English Composition Off-Topic Detection Based on LDA and word2vec: Postprint

Authors: Qu Qiang, Cui Rongyi, Zhao Yahui

Date: 2018-05-20T00:00:00+00:00

Abstract

To address the issue that current domestic English essay assisted grading systems lack accurate and efficient off-topic detection algorithms, this paper proposes an off-topic detection algorithm combining LDA and word2vec. The algorithm utilizes the LDA model for document modeling and employs word2vec for document training, leveraging the obtained document topics and semantic relationships between words to calculate probability-weighted sums for each topic and its feature words in the document, ultimately filtering out off-topic essays by setting a reasonable threshold. In experiments, the optimal number of topics was determined by varying the document topic numbers and analyzing the resulting F-values. Experimental results demonstrate that the proposed method is more effective than vector space model-based approaches, capable of detecting more off-topic essays with higher accuracy and achieving an F-value exceeding 89%, thereby realizing intelligent processing of essay off-topic detection that can be effectively applied in English essay instruction.

Full Text

Preamble

Off-Topic Detection for English Essays Based on LDA and word2vec

Qu Qiang, Cui Rongyi, Zhao Yahui†
(Intelligent Information Processing Laboratory, Department of Computer Science & Technology, Yanbian University, Yanji, Jilin 133002, China)

Abstract: Current English composition grading systems in China lack accurate and efficient off-topic detection algorithms. To address this problem, this paper proposes a novel off-topic detection algorithm that combines LDA and word2vec. The algorithm employs LDA to model documents and word2vec for document training. By leveraging the obtained semantic relationships between document topics and words, it calculates the probability-weighted sum of each topic and

its feature words within documents. Finally, by setting a reasonable threshold, off-topic essays are identified. In the experiments, the optimal number of topics was determined by evaluating different F-values obtained through varying the document topic count. Experimental results demonstrate that the proposed method is more effective than vector space model-based approaches, detecting more off-topic essays with higher accuracy and achieving an F-value above 89%. This approach enables intelligent processing of off-topic essay detection and can be effectively applied in English composition instruction.

Keywords: off-topic essay detection; vector space model (VSM); latent Dirichlet allocation (LDA); semantic relations between words

0 Introduction

Writing is a crucial means of expressing emotions and conveying information, with theme being the soul of any composition. A well-written essay must have a clear and appropriate central theme; otherwise, it can cause confusion and misunderstanding, or even result in off-topic writing. Essays may deviate from the topic for various reasons, whether intentional or due to unintentional submission errors [**?**].

Off-topic detection aims to determine whether an essay strays from its assigned topic, with its core task being the calculation of text similarity [**?**]. Text similarity serves as a metric for measuring the degree of resemblance between texts. Currently, the most commonly used and classic text representation model is the Vector Space Model (VSM), with the TF-IDF algorithm being the most widely adopted method for text similarity calculation. This approach characterizes word weights using term frequency in documents and computes text similarity through cosine similarity between vectors. Although the bag-of-words model is simple and somewhat effective, it ignores the semantic information inherent in words and fails to consider semantic similarity between terms. For instance, the English words "like" and "love" both express affection, yet in the vector space model, they are treated as two independent terms. To address this limitation, researchers have proposed word expansion methods using dictionaries such as WordNet and HowNet. Reference [**?**] proposed a method for calculating English word semantic similarity based on WordNet expansion, while reference [**?**] introduced a method using HowNet to compute lexical semantic similarity. However, these methods heavily rely on manually constructed dictionaries and encounter significant challenges when encountering new words.

To overcome these shortcomings, this paper proposes a novel text similarity calculation method for English essay off-topic detection. The algorithm employs LDA to model document collections, extracting each document' s topics and their characteristic words along with their probability distributions. By integrating the semantic relationships between words obtained through word2vec training, it calculates the probability-weighted sum of each topic in the document. This approach effectively identifies off-topic essays and, compared with

traditional vector space models, not only captures more semantic information between terms but also obtains topic distribution patterns through document modeling, thereby compensating for the limitation of ignoring word semantics in conventional VSM methods.

1 LDA Modeling

1.1 LDA Model

The LDA (Latent Dirichlet Allocation) model, proposed by Blei et al., is a three-layer Bayesian generative model comprising "document–topic–word" [**?**]. It extends the probabilistic latent semantic analysis (pLSA) model into a three-layer Bayesian probability framework containing word, topic, and document structures. This unsupervised machine learning algorithm can identify latent topic information in large-scale document collections or corpora. It adopts the bag-of-words approach, treating each document as a word frequency vector to transform textual information into numerical data suitable for modeling and computation.

The model operates under the premise that documents are composed of several latent topics, which in turn consist of specific vocabulary terms, while ignoring syntactic structure and word order [**?**]. The LDA topic model can be represented as a probabilistic graphical model, as shown in [Figure 1: see original paper].

In the figure, M represents the number of documents in the collection, K represents the number of topics, and N represents the number of feature words in each document. θ_m denotes the probability distribution of all topics in the m-th document, while ϕ_k represents the probability distribution of feature words under a specific topic.

For each document in the corpus, LDA provides the following generative process:

a) Sample the topic distribution θ_m for the m-th document from Dirichlet distribution Dir(α)

b) Sample the topic z_{mn} for the n-th word in the m-th document from the multinomial distribution Mult(θ_m)

c) Sample the word distribution $\phi_{z_{mn}}$ for topic z_{mn} from Dirichlet distribution Dir(β)

d) Sample the final word w_{mn} from the multinomial distribution Mult($\phi_{z_{mn}}$)

Since LDA assumes that an article contains multiple topics, with each topic corresponding to different words, the document generation process works as follows: first, select a topic with a certain probability, then select a word under that topic with a certain probability, generating the first word of the article. Repeating this process generates the entire document, with the assumption that words are independent of each other.

This paper employs the MCMC [**?**] method, specifically the Gibbs sampling [**?**] algorithm, for parameter estimation. This can be viewed as the inverse of the document generation process: given the document collection (the result of generation), parameter values are estimated. Based on the model diagram in [Figure 1: see original paper], the probability distribution of a document can be obtained.

1.2 Gibbs Sampling

Parameter estimation in LDA model construction requires approximate inference methods, with commonly used approaches including variational Bayes inference, expectation propagation, and collapsed Gibbs sampling. The Gibbs sampling-based parameter inference method is easy to understand, simple to implement, and highly effective for extracting topics from large-scale text collections [**?**]. Consequently, Gibbs sampling has become the most popular method for LDA model extraction.

Gibbs sampling is an MCMC (Markov chain Monte Carlo) algorithm, and Griffiths proposed applying Gibbs sampling to LDA model parameter estimation [**?**]. The two most important parameters in the LDA model are the probability distribution of feature words under each topic and the probability distribution of topics in each document.

The specific steps of the Gibbs sampling algorithm are as follows (for detailed derivation, see reference [**?**]):

a) Initialization: Topic z_i is initialized to a random integer between 1 and T, with i looping from 1 to N, where N is the total number of word occurrences in the corpus. This constitutes the initial state of the Markov chain.

b) Iterative sampling: After sufficient iterations, when the Markov chain approaches the target distribution, the topic z_i can be estimated according to the following formula:

$$P(z_i|z_{\neg i}, w) \propto \frac{n_{t,k}^{(\neg i)} + \beta_t}{\sum_{v=1}^{V}(n_{v,k}^{(\neg i)} + \beta_v)} \times \frac{n_{m,k}^{(\neg i)} + \alpha_k}{\sum_{j=1}^{K}(n_{m,j}^{(\neg i)} + \alpha_j)}$$

where $n_{t,k}$ represents the count of feature word t appearing in topic k, and $n_{m,k}$ represents the count of topic k appearing in document m. The values of ϕ and θ are indirectly obtained through Gibbs sampling, denoted as posterior probabilities $\hat{\phi}$ and $\hat{\theta}$, with calculation formulas:

$$\hat{\phi}_{k,t} = \frac{n_{t,k} + \beta_t}{\sum_{v=1}^{V}(n_{v,k} + \beta_v)}$$

$$\hat{\theta}_{m,k} = \frac{n_{m,k} + \alpha_k}{\sum_{j=1}^{K}(n_{m,j} + \alpha_j)}$$

1.3 LDA Modeling Process

Before conducting LDA modeling, the given document collection $\{d_1, d_2, \ldots, d_M\}$ must be preprocessed. For each document $d_m \in D$, preprocessing 主要包括分词、去停用词、去标点符号等操作，将处理后的每个词项用空格分隔保存，整理后获得对应的语料集，将其作为下一步的处理数据。

The preprocessed corpus is presented as a single document to construct the document-term matrix. The final text representation is shown in equation (4):

$$D = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{M1} & w_{M2} & \cdots & w_{Mn} \end{bmatrix}$$

where M represents the total number of documents and m represents the document index. w_{mn} denotes the n-th term in the m-th document.

For the document-term matrix obtained above, the LDA model is applied to model the preprocessed document collection D, yielding each document' s topics z and their probability distribution θ, as well as each topic' s feature words w and their probability distribution ϕ.

2 Topic Relevance Calculation Based on LDA and word2vec

2.1 word2vec

In recent years, with the rapid development of deep learning, word vector representation methods based on neural networks for automatic feature extraction have gained increasing attention among researchers. Mikolov et al. proposed the word2vec language model [**?**] for computing word vectors by drawing on Bengio' s NNLM (Neural Network Language Model) and Hinton' s Log-Linear model. In 2013, Google released word2vec as an open-source tool for training word vectors [**?**]. The word2vec [**?**] model can quickly and effectively represent a word as a real-valued vector based on a given corpus. By leveraging contextual information of words, it simplifies text content processing to K-dimensional vector operations, where similarity in vector space can represent semantic similarity in text.

The word vectors output by word2vec can be used for various NLP tasks such as sentiment classification, synonym finding, and part-of-speech analysis. Another characteristic of word2vec is its efficiency; Mikolov et al. [**?**] noted that an optimized single-machine version can train hundreds of billions of words per day, providing new approaches for applied research in natural language processing.

word2vec includes two training models with architectures: CBOW (Continuous Bag-Of-Words) and Skip-Gram. The schematic diagram is shown in [Figure 2: see original paper]. As evident from the figure, both CBOW and Skip-gram models consist of an input layer, projection layer, and output layer. The CBOW model predicts the current word' s vector from its context, representing the continuous words corresponding to the current word' s context in bag-of-words form and selecting the sum of context word vectors as the training target. Conversely, the Skip-gram model generates word vectors in the opposite manner, using only the current word to predict its context. Through these two models, word2vec comprehensively considers contextual information, thereby achieving better results.

2.2 Topic Relevance Calculation

Before calculating topic relevance for documents, the document collection must be trained using word2vec to obtain semantic information between terms. For the English corpus in this study, word2vec identifies different words based on spaces between them. After word2vec training, each word obtains a vector representation, and the cosine similarity between two vectors represents their semantic similarity distance. A larger cosine value indicates greater semantic similarity between the two words. For example, for two n-dimensional vectors $a = (x_{11}, x_{12}, \dots, x_{1n})$ and $b = (x_{21}, x_{22}, \dots, x_{2n})$, the cosine similarity is calculated as:

$$\cos(a, b) = \frac{\sum_{k=1}^{n} x_{1k} x_{2k}}{\sqrt{\sum_{k=1}^{n} x_{1k}^2} \sqrt{\sum_{k=1}^{n} x_{2k}^2}}$$

The trained word vector representations are stored in files for subsequent similarity calculations. For instance, specifying the word "woman" will display "man" as the closest word with a cosine distance of 0.685 after training. The training results transform semantic information between words in the document into vector representations for preservation.

Based on the above information, for each term w_j in each document, word2vec is used to compute the cosine similarity between the term and feature words w_t under topic z_i. The relevance between term w_j and topic z_i is the probability-weighted sum of cosine similarities with all feature words under z_i, calculated as:

$$S(w_j, t) = \sum_{n=1}^{N} \cos(w_j, w_n) \times P(w_n|t)$$

where N is the number of feature words in topic t.

The relevance between term w_j and document d_m is then the probability-weighted sum of its relevance to each topic:

$$S(w_j, d_m) = \sum_{i=1}^{K} S(w_j, t_i) \times P(t_i|d_m)$$

Finally, the total relevance of document d_m is the sum of $S(w_j, d_m)$ values for all its terms:

$$S(d_m) = \sum_{j=1}^{J} S(w_j, d_m)$$

3 Off-Topic Detection Algorithm

The off-topic detection algorithm first preprocesses the document collection to establish a document-term matrix. It then applies LDA modeling to obtain document topics z and their distribution θ, as well as feature words and their distribution ϕ under each topic. Next, word2vec trains the document collection and saves the results. Finally, the information from LDA and word2vec is combined, and a selected threshold is applied to filter each document and identify off-topic essays.

The algorithm leverages LDA to obtain document topic information while using word2vec-trained word vectors to capture more accurate semantic information contained in terms. These factors contribute significantly to effective off-topic essay detection.

The specific algorithm steps are designed as follows:

a) **Preprocess the document collection.** For English documents, preprocessing includes tokenizing by spaces, converting uppercase letters and words at sentence beginnings and proper nouns to lowercase, removing stop words (e.g., “the” , “a” , “an”), removing all punctuation, and extracting word stems (removing plural forms, -ing, -ed suffixes). For example, the sentence “we all like the book, it is so interesting.” becomes “like book interest” after preprocessing.

b) **Construct the document-term matrix.** The vectorized document representation takes the form of equation (4), where row i represents the i-th document, the number of columns in row i indicates the term count in that document, and column j in row i corresponds to the j-th term in the i-th document.

c) **Perform LDA modeling.** For each document in the document-term matrix, equations (1) and (2) are used to obtain the topic probability distribution θ_m for document m and the feature word probability distribution ϕ_k for topic k. Sorting by probability values yields each document’

s topics with their probability distributions and feature words with their probability distributions. For example, if an English document has 60% probability distribution on the education topic and 40% on children, feature words under the education topic would include "school", "students", "education", while the children topic would contain "children", "women", "family".

d) **Train word vectors with word2vec.** Using the preprocessed document collection as input, word2vec training outputs a vector representation for each word. The generated word vectors are used to calculate distances (similarities) with specified words via equation (6).

e) **Calculate topic relevance using LDA and word2vec.** For each term in each document, compute its cosine similarity with feature words under each topic obtained from LDA modeling using equation (7) to calculate the probability-weighted sum for each feature word. Then apply equation (8) to compute the weighted sum across topics. Finally, sum the topic relevance values for all terms in the document using equation (9) and identify off-topic essays based on the threshold.

The algorithm combines the advantages of LDA and word2vec. The word2vec training results enable more accurate expression of semantic relationships between words in documents, allowing the LDA modeling to effectively determine whether the document's topics are appropriate. This yields document topic relevance in a low-dimensional semantic space for off-topic detection.

4 Experimental Results and Comparative Analysis

The experiments collected 1230 university English essays across six different topics, with 205 essays per topic. Each essay was manually annotated with scores, with essays scoring below 5 out of 15 considered off-topic. The detected off-topic documents were compared against manually annotated results, evaluating accuracy, recall, and F-value to verify the algorithm's effectiveness and practicality.

Precision (P) is the ratio of correctly detected off-topic documents to all documents detected as off-topic. Recall (R) is the ratio of correctly detected off-topic documents to all actual off-topic documents. Let T represent correctly detected off-topic documents, A represent total detected off-topic documents, and B represent total actual off-topic documents:

$$P = \frac{T}{A} \times 100\%$$

$$R = \frac{T}{B} \times 100\%$$

As equations (10) and (11) imply, precision and recall typically have an inverse relationship. The F-value harmonizes this trade-off as a comprehensive metric balancing both:

$$F = \frac{2PR}{P + R}$$

Therefore, the F-value serves as the final evaluation metric. Experiments with different K values yielded corresponding F-values, with average F-values across different topic numbers shown in [Figure 4: see original paper].

The LDA model used Gibbs sampling in experiments. Initially assuming topic number $K = 2$, hyperparameters were set to empirical values [**?**] $\alpha = 50/K$ (varying with topic number) and $\beta = 0.01$ (fixed). To ensure accuracy, Gibbs sampling iterations were set to 1000.

When training the document collection with word2vec, multiple hyperparameters affect training quality and speed. Reference [**?**] provides guidance on parameter selection. Based on experimental requirements, word2vec training parameters were configured as shown in .

Table 1: word2vec Parameter Settings

Parameter	Value	Description
size	200	Word vector dimension
window	5	Context window size
min-count	3	Minimum word frequency threshold
cbow	0	Use CBOW model (0=yes)

Experiments revealed that hyperparameter α changes with document topic number K, exhibiting an inverse relationship. Larger K values result in smaller α values, indicating more topics per document. For feature words under each topic, reference [**?**] demonstrated that selecting 5 feature words yields good results, so this experiment uniformly selected 5 feature words per topic per document.

Using the optimal topic number determined from experiments, the algorithm detected off-topic documents. Comparing these with manually annotated off-topic documents yielded average precision of 91.86%, average recall of 88.78%, and average F-value of 89.81% across six topics.

A comparative experiment using the TF-IDF algorithm based on vector space model was conducted on the same English essay corpus. After preprocessing, TF-IDF represented documents as term vectors. Each essay' s cosine similarity with five given sample essays was computed and averaged as the final result, with threshold-based filtering identifying off-topic essays per topic. Using the same evaluation method, this baseline achieved an average F-value of 77.4% across six experiments.

Table 2: Off-Topic Detection Results with Topic Number = 2

Metric	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Average
Precision	94.74%	93.33%	93.75%	86.67%	61.54%	75%	84.17%
Recall	94.74%	100%	100%	86.67%	75%	89.40%	94.74%
F-value	96.55%	96.77%	86.67%	69.57%	75%	86.55%	-

With $K = 2$, the off-topic detection achieved average precision of 84.17%, average recall of 89.40%, and average F-value of 86.55%. To optimize performance, experiments varied the document topic number K across values $\{2, 3, 5, 10, 15, 20, 25, 30\}$. For each K, thresholds were selected to obtain off-topic detection results. Comparison with manually annotated scores yielded precision, recall, and F-value per topic, with final averages computed. Since the F-value comprehensively considers both precision and recall, it served as the primary evaluation metric.

[Figure 4: see original paper] clearly shows how the average F-value changes with topic number, peaking when $K = 15$. Therefore, the optimal topic number was determined to be 15. Additionally, experiments observed that iteration time increases with topic number.

The F-value comparison between the proposed method and TF-IDF is shown in [Figure 5: see original paper]. Analysis reveals that the proposed algorithm performs better, accurately analyzing semantic information of terms in documents while obtaining topic distribution patterns—both crucial for off-topic detection. While maintaining high precision, the algorithm detects more off-topic essays than the TF-IDF approach, showing significant F-value improvement and demonstrating reliability. In comparative experiments, the proposed method successfully identified all off-topic essays for two topics with high precision, whereas the TF-IDF algorithm missed some, including zero-score essays that were off-topic but not blank. This highlights TF-IDF' s primary limitation: it relies solely on term frequency and inverse document frequency, making it ineffective at capturing word semantics.

The proposed algorithm achieves over 88% off-topic detection with high precision, proving more effective than vector space model-based TF-IDF. It efficiently filters off-topic essays within short timeframes, saving teachers considerable grading effort.

5 Conclusion

This paper utilizes LDA for document modeling to conveniently extract document topics and their feature words. Word2vec training further enhances accurate expression of semantic relationships between words. LDA and word2vec

are then combined for topic relevance calculation. Experimental results demonstrate effective off-topic essay detection. The proposed algorithm provides intelligent assistance for English teaching and essay grading competitions, enabling computers to effectively simulate teachers' rapid, objective, fair, and automatic processing of English essays. It filters off-topic essays for each topic, reducing subjective factors in grading and improving efficiency while overcoming the limitation of manual methods that cannot quickly and effectively detect off-topic essays in large volumes.

A limitation of this work is that LDA topic number determination relies solely on F-value rather than computational theory for optimal topic selection. Given LDA' s extensibility, future work will focus on improving document modeling and topic number determination methods within the LDA framework.

References

[1] Chen Zhipeng, Chen Wenliang, Zhu Muhua. Improving off-topic essay detection using distributed word representations [**?**]. Journal of Chinese Information Processing, 2015, 29(5): 178-184, 203.

[2] Deane P. On the relation between automated essay scoring and modern views of the writing construct [**?**]. Assessing Writing, 2013, 18(1): 7-24.

[3] Zhai Yandong, Wang Kangping, Zhang Dongna, et al. A short text semantic similarity algorithm based on WordNet [**?**]. Acta Electronica Sinica, 2012, 40(03): 617-620.

[4] You Bin, Yan Yuesong, Sun Yingge, et al. An information content-based semantic similarity algorithm using HowNet [**?**]. Computer Systems & Applications, 2013, 22(01): 129-133.

[5] Zhang Zhifei, Miao Duoqian, Gao Can. Short text classification based on LDA topic model [**?**]. Journal of Computer Applications, 2013, 33(06): 1587-1590.

[6] Yao Quanzhu, Song Zhili, Peng Cheng, et al. Research on text classification based on LDA model [**?**]. Computer Engineering and Applications, 2011, 47(13): 150-153.

[7] Wang Zhenzhen, He Ming, Du Yongping. Text similarity computation based on LDA topic model [**?**]. Computer Science, 2013, 40(12): 229-232.

[8] Arora S, Ge R, Halpern Y, et al. A Practical Algorithm for Topic Modeling with Provable Guarantees [**?**]. Proc of International Conference on Machine Learning. 2012: 280-288.

[9] Farrahi K, Gaticaperez D. Discovering routines from large-scale human locations using probabilistic topic models [**?**]. ACM Trans on Intelligent Systems & Technology, 2011, 2(1): 1-27.

[10] Link W A, Eaton M J. On thinning of chains in MCMC [?]. Methods in Ecology & Evolution, 2012, 3(1): 112-115.

[11] Ma Haiyun. Research on test case generation technology based on Gibbs sampling [?]. Automation & Instrumentation, 2011, (02): 11+14.

[12] Tang Ming, Zhu Lei, Zou Xianchun. A document vector representation based on Word2Vec [?]. Computer Science, 2016, 43(06): 214-217, 269.

[13] Pennington J, Socher R, Manning C. Glove: Global Vectors for Word Representation [?]. Proc of Conference on Empirical Methods in Natural Language Processing. 2014: 1532-1543.

[14] Mikolov T, Chen K, Corrado G, et al. Efficient Estimation of Word Representations in Vector Space [?]. Computer Science, 2013.

[15] Mikolov T, Sutskever I, Chen K, et al. Distributed Representations of Words and Phrases and their Compositionality [?]. Advances in Neural Information Processing Systems. 2013: 3111-3119.

[16] Wang Peng, Gao Cheng, Chen Xiaomei. Research on text clustering based on LDA model [?]. Information Science, 2015, 33(01): 63-68.

[17] Hu Jiming, Chen Guo. Content theme mining and evolution based on dynamic LDA topic model [?]. Library and Information Service, 2014, 58(02): 138-142.

[18] Zhou Lian. Exploration of Word2vec working principle and applications [?]. Sci-Tech Information Development & Economy, 2015, 25(2): 145-148.

[19] Wu Kai, Wang Ying. Preliminary study on microblog user knowledge discovery based on mention relationships [?]. Library and Information, 2015(2): 123-127.

Note: Figure translations are in progress. See original paper for figures.