DOCUMENT COPY DETECTION BASED ON KERNEL METHOD

Bao Jun-Peng, Shen Jun-Yi, Liu Xiao-Dong, Liu Hai-Yan, Zhang Xiao-Di

Department of Computer Science and Engineering, Xi'an Jiaotong University,
Xi'an 710049, People's Republic of China
baojp@mail.xjtu.edu.cn

ABSTRACT

In this paper, we present Semantic Sequence Kernel (SSK) to detect document plagiarism, which is derived from String Kernel (SK) and Word Sequence Kernel (WSK). SSK first finds out semantic sequences in documents, and then it uses a kernel function to calculate their similarity. SK and WSK only calculate the gap between the first word and the last one. SSK takes into account each common word's position information. We believe SSK contains both local and global information so that it makes a great progress in small partial plagiarism detection. We compare SSK with Relative Frequency Model and Semantic Sequence Model, which is a word frequency based model. The results show that SSK is excellent on non-rewording corpus. It is also valid on rewording corpus with some impairment on the performance.

Keywords: Kernel Method, Document Copy Detection, String Kernel, Sequence Kernel

1. INTRODUCTION

The aim of Document Copy Detection (DCD) is to examine whether the whole document or part of it is the copy of another one. DCD plays an important role in Intellectual Property Protection. At present, the basic schemes of DCD are string matching and word frequency. The former stress local information and locate the plagiarized content precisely, but it is easily to be cheated by rewording. The latter stress global information and is able to resist light rewording noise, but it is hard to increase accuracy because it loses too much details.

In this paper we propose a Semantic Sequence Kernel (SSK) for DCD application, which is a kernel method based on the semantic density, not the common global word frequency. SSK is derived from word-sequence kernel[2] and string kernel[3]. Those word frequency based kernels are not suitable for DCD although they are popular in TC. Because the word frequency model takes only global semantic features of a document but loses the detailed local features and structural information. Consequently we can hardly find out the plagiarized document from a group of very similar documents by word frequency feature model.

In the next section, we introduce some related work on DCD and string kernel. We present our semantic sequence kernel in detail in section 3 and release experimental results in section 4. We discuss some aspects of our kernel method and future work in section 5. Finally we draw conclusions in section 6.

2. RELATED WORK

Joachims[1] provides both theoretical and empirical evidence that SVM works for TC very well. Lodhi et al.[3] proposed the string kernel method that divide

0-7803-7902-0/03/$17.00 © 2003 IEEE.

the text category according to the common sequences between documents. The string kernel exploits the structural information instead of word frequency, and can outperform the bag of words approach.

Cancedda et al.[2] introduced the word sequence kernel that extends the idea of string kernel. They greatly increase the number of symbols to be considered, as symbols are words rather than characters. It reduces the average number of symbols per document and yields a significant improvement in computing efficiency so that training on large corpus becomes feasible without approximation.

Brin et al.[4] proposed the first DCD prototype (i.e. COPS) that detects overlap based on sentence and string matching but it has some difficulties in detecting sentences and finding partial sentence copy. Before long, Shivakumar and Garcia-Molina[7] developed SCAM to improve COPS. The SCAM measures overlap based on word frequency.

Heintze[6] developed a KOALA system for plagiarism detection. Broder et al. proposed a *shingling* method[5] to determine the syntactic similarity of files. These 2 systems are similar to COPS. Monostori et al.[8] proposed the MDR (Match Detect Reveal) prototype to detect plagiarism in large collections of electronic texts. It is also based on string matching, but it uses suffix tree to find and store strings.

Si et al.[9] built a copy detection mechanism, called CHECK. CHECK parses each document to build an internal indexing structure called structural characteristic (SC), which is used in document registration and comparison modules. Song et al.[10] proposed an illegal coping and distributing detection algorithm for digital goods (CDSDG). Their algorithm indeed combines the SC structure of CHECK and the RFM of SCAM to discover the whole and partial illegal copying behaviors.

3. SEMANTIC SEQUENCE KERNEL

We think DCD is a typical binary category issue. We use kernel method to divide the documents pairs into copy class (found plagiarism) or non-copy class (no plagiarism) in a feature space. The *Mercer's condition* [11] guarantee that we need not know how to map string into some space as long as the kernel function meet the condition. Therefore, we collect all semantic sequences, which contains both the local and global features of the document, and then we get the longest η common semantic sequences between the documents pair, at last we use the semantic sequence kernel, similar to word sequence kernel, to discriminate the pairs.

3.1 Semantic Density and Semantic Sequence

Definition 1 Let S be a sequence of words, i.e. $S=w_1w_2...w_n$. We denote the position i in S by i_S, the word at i denoted by $w(i_S)$. The *word distance* of position i_S ($1\leq i\leq n$), denoted by $d(i_S)$, is the number of words between $w(h_S)$ and $w(i_S)$: $d(i_S)= i_S - h_S$, where $w(h_S)=w(i_S)$ and $w(k_S)\neq w(i_S)$ ($1\leq h<k<i\leq n$). If no $w(h_S)$ exists, i.e. $w(i_S)$ occurs for the first time, then $d(i_S)=\infty$.

Definition 2 Let S be a sequence of words, i.e. $S=w_1w_2...w_n$. The *semantic density* of position i_S ($1\leq i\leq n$), denoted by $\rho(i_S)$, is the reciprocal of $d(i_S)$: $\rho(i_S) = 1/d(i_S)$.

In fact, $d(i_S)$ is the distance of $w(i_S)$ to its last appearance in the sequence S, and $\rho(i_S)$ reflects its local frequency. A document is a long sequence of words so that in a given range the small distance means the high density of word in a local section. That is to say the smaller distance, the higher density of words in the local section. We believe that the high-density words in some section indicate the local semantic feature of the section.

Definition 3 Let S be a sequence of words, i.e. $S=w_1w_2...w_n$. A *semantic sequence* of S is a subsequence of S, denoted by $L(S)=w_iw_j...w_k$ ($1<i<j<k\leq n$), which satisfies the following conditions:

(1) $|L(S)|>1$

(2) $(w(l_{un})=w(x_s))\wedge(w(m_{un})=w(y_s))\wedge(m_{un}=l_{un}+1)\rightarrow(0<y_s-x_s\leq\varepsilon)$

$(3)\rho(l_S)\geq\delta,\ i_S\leq l_S\leq k_S$

$(4)(h_S = i_S - 1) \rightarrow (\rho(h_S)<\delta)$

$(5)(v_S = k_S+1) \rightarrow (\rho(v_S)<\delta)$

where δ and ε are user defined parameters.

In fact, a semantic sequence in S is a *continual* word sequence after omitting the low density words in S. Here *continual* means that if two words are adjacent in *L(S)*, then the difference between their positions in S must not be greater than a threshold (ε). A long S may have several semantics. We denote all of the semantic sequences in a document S by $\Omega(S)$, which then includes the global and local semantic features as well as local structural information. Hence we believe the semantic sequence can detect plagiarism in a fine granularity so that we can find n to 1 partial copy well.

There is a fact for DCD that if one section is shared in two documents, then we think they are a copy pair. That is to say we need not find all of the common sections between two documents in order to decide whether plagiarism exists. It is enough for DCD to compare similarity in several most possible common sequences. We know that the larger number of common words there are between two strings, the more similar they are. Therefore, we select candidate semantic sequences in $\Omega(S)$ and $\Omega(T)$ according to the number of common words between them.

Let $CL(S,T)$=$[(L_1(S),L_1(T)),\ldots,(L_n(S),L_n(T))]$ be the list of semantic sequence pairs on document S and T, which is sorted by $|L_i(S)\bigcap L_i(T)|$descendingly. We denote the first η semantic sequence pairs by $CL_\eta(S,T)$. We denote the set of all the common words between semantic sequence pairs in $CL_\eta(S,T)$ by $CP(S,T)$:

$$CP(S,T)=\bigcup_{i=1}^{\eta} L_i(S)\bigcap L_i(T),$$
$$(L_i(S),L_i(T))\in CL_\eta(S,T) \tag{1}$$

According to the fact that more common words between documents indicate more similarity between them, $|CP(S,T)|$ can measure document similarity. However, when two documents share the same words list, they must be very similar but may not identical. The different arrangements of the same words list often represent different documents on the same topic. Namely, they are similar and belong to the same category, but not identical (i.e. no plagiarism). $|CP(S,T)|$ regards all possible arrangements of the same words list as the same so that it makes high positive errors. In order to discriminate the different arrangement of the same words list, we add structure information of string in our plagiarism metric.

3.2 Semantic Sequence Kernel

String kernel and word sequence kernel calculate the dot product of two strings based on gaps between terms/words, i.e. $l(i)$, the length spanned by $s[i]$ in s, that is: $l(i) = i_n - i_1 + 1$. The $l(i)$ cannot exactly reflect different arrangement of the same words list because it considers only the first and the last term/word in the list, without any others. If we take into account the position of each word, then the detection will be more precise.

The semantic sequence kernel of two semantic sequences *L(S)* and *L(T)* is defined as:

$$k(L(S),L(T))=\sum_{i=1}^{|L(S)\bigcap L(T)|}\lambda^{x_i},$$
$$\lambda=\frac{|L(S)\bigcap L(T)|}{|L(S)|+|L(T)|} \tag{2}$$

where x_i is the difference of a common word's word distances in S and T, that is:

$$x=|d(p_S)-d(q_T)|,$$
$$w(p_S)=w(q_S)\in L(S)\bigcap L(T) \tag{3}$$

The λ is the proportion of common words between two semantic sequences in the total words of them. So the larger λ is, the larger kernel value is. The kernel value indeed sums up each common word's value according to their different word distances. That is the smaller the distance is (i.e. two words are closer), the larger the kernel value is. In order to keep the kernel values comparable and independent on the length of the strings, we normalize the kernel function as follows:

$$k^*(L(S),L(T)) = \frac{k(L(S),L(T))}{\sqrt{k(L(S),L(S))k(L(T),L(T))}} \qquad (4)$$

Indeed, $k(L(S),L(S)) = |L(S)|$, $k(L(T),L(T)) = |L(T)|$ So the normalized semantic sequence kernel of $L(S)$ and $L(T)$ is:

$$k^*(L(S),L(T)) = \frac{k(L(S),L(T))}{\sqrt{|L(S)||L(T)|}} \qquad (5)$$

For a document pair S and T, we may select several semantic sequence pairs in order to improve accuracy, i.e. $CL_\eta(S,T)$, $\eta \ge 1$. Thus, we define semantic sequence kernel function of document pair (S,T) as:

$$k(S,T) = \frac{1}{\eta}\sum_{i=1}^{\eta} k^*(L_i(S),L_i(T)), \quad (L_i(S),L_i(T)) \in CL_\eta(S,T) \qquad (6)$$

For document pair (S,T), we use the below discriminant to make decision.

$$f(S,T) = sign(ak(S,T)+b), \quad a,b \in R, a \ne 0 \qquad (7)$$

In the next section, we compare the semantic sequence kernel with Relative Frequency Model (RFM)[7]. RFM uses an asymmetric metric to find out subset copy, but it is not so valid for n to 1 partial copy. In order to contrast error trend of semantic sequence kernel with that of RFM, we vary the discriminating threshold manually.

4. EXPERIMENTAL RESULTS

In this section, we introduce the test corpus and the contrasting experiments.

4.1 Test Corpus

We collect 25 full text PDF documents about data mining from Internet and convert them into plain text files. These are original documents, from which we make our plagiarized documents. We make plagiarized documents as follows: (1) Exact self-copy: these copies are just renamed source files. (2) Self cross copy: we divide each source file into 10 blocks and then randomly reassemble these blocks into a new copy. (3) Subset copy: we select 2 files from the source files and divide each of them into 10 blocks, then we randomly reassemble these 20 blocks into a new copy. (4) N to 1 partial copy: we select n (n=5 in our experiment) files from the source files and divide each of them into 10 blocks, then we randomly select k (k=2 in our experiment) blocks from each 10 blocks, at last we randomly reassemble all selected blocks into a new copy. (5) We reword each word of the non-rewording copy files in certain probability to get the respective rewording files. Finally, we make a document pair with a copy file and a source file. In our corpus, we have 2500 non-rewording document pairs and 22500 rewording document pairs of different rewording probability in total.

We define the positive error as the proportion of non-copy document pairs (no plagiarism) above the threshold τ against the whole non-copy document pairs, i.e.

$$E^+_\Phi(\tau) = |\{\Phi_N\}_{\ge\tau}| / |\Phi_N| \qquad (8)$$

where Φ is some type of document pairs, and $\{\Phi_N\}_{\ge\tau}$ is those non-copy pairs of type Φ that plagiarism

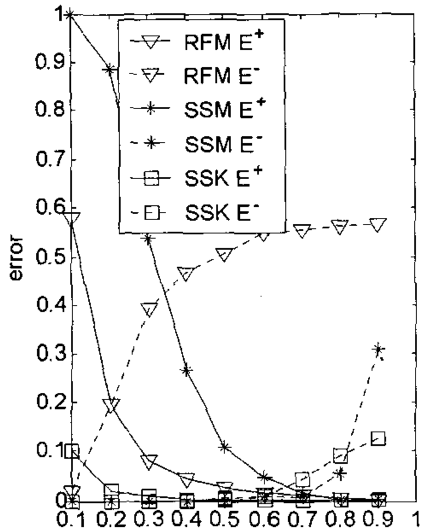

Fig 1. SSK, RFM and SSM error on the whole non-rewording corpus

score is greater than τ or equal to it. The negative error is the proportion of copy document pairs (existing plagiarism) below the threshold τ against the whole copy pairs, i.e.

$$E^-_{\Phi}(\tau) = |\{\Phi_C\}_{<\tau}| / |\Phi_C| \quad (9)$$

where $\{\Phi_C\}_{<\tau}$ is those copy pairs of type Φ that score is less than τ.

4.2 Contrasting Experiments

We have mentioned that we can use $|CP(S,T)|$ to distinguish plagiarism, and we call this approach as Semantic Sequence Model (SSM). In order to normalize $|CP(S,T)|$, we use the following equation to calculate the SSM plagiarism score of document S and T: $p_{SSM}(S,T)$.

$$p_{SSM}(S,T) = \frac{2}{\eta}\sum_{i=1}^{\eta}\lambda_i \quad (10)$$

Figure 1 shows the error trends of SSK, SSM and RFM on the whole non-rewording corpus. We see that the positive errors of SSK are always near 0 while its negative errors increase slowly. The negative errors of both SSK and SSM are very low, whereas the positive errors of SSM are higher than others. However, we can find the optimal value of discriminating threshold (τ) for SSM to make both positive and negative error low. So SSM is valid to detect plagiarism. SSK gains the lowest positive error without increasing its negative error largely. It proves that our semantic sequence kernel for DCD is successful on non-rewording corpus. Whatsoever, SSK is superior to RFM.

Figure 2 shows the error trends of SSM and RFM on the rewording corpus with each word rewording probability[1] θ=0.7. From figure 2, we see that their negative and positive error trends are both sharp, that is to say rewording action beats their detection ability. But the least error of SSM is still less than that of RFM and it is tolerable.

Figure 3 shows the error trends of SSK with the same rewording probability. We find that the valid discriminating threshold range of SSK is far less than that of SSM. The positive errors of SSK are still far less than that of SSM, but the negative errors of SSK increase greatly.

Figure 4 shows SSK error trends with different rewording probabilities (θ=0.2,0.4,0.6,0.8). We find that when the rewording probability increases, the positive errors of SSK are stable and the negative errors of SSK increase a little. Whatsoever, we can find the appropriate values of τ to make both the positive and negative error low. The least error of SSK is tolerable. It proves that SSK is valid for DCD even if the document is reworded in some degree.

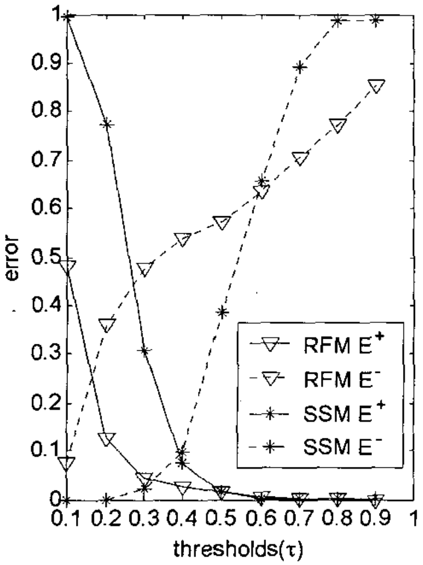

Fig 2. RFM and SSM error on the rewording corpus with rewording probability θ=0.7

5. DISCUSSIONS

From experiments we find that the positive error and the negative error are a pair of contradictory. That is to say a low positive error will lead to a high negative error and vice versa. Therefore we have to make a trade-off between the positive and negative error. On

[1] We use JWNL (http://sourceforge.net/projects/jwordnet) to reword each word in document. Because the synonym set of a word contains the word itself, the real rewording probability is lower than the labeled value.

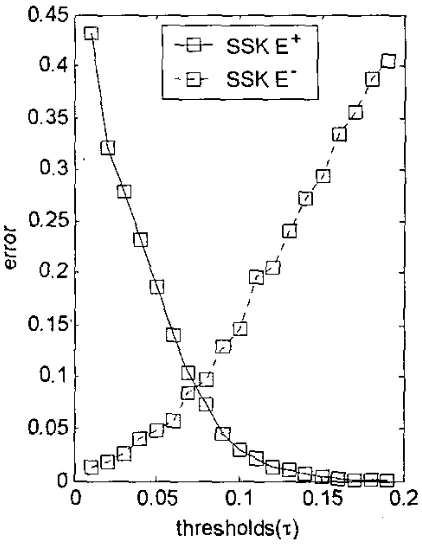

Fig 3. SSK error on the rewording corpus with rewording probability θ=0.7

the non-rewording corpus, the low negative errors of SSM lead to its high positive errors. But we can use SSK to reduce the positive error without largely increasing negative errors. On the rewording corpus, their detection capability declines. Whatsoever, they are valid for DCD even on the largely rewording corpus and they are superior to RFM.

SSM stresses the total amount of common words, which reflects more global feature. SSK takes into account both local structural information and magnitude of common words, i.e. it contains both local and global features. Both SSK and SSM conform to the principle that the more common words there are between documents the more similar they are. Additionally, SSK satisfies more strict structural condition, i.e. the common words' word distance must be similar, otherwise the word will be penalized. We know that the plagiarized documents must contain many common words, but sharing of many words between documents may not necessarily mean plagiarism. SSM cannot distinguish different arrangement of the same word list so its positive errors are high. SSK adds structural information to make detection in more details. That diminishes positive errors greatly. SSK is excellent on

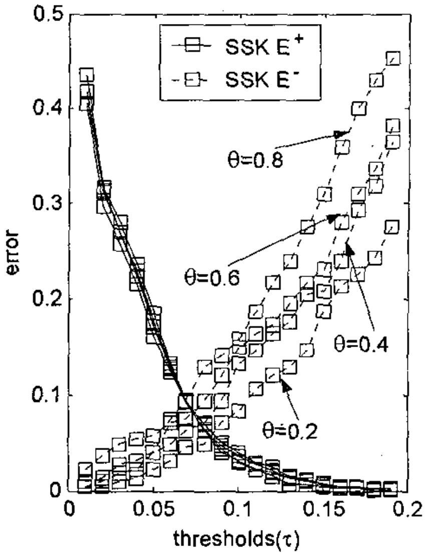

Fig 4. SSK error on the rewording corpus

non-rewording documents even if the plagiarism exists only in a small section.

However, the rewording action may not only reduce the common words but also disorder the words sequence. That worsens detection environment and increases the negative error of SSK and SSM largely, namely they may fail in detecting some plagiarized documents. In the future, we have to improve detection accuracy on rewording corpus. We think it is hard to make progress with the method that directly compares features of raw rewording documents. We are going to add rewording probability to the kernel in order to take into account rewording information. We will use synonymy of words instead of identity of words, namely, if two words are synonym, we consider they are common in some probability.

6. CONCLUSIONS

String matching and word frequency are two basic approaches of DCD though they have their limitation. The former stress local information whereas the latter stress global information. In this paper, we present semantic sequence kernel to combine the two basic approaches and make a trade-off between global and local information. We first find out semantic sequences of documents based on semantic density, and then we use kernel method to calculate the similarity of two semantic sequences. SSK is very similar to string kernel and word sequence kernel, but we contain each common word's position information (word distance). On the one hand, the amount of common words takes effect on SSK, i.e. the more common words, the larger SSK value, which complies with word frequency models. On the other hand, SSK takes account of word sequence, i.e. different arrangements of the same words list have different value, which conforms to the characteristic of string matching models. The experiments show that SSK is excellent on non-rewording corpus. The rewording action beats performance of SSK though it is still valid on rewording corpus. Our next goal is to improve detection accuracy on rewording corpus by considering rewording probability in the new model.

7. ACKNOWLEDGEMENTS

Our study is sponsored by NSFC (National Natural Science Foundation of China, ID: 60173058) and Xi'an Jiaotong University Science Research Fund (ID: 573031).

References

[1] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In Proceedings of the European Conference on Machine Learning (ECML98), number 1398 in Lecture Notes in Computer Science, pages 137-142. Springer Verlag, 1998.

[2] N. Cancedda, E. Gaussier, C. Goutte, J. M. Renders. Word-Sequence Kernels. Journal of Machine Learning Research, 3:1059-1082, 2003

[3] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, C. Watkins. Text Classification using String Kernels. Journal of Machine Learning Research, 2(Feb):419-444, 2002

[4] S. Brin, J. Davis, and H. Garcia-Molina. Copy detection mechanisms for digital documents. In Proceedings of the ACM SIGMOD Annual Conference, San Francisco, CA, May 1995.

[5] Broder A.Z., Glassman S.C., Manasse M.S., Syntactic Clustering of the Web. Sixth International Web Conference, Santa Clara, California USA, April 7-11, 1997.

[6] Heintze N. Scalable Document Fingerprinting. In Proceedings of the Second USENIX Workshop on Electronic Commerce, Oakland, California, 18-21 November, 1996.

[7] N. Shivakumar and H. Garcia-Molina. SCAM: A copy detection mechanism for digital documents. In Proceedings of 2nd International Conference in Theory and Practice of Digital Libraries (DL'95), Austin, Texas, June 1995.

[8] K. Monostori, A. Zaslavsky, H. Schmidt. MatchDetectReveal: Finding Overlapping and Similar Digital Documents. In Proceedings of Information Resources Management Association International Conference (IRMA2000), Anchorage, Alaska, USA, 21-24 May, 2000

[9] Si A., Leong H.V., Lau R. W. H. CHECK: A Document Plagiarism Detection System. In Proceedings of ACM Symposium for Applied Computing, pp.70-77, Feb. 1997.

[10] Song Qinbao, Shen Junyi. On illegal coping and distributing detection mechanism for digital goods. Journal of Computer Research and Development. 38(1): 121-125, 2001

[11] V. Vapnik. The Nature of Statistical Learning Theory. Springer-Verlag, New York, 1995.