SURVEY ON DOCUMENT CLASSIFICATION TECHNIQUES AND DEEP LEARNING

Michael Shell, *Member, IEEE,* John Doe, *Fellow, OSA,* and Jane Doe, *Life Fellow, IEEE*

***Abstract*—The exponential growth of the Internet has led to great deal of interest in developing useful and efficient tools and software to assist users in searching the web for relevant documents. Document classification is generally defined as content-based assignment of one or more predefined categories to documents. Document classification appears in may applications, including email-filtering, news monitoring, etc. It is not feasible to classify these documents manually and present automated classification methods have drawbacks like low accuracy and dependency on humans for document tagging.**

This paper describes different document summarization and classification methods. It also compares different classifier models and explains why Deep Learning methods are better than the conventional ones. Further, it proposes the applications of document classification in real world.

Keywords*—*Classification, Summarization, Neural networks, RCNN , Survey, Deep Learning.

I. Introduction

With the increasing availability of digital documents from diverse sources, text classification is gaining popularity day in and day out. There is a mushroom growth of digital data made available in the last few years, data discovery and data mining have worked together to extract meaningful data into useful information and knowledge. Text summarization refers to the process of deriving high quality information from text. It is conducive in utilizing information contained in textual documents in various ways including discovery of patterns, association among entities etc. and this is done with the amalgamation of NLP(Natural Language Processing), Data Mining and Machine learning techniques. Infeasibility of human beings to go through all the available documents to find the document of interest precipitated the rise of document classification. Automatically categorizing documents could provide people a significant ease in this realm. Text classification assigns documents one or more predefined categories. The notion of classification is very general and has many applications within and beyond information retrieval (IR). For instance, text classification finds its application in automatic spam detection, sentiment analysis, automatic detection of obscenity, personal email sorting and Topic specific or Vertical Searches. An example of classification would be automatically labeling news stories with subjects like business, entertainment, sports etc.

M. Shell is with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332 USA e-mail: (see http://www.michaelshell.org/contact.html).

J. Doe and J. Doe are with Anonymous University.

Manuscript received April 19, 2005; revised January 11, 2007.

A. Machine Learning in Automated text Categorization

The automated categorization (or classification) of texts into predefined categories has witnessed a booming interest in the last 10 years, due to the increased availability of documents in digital form and the ensuing need to organize them. In the research community the dominant approach to this problem is based on machine learning techniques: a general inductive process automatically builds a classifier by learning, from a set of preclassified documents, the characteristics of the categories. The advantages of this approach over the knowledge engineering approach (consisting in the manual definition of a classifier by domain experts) are a very good effectiveness, considerable savings in terms of expert labor power, and straightforward portability to different domains.

Until the late 80s the most popular approach to TC, at least in the operational (i.e., real-world applications) community, was a knowledge engineering (KE) one,consisting in manually defining a set of rules encoding expert knowledge on how to classify documents under the given categories. In the 90s this approach has increasingly lost popularity (especially in the research community) in favor of the machine learning (ML) paradigm, according to which a general inductive process automatically builds an automatic text classifier by learning, from a set of preclassified documents, the characteristics of the categories of interest. The advantages of this approach are an accuracy comparable to that achieved by human experts, and a considerable savings in terms of expert labor power, since no intervention from either knowledge engineers or domain experts is needed for the construction of the classifier or for its porting to a different set of categories.

B. Intended Audience and Reading Suggestions

This paper can be used as a reference by researchers who wish to study different summarization/classification techniques or study the advantages of deep learning methods to increase efficiency of document classification. Developers can refer this paper to select a classification method suitable to their application type. It also helps application users to understand why deep learning is superior to other classification techniques used in other similar applications.

II. Summarization Techniques

A. Genetic Clustering Algorithm for Extractive Text Summarization

B. K nearest Neighbor for Text Summarization using feature Similarity (2017)

Traditional KNN - In the traditional KNN version, the sample words which are labeled with the positive class or the negative class are encoded into numerical vectors. The similarities of the numerical vector which represents a novice word with those representing sample words are computed using the Euclidean distance or the cosine similarity. The k most similar sample words are selected as the k nearest neighbors and the label of the novice entity is decided by voting their labels.

Alternate KNN - In another KNN version, a text is given as an input and it is encoded into a numerical vector. The similarities of the novice text with the sample ones are computed, k most similar samples are selected as the nearest neighbors, and the label of the novice is decided by voting their labels. The scheme of computing the similarity between numerical vectors is the essential difference between the two versions.

C. Graph Based Text Summarization

The proposed method is multi-graph based. The number of edges in the graph between two sentences (i.e., two nodes) is equal to the number of same words in both sentences. According to our assumption, a word may occur in a sentence more than once as in the example of Figure 1, such occurrence will be added in the symmetric matrix. The total number of edges is stored in a symmetric matrix that represents the text being summarized. Then, we sum the values of rows (or columns sym metric matrix) of the matrix to generate what we call a sum vector, which is then used for ranking the sentences as show n in table 1. This approach is used to replace other graph-based methods measures: term frequency, tf, and inverse document

frequency, idf, which are used for more than half a century by researchers until this date. This new approach can be summarized as follows:

- Generate the symmetri c matrix with the exact number of edges between the nodes (i.e., the sentences)
- Algebraically sum the rows of the matrix to produce rand vector
- Sort the ranking vector to get the sentence rank
- Apply the cut-off m echanism using the required threshold to produce th e summary

This method is different from traditional Graph based methods in the sense that it does not use the cosine equation to find the similarity between the sentences. For the calculation of cosine equation, researchers have been using tf-idf but these factors for each word within a sentence are not calculated here. The other difference is that this method is not identifying any relation within a sentence because within a sentence we don't have to find edges. Those edges may contribute towards redundant information in the summary. We are not tracking each word for the calculation of matrix. We are focusing on the significant words only.

The proposed method is efficient, simple and fast. Almost all methods have pre-processing schemes. For, this algorithm, its performance has been seen to increase significantly when the preprocessing scheme was employed. Pre-processing reduces the size of the matrix by a considerable amount, which increases the performance and accuracy of the algorithm. Preprocessing mainly includes removal of articles, prepositions and meaningless words (like a sentences starting with a bracket or any special character).

III. Classification Techniques

A. SVM with TF-IDF

SVM is a partial case of kernel-based methods. It binds feature vectors into a higher-dimensional space using a kernel function and builds an optimal linear discriminating function in this space or an optimal hyper-plane that is congruent with the training data. The kernel is not explicitly defined in case of SVM. Instead, a distance between any 2 points in the hyper-space needs to be defined. The key features of SVMs are the use of kernels, the absence of local minima, the sparseness of the solution and the capacity control obtained by optimizing the margin. Besides the advantages of SVMs - from a practical point of view they have some drawbacks. An important practical question that is not entirely solved, is the selection of the kernel function parameters - for Gaussian kernels.

B. Naive Bayes

The Naive Bayes classifier is a probabilistic classifier based on Bayes theorem with strong and nave independence assumptions. It is supposed to be one of the most basic text classification techniques with various applications in email spam detection, personal email sorting, document categorization, sexually explicit content detection, language detection and sentiment detection. Experiments witness that this algorithm performs well on numeric and textual data. Though it is often outperformed by other techniques such as boosted trees, random forests, Max Entropy, Support Vector Machines etc., Naive Bayes classifier is quite efficient since it is less computationally intensive (in both CPU and memory) and it necessitates a small amount of training data. The assumption of conditional independence is breached by real-world data with highly correlated features thereby degrading its performance.

C. Convolutional Neural Networks

Convolutional neural networks (CNN) utilize layers with convolving filters that are applied to local features. Originally invented for computer vision, CNN models have subsequently been shown to be effective for NLP and have achieved excellent results in semantic parsing, search query retrieval, sentence modeling, and other traditional NLP tasks.

In the present work, the proposed method trains a simple CNN with one layer of convolution on top of word vectors obtained from an unsupervised neural language model.

These vectors were trained by Mikolov et al. (2013) on 100 billion words of Google News, and are publicly available. Initially keep the word vectors static and learn only the other parameters of the model. Despite little tuning of hyperparameters, this simple model achieves excellent results on multiple benchmarks, suggesting that the pre-trained vectors are universal feature extractors that can be utilized for various classification tasks. Learning task-specific vectors through fine-tuning results in further improvements. It finally describes a simple modification to the architecture to allow for the use of both pre-trained and task-specific vectors by having multiple channels.

D. Recurrent - Convolutional Neural Networks

This technique proposes an ensemble application of convolutional neural network (CNN) and recurrent neural network (RNN) to tackle the problem of multi-label text categorization. A CNN-RNN architecture is developed to model the global and local semantic information of texts, and then utilize such label correlations for prediction. In particular, it employs the state-of-the-art word-vector based CNN feature extraction and deal with high-order label correlation while keeping a tractable computational complexity by using RNN.

This method overcomes the drawback of CNN where it does not consider feedback from the last layer to improve efficiency of the classification model.

IV. Conclusion

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Appendix A
Proof of the First Zonklar Equation

Some text for the appendix.

Acknowledgment

The authors would like to thank...

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.

John Doe Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.