An Improved Model of Dotplotting for Text Segmentation

Na Ye[1], Jingbo Zhu[1], Huizhen Wang[1], Matthew Y. Ma[2], Bin Zhang[1]
[1] Dept. of Computer Science and Engineering, Northeastern University, 110004, China
[2] IPVALUE Management Inc. 991 Rt. 22 West, Bridgewater, NJ 08807
yn.yena@gmail.com

Abstract

The Dotplotting method has been widely used for text segmentation for its merits in detecting lexical repetition in global context. However, a theoretical analysis of its segmentation criterion function finds several deficiencies. The original function can not make full use of the text structure features and does not suit the text segmentation task very well. We propose an improved model (MMD model) that resolves these deficiencies. Comparative experimental results on the synthetic corpus and real corpus have shown that MMD model reduces the error rate of the original Dotplotting method by more than 20 percent, and outperforms other existing methods derived from Dotplotting.

Keywords

Text Segmentation; Dotplotting; between-segment similarity; within-segment similarity; word distance; segment length.

1 Introduction

A natural language discourse is usually composed of multiple subtopics, which in turn may convey only one main topic. In traditional text processing tasks such as information retrieval(IR), question answering(QA) and text summarization, if the subtopic structure of a text can be identified and consequently its semantic segments can be used in the basic processing unit, the performance of the system will be greatly improved (Hearst, 1994; Boguraev et al., 2000). In addition, the segment-based IR will provide users with answers of higher accuracy and less redundancy results. The core technology involved in the identification of subtopic structure and therefore semantic segments of a text is called text segmentation, which is the focus of this paper.

In recent years, many text segmentation algorithms have been developed. Some use local information, such as lexical similarity between adjacent parts of the text (Hearst, 1994; Kan et al., 1998; Brants et al., 2002), to detect subtopic changes. These algorithms measure topical coherence between local contexts, however, can not achieve global optimization over the whole discourse. Other methods adopt global optimization algorithm to find the best segmentation (Reynar, 1994 and 1998; Heinonen, 1998; Choi et al., 2000 and 2001; Utiyama and Isahara, 2001; Ji and Zha, 2003; Fragkou et al., 2004; Zhu et al., 2005; Malioutov and Barzilay, 2006). Among these approaches, Dotplotting (Reynar, 1998) is a well-known text segmentation algorithm, which is widely cited and studied because it can detect lexical

repetition in global context. Another model under the Dotplotting framework was designed by Choi (2000) using a ranking scheme and achieved better performance. Later Choi (2001) incorporated latent semantic analysis (LSA) to improve his previous work. However, in order to get parameters of LSA, training corpus is required.

In this paper we conduct theoretical analysis on the original Dotplotting method. Several deficiencies are found in its segmentation criterion function. First, the function is asymmetric, leading to the apparent illogical phenomenon that forward scan may result in different segmentation with backward scan. Second, while determining segment boundaries, the assessing strategy does not adequately take the previously located boundaries into account. Third, the criterion function only focuses on minimizing the similarity between adjacent segments and ignores the maximization of similarity within segments. Fourth, the effect of word distance on lexical similarity computation is not considered. Fifth, the function may result in segmentation with abnormally short segments.

On the basis of our analysis, an improved model called MMD model (Min-Max similarity Dotplotting model) is proposed to resolve these deficiencies. Detailed analysis of the proposed model is given and a complete text segmentation algorithm employing the Dotplotting search strategy is described. Comparative experimental results on the Choi benchmark corpus and real corpus have shown that MMD model outperforms both original and improved Dotplotting methods by Choi.

The remainder of this paper is organized as follows. In Section 2, we briefly introduce the Dotplotting algorithm and thoroughly analyze the problems of Dotplotting. In Section 3, we propose a MMD model to overcome these shortcomings. In Section 4, experimental results are given to compare our model with other relevant methods. Finally in Section 5, we draw conclusion and address our future work.

2 Analysis of the Dotplotting Method

2.1 The Dotplotting Method

Reynar (1994 and 1998) introduces a graphical technique of locating subtopic boundaries based on lexical cohesion called Dotplotting. Dotplotting is based on a plot called Dotplot that reflects lexical repetition over all parts of a text. On the Dotplot, diagonal corresponds to the whole text and the points that distinguish lightest regions off of the diagonal act as segment boundaries. In Dotplotting, segment boundaries are found one by one. In each cycle, the candidate boundary that yields minimum overall density outside the diagonal is selected as the best boundary to be inserted. The density function is as follows:

$$f_D = \sum_{j=2}^{|P|} \frac{V_{P_{j-1},P_j} \cdot V_{P_j,n}}{(P_j - P_{j-1})(n - P_j)} \tag{1}$$

where n is the length of the whole text, P_j is the position of the jth boundary, |P| is the number of segments in the text, and $V_{x,y}$ is a vector containing the word counts associated with word positions x through y in the article. This model is called minimization model[1].

[1] Reynar also proposed maximization model (Reynar, 1998). However, since experimental results show that minimization model works better than maximization model (see section 4.3 and 4.4), we choose the minimization model as the original model to be studied and improved.

In Dotplotting, the density function acts as a segmentation criterion function through which segmentations are scored and ranked. However, there are several flaws in the function of Dotplotting. For each candidate boundary, the corresponding individual density item in Eq.**Error! Reference source not found.** is $\sum_{j=2}^{|P|} \frac{V_{P_{j-1},P_j} \cdot V_{P_j,n}}{(P_j - P_{j-1})(n - P_j)}$, which is simulated in **Error! Reference source not found.**.

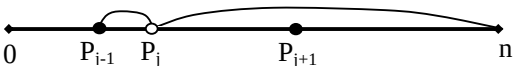

Figure 1. An illustration of the density measurement of Dotplotting method

As shown in **Error! Reference source not found.**, the line indicates the text; circles indicate the segment boundaries (solid circles indicate the boundaries already located; and the hollow circle indicates the candidate boundary position being examined). Each curve indicates the vector associated with text segment located from the starting circle to the ending circle. We can see from **Error! Reference source not found.** that each individual density item at position P_j measures the lexical similarity between the immediate proceeding segment and the whole text that follows it. Therefore, the individual density at P_j is decided by its previous segment boundary location P_{j-1} and the end of the whole text n.

2.2 Problems of the Dotplotting Method

In this section, detailed analysis regarding each deficiency in the Dotplotting method is given.

2.2.1 Symmetry of the density function

Suppose we are faced with a text, the topical coherence of the text should be independent of the scan direction. It is apparent that when we scan the text from the start towards the end to divide it into segments, we should get the same segmentation result as we scan from the end towards the start. However, with the density function of Dotplotting different segmentation results will be generated.

For example, in the testing corpus created by Choi (2000), for article #0 in data set 3-5[2], the segmentation acquired with Eq. **Error! Reference source not found.** is:

{0 3 4 15 16 18 20 21 22 23 39}

When we reverse the sentence order in the article, the segmentation acquired with Eq. **Error! Reference source not found.** is:

{0 4 16 17 18 19 21 22 23 25 39}

The two segmentations are quite different. The reason that has caused such phenomenon is that the density function of Dotplotting is not symmetric. As previously mentioned, the density item at position P_j is decided by its previous segment boundary location P_{j-1} and the

[2] There are 4 data sets in the corpus. In this data set there are 100 articles. See section 3.1 for detailed introduction.

end of the whole text *n*. This is illustrated in **Error! Reference source not found.** by the curve connection above the line.

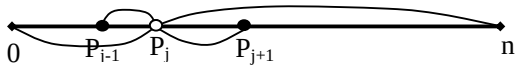

Figure 2. An illustration of the asymmetric density function

With this strategy, if we scan from the end to the start of the text, we will get a "backward" density in the following form:

$$f_{D'} = \sum_{j=2}^{|P|} \frac{V_{0,P_j} \cdot V_{P_j,P_{j+1}}}{P_j(P_{j+1} - P_j)} \tag{1}$$

This is demonstrated by the curves below the line in **Error! Reference source not found.**. The individual item in Eq. (1) measures the lexical similarity between segments [0, P_j] and [P_j, P_{j+1}]. Thus the density item at position P_j is decided by the start of the whole text 0 and its next segment boundary position P_{j+1}. Now in each cycle, the best boundary selected may be different from that selected with Eq. **Error! Reference source not found.**, and consequently forward scan may lead to different segmentation with backward scan.

2.2.2 Prior boundaries

In text segmentation, segment boundaries are located to distinguish the two segments before and after them. So the selection of each segment boundary should be dependent on its immediate proceeding and next segment boundary. Therefore, in the selection of the boundary, for each candidate boundary P_j being examined, the adjacent boundaries that are already located (P_{j-1} and P_{j+1}) should have direct effect.

However, the Dotplotting algorithm doesn't adequately make use of the restriction of the two boundaries. As mentioned above, the individual density item measures the topical coherence between segments [P_{j-1}, P_j] and [P, n] (see **Error! Reference source not found.**). This strategy only takes the previous segment boundary P_{j-1} into consideration, and may work less effectively.

2.2.3 The similarity within segments

It will reasonably hold true that in an appropriately segmented text, sentences within a single segment are topically related and sentences that belong to adjacent segments are topically unrelated conveying different subtopics. If two sentences describe the same topic, words used in them tend to be related to one another. Thus, within a segment, vocabulary tends to be cohesive and repetitive, leading to significant within-segment lexical similarity; whereas between adjacent segments, the vocabulary tends to be distinct, leading to dismal between-segment similarity. We believe that the above lexical similarity property must exist for a good segmentation strategy.

For example, the following text fragments are extracted from article #1 in data set of 3-5[3] in the Choi(2000) testing corpus for text segmentation:

> This **theorem** is similar to the **theorem** of Kakutani that there exists a circumscribing **cube** around any closed , bounded convex set in **Afj** . The latter **theorem** has been generalized by Yamabe and Yujobo , and Cairns to show that in **Afj** there are families of such **cubes** .
> ==========
> Several **defendants** in the Summerdale **police** burglary **trial** made statements indicating their guilt at the time of their arrest , Judge James B. Parsons was told in Criminal court yesterday . The disclosure by Charles Bellows , chief **defense** counsel , startled observers and was viewed as the prelude to a quarrel between the six attorneys representing the eight former **policemen** now on **trial** .

We can see that in the above two segments, there exists highly repetitive vocabulary within each segment whereas there is great difference in the vocabulary comprising the adjacent segments.

However, as seen from Eq.**Error! Reference source not found.**, the density function of Dotplotting is in fact a measure of between-segment similarity. Each individual density item counts the lexical repetition between the two contiguous segments $[P_{j-1}, P_j]$ and $[P_j, n]$. No measurement of lexical similarity within segments is treated in the evaluation of segmentation.

2.2.4 The effect of word distance

If we randomly select two words from a discourse, the probability of them belonging to the same segment varies greatly with the distance between them. Two words far apart are unlikely to belong to the same segment, whereas two adjacent words are much more likely. Therefore, the distances between words should have influence on the computation of lexical similarity. The farther away the two words, the less their repetition contributes to the overall similarity. In Dotplotting, words are regarded as the same no matter how far apart they are.

2.2.5 Short segments

In text segmentation, text pieces that are too short do not adequately describe an independent subtopic. In fact, too short text pieces can hardly express an independent subtopic. In practice, we do not consider text fragments with only a few sentences as segments. This is because these text fragments may indicate the presence of a short

[3] There are 4 data sets in the corpus. In this data set there are 100 articles. See Section 4.3 for detailed introduction.

transitive paragraph, and it often brings great detriment to the overall segmentation performance. However, the segmentation criterion function of Dotplotting only takes lexical densities of segments into account and no extra restriction over lengths of segments is employed to address the problem. This strategy may result in segmentation with abnormally short segments.

3 Our Proposed MMD Model

3.1 Segmentation Criterion Function

To address the deficiencies in the Dotplotting as analyzed above, an improved model (MMD model) is proposed. The modified segmentation criterion function is as follows:

$$J = (\alpha \cdot \sum_{j=2}^{|P|} \frac{\sum_{m=P_j+1}^{P_{j+1}} \sum_{n=P_{j-1}+1}^{P_j} W_{m,n} D_{m,n}}{(P_{j+1} - P_j)(P_j - P_{j-1})} - \beta \cdot \sum_{j=2}^{|P|} \frac{\sum_{m=P_{j-1}+1}^{P_j} \sum_{n=P_{j-1}+1}^{P_j} W_{m,n} D_{m,n}}{(P_j - P_{j-1})^2}) / \prod_{i=1}^{|P-1|} \frac{L_i}{L} \quad (2)$$

where α and β are the relative weights of between-segment lexical similarity and within-segment lexical similarity, respectively. $\alpha + \beta = 1$. L_i is the length of the ith segment, and the length of the whole text is L. m and n are the mth and nth sentence in the text. $D_{m,n}$ is the lexical similarity between sentence m and sentence n. The value of $D_{m,n}$ equals to one if there exist one or more words in common between sentence m and n, and zero otherwise. $W_{m,n}$ is the weighting factor, and is based on the distance between the sentence m and sentence n. The values of m and n represent the positions of each corresponding sentence. An exemplary definition of $W_{m.n}$ is given as:

$$W_{m,n} = \begin{cases} 1 & if\ |m-n| \le 2 \\ \dfrac{1}{\sqrt{|m-n|-1}} & else \end{cases}$$

With the new function, forward and backward scan lead to the same density value, and consequently to the same segmentation. This function also directly takes the restriction of existing adjacent boundaries (P_{j-1} and P_{j+1}) into consideration and strengthens the restriction from the previously located boundaries. In MMD model, simultaneous maximization of within-segment lexical similarity as well as minimization of between-segment lexical similarity are attempted to achieve in order to discover topical coherence precisely. Unlike Dotplotting which is based on word-based similarity, the above function takes sentence-based lexical similarity. This is because sentences act as the smallest units that can express a complete meaning in a natural language discourse, and in practice segment boundaries are set at the ends of sentences.

A length penalty factor is incorporated to penalize segmentation choices with short segments by assigning a small evaluation function score to it. The length penalty factor

is defined as $\prod_{i=1}^{|P-1|} \frac{L_i}{L}$, where $L = \sum_{i=1}^{|P-1|} L_i$ and it achieves maximum value when $L_1=L_2=\ldots=L_{|P-1|}$. We also add a distance-based weighting factor $W_{m,n}$ to the function, enabling the lexical similarity of two sentences to fluctuate as the distance between them varies.

3.2 Text Segmentation Algorithm

To optimize the segmentation evaluating function (Eq. (2)) globally, we provide an implementation using the Dotplotting searching strategy (Reynar 1994, 1998) to find the best segmentation. The complete text segmentation algorithm is shown in Figure 3. MMD Text segmentation algorithm

, followed by detailed explanation.

Given a text *S*, *N* is the desired number of segments.

Initialization: $B=\{\}$, $P=\{\}$, $J_{min}=+\infty$, $C=\{i \mid i$ is the potential boundary in $S\}$, $G_{best}=0$.

Segmentation:

For k=1 to N

For each i in C

1) $P= B \cup \{i\}$
2) Use evaluating function J to compute the score of the segmentation derived from P.
3) If $J_{min}>J$ Then

$J_{min}=J$ and $G_{best}=i$

Endif

Endfor

$B= B \cup \{G_{best}\}$

$C= C - \{G_{best}\}$

Endfor

Output the best segmentation B

Figure 3. MMD Text segmentation algorithm

In the above procedure, segment boundaries are inserted sequentially until the desired number of segments is achieved. Sentence boundaries act as candidate segment boundaries. To determinate a new segment boundary, each candidate position is examined. The candidate position is hypothetically added to the boundary set *B*, and constitutes current segmentation set *P*. Then the value of the segmentation evaluating function *J* is computed using the boundaries in *P*. The boundary position achieving the lowest value is selected as the next boundary, to be inserted to the boundary set.

4 Evaluation

The evaluation has been conducted systematically under a strict guideline in order to compare our approach with other state of the art algorithms on a fair basis. The key requirements are: 1) Evaluation should be conducted using a sizable testing data in order to generate meaningful results; 2) The testing data should be publicly available; 3) In order to compare with other people's work, we attempt to use their own implementations or published results as these are likely optimized for taking maximum advantages of their merits.

4.1 Experiment Settings

In our experiments, the first testing corpus is the widely used publicly available corpus developed by Choi[4]. In this corpus, each article is a concatenation of ten text segments. A segment is the first n sentences of a randomly selected document from the Brown corpus. The data set is divided into four subsets depending on the range of n. The number of articles in each subset is listed in **Error! Reference source not found.**.

Range of *n*	3-11	3-5	6-8	9-11
Number of samples	400	100	100	100

Table 1. Testing data set 1 (Choi's collection)

Due to the restriction of synthetic corpus, we also conducted experiments on real corpus. The second testing data set is selected from *Mars* written by Percival Lowell in 1895. We present the results with Section 1 (As a Star) of Chapter 1 (General Characteristics), Section 2 (Clouds) of Chapter 2 (Atmosphere), and Section 1 (First Appearances) of Chapter 4 (Canals). Heinonen (1998) and Ji and Zha (2003) also used some texts from *Mars* to evaluate their methods.

To evaluate text segmentation algorithms, using precision and recall is inadequate because inaccurately identified segment boundaries are penalized equally regardless of their distance from the correct segment boundaries. We instead use the P_k metric (Beeferman et al., 1999). P_k is the probability that a randomly chosen pair of words with a distance of k words apart is incorrectly segmented[5]. Low P_k value indicates high segmentation accuracy. P_k metric is defined as:

$$P_k = \sum_{1 \le i \le j \le k} D_k(i, j)(\delta_{ref}(i, j) \overline{\oplus} \delta_{hyp}(i, j))$$

where $\delta_{ref}(i, j)$ is an indicator function whose value is one if sentences i and j belong to the same segment and zero otherwise. Similarly, $\delta_{hyp}(i, j)$ is one if the two sentences are hypothesized as belonging to the same segment and zero otherwise. The $\overline{\oplus}$ operator is the XNOR operator. The function D_k is the distance probability distribution that uniformly concentrates all its mass on the sentences which have a distance of k. The value of k is usually selected as half the average segment length.

4 www.lingware.co.uk/homepage/freddy.choi/index.htm

5 We use the implementation of P_k in Choi's software package. (www.lingware.co.uk/homepage/freddy.choi/index.htm)

In fact, this error metric was recently criticized (Penvzner and Hearst, 2002). It was mentioned to have several biased flaws such as penalizing missed boundaries more than erroneous additional boundaries. A new metric called WindowDiff (Penvzner and Hearst, 2002) was proposed. However, since previous publications only present P_k evaluation value on Choi corpus, we will make comparison under P_k metric on this synthetic corpus and both metrics (P_k and WindowDiff) on the *Mars* real corpus.

Punctuation marks and stopwords are removed (using the stopword list offered in the Choi software package), and the Porter (1980) stemming algorithm is applied to the remaining words to obtain word stems.

4.2 Experiment I - Parameter Selection

In this experiment the goal is to determine the influence of α (the relative weight of between-segment lexical similarity in Eq.(2)) on segmentation performance (measured by P_k). We determine appropriate α values using the texts of testing data set 1.

We let α take the values 0.00, 0.01, 0.02 , . . . , 0.09, 0.1, 0.2, 0.3 , . . . , 1.0. For each value we run the segmentation algorithm on the whole corpus. In **Error! Reference source not found.** we give detailed results for the influence of α on each subset.

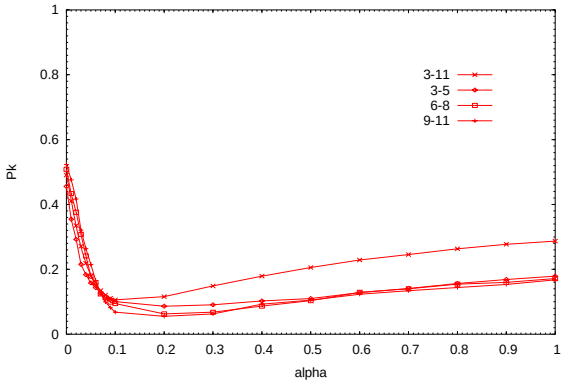

Figure 4. P_k plotted as a function of α for the texts of testing corpus 1

It can be seen that the best performance of our algorithm has been achieved for α in the range [0.1, 0.2]. We set α to 0.2 in our following experiments.

The performance of our algorithm (as obtained by the validated parameter values) on Choi's corpus is presented in the next section.

4.3 Experiment II - Experimental Results on Synthetic Corpus

Since MMD model employs the searching strategy of Dotplotting (Reynar, 1994 and 1998), we evaluate MMD model in comparison to the Dotplotting method including maximization algorithm (D_Max) and minimization algorithm (D_Min). Two other approaches derived from Dotplotting, namely C99 (Choi et al., 2000) and CWM (Choi et al., 2001) are also included in our evaluation. The experimental results of Dotplotting come from Choi's software package, which is an exact implementation

of the Dotplotting method[6] published in (Reynar, 1994 and 1998). In our experiments the desired number of segments is given in advance. **Error! Reference source not found.** shows the P_k evaluation of the methods on the synthetic testing data set 1.

Method	3-11	3-5	6-8	9-11	Training
D_Max	48.50%	48.79%	49.73%	51.26%	No
D_Min	34.75%	33.86%	36.49%	38.47%	No
C99	12%	11%	10%	9%	No
CWM	9%	10%	7%	5%	Yes
MMD	**10.60**%	**8.63**%	**6.24**%	**5.51**%	No

Table 2. P_k Comparison with Dotplotting methods on testing data set 1

From experimental results we can see that our MMD model performs significantly better with more than 20% reduction on error rate for max and min Dotplotting methods on synthetic corpus. The results indicate the improved technique in our method is very effective. With the results in **Error! Reference source not found.**, it is also demonstrated that our method is more favorable for dealing with long text segments. The smallest error rate is achieved when the average segment size is the largest among all subsets of data.

Comparing to C99 algorithm, MMD model achieves better performance up to 4%. In C99 algorithm, only lexical similarities within segments and no other indicators such as segment lengths that can detect subtopic changes are considered for segmentation. In contrast, MMD takes advantage of rich information to discover topical coherence in the discourse.

In comparison with the CWM algorithm, our model achieves comparable or improved results but requires no training data. CWM achieves good performance based on large amount of training data in some domains. However, MMD tends to perform equally well when it is applied to a different text domain without requiring training data.

4.4 Experiment III - Illustration of the Impact of Each Factor

The third suite of experiments aims to illustrate the effectiveness of each element incorporated in the segmentation criterion function. We examine three factors: within-segment similarity, segment length penalty and distance-based similarity weighting strategy. We report the results of the MMD model with one of the factors removed each time, and comparing them with the original model.

In the experiment segment number is given in advance. P_k metric is used to measure the performance. The experiments are also done on testing data set 1 (Choi's collection).

[6] Choi (2000) also published experimental results of Dotplotting, which were his own interpretation of the algorithm. However, since our modifications are on the original Dotplotting method, we cite the experimental results of the original Dotplotting method. Although not published, Choi developed a package that includes both the implementation of the original Dotplotting method and his interpretation. The experimental results come from the publicly available software package. (www.lingware.co.uk/homepage/freddy.choi/index.htm)

Error! Reference source not found. presents the segmentation performance for each of the removed factors ("-" denotes "removing").

Method	3-11	3-5	6-8	9-11
MMD	**10.60**%	**8.63**%	**6.24**%	**5.51**%
-Within_Sim	11.50%	14.00%	9.13%	5.98%
-Len-Pen	20.16%	19.56%	20.50%	21.14%
-Dist_Wgt	11.79%	10.07%	8.93%	7.53%

Table 3. Experimental results of MMD when one of the factors is removed

We see that using all factors have contributed to yield better performance. Without within-segment similarity, the overall performance decreases by up to 18%. This validates combination of within-segment and between-segment similarity, as stated in section 2.2. Length penalty factor also benefits greatly. This is strong evidence that segment length is a good predictor of segment boundaries. Without distance-based weighting strategy, the performance is also slightly lower, indicating that this strategy helps to capture lexical distribution in the discourse more precisely.

4.5 Experiment IV - Experimental Results on Real Corpus

In this section we present comparative experimental results on the *Mars* real corpus. Paragraphs in the corpus are regarded as the correct segmentation. In this suite of experiment, both P_k and WindowDiff metrics are examined to demonstrate the effectiveness of algorithms. Experimental results of Dotplotting and C99 come from the Choi's software package aforementioned in section 4.1. CWM algorithm is excluded from the comparison because the exact implementation package is not publicly available. In addition, because CWM is rather domain dependent and requires training, it is not possible to acquire and duplicate training corpus in our experiment. In our experiments the desired number of segments is given in advance for all algorithms. **Error! Reference source not found.** summarizes the evaluation values (α is set to 0.2).

Method	P_k	WindowDiff
D_Max	47.92%	55.27%
D_Min	43.47%	49.14%
C99	41.47%	45.05%
MMD	**40.09**%	**43.91**%

Table 4. P_k and WindowDiff Comparison with Dotplotting methods on testing data set 2

As shown in Table 4, our MMD method outperforms both original Dotplotting methods and the improved version C99 algorithm. This assessment on real corpus indicates the robustness of our model.

5 Conclusion and Future Work

In this paper we conducted theoretical analysis of the Dotplotting method for text segmentation. We have identified several deficiencies with respect to the density function of Dotplotting and proposed an improved model that remedies these problems. We have given an analytical form of the improved segmentation evaluation function and described a complete text segmentation algorithm using Dotplotting searching scheme.

Experimental results on public available synthetic corpus (Choi) and real corpus (*Mars*) are provided and compared with other systems using the same searching scheme. Our proposed MMD model, is shown to be promising and effective in text segmentation as it outperforms all other systems in most testing data sets, for both domain independent and domain dependent systems. In comparing with the best comparable system (C99, domain independent), the MMD model has achieved a 20% improvement in performance on the Choi benchmark corpus.

In the future we plan to optimize our algorithm in automatically determining the number of segments, and improve it when large variation of segment numbers exists in a given text. The searching strategy is also worth of further studying since dynamic programming has shown to be a promising searching strategy in some works (Ji and Zha, 2003). The critical problem to be solved is an appropriate combination of segmentation evaluating function and the optimization algorithm. In addition, more adequate segment length factor needs to be investigated.

It is demonstrated in (Bestgen, 2006) that semantic information trained from background corpus can help improve text segmentation performance. In the future we will also consider introducing semantic knowledge in the model.

6 Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant No.60473140, the National 863 High-tech Project No. 2006AA01Z154 ; the Program for New Century Excellent Talents in University No. NCET-05-0287 ; and the National 985 Project No.985-2-DB-C03 .

We would like to thank Yan Zheng and Haitao Luo for their assistance.

7 References

Beeferman, D., Berger, A., and Lafferty, J. 1997. Text Segmentation Using Exponential Models. In Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing, pp. 35–46.

Boguraev, Branimir and Mary Neff. 2000. Discourse segmentation in aid of document summarization. In Proceedings of the 33rd Hawaii International Conference on System Sciences, Maui, HI.

Choi, F.Y.Y. 2000. Advances in domain independent linear text segmentation. In Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics, pp. 26-33.

Choi, F.Y.Y., Wiemer-Hastings, P., and Moore, J. 2001. Latent Semantic Analysis for Text Segmentation. In Proceedings of the 6th Conference on Empirical Methods in Natural Language Processing, pp. 109–117.

Heinonen, O. 1998. Optimal Multi-Paragraph Text Segmentation by Dynamic Programming. In Proceedings of 17th International Conference on Computational Linguistics pp. 1484–1486.

Igor Malioutov and Regina Barzilay. Minimum Cut Model for Spoken Lecture Segmentation. In Proceedings of the ACL'06, pp. 25-32.

Jeffrey C. Reynar. 1994. An automatic method of finding topic boundaries. In Proceedings of ACL'94 (Student session).

Jeffrey C. Reynar. 1998. Topic segmentation: Algorithms and applications. Ph.D. thesis, Computer and Information Science, University of Pennsylvania.

Ji, X., Zha, H., 2003. Domain-independent Text Segmentation Using Anisotropic Diffusion and Dynamic Programming. in Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp.322-329.

L. Pevzner and M. Hearst. A Critique and Improvement of an Evaluation Metric for Text Segmentation. Computational Linguistics, 28 (1), pp.19-36, 2002.

Marti A. Hearst. 1994. Multi-paragraph segmentation of expository text. In Proceedings of the ACL'94. Las Crces, NM.

Min-Yen Kan, Judith L. Klavans, and Kathleen R. McKeown. 1998. Linear segmentation and segment significance. In Proceedings of the 6th International Workshop of Very Large Corpora(WVLC-6), pp. 197–205.

P. Fragkou, V. Petridis and Ath. Kehagias. 2004. A Dynamic Programming Algorithm for Linear Text Segmentation. Journal of Intelligent Information Systems, 23:2, 179–197

Porter, M.F. 1980. An Algorithm for Suffix Stripping. Program, 14, 130–137.

T. Brants, F. Chen, and I. Tsochantarides. 2002. Topic-based document segmentation with probabilistic latent semantic analysis. In Proceedings of the 11th International Conference on Information and Knowledge Management, pp.211-218.

Utiyama, M. and Isahara, H. 2001. A Statistical Model for Domain-Independent Text Segmentation. In Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics, pp. 491-498

Zhu Jingbo, Ye Na, Chang Xinzhi, Chen Wenliang and Benjamin K Tsou. 2005. Using Multiple Discriminant Analysis Approach for Linear Text Segmentation. In Proceedings of the Second International Joint Conference on Natural Language Processing, pp. 292-301