Submitted to Pattern Recognition (BRI623)

Document classification using 3-view of document representations and ensemble: TF-IDF, LDA and Doc2Vec

Eunji Jun[1], Deokseong Seo[2], Honggyu Jung[1]

[1]Department of Brain and Cognitive Engineering, Korea University
[2]Division of Industrial Management Engineering, Korea University

Abstract. Document classification is to assign a document to one or more predefined categories, which is one of the important research issues in the field of text mining. The documents can be classified using document representation. There are 3 widely used methods for document representation which are TF-IDF (Term Frequency Inverse Document Frequency), LDA (Latent Dirichlet Allocation) and Doc2Vec (Document-to-Vector). In this report, we propose an ensemble model that combines all the three document representation methods and analyze the performance. By experiments, we verify that the ensemble model is superior to others in terms of accuracy for Economic dataset.

1 Introduction

As the volume of data drastically grows, people are recently concerned about how to process and manage such big data. Especially, nowadays the number of documents that public institutions and enterprises hold for their business is also increasing rapidly. Moreover, documents are directly related to valuable information such as human resources, distribution, sales, and marketing. In this light, document management is one of the crucial issues to companies for their organized business activities.

One of the key points to manage documents is to classify the documents. The documents can be classified using document representation. Document representation is to transform the document from the full text version to a document vector which describes the contents of the document for reducing the complexity of the documents and making them easier to handle [1]. There are 3 widely used methods for document representation which are TF-IDF (Term Frequency Inverse Document Frequency) [2], LDA (Latent Dirichlet Allocation) [3] and Doc2Vec (Document-to-Vector) [4].

However, the document representation methods do not guarantee superior performance for various text datasets universally. To tackle this issue, we propose an ensemble model that combines all the three document representation methods. In order to classify documents, we use the Naïve Bayes and the decision tree classifier, and then compare the performances.

The simulation results show that the ensemble model is superior to others in terms of accuracy for Economic dataset. This result suggests that the proposed model can be effectively used for text data sets related to economy.

The rest of this report is organized as follows. In Section 2, we introduce relate works. In Section 3, the three document representation methods and the proposed model are described. Simulation results are presented in Section 4 and discussions and conclusions are drawn in Section 5 and 6.

2 Related Works

We investigate three types of recent researches. We first introduce recent works related to document classification algorithms that use TF-IDF. Then, we provide the overview of the works that use LDA. Lastly, researches that utilize Doc2Vec are introduced.

2.1 TF-IDF for Document Classification

TF-IDE is a statistical numerical value that indicates how important a word is in a document when there are several documents. Khan et al. [2] uses TF-IDF to reduce the dimension of documents and classify the documents on Internet. Zang et al. [5] compares the performance of TF-IDF with LSI (Latent Semantic Indexing) and Multi-words methods for document classification. Jing et al. [6] propose a variable selection method based on TF-IDF and a VSM (Value Stream Mapping) model for classification. Yun-tao et al. [7] also propose a variation of TF-IDF.

2.2 LDA for Document Classification

LDA is a generative probabilistic model of a corpus. Its idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words [3]. Discrete data such as words in a document is suitable for LDA. The goal of LDA is to define latent topics based on the frequency of words in a document in a probabilistic manner and express the document in terms of the distribution of topics. LDA is utilized for similarity detection, classification, and abnormality search [3].

2.3 Doc2Vec for Document Classification

Doc2Vec, or paragraph vector is an unsupervised algorithm that learns fixed-length feature representations from variable-length pieces of texts such as sentences, paragraphs, and documents. It represents each document by a dense vector which is trained to predict words in the document [4]. Note that Doc2Vec is independent to the order of words, which is not true for models using RNN [4].

3 Proposed Method

In this report, we propose an ensemble model that utilizes TF-IDF, LDA and Doc2Vec. Fig. 1 shows the framework of the proposed methodology. In the following sections, we explain each individual representation and introduce the ensemble model.

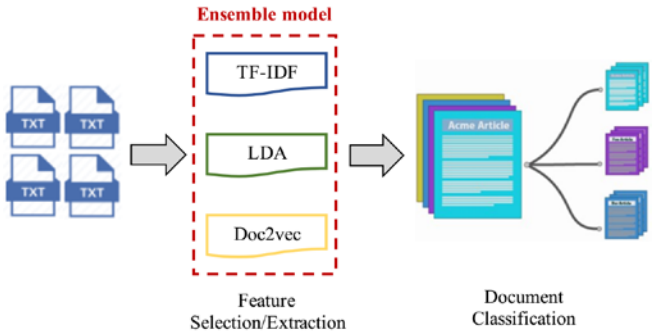

Figure 1. The overview of the proposed method

3.1 TF-IDF

TF-IDF can be calculated by multiplying term frequency and its inverse document frequency. This is a statistical indicator to find what words are important in a document.

Word frequency often refers to the indicator associated with how often a word appears in the entire set of documents. The higher the value, the more important the word is to characterize the document. However, sometimes, words with high frequencies are necessarily important. This is because there is a case where such repeated words do not represent the characteristics of a document. For this case, we use the concept of inverse document frequency. This factor is calculated by dividing document frequency into the total number of documents and taking the logarithm. Thus, given *i*-th word and *j*-th document, the weight w_{ij} can be obtained as follows.

$$w_{ij} = tf_{ij} \times \log\left(\frac{N}{df_i}\right)$$

, where tf_{ij} is the term frequency of *i*-th word and *j*-th document, df_i is the document frequency of *i*-th word, and idf_i is defined by $\log\left(\frac{N}{df_i}\right)$.

TF-IDF is widely used as a traditional probabilistic model for information retrieval. There exist several works to define the term frequency and the inverse document frequency terms. In this report, we use the term frequency by counting the word frequency and use $\log_2$ for the inverse document frequency. Fig. 2 shows the table that is a common method to illustrate the weights of TF-IDF.

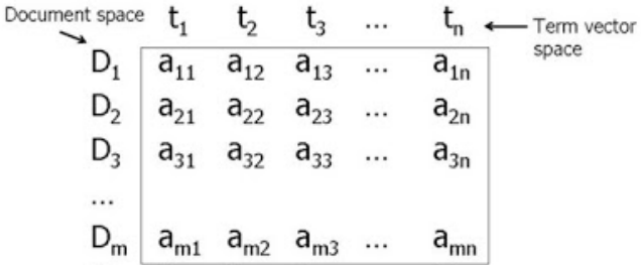

Figure 2. An example of the table that expresses TF-IDF weights

3.2 LDA

LDA defines latent variables for topics probabilistically in terms of word frequencies. This is also called an unsupervised generative topic model. LDA assumes that word frequencies are from latent topics, and documents is composed of mixture distributions of topics. By considering a joint probability, we infer the probability distribution θ of topics and the probability distribution β of generating specific words. Then, using the above process, we generate a document.

To be specific, we first define a set of documents $D = \{d_1, d_2, \ldots, d_{|D|}\}$, a set of labels $C = \{C_1, C_2, \ldots, C_M\}$, each document $d = \{w_1, w_2, \ldots, w_N\}$, and *i*-th word $w_i = (0, 0, 0, \ldots, 1, 0, 0, \ldots, 0)$. Fig. 3 shows the graphical model for LDA.

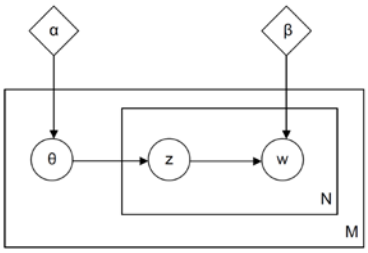

Figure 3. The graphical model for LDA

Then, we assume that the latent variables θ related to topics follow Dirichlet distribution with parameter α,

$$p(\theta|\alpha) = \frac{\Gamma(\Sigma_{i=1}^{k}\alpha_i)}{\Pi_{i=1}^{k}\Gamma(\alpha_i)}\prod_{i=1}^{k}\theta_i^{\alpha_i-1}$$

and the latent variables z_n of word w_n follow Multinomial distribution with parameter θ.

$$p(z|\theta) = \prod_{i=1}^{k}\theta_i^{z_i},\ \sum_{i=1}^{k}\theta_i = 1,\ \theta_i \geq 0$$

The word w_n follows Multinomial distribution with parameters z_n and β where $\beta = k \times V$ is a parameter matrix.

$$p(w_n|z_n, \beta) = \prod_{j=1}^{V} \left(\sum_{i=1}^{k} z_{n,i} \beta_{i,j} \right)^{w_{n,j}}$$

Moreover, given α, β, we first generate a joint distribution of θ, z, d before obtaining the distribution of d.

$$p(\theta, z, d|\alpha, \beta) = p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta) p(w_n|z_n, \beta)$$

Then, we generate a marginal distribution of d as follows.

$$\begin{aligned} p(d|\alpha, \beta) &= \int_{\theta}^{1} \sum_{z}^{1} p(\theta, z, w|\alpha, \beta) \\ &= \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \int_{\theta} \left(\prod_{i=1}^{k} \theta_i^{\alpha_i - 1} \right) \left(\prod_{n=1}^{N} \sum_{i=1}^{k} \prod_{j=1}^{V} (\theta_i \beta_{i,j}^{1})^{w_{n,j}} \right) \end{aligned}$$

By this, we define a log likelihood function by

$$L(D; \alpha, \beta) = \log p(D|\alpha, \beta) = \sum_{d \in D}^{1} \log p(d|\alpha, \beta)$$

Since it is difficult to maximize the above function, we calculate a lower bound by using Jensen's inequality $f(E[X]) \geq E[f(X)]$ for all concave functions.

$$\begin{aligned} \log p(d|\alpha, \beta) &= \log \int_{\theta}^{1} \sum_{z}^{1} p(\theta, z, d|\alpha, \beta) \, d\theta \\ &= \log \int_{\theta}^{1} \sum_{z}^{1} \frac{p(\theta, z, d|\alpha, \beta)}{q(\theta, z)} q(\theta, z) \, d\theta \\ &= \log E_q \left[\frac{p(\theta, z, d|\alpha, \beta)}{q(\theta, z)} \right] \\ &\geq E_q[\log p(\theta, z, d|\alpha, \beta)] - E_q[\log q(\theta, z)] \triangleq L \end{aligned}$$

To further facilitate the maximization of L, we set variational parameters γ, ϕ, and the approximation of probability distributions θ and z is defined by $q(\theta, z|\gamma, \phi) = q(\theta|\gamma) \prod_{n=1}^{N} q(z_n|\phi_n)$ where $q(\theta|\gamma) \sim Dir(\gamma)$, $q(z_n|\phi_n) \sim Multi(\phi_n)$. Fig. 4 shows the graphical model of variational distribution.

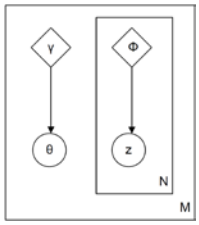

Figure 4. The graphical model of variational distribution

By the approximation above, L can be expressed by

$$L(\gamma, \phi;\ \alpha,\ \beta) = E_q[\log p(\theta|\alpha)] + \sum_{n=1}^{N} E_q[\log p(z_n|\theta)] + \sum_{n=1}^{N} E_q[\log p(w_n|z_n,\ \beta)]$$
$$-E_q[\log q(\theta|\gamma)] - \sum_{n=1}^{N} E_q[p(z_n|\phi_n)]$$

To maximize L, we calculate the derivatives of $\gamma,\ \phi,\ \alpha,\ \beta$ and find the update rules as follows.

$$\phi_{n,\,i} \propto \beta_{i,w_n} \exp\left(\Psi(\gamma_i)\right)$$
$$\gamma = \alpha + \sum_{n=1}^{N} \phi_n$$
$$\beta_{i,\,j} \propto \sum_{d=1}^{M} \sum_{n=1}^{1} N_d \phi_{d,\,n,\,i} w_{d,\,n,\,j}$$

where α is estimated using the Newton's method.

Fig. 5 shows an illustration of LDA for document representation.

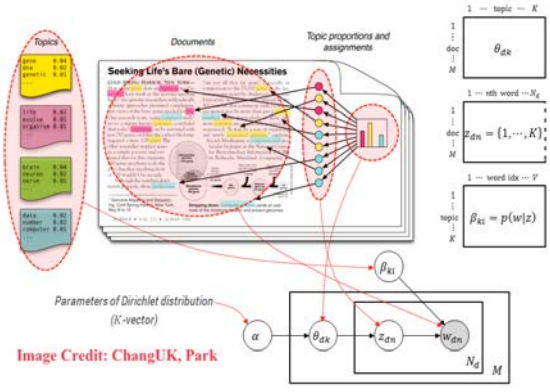

Figure 5. An illustration of LDA for document representation

3.3 Doc2Vec

When using a feature vector in terms of texts, BoW (Bag of Words) is one of a common approach. However, BoW has several disadvantages. First, BoW does not consider the sequence of words. Thus, a different sentence that has the same words can be expressed in the same BoW vectors. Second, the number of dimensions is high because only the number of word frequencies is considered. Lastly, it is difficult to understand the order of words in a sentence, or the exact meaning of a word [4].

To overcome these limitations, various ideas came from word vectors to paragraph vectors. The paragraph vector is obtained by adding paragraph information to the existing word vector method as follows.

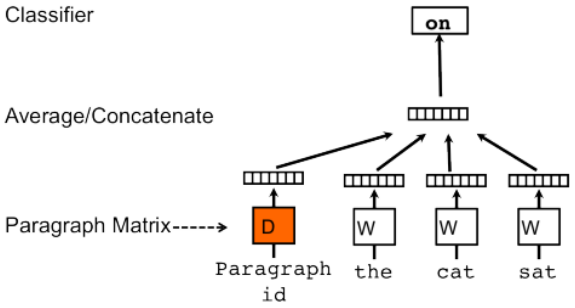

Figure 6. An illustration of the paragraph vector

Doc2vec is a concept that extends the vector interpretation from word to sentence, paragraph, and document. We learn to predict the next word in a given context by associating a few word vectors in a paragraph with a paragraph vector.

The operation of this algorithm consists of a learning step for updating parameters, and a reasoning step for obtaining a paragraph vector for a new paragraph. First, in the learning step, the word vector W, the Softmax weights U, and the paragraph vector D are updated with respect to the given paragraph. Every paragraph is linked to a vector represented by a column of matrix D. Further, all words are linked to a unique vector represented by a column of matrix W. A separate paragraph is regarded as a different word and serves as a memory for memorizing the missing part of the content and is referred to as a Distributed Memory Model of Paragraph Vectors (PV-DM). Paragraph vectors and word vectors are learned by the stochastic gradient descent method and the gradient is obtained by backpropagation. In the stochastic gradient descent, a fixed length of context is extracted from arbitrary paragraphs, the error gradient is calculated, and the parameters are updated.

Given words $w_1, w_2, w_3, \ldots, w_T$ from training data, the purpose of the paragraph vector model is to maximize the average log probability.

$$\frac{1}{T}\sum_{t-k}^{T-k} \log p(w_t | w_{t-k}, \ \ldots, \ w_{t+k})$$

, where $\log p(w_t|w_{t-k}, \ldots, w_{t+k}) = \frac{e^{y w_t}}{\sum_i e^{y_i}}$.

Since y_i is a non-normalized log probability for each result, it is calculated as follows.

$$y = b + Uh(w_{t-k}, \ldots, w_{t+k}; W, D)$$

After fixing W, U, and b to predict the word, then we perform the gradient descent on D and add more columns to obtain the D vector for the new paragraph. These learned paragraph vectors can be used as features for paragraphs [4].

3.4 Proposed Ensemble Model

We propose ensemble models that utilize the above three document representations. We basically consider Naïve Bayes classifier.

Based on the classification result, we first perform majority voting. To be specific, we obtain the classification results of all the three document representations and make the final decision with the majority. The second ensemble method is to use the probabilities from Naïve Bayes. We obtain the average probability from the three representations. Then, we make the decision by considering the highest average probability. We also use the two ensemble models for the decision tree classifier.

4 Experiment

For experiments, we used three datasets as follows.

Table 1. The three datasets for document classification

Dataset	Description	Range	Row	Source
Economic	Whether a news article data is associated with the US economy	No : 6,458 (82.12%) Yes : 1,406 (17.88%)	7,864	http://www.crowdflower.com /data-for-everyone
Ohsumed	Articles related abstracts of medical data	C04 : 2,630 (50.77%) C14 : 2,550 (49.23%)	5,180	http://disi.unitn.it/moschitti /corpora.htm
Reuters	21578 documents obtained by the Reuters news data	Earn : 3,953 (51.67%) Non-earn : 4,697 (48.33%)	7,650	http://www.daviddlewis.com/resources /testcollections/reuters21578/

We evaluated the performance of the proposed Ensemble models (Voting and probability method) and compared with individual models for three datasets. The code for the experiments can be downloaded at GitHub[1]. The following table shows the results.

1. https://github.com/PRMLteam5/PRML-final-project

Table 2. Results for Economy dataset

Classifier	Representation	Accuracy(%)	Recall(%)	Precision(%)	F1-measure(%)
Naive Bayesian	TF-IDF	63.68±0.87	68.07±1.97	28.57±1.09	40.23±1.19
	LDA	50.85±1.08	73.80±2.07	22.93±1.06	34.98±1.33
	Doc2Vec	75.56±0.93	40.79±1.86	34.71±1.83	37.47±1.46
	Ensemble(Voting)	79.64±8.83	4.02±13.33	18.35±16.02	3.24±6.04
	Ensemble(Prob.)	78.35±10.06	6.24±16.15	18.37±17.60	4.40±8.04
Decision tree	TF-IDF	73.45±0.86	27.37±2.38	26.53±1.94	26.91±1.87
	LDA	70.95±0.91	23.21±2.19	21.28±1.74	22.17±1.76
	Doc2Vec	71.00±0.86	23.17±2.04	21.53±1.77	22.31±1.75
	Ensemble(Voting)	80.14±7.06	3.10±9.95	17.14±12.85	3.02±5.09
	Ensemble(Prob.)	78.97±10.01	5.39±16.12	22.26±18.91	3.85±7.00

Table 3. Results for Ohsumed dataset

Classifier	Representation	Accuracy(%)	Recall(%)	Precision(%)	F1-measure(%)
Naive Bayesian	TF-IDF	86.85±0.71	85.89±1.05	87.17±1.17	86.52±7.79
	LDA	75.20±0.91	77.04±1.50	73.70±1.37	75.32±1.02
	Doc2Vec	65.31±1.26	59.64±1.89	66.58±1.94	62.90±1.60
	Ensemble(Voting)	52.92±3.25	31.42±30.84	57.53±11.29	31.45±22.48
	Ensemble(Prob.)	52.66±2.35	27.00±29.03	57.64±11.93	28.35±21.16
Decision tree	TF-IDF	86.66±0.77	85.85±1.23	86.87±1.07	86.35±0.80
	LDA	75.16±0.95	77.23±1.44	73.60±1.43	75.36±1.01
	Doc2Vec	65.54±1.35	59.72±1.84	66.94±1.89	63.10±1.51
	Ensemble(Voting)	52.66±2.83	33.56±33.09	58.80±11.59	32.47±21.72
	Ensemble(Prob.)	52.45±2.61	29.73±31.33	56.94±9.94	29.93±21.50

Table 4. Results for Reuters dataset

Classifier	Representation	Accuracy(%)	Recall(%)	Precision(%)	F1-measure(%)
Naive Bayesian	TF-IDF	94.24±0.39	97.25±0.53	92.69±0.73	94.91±0.35
	LDA	82.48±0.67	79.88±1.14	87.38±0.99	83.45±0.72
	Doc2Vec	65.72±0.76	56.69±1.18	75.14±1.18	64.61±0.92
	Ensemble(Voting)	53.37±5.41	29.85±18.96	71.14±10.74	38.45±14.83
	Ensemble(Prob.)	53.26±4.80	28.89±18.18	72.18±10.26	37.76±14.57
Decision tree	TF-IDF	94.19±0.40	97.28±0.54	92.57±0.74	94.86±0.36
	LDA	82.61±0.70	79.97±1.11	87.48±0.95	83.55±0.74
	Doc2Vec	65.70±0.84	56.78±1.27	75.15±1.31	64.68±1.03
	Ensemble(Voting)	54.23±5.42	30.82±19.28	73.93±9.82	39.76±14.69
	Ensemble(Prob.)	53.83±5.15	29.96±19.43	72.43±10.66	38.66±15.09

5 Discussion

We identified that from the results, the performance of the two classifiers is similar, except for the economic dataset and most individual models outperform ensemble models, but only accuracy in the economic dataset. Hence, we need different approaches to respective dataset for effective document representation.

6 Conclusion

In this work, we constructed an ensemble model that combines all three document representation methods and compare the performance using the proposed ensemble model with the individual models that use TF-IDF, LDA and Doc2Vec, respectively. Furthermore, we also compared performances of two different classifiers used for document classification.

References

[1] Charles T. Meadow, "Text Information Retrieval Systems," Academic Press, 1992

[2] A. Khan, B. Baharudin, L. Lee, and K. khan, "A Review of Machine Learning Algorithms for Text-Documents Classification", Journal of Advances in Information Technology, Vol. 1, No. 1, pp. 4-20, Feb 2010.

[3] D. Blei, A. Ng, and M. Jordan, "Latent Dirichlet Allocation," Journal of Machine Learning Research, Vol.3, pp. 993-1022, Jan. 2003.

[4] Q. Le, and T. Mikolov, "Distributed Representations of Sentences and Documents," ICML, Vol. 14, pp. 1188-1196, May 2014.

[5] W. Zhang, T. Yoshida, and X. Tang, "A Comparative Study of TF-IDF, LSI, and Multi-Words for Text Classification", Expert Systems with Applications, Vol. 38, No. 3, pp. 2758–2765, Mar. 2011.

[6] L. JING, H. HUANG, and H. SHI, “Improved Feature Selection Approach TF-IDF in Text Mining", in International Conferene on Machine Learning and Cybernetics, Vol. 2, Nov. 2002.

[7] Z. Tao, G. Ling, and W. Cheng, "An Improved TF-IDF Approach for Text Classification", Journal of Zhejiang University SCIENCE A, Vol. 6, No. 1, pp. 49-55, Aug. 2005.