Summary-Based Document Classification

P. P. Assainar Hafnan and Anuraj Mohan

Abstract Document classification is one among the major NLP tasks that facilitate mining of text data and retrieval of relevant information. Most of the existing works use pre-computed features for building the classification model. Large-scale document classification relies on the efficiency or appropriateness of feature selection for document representation. The proposed system uses text summarization for automated feature selection to build the classification model. This work considers feature selection as a sentence extraction task which can be done using extractive text summarization. The method will have the advantage of reduced feature space, as classifier will be trained on shorter summary than the original document. Also, deep learning-based summarization generates the most relevant features resulting in improved efficiency and accuracy of the classifier. Experiments showed that classification based on features generated using deep learning provides better classification accuracy.

Keywords Deep learning · Document classification · Text summarization

1 Introduction

Availability of information in the form of documents and other forms are increasing at an exponential pace everyday making text management a tedious task. Searching for and retrieval of relevant information through this vast amount of information sources is a challenging problem. Automation of these processes is carried out using natural language processing techniques. Summarization is such an NLP task for capturing the main aspects covering a document or a collection of documents. In this work, a newer approach based on document summarization is proposed for document classification. Summarization is the process of deriving a reduced representation of

P. P. Assainar Hafnan (✉) · A. Mohan
Department of Computer Science and Engineering, NSS Engineering College, Palakkad, India
e-mail: hafnu.hafnu@gmail.com

P. K. Sa et al. (eds.), *Recent Findings in Intelligent Computing Techniques*, Advances in Intelligent Systems and Computing 709,
https://doi.org/10.1007/978-981-10-8633-5_16

a document or a collection of documents, which conveys what is actually intended by the original document or the collection. The summary can be extracted or generated from the documents. Documents can be classified based on its content or based on its attributes such as document length, author name, etc. which facilitates users to search for information based on the category. Content-based classification assigns documents to classes based on the weights assigned to the textual units in its content.

This paper presents a supervised content-based document classification system built on summary features extracted via deep learning. Existing classification models require explicit specification of features upon which the classifier needs to be trained. The idea behind this work is to determine relevant features through summarization and use these features for document classification in order to increase the accuracy and efficiency of the classifier. This work is an attempt to automate the feature selection by summarizing the documents via deep learning. These features are then used as attributes for building the classifier.

Deep learning techniques are the most efficient methods for automatically finding features and learning higher level representations from the data provided to it. Building a summary-based classifier has several advantages. As the size of the summary is much less than the size of the corresponding document, the dataset size gets reduced. The classifier needs to be trained on a smaller dataset with lesser time. Hence efficiency is ought to be increased. There will be a reduction in the dimensionality of input representations. Also, feature space for summaries will be small as compared to the feature space for the entire document collection. Hence, a smaller feature space needs to be explored which contains the most relevant features extracted from the document. If the classifier can be built on the most relevant features extracted, accuracy can be increased.

2 Literature Survey

2.1 Text Summarization

This section discusses the various methods for implementing text summarization. Sentence ranking is the major step in selecting important sentences in the extractive summarization of a document. The most traditional method in sentence ranking is the use of statistical features derived from the document [1, 2]. The statistical features used include the position of a sentence in the document, similarity of a sentence with the title, sentence length, presence of cue words or phrases, presence of frequent words, presence of words with high Tf-idf values, presence of title words in the sentence, presence of proper nouns, etc. Graph-based methods are the next approach towards the extraction of summaries from the original document. The core of all graph-based methods is almost the same which includes preprocessing tasks,

building a graph model, applying ranking algorithm and finally summary generation task. Documents are represented using simple graphs or bipartite graphs [3, 4]. Graph is constructed with nodes representing the textual units and edges are drawn between the nodes and is weighted with cosine similarity, eigenvector similarity, content-based features, other features such as cue words, length of sentences, etc. Graph-based shortest path algorithms, HITS algorithm, page rank algorithm, etc. are used for ranking graph nodes.

Another approach which gained interest in recent years towards summarization employs natural language processing and machine learning techniques. The work by [5] employs support vector machine cascaded with clustering technique to obtain much better summary. A machine learning-based single document summarization system uses a set of features based on groups of words which often co-occur with each other for obtaining sentence vectors and a classifier is trained in order to make a global combination of these scores in the vector [6]. The sentences are represented as vector of features including sentence length, sentence position, etc. A concept-based automatic multi-document summarizer is built using learning mechanism [7]. Concepts extracted using mutual information combined with statistical features derived from the document are fed to appropriate learning model.

A summarization system for news articles is implemented in three phases including neural network training, feature fusion and sentence selection [8]. A system is built that utilizes bisect K-means clustering to improve the time and neural networks to improve the accuracy of the summary generated by NEWSUM algorithm [9]. A pair-based sentence ranker for the summarization of newswire documents is built using RankNet learning algorithm to score every sentence in the document and identify the most important sentences [10].

A summarization model is built using embeddings derived for sentence level and document level using convolutional neural networks with backpropagation [11]. A newer technique for summarization employs neural network and rhetorical structure theory [12]. Neural network-based training is used to extract sentences, and these sentences are fed to rhetorical structure to find relation between sentences that facilitates generation of better summary. In addition to handcrafted feature vectors of words as input, recursive neural networks are used to automatically learn ranking features [13]. A deep learning-based approach for extractive summarization uses restricted Boltzmann machine to reduce redundancy. The four features title similarity, positional feature, term weight and concept feature computed for document is the input to the training algorithm.

All the above works require explicit specification of features to be used for building the summarizer. With the development of deep learning techniques, automated generation of features can be done. One such work performs query-oriented summarization on multiple documents [14]. It employs deep learning model with three stages: concept extraction, reconstruction validation and summary generation.

2.2 *Document Classification*

Similar to text summarization, text classification has also been widely explored. Traditional machine learning algorithms such as decision trees, *K*-nearest neighbour, naive Bayes, support vector machines, etc. are used for text categorization [15–17]. In recent years, neural network-based classifiers are also built [18].

Accuracy of all the above methods relies on the accuracy of the feature selection process. Dimensionality reduction, i.e. selecting only the most representative features during feature selection is also an important consideration. The method proposed here aims to use summarization to extract relevant features. The focus in this approach is to replace manual feature selection with automated feature selection using deep learning in order to increase the accuracy of classification. To the best of our knowledge, this proposal is the first attempt at using summarization to assist the classification task.

3 Proposed System

This section gives the detailed description of summary-based document classifier. Figure 1 gives the architecture of the proposed system with two important modules: summarizer and classifier.

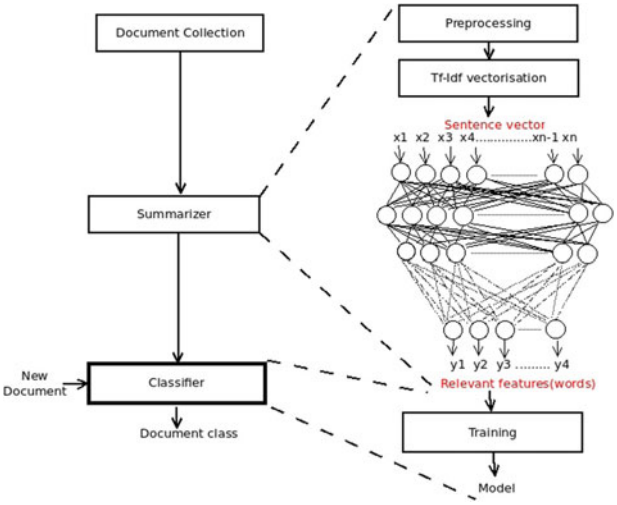


Fig. 1 Summary-based classifier

3.1 Summarizer

Summarization module works as the feature extractor for building the classifier. Documents are represented using the bag-of-words model. Deep learning technique is the core of the summarization process presented here.

Text Preprocessing. The text document is initially preprocessed to clean the textual content and produce preprocessed sentences. Preprocessing involves removal of punctuations, stop words, special characters, etc.

Tf-idf Vectorizer. Vectorizer then converts these textual sentences from preprocessing stage into corresponding tf-idf vectors. The vectorizer first builds a local vocabulary for the input document and creates a representation vector with size same as that of the vocabulary, i.e. $x_1, x_2, x_3, \ldots, x_n$, where each x_i denotes the $tf - idf$ value of the word i in the vocabulary built over the document and n denotes the vocabulary size.

Deep Architecture. The deep architecture uses restricted Boltzmann machines as its basic building block. RBM has a visible and hidden layer with weighted connections between nodes in each layer. Each possible joint configuration of the visible and hidden units has an energy determined by weights and biases. The learning procedure starts by setting the states of visible units to a training vector. Each possible visible layer and hidden layer configurations are reconstructed based on the joint probabilities. The probability computation is based on the logistic sigmoid function. Each time the configurations are updated, the weighted connections are also updated. Training vector is input to the visible layer and activations for reduced representation of document is produced as output at the hidden layer.

Summarizer is built by stacking RBMs one over the other. Figure 2 gives the stacked structure of multiple RBM units. Each layer performs computation over

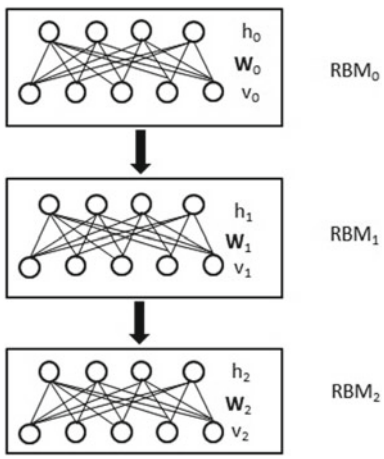


Fig. 2 Stacked RBMs

these input sentence vectors to filter out relevant features. Each RBM is fine-tuned to find the most relevant features which are fed as input to the next higher level RBM. The whole operation with RBM starts with giving the sentence vector as input. The number of nodes in the initial visible layer is determined by the length of the input sentence vector. The number of nodes is equal to the dimension of the tf-idf vector representing a document sentence. Each node in the visible layer takes as input the tf-idf value for the corresponding word or feature. With these values as input, the RBM performs its joint probability computations and computes the activations for its nodes as described previously. This processing continues until an acceptable error rate is reached. The output from the hidden layer of the last unit gives the relevant feature vector $y_1, y_2, \ldots, y_m$ with $m < n$, i.e. a vector with a lower dimension. This vector serves as the input to the classifier training. Thus, the summarizer module facilitates dimensionality reduction.

3.2 Classifier

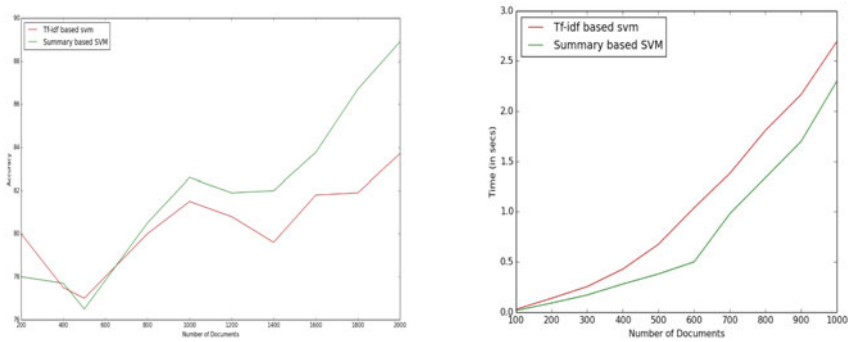
The classification model is built using SVM classifier which takes the features extracted from the deep architecture as input. It then learns from the training samples and builds the model. After the model has been constructed, newly arriving documents can be classified based only on the known labels.

4 Experimental Setup

This work is implemented using Python language and its associated libraries upon the TensorFlow framework to implement deep learning. TensorFlow is an open source software library for machine learning in various kinds of language understanding tasks. Preprocessing and vectorization are carried out using NLTK and Scikit-learn library. LIBSVM tool is used for building the classifier. The dataset used for the task is Reuters-21578 text categorization test collection.

5 Experimental Result

The text collection without considering the label is first fed through the summarizer. The result together with the class labels is then given as input to the classifier for training. The number of visible units in RBM is set equal to the dimension of input vectors. The summarizer module can be optimized by adjusting the learning rate and



(a) Accuracy **(b)**Execution Time

Fig. 3 Experimental results

number of hidden units for each RBM. Optimization aims at minimizing the error rate. The system shows better results than using normal tf-idf vectors for representing the document text. The system has an improved classification time and accuracy when compared to tf-idf-based SVM document classifier and is illustrated in Fig. 3.

6 Conclusion

The work proposed here is a supervised content-based approach for text classification. This work utilizes summarization to improve the accuracy of the text classification task. The proposed method uses deep learning which is the most recent approach for extracting the important features from the data. Feature extraction is done using the summarizer module, and these features are further utilized in text classification. Since the classifier is built on a reduced feature space, the time for building the classifier gets reduced and efficiency is improved. Accuracy is also increased as the classification model is built by training upon the most relevant features of the training sample. Hence, the accuracy of classification is limited only to the seen data. This approach can be improved further by implementing both feature extraction and classification using deep architecture.

References

1. Edmundson, H.P.: New methods in automatic extracting. J. ACM (JACM) **16**, 264–285 (1969)
2. Luhn, H.P.: The automatic creation of literature abstracts. IBM J. Res Dev. **2**, 159–165 (1958)
3. Mihalcea, R., Tarau, P.: TextRank: bringing order into texts. Assoc. Comput. Linguist. (2004)

4. Parveen, D., Strube, M.: Integrating importance, non-redundancy and coherence in graph-based extractive summarization. In: proceedings of International Joint Conference on Artificial Intelligence, pp. 1298–1304 (2015)
5. Patil, M.S., Bewoor, M.S., Patil, S.H.: A hybrid approach for extractive document summarization using machine learning and clustering technique. Int. J. Comput. Sci. Inf. Technol. **5**, 1584–1586 (2014)
6. Amini, M.R., Usunier, N., Gallinari, P.: Automatic text summarization based on word-clusters and ranking algorithms. In: European Conference on Information Retrieval, pp. 142–156. Springer, Berlin, Heidelberg (2005)
7. PadmaPriya, G., Duraiswamy, K.: An approach for concept-based automatic multi-document summarization using machine learning. Int. J. Appl. Inf. Syst. **3**, 49–55 (2012)
8. Kaikhah, K.: Text Summarization using Neural Networks (2004)
9. Igave, M.S., Gaikwad, C.M.: Int. J. Adv. Eng. Manag. Sci. **2**, 0952–0957 (2016)
10. Svore, K.M., Vanderwende, L., Burges, C.J.C.: Enhancing single-document summarization by combining RankNet and third-party sources. In: Emnlp-conll, pp. 448–457 (2007)
11. Denil, M., Demiraj, A., Kalchbrenner, N., Blunsom, P., de Freitas, N.: Modeling, Visualizing and Summarizing Documents with a Single Convolutional Neural Network (2014). arXiv:1406.3830
12. Kulkarni, A.R., Sarda, A.: Text summarization using neural networks and rhetorical structure theory. Int. J. Adv. Res. Comput. Commun. Eng. **4**, 49–52 (2015)
13. Cao, Z., Wei, F., Dong, L., Li, S., Zhou, M.: Ranking with recursive neural networks and its application to multi-document summarization. In: proceedings of the Association for the Advancement of Artificial Intelligence conference, pp. 2153–2159 (2015)
14. Zhong, S.-H., Liu, Y., Li, B., Long, J.: Query-oriented unsupervised multi-document summarization via deep learning model. Expert Syst. Appl. **42**, 8146–8155 (2015)
15. Basu, A., Watters, C., Shepherd, M.: Support vector machines for text categorization. In: Proceedings of the 36th Hawaii International Conference on System Sciences (2002)
16. kim, S.-B., Han, K.-S. Rim, H.-C., Myaeng, S.H.: Some effective techniques for naive Bayes text classification. IEEE Trans. Knowl. Data Eng. (2006)
17. Bijalwan, V., Kumar, V., Kumari, P., Pascual, J.: KNN based machine learning approach for text and document mining. J. Database Theory Appl. **7**, 61–70 (2014)
18. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. In: Advances in Neural Information Processing Systems, pp. 649–657 (2015)
19. Balaji, J., Geetha, T.V., Parthasarathi, R.: A Graph based query focused multi-document summarization. Int. J. Intell. Inf. Technol. (IJIIT) **10**, 16–41 (2014)
20. Jeonghun, Y.O.O.N., Dae-Won, K.: Classification based on predictive association rules of incomplete data. IEICE Trans. Inf. Syst. **95**, 1531–1535 (2012)
21. Kim, Y.: Convolutional Neural Networks for Sentence Classification (2014). arXiv:1408.5882
22. Lertnattee, V., Theeramunkong, T.: Class normalization in centroid-based text categorization. Inf. Sci. **176**, 1712–1738 (2006)
23. Li, W., Han, J., Pei, J.: CMAR: accurate and efficient classification based on multiple class-association rules. In: Proceedings of the IEEE International Conference on Data Mining series, pp. 369–376 (2001)
24. PadmaPriya, G., Duraiswamy, K.: An approach for text summarization using deep learning algorithm. J. Comput. Sci. **10**, 1–9 (2014)
25. Rahman, C.M., Sohel, F.A., Naushad, P., Kamruzzaman, S.M.: Text classification using the concept of association rule of data mining (2010). arXiv:1009.4582
26. Tan, S.: An improved centroid classifier for text categorization. Expert Syst. Appl. **35**, 279–285 (2008)
27. Thakkar, K.S., Dharaskar, R.V., Chandak, M.B.: Graph-based algorithms for text summarization. In: Proceedings of the 3rd International Conference on Emerging Trends in Engineering and Technology, pp. 516–519 (2010)