ISSN 1870-4069

A Method on Similar Text Finding and Plagiarism Detection based on Topic Model

Li Zhang, Jun Li

University of Science and Technology of China,
School of Information Science and Technology,
China

zlahu@foxmail.com, ljun@ustc.edu.cn

Abstract. This paper proposes a method on similar text finding and plagiarism detection among mass texts, which is based on the LDA mode, when a text need to detect, LDA generates its topics distribution, then calculate the similarity with the text in topics distribution library, we define the most similar text set as SimiSet, plagiarism detection is based on SimiSet, the coming text compares with the text in SimiSet using the method of fingerprint matching. In this paper, we compared several kinds of similarity calculation algorithm; experiment found that the combined algorithm of KNN and JS distance has higher recall rate and lower AIV value on plagiarism detection. We also compared proposed method with the traditional method on plagiarism detection, the result shows our method has higher F1 and smaller search range.

Keywords. Plagiarism, text similarity, LDA, SimiSet, TF-IDF, fingerprint.

1 Introduction

With the rapid development of the computing and Internet technology, the amount of multimedia data including digital text resource increases in a surprising speed, there are many complex relationships among these explosive texts, one of the performances is plagiarism, especially for popular articles and thesis, these misconduct acts like reproducing sections in absence of quoted reference or statements is no longer rare in big data times. To purify the atmosphere of the Internet and academic, it is important to solve plagiarism.

At present, TF-IDF (Term Frequency-Inverse Document Frequency) based on VSM (Vector Space Model) or its modified form is a widely used algorithm in similar text detection, a text's feature weight is expressed by combining term frequency and Inverse document frequency. However, TF-IDF method has several limits, such as ignoring semantic information, sparse problem, especially for short text; sometimes, the word able to characterize text semantics does not have highest weight, a word with a higher weight may has no practical meaning, these problems lead to low computational efficiency.

In order to avoid above problems, we take advantage of LDA [1] (Dirichlet Allocation Latent) topic model to calculate similarity, in which the feature of each text is represented by topic distribution, this representation reveals text semantics and

pp. 5–15; rec. 2017-01-27; acc. 2017-04-12

content information, due to the vector is relatively short, LDA solve high dimensional sparse problem, moreover, calculation procedure has speed up, the computational efficiency has improved. The proposed method of plagiarism detection in this paper is based on one assumption; the texts with plagiarism are similar in some content or topics. The procedure is to find target texts which constitute plagiarism with the source text in similar text set (SimiSet), so the proposed method is two-step method.

2 Related Work

2.1 Similar Text Detection

Similar text detection methods can be divided into three catalogues: VSM based, lexicon based and semantic based, in which lexicon based and semantic based approaches belong to semantic similarity detection [2].

In VSM based methods, the text is divided into many keywords, each weight of the keywords can be calculated by TF-IDF, chi-square statistics (CHI), mutual information and entropy. Guo [3] proposed an improved DF and TF-IDF algorithm, which increase the precision of the similarity calculation by adding keywords to reduce the incorrect filtering of information.

The lexicon-based approach figures the similarity between lexical items by referring to semantic lexicon, WordNet [4] is the most popular semantic lexicon, apart from WordNet, HowNet [5] is a common used Chinese semantic lexicon.

Rada [6] presented a method for measuring text semantic similarity, use corpus-based and knowledge-based similarity measures, experiments show that the proposed method outperforms the method based on simple lexical matching and the traditional vector-based similarity metric.

According to the information gain in corpus, semantic based methods are used to determine the similarity between lexical items. LSI (Semantic Index Latent) [7] and LDA are commonly used. Islam [8] proposed a semantic similarity method based on corpus and improved LCS algorithm, which is suitable for various situations of text representation and similarity discovery, experiment shows that the method has better performance than TF-IDF.TF-IDF cannot utilize semantics, the vector has very high dimensions and is extremely sparse, resulting in inefficient calculation and detection accuracy.

TF-IDF is not suitable for large text set, because of time consuming. The lexicon-based approach utilizes external knowledge, the detection efficiency depends on dictionary quality, which introduces uncertain factor. LSI uses matrix singular value decomposition (SVD) to convert word frequency into singular matrix, although the latent semantic relation is mined, but the parameters increase with corpus size, apart from this, complex calculation and high dimension are another limitation.

Compared with LSI, LDA also uses semantic information, but it can solve high dimension and sparseness problems. There are few parameters in LDA, it is suitable for large corpus.

2.2 Plagiarism Detection

Plagiarism detection methods can be divided into two catalogues: frequency statistic, string matching (also known as fingerprint).

The idea of frequency statistic is to calculate word frequency in two texts, Plagiarism detection become the problem of frequency similarity. Shivakumar developed SCAM [9], the system used VSM and frequency statistic method, and it could solve conflict of intellectual property rights in a certain degree. Si established CHECK [10] system, used the keyword statistic approach to measure text similarity, document structure information is introduced to achieve more accurate detection for the first time. CHENG [11] proposed a method in a mathematic way, the method is based on the representation of Chinese characters in computer, and used frequency statistic to measure text similarity, experiment proved that the algorithm is effective.

The string-matching method selects strings from document, these strings are a sequence of N consecutive words [12], each of which can be represented by a hash value (fingerprint) [13, 14]. Manber implemented Siff [15] tool and presented the approximate concept of fingerprint. Since then, many plagiarism detection systems have adopted this idea, such as COPS [16], KOALA [17], shingling [18], MDR [19], YAP [20]. In addition to the two methods mentioned above, Jamal[21] proposed a method for paragraph plagiarism detection, which is based on semantic analysis and part-of-speech (POS) tagging, to facilitate rapid detection, they put topics, synonyms and POS of paragraphs into the corresponding XML files, experiment showed the method can detect almost all type of plagiarism text.

At present, most of plagiarism detection algorithms are based on frequency statistics and string-matching methods. Statistics method does not consider semantic and text structure, which affects the accuracy of plagiarism detection. Although string matching method also does not use semantic, but it can quickly achieve plagiarism detection, the time and space required of generate fingerprints are relatively small, it can handle more files at the same time, for the case of full plagiarism, the algorithm will detect it quickly. Therefore, we use string matching method for plagiarism detection, experiments showed the method can make quick and accurate judgments for coming detected.

3 Similar Text Detection and Plagiarism Detection Method

3.1 LDA Model and Parameter Estimation

The origin of topic model is from LSI, which considers semantic relations, but it may filter out important features, resulting in poor classification performance. In 2003, Blei proposed LDA, which further improve topic model. LDA has prominent advantages, it has a clear hierarchical structure, is a probability model of word-topic- article layers.

A text has multiple latent topics mixed, every topic is expressed in probability distribution, and therefore text information is transformed into digital information which is easy to model. LDA introduces Dirichlet prior parameters in topic and word layers, both article-topic and the topic-word layer obey polynomial distribution, the parameters do not linearly increase with the growth of training set, which avoids over-fitting, especially appropriate for large corpus.

In this paper, Gibbs sampling is used to estimate model parameters, the ideology is to estimate the topics of each word, then the problem becomes calculating the conditional probability in equation (1), once the topic of each word is determined, the topic-word matrix and the article-topic matrix can be estimated by counting term frequency using equation (2) and (3) [22]:

$$p(z_i = k \mid \overrightarrow{Z_{\neg i}}, \vec{W}) = \frac{p(\vec{W}, \vec{Z})}{p(\vec{W}, \overrightarrow{Z_{\neg i}})} \propto \frac{n_{k,\neg i}^{t} + \beta_t}{\sum_{t=1}^{V} n_{k,\neg i}^{t} + \beta_t} \times \frac{\left(n_{m,\neg i}^{k} + \alpha_k\right)}{\left[\sum_{k=1}^{K} n_m^k + \alpha_k\right] - 1}, \tag{1}$$

$$\varphi_{k,t} = \frac{n_k^t + \beta_t}{\sum_{t=1}^{V} n_k^t + \beta_t}, \tag{2}$$

$$\theta_{m.k} = \frac{n_m^k + \alpha_k}{\sum_{t=1}^{K} n_m^k + \alpha_k}. \tag{3}$$

z_i denotes topic variables correspond with the i^{th} words, $\neg_i$ denotes the i^{th} excluded term, n_k^t denotes the number of word *t* occurrence in a specific topic *k*, n_m^k denotes the number of topic *k* occurrence in a specific text *m*.

3.2 Text Similarity

The topics generated by LDA model are not always meaningfully, because of lot of noise in training corpus, there are spam topics in topic mining results, especially for large training texts, which affect the performance of text similarity. We remove the stop words, names of individuals, words which only contains a single character, illegal English words, words whose document frequency over 60% and some lowest frequency words.

These words often appear as meaningless words in certain topics, have an impact on subsequent topic distribution extraction. This paper realizes the automatic filtering of spam topic. From the perspective of information theory, the higher the quality of a topic, the word distribution of the topic is more unbalanced, so smaller topic entropy is preferred, use (4) to calculate topic entropy, when the entropy is greater than the preset threshold, then it can be regarded as a spam topic. Suppose the probability distribution of a topic is: $p = \{p_1, p_2, p_3, \ldots, p_n\}$, The entropy of P is:

$$H(p) = H(p_1, p_2, p_{3,} \ldots, p_n) = -\sum_{i=1}^{n} p_i \times \log p_i\,. \tag{4}$$

We define plagiarism text source collection as CopiedSet, refers to the collection of the text which plagiarism text copied from. for example, assuming that the plagiarism text set only contains text *A*, apart from its original content, the rest is obtained by copying text *B*, *C*, *D*, then *CopiedSet (A)* = {*B*, *C*, *D*}. The definition of SimiSet refers to a collection of some most similar text with the coming text, each text is represented by corresponding topic distribution. The common used methods of measuring differences between two probability distributions are KL distance, JS distance and cosine similarity.

For two probability distributions, $p=\{p_1,p_2,p_3,\ldots,p_n\}$ $q=\{q_1,q_2,q_3,\ldots,q_n\}$:

$$D_{KL}(p,q)=\sum_{j=1}^{n} p_j \frac{p_j}{q_j} . \tag{5}$$

KL distance is not symmetrical, JS distance is symmetric version of KL:

$$D_{JS}(p,q)=\frac{1}{2}\times\left[D_{KL}(p,\frac{p+q}{2})+D_{KL}(q,\frac{p+q}{2})\right]. \tag{6}$$

Cosine similarity is:

$$D_{cosin}(p,q)=\cos\theta=\frac{\sum_{j=1}^{n} p_j\times q_j}{\sqrt{\sum_{j=1}^{n} p_j^2\times\sum_{j=1}^{n} q_j^2}} . \tag{7}$$

Considering that a text may have multiple topics, one text may not only copy another text which is basically the same in the whole topics, but also may plagiarize text which are similar in several topics, such as political and economic texts, A political text may not copy political texts, on the contrary it may plagiarize some economic texts. That is because political and economic texts have certain correlation in some aspects.

If the similarities are measured directly over the entire topic distribution according to (8) and (9), it is possible some text which have high similarity in partial topics are not considered, and therefore the similarity between topic distributions on common non-zero topics must be taken into account.

The modified text similarity measure proposed in this paper fully considers this problem, suppose the topic distribution of coming text *A* is *p*, text *B* is in training corpus, its topic distribution is *q*. We set the topic distribution with very low proportion of probabilities and the spam topic to zero. *sum(A)* represents the sum of the common topic probabilities present in *p*, and *sum(B)* represents the sum of the common topic probabilities present in *q*. The modified text similarity equation is:

$$modifyDist_{JS}(p,q)=\frac{sum(A)\times sum(B)}{D_{JS}(p,q)+const} , \tag{8}$$

$$modifyDist_{cosin}(p,q)=sum(A)\times sum(B)\times D_{cosin}(p,q) . \tag{9}$$

const in (9) is a constant to prevent overflow in memory, section 4.4 will compare the effect of four methods indicated by formula (6), (7), (8) and (9).

3.3 Plagiarism Detection Method

In this paper, the string-matching method is used to generate unique values (fingerprints) for a text, we first selected feature by using certain strategies, and then adopted Hash function to generate fingerprints, text *A* is split into several characteristic statements and sentences. $A=\{a_1,a_2,a_3,\ldots,a_m\}$, *Hash(A)* expresses fingerprints of *A*:

$$Hash(A) = \{h_1, h_2, h_3, \ldots, h_m\} . \tag{10}$$

Fingerprint is a mainstream and efficient approach for plagiarism detection, which can not only determine plagiarism, but also can mark details. The main idea of plagiarism detection is matching the fingerprints of two texts, the coincidence times of fingerprints can be expressed by the number of intersection elements of two Hash set, the overlap proportion is ratio of coincidence times to the number of fingerprints, if the overlap proportion is above a certain threshold, it will be determined as plagiarism.

4 Experimental Steps and Results

4.1 Experimental Steps

Our experimental tool is mainly implemented in Java include LDA model, we choose Fudan[1] corpus, which is a Chinese classified corpus, the training corpus contains nearly 10,000 articles in 20 categories.

However, the text distribution is seriously unbalanced, so 11 categories are removed due to containing very few texts, we extract 7480 texts from the remaining corpus, and then filter out short text which contains less than 1000 words, finally the training corpus has 5491 texts. In this paper, firstly, we determined the optimal number of topics for the training corpus and filtered out spam topic, secondly, we given the optimal text similarity calculation method, used the given method to construct SimiSet.

Finally, we compare the proposed method with the traditional method on plagiarism detection. A downsizing dictionary is constructed on the basis of stop words and other meaningless words removing, the size of the dictionary is much smaller than the original one, and so the training set contains less noise and light-weighted, experiment shows that the model contains less spam topics and has better plagiarism detection result.

4.2 Experimental Evaluation

We use three performance evaluation indicators: *F1*, recall rate *R* (Recall), *AIV* (Average Index Value), the indicator *AIV* is defined by us:

$$R = \frac{\left|SimiSet(A) \cap CopyedSet(A)\right|}{\left|CopyedSet(A)\right|} , \tag{11}$$

$$AIV = \frac{\sum Index\left\{SimiSet(A) \cap CopyedSet(A)\right\}}{\left|SimiSet(A) \cap CopyedSet(A)\right|} . \tag{12}$$

Suppose $SimiSet(A) = \{t_1, t_2, t_3, \ldots t_m\}$, $CopyedSet(A) = \{c_1, c_2, c_3, \ldots c_n\}$, *SimiSet (A)* is a set sorted in descending order. *A* may copy more than one piece of text, these texts are added to *SimiSet (A)* in the similarity calculation stage, *R* is the ratio of the number of

[1] http://www.nlpir.org/?action-viewnews-itemid-103

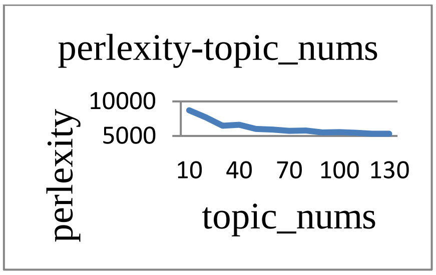

Fig. 1. Perplexity with different topic numbers.

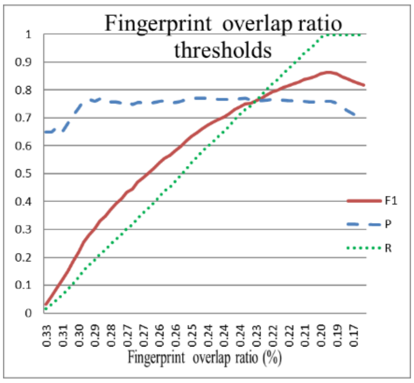

Fig. 2. Fingerprint overlap coincidence.

found plagiarism source text in *SimiSet (A)* and the number of all text in *CopiedSet (A)*, which is formulated in equation (11), operator $|*|$ means obtaining the number of collection elements.

In addition, *AIV* is defined as an average of all index of plagiarism source text in *SimiSet (A)*, which reflects the average rank of plagiarism source text, the higher the rank is, the less fingerprint matching times is used to find plagiarism source text. In equation (12), the operator $Index\{*\}$ indicates the index of plagiarism source text in *SimiSet (A)*.

4.3 Topic Number Selection and Spam Topic Filtering

LDA has few parameters, α, β is super parameter, which have empirical values，*K* is the number of topics, how to determine the size of *K*? Blei suggests Perplexity.

To find the optimal *K*, we draw the curve of perplexity with different number of topics, as shown in Fig.1, perplexity decreases monotonously with the number of topics increased, lower perplexity means better model, but the model is more prone to over-fitting, so we choose the number of topic when the curve begins to converge, for this model $K = 100$. Experiment indeed achieved good results when *K* set to 100.

Table 1. *R* compared on four similarity algorithms.

Simset Size	modify_cosin	modify_js	cosin	js
50	0.298	0.257	0.266	0.292
100	0.445	0.430	0.385	0.483
150	0.560	0.563	0.510	0.592
200	0.655	0.655	0.601	0.680
250	0.730	0.732	0.672	0.745
300	0.785	0.806	0.760	0.810
400	0.881	0.890	0.847	0.893
500	0.943	0.933	0.911	0.931
700	0.982	0.973	0.983	0.977
900	0.993	0.991	0.993	0.990
1200	0.996	0.996	0.997	0.994

Table 2. *AIV* compared on four similarity algorithms.

Simset Size	modify_cosin	modify_js	cosin	js
50	19.803	20.587	19.474	20.726
100	37.179	42.020	36.204	41.062
150	54.263	61.456	57.551	56.341
200	71.616	76.839	75.099	71.477
250	87.207	92.368	90.462	84.902
300	100.226	108.699	111.748	100.287
400	127.034	131.311	136.395	123.356
500	148.312	146.073	157.556	136.429
700	166.133	164.285	188.979	157.392
900	172.935	175.356	194.830	164.911
1200	175.192	179.067	197.733	168.108

4.4 Comparison of Several Similarity Methods

The test set consists of 200 plagiarized texts, which copied from 800 source texts, Table 1 shows the trend of *R* when different similarity methods is adopted, these similarity values is calculated by using equation (6), (7), (8) and (9) respectively.

As shown in Table 1, when SimiSet size within [0,700], *R* increases slowly with the SimiSet increases, when the size exceeds 700, *R* converges very close to 1. *R* of the four methods in Table 1 are basically the same except for cosine distance, but *AIV* is different as shown in Table 2. Lower *AIV* value indicates higher rank in SimiSet, it means less number of matches required to achieve the same *R* rate. JS distance is better than other methods, so this paper will use JS distance as text similarity measure.

4.5 Fingerprint Matching Threshold

Different fingerprint coincidence proportion thresholds affect the accuracy of plagiarism detection, to find the appropriate threshold, we also use section 4.4 test set for threshold selection experiment, and the SimiSet size is set to 800. As shown in Fig. 2, with the expansion of the threshold, *F1* value decreases after slowly increase, because the accuracy *P* is less affected by threshold change, the decline of *R* occurs because the higher the threshold, the less the number of plagiarized source texts will be found in SimiSet, so *F1* is mainly affected by *R*. *F1* reaches the peak value 0.863 at the threshold 19.8%.

This shows that the fingerprint method based on string matching has high reliability in finding the plagiarism source text from SimiSet.

4.6 Proposed Method and Traditional Method on Plagiarism Detection

In order to validate effectiveness of the proposed method, we compare it with the traditional plagiarism detection method; in order to find all plagiarism texts, traditional method use one-by-one comparison. Test set in this experiment is from section 4.4. As shown in Table 3, *F1* of the proposed method (0.863) is slightly high then the traditional method (0.828), but our method greatly reduced matching times. The training corpus contains 5,491 texts, the proposed method carries on SimiSet which contains only 800 texts, which is different from traditional plagiarism detection, the number of texts need to match is 14.6% of the original size, so our methods has greatly improved detection speed enhanced accuracy.

If want to further improve *F1* value, one need to make compromise on SimiSet size, which means if SimiSet size is larger, the detection process will be more time consuming, but *F1* value will be promoted. Experiment has found that when SimiSet size is 800, the accuracy and speed can both achieve satisfactory results.

5 Conclusion

This paper introduces the LDA topic model, which solves sparse problem effectively. LDA utilizes semantics of text, which greatly reduces the dimension of text vector representation and enhances its meaning, therefore, the effect of similarity calculation is more accurate and effectively.

In this paper, we find the optimal method to measure text similarity among four similarity algorithms, then we construct SimiSet for similar texts, the size of SimiSet in the experiment is far smaller than the total number of texts in the corpus, the plagiarism source text is mainly concentrated in SimiSet, which effectively reduces the number of texts to match and improves the detection performance and accuracy.

Next, we use fingerprint method to further detect plagiarism details from SimiSet. Finally, we compare our methods with traditional method in plagiarism detection, experiment proof the proposed method is more effective. The research mainly focus on long text, such as thesis, dissertation or web blog, the proposed method is ineffective in short text.

Table 3. Comparison between the proposed method and traditional method.

Methods	**collection size**	***F1***
proposed method	800	0.863
traditional method	5491	0.828

The subsequent work mainly aimed at short text plagiarism detection and similarity detection, how to effectively distinguish the boundaries between similarity and plagiarism need further study. LDA has many extensions, we will also explore new methods of text modeling and text mining based on LDA.

Acknowledgments. This research was supported by China Academy of Telecommunication Research of MIIT, the project Experimental verification of the basic commonness and key technical standards of the Industrial Internet network architecture. The authors would like to thank the reviewers for their valuable comments.

References

1. Blei, D., Ng, A., Jordan, M.: Latent Dirichlet allocation. Journal of Machine Learning Research, pp. 993–1022 (2003)
2. Gomaa, W. H., Fahmy, A. A.: A survey of text similarity approaches. International Journal of Computer Applications, vol. 68, no. 13 (2013)
3. Guo, Q.: The similarity computing of documents based on VSM. International Conference on Network-Based Information Systems, Springer, pp. 142–148 (2008)
4. Miller, G. A.: WordNet: a lexical database for English. Communications of the ACM, vol. 38, no. 11 pp. 39–41 (1995) doi: 10.1145/219717.219748
5. HowNet: http://www.keenage.com/html/c_index.html
6. Mihalcea, R., Corley, C., Strapparava.: Corpus based and knowledge-based measures of text semantic similarity. American Association for Artificial Intelligence, pp. 775–780 (2006)
7. Dumais, S. T.: Latent semantic analysis. Annual review of information science and technology, vol. 38, pp. 188–230 (2004)
8. Islam, A., Inkpen, D.: Semantic text similarity using corpus-based word similarity and string similarity. ACM Transactions on Knowledge Discovery from Data 2, no. 2, pp. 1–25 (2008) doi: 10.1145/1376815.1376819
9. Shivakumar, N., Garcia-Molina, H.: SCAM: A copy detection mechanism for digital documents. ACM SIGMOD Record, pp. 11–13 (1995)
10. Si, A., Leong, H. V., Lau, R. W. H.: Check: a document plagiarism detection system. In: Proceedings of the 1997 ACM Symposium on Applied Computing, pp. 70–77 (1997)
11. Cheng, Y., Zhang, J.: An algorithm for the illegal copying detection of digital documents. In: International Conference on Natural Language Processing and Knowledge Engineering, IEEE Press, pp. 384–387 (2005) doi: 10.1109/NLPKE.2005.1598767.
12. Lukashenko, R., Šakele, V., Grundspenkis, J.: Computer-based plagiarism detection methods and tools: an overview. In: Proceedings of the 2007 International Conference on Computer Systems and Technologies, ACM, vol. 285 pp. 40 (2007) doi: 10.1145/1330598.1330642
13. Stein, B., Zu-Eissen, S. M.: Near similarity search and plagiarism analysis. From data and information analysis to knowledge engineering, In: Proceedings of the 29th Annual Conference of the Gesellschaft für Klassifikation e.V, pp. 430–437 (2006) doi: 10.1007/3-540-31314-1_52

14. Schleimer, S., Wilkerson, D. S., Aiken, A.: Winnowing: Local algorithm for document fingerprinting. In: Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, pp. 76–85 (2003) doi: 10.1145/872757.87277
15. Manber, U.: Finding similar files in a large file system. In: Winter USENIX Technical Conference, pp. 1–10 (1994)
16. Brin, S., Davis, J., Garcia-Molina, H.: Copy detection mechanisms for digital documents. ACM SIGMOD Record, pp. 398–409 (1995)
17. Heintze, N.: Scalable document fingerprinting. In: Proceedings of the 2nd USENIX Workshop on Electronic Commerce, pp. 3 (1996)
18. Broder, A. Z., Glassman, S. C., Manasse, M. S., Zweig, G.: Syntactic clustering of the Web. In: Proceedings of the 6th International Web Conference, vol. 29, no. 8-13, pp. 1157–1166 (1997) doi: 10.1016/S0169-7552(97)00031-7
19. Monostori, K., Zaslavsky, A., Schmidt, H.: Match detect reveal: finding overlapping and similar digital documents. In: Proceedings of the Information Resources Management Association International Conference, IDEA Group Publisher, pp. 541–552 (2000)
20. Wise, M. J.: YAP3: Improved detection of similarities in computer programs and other texts. In: SIGCSE technical symposium on Computer science education, vol. 28, no. 1, pp. 130–134 (1996) doi: 10.1145/236462.236525
21. Heinrich, G.: Parameter estimation for text analysis. University of Leipzig, Tech. Rep, Hannover (2008)