TopicTiling: A Text Segmentation Algorithm based on LDA

Martin Riedl and **Chris Biemann**
Ubiquitous Knowledge Processing Lab
Computer Science Department, Technische Universität Darmstadt
Hochschulstrasse 10, D-64289 Darmstadt, Germany
{riedl,biemann}@cs.tu-darmstadt.de

Abstract

This work presents a Text Segmentation algorithm called *TopicTiling*. This algorithm is based on the well-known TextTiling algorithm, and segments documents using the *Latent Dirichlet Allocation (LDA)* topic model. We show that using the mode topic ID assigned during the inference method of LDA, used to annotate unseen documents, improves performance by stabilizing the obtained topics. We show significant improvements over state of the art segmentation algorithms on two standard datasets. As an additional benefit, TopicTiling performs the segmentation in linear time and thus is computationally less expensive than other LDA-based segmentation methods.

1 Introduction

The task tackled in this paper is Text Segmentation (TS), which is to be understood as the segmentation of texts into topically similar units. This implies, viewing the text as a sequence of subtopics, that a subtopic change marks a new segment. The challenge for a text segmentation algorithm is to find the sub-topical structure of a text.

In this work, this semantic information is gained from Topic Models (TMs). We introduce a newly developed TS algorithm called *TopicTiling*. The core algorithm is a simplified version of *TextTiling* (Hearst, 1994), where blocks of text are compared via bag-of-word vectors. *TopicTiling* uses topic IDs, obtained by the LDA inference method, instead of words. As some of the topic IDs obtained by the inference method tend to change for different runs, we recommend to use the most probable topic ID assigned during the inference. We denote this most probable topic ID as the mode (most frequent across all inference steps) of the topic assignment. These IDs are used to calculate the cosine similarity between two adjacent blocks of sentences, represented as two vectors, containing the frequency of each topic ID. Without parameter optimization we obtain state-of-the-art results based on the Choi dataset (Choi, 2000). We show that the mode assignment improves the results substantially and improves even more when parameterizing the size of sampled blocks using a window size parameter. Using these optimizations, we obtain significant improvements compared to other algorithms based on the Choi dataset and also on a more difficult Wall Street Journal (WSJ) corpus provided by Galley et al. (2003). Not only does TopicTiling deliver state-of-the-art segmentation results, it also performs the segmentation in linear time, as opposed to most other recent TS algorithms.

The paper is organized as follows: The next section gives an overview of text segmentation algorithms. Section 3 introduces the TopicTiling TS algorithm. The Choi and the Galley datasets used to measure the performance of TopicTiling are described in Section 4. In the evaluation section, the results of TopicTiling are demonstrated on these datasets, followed by a conclusion and discussion.

2 Related Work

TS can be divided into two sub-fields: (i) linear TS and (ii) hierarchical TS. Whereas linear TS deals with the sequential analysis of topical changes,

Proceedings of the 2012 Student Research Workshop, pages 37–42,

hierarchical segmentation is concerned with finding more fine grained subtopic structures in texts. One of the first unsupervised linear TS algorithms was introduced by Hearst (1994): *TextTiling* segments texts in linear time by calculating the similarity between two blocks of words based on the cosine similarity. The calculation is accomplished by two vectors containing the number of occurring terms of each block. *LcSeg* (Galley et al., 2003), a TextTiling-based algorithm, uses tf-idf term weights and improved TS results compared to TextTiling. Utiyama and Isahara (2001) introduced one of the first probabilistic approaches using Dynamic Programming (DP) called *U00*. Related to our work are the DP approaches described in Misra et al. (2009) and Sun et al. (2008): here, topic modeling is used to alleviate the sparsity of word vectors. This approach was extended by (Misra et al., 2009) and (Sun et al., 2008) using topic information achieved from the LDA topic model. The first hierarchical algorithm was proposed by Yaari (1997), using the cosine similarity and agglomerative clustering approaches. A hierarchical Bayesian algorithm based on LDA is introduced with Eisenstein (2009). In our work, however, we focus on linear TS.

LDA was introduced by Blei et al. (2003) and is a generative model that discovers topics based on a training corpus. Model training estimates two distributions: A topic-word distribution and a topic-document distribution. As LDA is a generative probabilistic model, the creation process follows a generative story: First, for each document a topic distribution is sampled. Then, for each document, words are randomly chosen, following the previously sampled topic distribution. Using the Gibbs inference method, LDA is used to apply a trained model for unseen documents. Here, words are annotated by topic IDs by assigning a topic ID sampled by the document-word and word-topic distribution. Note that the inference procedure, in particular, marks the difference between LDA and earlier dimensionality reduction techniques such as Latent Semantic Analysis.

3 TopicTiling

This section introduces the *TopicTiling* algorithm, first introduced in (Riedl and Biemann, 2012a). In contrast to the quite similar TextTiling algorithm, TopicTiling is not based on words, but on the last topic IDs assigned by the Bayesian Inference method of LDA. This increases sparsity since the word space is reduced to a topic space of much lower dimension. Therefore, the documents that are to be segmented have first to be annotated with topic IDs. For useful topic distinctions, however, the topic model must be trained on documents similar in content to the test documents. Preliminary experiments have shown that repeating the Bayesian inference, often leads to different topic distributions for a given sentence in several runs. Memorizing each topic ID assigned to a word in a document during each inference step can alleviate this instability, which is rooted in the probabilistic nature of LDA. After finishing the inference on the unseen documents, we select the most frequent topic ID for each word and assign it to the word. We call this method the mode of a topic assignment, denoted with $d = true$ in the remainder (Riedl and Biemann, 2012b). Note that this is different from using the overall topic distribution as determined by the inference step, since this winner-takes-it-all approach reduces noise from random fluctuations. As this parameter stabilizes the topic IDs at low computational costs, we recommend using this option in all setups where subsequent steps rely on a single topic assignment.

TopicTiling assumes a sentence s_i as the smallest basic unit. At each position p, located between two adjacent sentences, a *coherence score* c_p is calculated. With w we introduce a so-called *window parameter* that specifies the number of sentences to the left and to the right of position p that define two *blocks*: $s_{p-w}, \ldots, s_p$ and $s_{p+1}, \ldots, s_{p+w+1}$. In contrast to the mode topic assignment parameter d, we cannot state a recommended value for w, as this parameter is dependent on the number of sentences a segment should contain. This is conditioned on the corpus that is segmented.

To calculate the coherence score, we exclusively use the topic IDs assigned to the words by inference: Assuming an LDA model with T topics, each block is represented as a T-dimensional vector. The t-th element of each vector contains the frequency of the topic ID t obtained from the according block. The coherence score is calculated by the vector dot product, also referred to as *cosine similarity*. Val-

ues close to zero indicate marginal relatedness between two adjacent blocks, whereas values close to one denote a substantial connectivity. Next, the coherence scores are plotted to trace the local minima. These minima are utilized as possible segmentation boundaries. But rather using the c_p values itself, a *depth score* d_p is calculated for each minimum (cf. TextTiling, (Hearst, 1994)). In comparison to TopicTiling, TextTiling calculates the depth score for each position and than searches for maxima. The depth score measures the deepness of a minimum by looking at the highest coherence scores on the left and on the right and is calculated using following formula: $d_p = 1/2(hl(p) - c_p + hr(p) - c_p)$.

The function $hl(p)$ iterates to the left as long as the score increases and returns the highest coherence score value. The same is done, iterating in the other direction with the $hr(p)$ function. If the number of segments n is given as input, the n highest depth scores are used as segment boundaries. Otherwise, a threshold is applied (cf. TextTiling). This threshold predicts a segment if the depth score is larger than $\mu - \sigma/2$, with μ being the mean and σ being the standard variation calculated on the depth scores.

The algorithm runtime is linear in the number of possible segmentation points, i.e. the number of sentences: for each segmentation point, the two adjacent blocks are sampled separately and combined into the coherence score. This, and the parameters d and w, are the main differences to the dynamic programming approaches for TS described in (Utiyama and Isahara, 2001; Misra et al., 2009).

4 Data Sets

The performance of the introduced algorithm is demonstrated using two datasets: A dataset proposed by Choi and another more challenging one assembled by Galley.

4.1 Choi Dataset

The Choi dataset (Choi, 2000) is commonly used in the field of TS (see e.g. (Misra et al., 2009; Sun et al., 2008; Galley et al., 2003)). It is a corpus, generated artificially from the Brown corpus and consists of 700 documents. For document generation, ten segments of 3-11 sentences each, taken from different documents, are combined forming one document. 400 documents consist of segments with a sentence length of 3-11 sentences and there are 100 documents each with sentence lengths of 3-5, 6-8 and 9-11.

4.2 Galley Dataset

Galley et al. (2003) present two corpora for written language, each having 500 documents, which are also generated artificially. In comparison to Choi's dataset, the segments in its 'documents' vary from 4 to 22 segments, and are composed by concatenating full source documents. One dataset is generated based on WSJ documents of the Penn Treebank (PTB) project (Marcus et al., 1994) and the other is based on Topic Detection Track (TDT) documents (Wayne, 1998). As the WSJ dataset seems to be harder (consistently higher error rates across several works), we use this dataset for experimentation.

5 Evaluation

The performance of TopicTiling is evaluated using two measures, commonly used in the TS task: The P_k measure and the WindowDiff (WD) measure (Beeferman et al., 1999; Pevzner and Hearst, 2002). Besides the training corpus, the following parameters need to be specified for LDA: The number of topics T, the number of sample iterations for the model m and two hyperparameters α and β, specifying the sparseness of the topic-document and the topic-word distribution. For the inference method, the number of sampling iterations i is required. In line with Griffiths and Steyvers (2004), the following standard parameters are used: $T = 100$, $\alpha = 50/T$, $\beta = 0.01$, $m = 500$, $i = 100$. We use the JGibbsLDA implementation described in Phan and Nguyen (2007).

5.1 Evaluation of the Choi Dataset

For the evaluation we use a 10-fold Cross Validation (CV): the full dataset of 700 documents is split into 630 documents for training the topic model and 70 documents that are segmented. These two steps are repeated ten times to have all 700 documents segmented. For this dataset, no part-of-speech based word filtering is necessary. The results for different parameter settings are listed in Table 1.

When using only the window parameter without the mode (d=false), the results demonstrate a sig-

seg. size	3-5		6-8		9-11		3-11	
	P_k	WD	P_k	WD	P_k	WD	P_k	WD
d=false,w=1	2.71	3.00	3.64	4.14	5.90	7.05	3.81	4.32
d=true,w=1	3.71	4.16	1.97	2.23	2.42	2.92	2.00	2.30
d=false,w=2	1.46	1.51	1.05	1.20	1.13	1.31	1.00	1.15
d=true,w=2	**1.24**	**1.27**	**0.76**	**0.85**	**0.56**	**0.71**	**0.95**	**1.08**
d=false,w=5	2.78	3.04	1.71	2.11	4.47	4.76	3.80	4.46
d=true,w=5	2.34	2.65	1.17	1.35	4.39	4.56	3.20	3.54

Table 1: Results based on the Choi dataset with varying parameters.

nificant error reduction when using a window of 2 sentences. An impairment is observed when using a too large window (w=5). This is expected, as the size of the segments is in a range of 3-11 sentences: A window of 5 sentences therefore leads to blocks that contain segment boundaries. We can also see that the mode method improves the results when using a window of one, except for the documents having small segments ranging from 3-5 sentences. The lowest error rates are obtained with the mode method and a window size of 2.

As described above, the algorithm is also able to automatically estimate the number of segments using a threshold value (see Table 2).

	3-5		6-8		9-11		3-11	
	P_k	WD	P_k	WD	P_k	WD	P_k	WD
d=false,w=1	**2.39**	**2.45**	4.09	5.85	9.20	15.44	4.87	6.74
d=true,w=1	3.54	3.59	1.98	2.57	3.01	5.15	2.04	2.62
d=false,w=2	15.53	15.55	0.79	0.88	1.98	3.23	1.03	1.36
d=true,w=2	14.65	14.69	**0.62**	**0.62**	**0.67**	**0.88**	**0.66**	**0.78**
d=false,w=5	21.47	21.62	16.30	16.30	6.01	6.14	14.31	14.65
d=true,w=5	21.57	21.67	17.24	17.24	6.44	6.44	15.51	15.74

Table 2: Results on the Choi dataset without given number of segments as parameter.

The results show that for small segments, the number of segments is not correctly estimated, as the error rates are much higher than with given segments. As the window parameter has a smoothing effect on the coherence score function, less possible boundary candidates are detected. We can also see that the usage of the mode parameter leads to worse results with w=1 compared to the results where the mode is deactivated for the documents containing segments of length 3-5. Especially, results on these documents suffer when not providing the number of segments. But for the other documents, results are much better. Some results (see segment lengths 6-8 and 3-11 with d=true and w=2) are even better than the results with segments provided (see Table 1). The threshold method can outperform the setup with given a number of segments, since not recognizing a segment produces less error in the measures than predicting a wrong segment.

Table 3 presents a comparison of the performance of TopicTilig compared to different algorithms in the literature.

Method	**3-5**	**6-8**	**9-11**	**3-11**
TT (Choi, 2000)	44	43	48	46
C99 (Choi, 2000)	12	9	9	12
U00 (Utiyama and Isahara, 2001)	9	7	5	10
LCseg (Galley et al., 2003)	8.69			
F04 (Fragkou et al., 2004)	5.5	3.0	1.3	7.0
M09 (Misra et al., 2009)	2.2	2.3	4.1	2.3
TopicTiling (d=true, w=2)	**1.24**	**0.76**	**0.56**	**0.95**

Table 3: Lowest P_k values for the Choi data set for various algorithms in the literature with number of segments provided

It is obvious that the results are far better than current state-of-the-art results. Using a one-sampled t-test with $\alpha = 0.05$ we can state significant improvements in comparison to all other algorithms.

While we aim not using the same documents for training and testing by using a CV scheme, it is not guaranteed that all testing data is unseen, since the same source sentences can find their way in several artificially crafted 'documents'. We could detect reoccurring snippets in up to 10% of the documents provided by Choi. This problem, however, applies for all evaluations on this dataset that use any kind of training, be it LDA models in Misra et al. (2009) or tf-idf values in Fragkou et al. (2004) and Galley et al. (2003).

5.2 Evaluation on Galley's WSJ Dataset

For the evaluation on Galley's WSJ dataset, a topic model is created from the WSJ collection of the PTB project. The dataset for model estimation consists of 2499 WSJ articles, and is the same dataset Galley used as a source corpus. The evaluation generally leads to higher error rates than in the evaluation for the Choi dataset, as shown in Table 4.

This table shows results of the WSJ data when using all words of the documents for training a topic model and assigning topic IDs to new documents and also filtered results, using only nouns (proper

Parameters	All words		Filtered	
	P_k	WD	P_k	WD
d=false,w=1	37.31	43.20	37.01	43.26
d=true,w=1	35.31	41.27	33.52	39.86
d=false,w=2	22.76	28.69	21.35	27.28
d=true,w=2	21.79	27.35	19.75	25.42
d=false,w=5	14.29	19.89	12.90	18.87
d=true,w=5	**13.59**	**19.61**	**11.89**	**17.41**
d=false,w=10	14.08	22.60	14.09	22.22
d=true,w=10	13.61	21.00	13.48	20.59

Table 4: Results for Galley's WSJ dataset using different parameters with using unfiltered documents and with filtered documents using only verbs, nouns (proper and common) and adjectives.

and common), verbs and adjectives[1]. Considering the unfiltered results we observe that results improve when using the mode assigned topic ID and a window of larger than one sentence. In case of the WSJ dataset, we find the optimal setting for w=5. As the test documents contain whole articles, which consist of at least 4 sentences, a larger window is advantageous here, yet a value of 10 is too large. Filtering the documents for parts of speech leads to $\sim$ 1% absolute error rate reduction, as can be seen in the last two columns of Table 4. Again, we observe that the mode assignment always leads to better results, gaining at least 0.6%. Especially the window size of 5 helps TopicTiling to decrease the error rate to a third of the value observed with d=false and w=1. Similar to the previous findings, results decline when using a too large window.

Table 5 shows the results we achieve with the threshold-based estimation of segment boundaries for the unfiltered and filtered data.

Parameters	All words		Filtered	
	P_k	WD	P_k	WD
d=false,w=1	53.07	72.78	52.63	72.66
d=true,w=1	53.42	74.12	51.84	72.57
d=false,w=2	46.68	65.01	44.81	63.09
d=true,w=2	46.08	64.41	43.54	61.18
d=false,w=5	30.68	43.73	28.31	40.36
d=true,w=5	28.29	38.90	26.96	36.98
d=false,w=10	19.93	32.98	18.29	29.29
d=true,w=10	**17.50**	**26.36**	**16.32**	**24.75**

Table 5: Table with results the WSJ dataset without number of segments given, using all words and content words only.

[1]The Treetagger `http://code.google.com/p/tt4j/` is applied to POS-tag the data

In contrast to the results obtained with the Choi dataset (see Table 2) no decline is observed when the threshold approach is used in combination with the window approach. We attribute this due to the small segments and documents used in the Choi setting. Comparing the all-words data with pos-filtered data, an improvement is always observed. Also a continuous decreasing of both error rates, P_k and WD, is detected when using the mode and using a larger window size, even for w=10. The reason for this is that too many boundaries are detected when using small windows. As the window approach smoothes the similarity scores, this leads to less segmentation boundaries, which improve results.

For comparison, we present the evaluation results of other algorithms, shown in Table 6, as published in Galley et al. (2003).

Method	P_k	WD
C99 (Choi, 2000)	19.61	26.42
U00 (Utiyama and Isahara, 2001)	15.18	21.54
LCseg (Galley et al., 2003)	12.21	18.25
TopicTiling (d=true,w=5)	**11.89**	**17.41**

Table 6: List of results based on the WSJ dataset. Values for C99, U00 and LCseg as stated in (Galley et al., 2003).

Again, TopicTiling improves over the state of the art. The improvements with respect to LCseg are significant using a one-sample t-test with $\alpha = 0.05$.

6 Conclusion and Further Work

We introduced *TopicTiling*, a new TS algorithm that outperforms other algorithms as shown on two datasets. The algorithm is based on TextTiling and uses the topic model LDA to find topical changes within documents. A general result with implications to other algorithms that use LDA topic IDs is that using the mode of topic assignments across the different inference steps is recommended to stabilize the topic assignments, which improves performance. As the inference method is relatively fast in comparison to building a model, this mechanism is a useful and simple improvement, not only restricted to the field of TS. Using more than a single sentence in inference blocks leads to further stability and less sparsity, which improves the results further. In contrast to other TS algorithms using topic models (Misra et al., 2009; Sun et al., 2008), the runtime of TopicTiling is linear in the number of sentences. This

makes TopicTiling a fast algorithm with complexity of $O(n)$ (n denoting the number of sentences) as opposed to $O(n^2)$ of the dynamic programming approach as discussed in Fragkou et al. (2004).

Text segmentation benefits from the usage of topic models. As opposed to general-purpose lexical resources, topic models can also find fine-grained subtopical changes, as shown with the segmentation results of the WSJ dataset. Here, most articles have financial content and the topic model can e.g. distinguish between commodity and stock trading. The topic model adapts to the subtopic distribution of the target collection, in contrast e.g. to static WordNet domain labels as in Bentivogli et al. (2004).

For further work, we would like to devise a method to detect the optimal setting for the window parameter w automatically, especially in a setting where the number of target segments is not known in advance. This is an issue that is shared with the original TextTiling algorithm. Moreover, we will extend the usage of our algorithm to more realistic corpora.

Another direction of research that is more generic for approaches based on topic models is the question of how to automatically select appropriate data for topic model estimation, given only a small target collection. Since topic model estimation is computationally expensive, and topic models for generic collections (think Wikipedia) might not suit the needs of a specialized domain (such as with the WSJ data), it is a promising direction to look at target-domain-driven automatic corpus synthesis.

Acknowledgments

This work has been supported by the Hessian research excellence program "Landes-Offensive zur Entwicklung Wissenschaftlich-konomischer Exzellenz" (LOEWE) as part of the research center "Digital Humanities".

References

D. Beeferman, A. Berger, and J. Lafferty. 1999. Statistical models for text segmentation. *Mach. learn.*, 34(1):177–210.

L. Bentivogli, P. Forner, B. Magnini, and E. Pianta. 2004. Revising the wordnet domains hierarchy: semantics, coverage and balancing. In *Proc. COLING 2004 MLR*, pages 101–108, Geneva, Switzerland.

D. M. Blei, A. Y Ng, and M. I. Jordan. 2003. Latent Dirichlet Allocation. *JMLR '03*, 3:993–1022.

F. Y. Y. Choi. 2000. Advances in domain independent linear text segmentation. In *Proc 1st NAACL '00*, pages 26–33, Seattle, WA, USA.

J. Eisenstein. 2009. Hierarchical text segmentation from multi-scale lexical cohesion. In *Proc. NAACL-HLT '09*, pages 353–361, Boulder, CO, USA.

P. Fragkou, V. Petridis, and A. Kehagias. 2004. A Dynamic Programming Algorithm for Linear Text Segmentation. *JIIS '04*, 23(2):179–197.

M. Galley, K. McKeown, E. Fosler-Lussier, and H. Jing. 2003. Discourse segmentation of multi-party conversation. In *Proc 41st ACL '03*, volume 1, pages 562–569, Sapporo, Japan.

T. L. Griffiths and M. Steyvers. 2004. Finding scientific topics. *PNAS*, 101:5228–5235.

M. A. Hearst. 1994. Multi-paragraph segmentation of expository text. In *Proc. 32nd ACL '94*, pages 9–16, Las Cruces, NM, USA.

M. Marcus, G. Kim, M. A. Marcinkiewicz, R. Macintyre, A. Bies, M. Ferguson, K. Katz, and B. Schasberger. 1994. The Penn treebank: Annotating predicate argument structure. In *Proc. ARPA-HLT Workshop '94*, pages 114–119, Plainsboro, NJ, USA.

Hemant Misra, Joemon M Jose, and Olivier Cappé. 2009. Text Segmentation via Topic Modeling : An Analytical Study. In *Proc. 18th CIKM '09*, pages 1553–1556, Hong Kong.

L. Pevzner and M. A. Hearst. 2002. A Critique and Improvement of an Evaluation Metric for Text Segmentation. *Computational Linguistics*, 28.

X.-H. Phan and C.-T. Nguyen. 2007. GibbsLDA++: A C/C++ implementation of latent Dirichlet allocation (LDA). http://jgibblda.sourceforge.net/.

M. Riedl and C. Biemann. 2012a. How text segmentation algorithms gain from topic models. In *Proc. NAACL-HLT '12*, Montreal, Canada.

M. Riedl and C. Biemann. 2012b. Sweeping through the Topic Space: Bad luck? Roll again! In *ROBUS-UNSUP at EACL '12*, Avignon, France.

Q. Sun, R. Li, D. Luo, and X. Wu. 2008. Text segmentation with LDA-based Fisher kernel. In *Proc. 46th ACl-HLT '08*, pages 269–272, Columbus, OH, USA.

M. Utiyama and H. Isahara. 2001. A statistical model for domain-independent text segmentation. In *Proc. 39th ACL '00*, pages 499–506, Toulouse, France.

C. Wayne. 1998. Topic detection and tracking (TDT): Overview & perspective. In *Proc. DARPA BNTUW*, Lansdowne, Virginia.

Y. Yaari. 1997. Segmentation of expository texts by hierarchical agglomerative clustering. In *Proc. RANLP '97*, Tzigov Chark, Bulgaria.