: A collection of messages, from 20 different newsgroups, with one thousand messages from each newsgroup. The data set has a vocabulary of 64,766 words [21,22,23,24] and splits in two subsets: one for training (or development) and the other one for testing (or for performance evaluation). The split between the train and test set is based upon a messages posted before and after a specific date. We processed the text of this collection using the Rainbow toolbox (McCallum 1996) to extract terms [25].

5. Experimental Results

To evaluate the performance of our proposed algorithm, we have compared it with the traditional KNN algorithm. The most common performance metrics used in text categorization are Recall, Precision and F-measure [26]. They can be calculated as follows: In our experiment, if we define TP as the number of true positive samples predicted as positive, FN as the number of true positive samples predicted as negative, FP as the number of true negative samples predicted as positive and D as the number of true negative samples predicted as negative, then Precision, Recall, F-measure can be expressed as follows:

$$Precision=TP / (TP+FP) \tag{5}$$

$$Recall = TP / (TP+FN) \tag{6}$$

$$F\text{-}measure = (2* Precision * Recall) / (Precision + recall) \tag{7}$$

We have conducted a comparative study of the performance of traditional KNN and CLF KNN method with the different K values. The value of parameter A is determined equal to a fraction of the number of train data which is empirically set to 10% of the train size. The classification results with the different *K* values are shown in Fig. 1 for Ohsumed collection and in Fig. 2 for 20 Newsgroup collection. The results listed are the best results we get for each algorithm from our experiments.

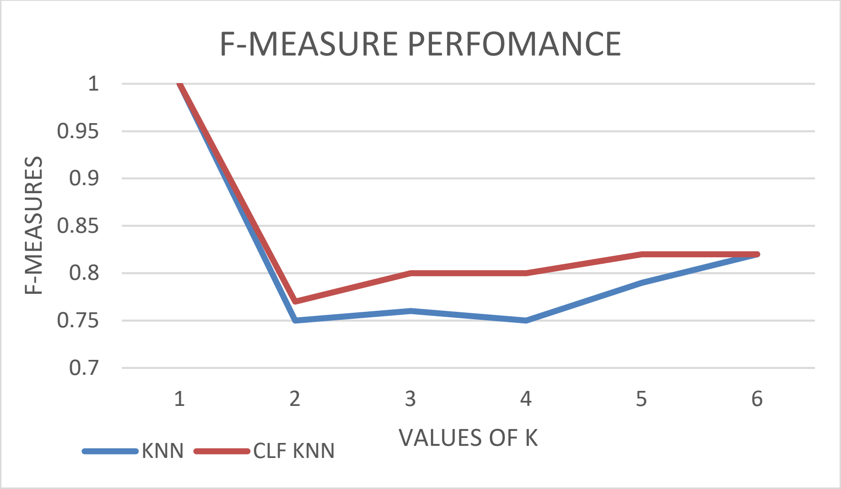

Figure 1: Comparison between traditional KNN classification and CLF KNN Classification. Ohsumed collection

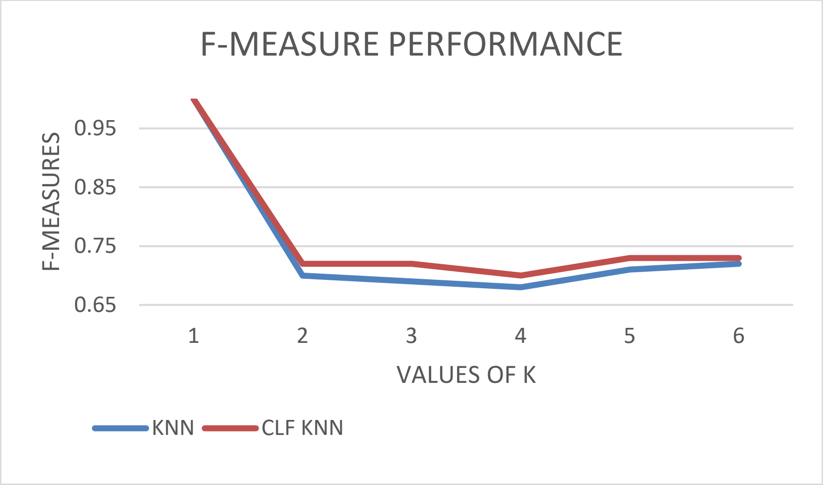

Figure 2: Comparison between traditional KNN classification and CLF KNN Classification. 20 Newsgroup collection

6. Discussion

Counting class label frequencies is based on the assumption that this conveys very simple and important information in KNN classification process. This information can be used in a sophisticated way to devise a weighting ranking scheme in the weighted KNN classification algorithm inspired from traditional information retrieval task. We call the method the CLF-KNN approach since the class label neighbors are weighted by using raw frequencies derived from the neighborhood of a train sample. The experiments show that the proposed CLF KNN method significantly outperforms traditional KNN method, while using different choices of value K, over different datasets. The performance is better because the proposed classification is based on ranked neighbors using local and global class label frequency which have more information in comparison with simple class labels.

One of the limitations of this work is that it employs a single class label frequency logarithmic normalization mechanism that does not take into account various aspects of a class label's saliency among the *A* nearest neighbors of a train sample.

7. Conclusion and Recommendations

In this work, we present a novel lexical approach to text categorization in the text classificarion domain. We propose an algorithm CLF KNN which automatically classifies journal articles of text documents into various categories. The concept of tokens is used to represent a journal article. Each journal article is represented as a vector of tokens in the medical articles of Ohsumed collection extracted from the abstract part, and from the body text in 20 Newsgroup collection. The CLF KNN algorithm is used as a text classifier. The proposed algorithm has outperformed the traditional KNN. This is shown by calculating Recall, Precision and F-measure values. In the future, the conducted pilot study could be extended for the various weighting schemes derived from traditional information retrieval field. Also, the algorithm could be tested for various other text documents or text datasets.

Acknowledgements

This research has been co-funded partially by the European Union (Social Fund) and Greek national resources under the framework of the "Archimedes III: Funding of Research Groups in TEI of Athens" project of the "Education & Lifelong Learning" Operational Programme.

References

[1] K. Fragos, C. Skourlas. "Smoothing Class Frequencies for KNN Medical Article Classification," in Proc of 20th Pan-Hellenic Conference on Informatics. PCI 16, 2016, Article No. 79.

[2] K. Fragos, C. Skourlas. "Ranking tokens with class label frequencies for medical article classification," in Proc of 19th Panhellenic Conference on Informatics, 2015 pp. 359-360.

[3] A. Wulamu et al. "A Robust Text Classifier Based on Denoising Deep Neural Network," Analysis of Big Data Scientific Programming Vol 33, 2017, Article ID 3610378, 10 pages: https://doi.org/10.1155/2017/3610378.

[4] GH. A. Z. Mohammed and A. B. Can. "ROLEX-SP: Rules of lexical syntactic patterns for free text categorization," journal of Knowledge-Based Systems, Elsevier vol 24, 58-65, 2011.

[5] P. Semberecki and H. Maciejewski. "Deep learning methods for subject text classification of articles,"

presented at Federated Conference on Computer Science and Information Systems (FedCSIS), Prague, Czech Republic, 3-6 Sept. 2017.

[6] L. H. Lee, D. Isa, W. O. Choo, and W. Y. Chue, "High Relevance Keyword Extraction facility for Bayesian text classification on different domains of varying characteristic" Expert Systems with Applications, vol. 39, no. 1, pp. 1147–1155, 2012.

[7] K. Fragos, C. Skourlas. "Towards Improving Classification of Real World Biomedical Articles." Presented at 18th Int. Conf. of Panhellenic Conference Informatics, Harokopio University of Athens, 2nd - 4th October, 2014.

[8] S. Jaffali and S. Jamoussi. "Principal Component Analysis neural network for textual document categorization and dimension Reduction," presented at 6th International Conference on Sciences of Electronics, Technologies of Information and Telecom, IEEE Xplore, 2012.

[9] D. W. Aha, K. Dennis and A. K. Marc. "Instance-Based Learning Algorithms," Machine Learning, vol. 6, pp. 37-66, 1991.

[10] Q. Zhang, et al. "Machine Learning Methods for Medical Text Categorization," in Proc. Pacific-Asia Conference on Circuits, Communications and Systems, 2009, pp. 494 – 497.

[11] H. Parvin, H. Alizadeh and B. Minaei-Bidgoli. "Modification on K-Nearest Neighbor Classifier," Global Journal of Computer Science and Technology, vol.10, pp. 37-41, Nov. 2010.

[12] H. Parvin, H. Alizadeh and B. Minaei-Bidgoli. "MKNN: Modified K-Nearest Neighbor," presented at the World Congress on Engineering and Computer Science 2008 WCECS, October 22 - 24, San Francisco, USA, 2008.

[13] S. Belongie, J. Malik, and J. Puzicha. "Shape matching and object recognition using shape contexts," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24 (4), pp. 509–522, 2002.

[14] P. Y. Simard, Y. LeCun, and J. Decker. "Efficient pattern recognition using a new transformation distance," in S. Hanson, J. Cowan, and L. Giles, editors, Advances in Neural Information Processing Systems 6, pp. 50–58, San Mateo, CA, Morgan Kaufman, 1993.

[15] R. O. Duda, P. E. Hart and D. G. Stork. "Pattern Classification," John Wiley & Sons, 2000.

[16] A. Lambda, D. Kumar. "Survey on KNN and Its Variants," International Journal of Advanced Research in Computer and Communication Engineering, vol. 5, Issue 5, May 2016

[17] G. Salton and C. Buckley. "Term-weighing approaches in automatic text retrieval," Information Processing & Management, vol. 24(5), pp. 513-523, 1988.

[18] A. Berger, et al. “Bridging the Lexical Chasm: Statistical Approaches to Answer Finding,” in Proc. Int. Conf. Research and Development in Information Retrieval, 2000, pp. 192-199.

[19] K. Sparck Jones. “Document retrieval systems, A statistical interpretation of term specificity and its application in retrieval”, Taylor Graham Publishing, pp. 132–142, London, UK, UK, 1988.

[20] W. Hersh, C. Buckley, T. Leone and D. Hickman. “OHSUMED: An interactive retrieval evaluation and new large text collection for research,” in Proc. 17th ACM International Conference Research and Development in Information Retrieval 1994, pp. 192–201.

[21] A. McCallum and K. Nigam. “A comparison of event models for naive bayes text classification,” presented at AAAI-98 Workshop on Learning for Text Categorization, 1998.

[22] S. Eyheramendy and D. Lewis. “On the Naive Bayes model for text categorization,” in Proc. Ninth International Workshop on Artificial Intelligence & Statistics, Key West, FL, 2002

[23] M. D. J. Rennie, et al. “Tackling the poor assumptions of naive Bayes text classifiers,” presented at Twentieth International Conference on Machine Learning, Washington, DC, 2003.

[24] E. R. Madsen, et al. “Modeling word burstiness using the Dirichlet distribution,” presented at 22nd International Conference on Machine Learning. Bonn, Germany, ACM Press, 2005.

[25] K. A. McCallum. “Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering,” presented at Carnegie Mellon University, 1996.

[26] L. Baoli, et al. “An Improved k-Nearest Neighbor Algorithm for Text Categorization,” presented at the 20th International Conference on Computer Processing of Oriental Languages, Shenyang. China. 2003.