Using Text Segmentation Algorithms for the Automatic Generation of E-Learning Courses

Can Beck, Alexander Streicher and Andrea Zielinski
Fraunhofer IOSB
Karlsruhe, Germany
{can.beck, alexander.streicher,
andrea.zielinski}@iosb.fraunhofer.de

Abstract

With the advent of e-learning, there is a strong demand for tools that help to create e-learning courses in an automatic or semi-automatic way. While resources for new courses are often freely available, they are generally not properly structured into easy to handle units. In this paper, we investigate how state of the art text segmentation algorithms can be applied to automatically transform unstructured text into coherent pieces appropriate for e-learning courses. The feasibility to course generation is validated on a test corpus specifically tailored to this scenario. We also introduce a more generic training and testing method for text segmentation algorithms based on a Latent Dirichlet Allocation (LDA) topic model. In addition we introduce a scalable random text segmentation algorithm, in order to establish lower and upper bounds to be able to evaluate segmentation results on a common basis.

1 Introduction

The creation of e-learning courses is generally a time consuming effort. However, separating text into topically cohesive segments can help to reduce this effort whenever textual content is already available but not properly structured according to e-learning standards. Since these segments textually describe the content of learning units, automatic pedagogical annotation algorithms could be applied to categorize them into introductions, descriptions, explanations, examples and other pedagogical meaningful concepts (K.Sathiyamurthy & T.V.Geetha, 2011).

Course designers generally assume that learning content is composed of small inseparable learning objects at the micro level which in turn are wrapped into Concept Containers (CCs) at the macro level. This approach is followed, e.g., in the Web-Didactic approach by Swertz et al. (2013) where CCs correspond to chapters in a book and Knowledge Objects (KOs) correspond to course pages. To automate the partition of an unstructured text source into appropriate segments for the macro and micro level we applied different text segmentation algorithms (segmenters) on each level.

To evaluate the segmenters in the described scenario, we created a test corpus based on featured Wikipedia articles. For the macro level we exploit sections from different articles and the corresponding micro structure consists of subsequent paragraphs from these sections. On the macro level the segmenter TopicTiling (TT) by Riedl and Biemann (2012) is used. It is based on a LDA topic model which we train based on the articles from Wikipedia to extract a predefined number of different topics. On the micro level, the segmenter BayesSeg (BS) is applied (Eisenstein & Barzilay, 2008).

We achieved overall good results measured in three different metrics over a baseline approach, i.e., a scalable random segmenter, that indicate

This work is licensed under a Creative Commons Attribution 4.0 International Licence. Page numbers and proceedings footer are added by the organisers. Licence details: http://creativecommons.org/licenses/by/4.0/

*Proceedings of the Third Joint Conference on Lexical and Computational Semantics (*SEM 2014)*, pages 132–140,
Dublin, Ireland, August 23-24 2014.

text segmentation algorithms are ready to be applied to facilitate the creation of e-learning courses.

This paper is organized as follows: Section 2 gives an overview of related work on automatic course generation as well as text segmentation applications. In the main sections 3 and 4 we describe our approach and evaluation results on our corpus. In the last section we summarize the presented findings and give an outlook on further research needed for the automatic generation of e-learning courses.

2 Related Work

Automatic course generation can roughly be divided into two different areas. One is concerned with generation from existing courses and is mainly focused on adaption to the learner or instructional plans see Lin et al. (2009), Capuno et al. (2009) and Tan et al. (2010). The other area is the course creation itself on which we focus on in this paper.

Since the publication of the segmenter TextTiling by Hearst (1997) at least a dozen different segmenters have been developed. They can be divided into linear and hierarchical segmenters. Linear segmenters process the text sequentially sentence by sentence. Hierarchical segmenters first process the whole text and extract topics with varying granularities. These topics are then agglomerated based on a predefined criterion.

Linear segmenters have been developed by Kan et al. (1998) and Galley et al. (2003). One of the first probabilistic algorithms has been introduced by Utiyama and Isahara (2001). LDA based approaches were first described by Sun et al. (2008) and improved by Misra et al. (2009). The newest LDA based segmenter is TT. It performs linear text segmentation based on a pretrained LDA topic model and calculates the similarity between segments (adjacent sentences) to measure text coherence on the basis of a topic vector representation using cosine similarity. For reasons of efficiency, only the most frequent topic ID is assigned to each word in the sentence, using Gibbs sampling.

Hierarchical text segmentation algorithms were first introduced by Yaari (1997). The latest approach by Eisenstein (2008) uses a generative Bayesian model BS for text segmentation, assuming that a) topic shifts are likely to occur at points marked by cue phrases and b) a linear discourse structure. Each sentence in the document is modeled by a language model associated with a segment. The algorithm then calculates the maximum likelihood estimates of observing the whole sequence of sentences at selected topic boundaries.

The applications of text segmentation algorithms range from information retrieval (Huang, et al., 2002) to topic tracking and segmentation of multi-party conversations (Galley, et al., 2003).

Similar to our work Sathiyamurthy and Geetha (2011) showed how LDA based text segmentation algorithms combined with hierarchical domain ontology and pedagogical ontology can be applied to content generation for e-learning courses. They focussed on the segmentation of existing e-learning material in the domain of computer science and introduced new metrics to measure the segmentation results with respect to concepts from the ontologies. Our work focusses on the appropriate segmentation of unstructured text instead of existing e-learning material. Although the usage of domain models is an interesting approach the availability of such models is very domain dependent. We rely on the LDA model parameters and training to accomplish a word to topic assignment.

Rather than introducing new aspects such as pedagogical concepts we investigated the general usability of segmentation algorithms with focus on the macro and micro structure which is characteristic for most e-learning content.

3 Automatic Generation of E-Learning Courses

The main objective is to provide e-learning course designers with a tool to efficiently organize existing textual content for new e-learning courses. This can be done by the application of text segmenters that automatically generate the basic structure of the course. The intended web-didactic conform two-level structure differentiates between macro and micro levels. The levels have different requirements with respect to thematic coherence: the CCs are thematically rather independent and the KOs within each CC need to be intrinsically coherent but still separable.

We chose the linear LDA-based segmenter TT to find the boundaries between CCs. The LDA-based topic model can be trained on content which is topically related to the target course. This approach gives the course creator flexibility in the generation of the macro level structure by either adjusting the training documents or by

changing the number and size of topics that should be extracted for the topic model.

On the micro level we did not use TT. The training of an appropriate LDA model would have to be done for every CC separately since they are thematically relatively unrelated. Apart from that the boundaries between the KOs should be an optimal division for a given number of expected boundaries. The reason for this is that the length of KOs should be adapted to the intended skill and background of the learners. This is why we decided to use the hierarchical segmenter BS.

3.1 Application Setting and Corpus

To evaluate segmenters many different corpora have been created. The most commonly used corpus was introduced by Choi (2000). It is based on the Brown Corpus and contains 700 samples, each containing a fixed number of sentences from 10 different news texts, which are randomly chosen from the Brown Corpus. Two other widely tested corpora were introduced by Galley et al. (2003). Both contain 500 samples, one with concatenated texts from the Wall Street Journal (WSJ) and the other with concatenated texts from the Topic Detection and Tracking (TDT) corpus (Strassel, et al., 2000). A standard for the segmentation of speech is the corpus from the International Computer Science Institute (ICSI) by Janin et al. (2003). A medical text book has been used by Eisenstein and Barzilay (2008). The approaches to evaluate segmenters are always similar: they have to find the boundaries in artificially concatenated texts.

We developed our own dataset because we wanted to use text that potentially could be used as a basis for creating e-learning courses. We therefore need samples which, on the one hand, have relatively clear topic boundaries on the macro level and, on the other hand resemble the differences in number of topics and inter-topic cohesion on the micro level.

We based our corpus on 530 featured[1] articles from 6 different categories of the English Wikipedia. It can be assumed that Wikipedia articles are often the source for learning courses. We used featured articles because the content structure is very consistent and clear, i.e., sections and paragraphs are well defined.

The corpus is divided into a macro and micro dataset in the following manner: The macro dataset contains 1200 samples. Each sample is a concatenation of paragraphs from 6-8 different sections from featured articles. Each topic in a sample consists of 3-6 subsequent paragraphs from a randomly selected section. We propose that one paragraph describes one KO. One CC contains all KOs which are from the same section in the article. Thus, one sample from the macro dataset contains 6-8 CCs, each containing 3-6 KOs. The segmentation task is to find the topic boundaries between the CCs. The macro dataset is quite similar in structure to the Choi-Corpus.

The micro dataset is extracted from the macro dataset. It contains 8231 samples, where each sample contains all KOs from one CC of the macro dataset. The segmentation task is to find the topic boundaries between the KOs, i.e, subsequent paragraphs of one section, see Figure 1.

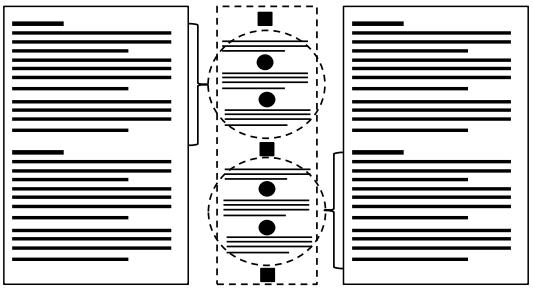


Figure 1: Schema for corpus samples: left and right Wikipedia articles with sections and paragraphs, in the middle three samples, dashed rectangle is a macro sample and dashed circles are micro samples. Filled squares indicate topic boundaries in the macro sample and filled circles in the micro samples.

All texts in our corpus are stemmed and stopwords are removed with the NLP-Toolkit for Python (Bird, et al., 2009) using an adapted variant[2] of the keyword extraction method by Kim et al. (2013).

The macro and micro dataset themselves are divided into multiple subsets to evaluate the stability of the segmenters when the number of sentences per topic or the number of topics per sample have changed. The detailed configuration is shown in Table 1 and 2. Each subset is identified by the number of CCs per sample and the number of KOs per CC (the subset is denoted as #CC_#KO). Subsets of the micro dataset are identified by a single value which is the number

[1] http://en.wikipedia.org/wiki/Wikipedia:Featured_articles

[2] https://gist.github.com/alexbowe/879414

of KOs per sample (#KO). In Table 1 the identifier *R* means that the number of CCs or KOs is not the same for all samples, it is chosen randomly from the set depicted by curly brackets.

ID	CCs per sample	KOs per CC	mean sentences per CC
7_3	7	3	20
7_4	7	4	27
7_5	7	5	33
7_6	7	6	40
7_R	7	{3,4,5,6}	30
R_R	{6,7,8}	{3,4,5,6}	30

Table 1: Macro dataset and its subsets each with 200 samples.

ID	KOs per sample	mean sentences per KO
3	3	9
4	4	8
5	5	7
6	6	7

Table 2: Micro dataset and its subsets.

The important difference between the macro and micro dataset is that every subset of the macro dataset contains a constant number of topics which differ in number of sentences per topic between 20 and 40, except the subset *R_R* which contains a random number of topics between 6 and 8. In contrast, each micro-level subset differs in number of topics but not significantly in the number of sentences per topic.

This difference between the datasets allows us to focus on the different level-specific aspects. On the macro dataset we can evaluate the stability of TT over topics with highly varying lengths and on the micro dataset we can evaluate BS when the number of strongly coherent topics changes.

3.2 Text Segmentation Metrics

The performance of a segmenter cannot simply be measured by false positive and false negative boundaries compared to the true boundaries because, if the predicted boundary is only one sentence away from the true boundary this could still be very close, e.g., if the next true topic boundary is 30 sentences away. Thus, the relative proximity to true boundaries should also be considered. There is an ongoing discussion about what kind of metric is appropriate to measure the performance of segmenters (Fournier & Inkpen, 2012). Most prominent and widely used are WindowDiff *wd* (Pevzner & Hearst, 2002) and the probabilistic metric *pk* (Beeferman, et al., 1999). The basic principle is to slide a window of fixed size over the segmented text, i.e., fixed number of words or sentences, and assess whether the sentences on the edges are correctly segmented with respect to each other. Both metrics *wd* and *pk* are penalty metrics, therefore lower values indicate better segmentations. The problem with these metrics is that they strongly depend on the arbitrarily defined window size parameter and do not penalize all error types equally, e.g., *pk* penalizes false negatives more than false positives and *wd* penalizes false positive and negative boundaries more at the beginning and end of the text (Lamprier, et al., 2007). Because of that we also used a rather new metric called BoundarySimilarity *b*. This metric is parameter independent and has been developed by Fournier and Inkpen (2013) to solve the mentioned deficiencies. Since *b* measures the similarity between the boundaries, higher values indicate better segmentations. We used the implementations of *wd*, *pk* and *b* by Fournier[3] (*wd* and *pk* with default parameters).

3.3 LDA Topic Model Training

Riedl and Biemann evaluated TT on the Choi-Corpus based on a 10-fold cross validation. Thus, the LDA topic model was generated with 90% of the samples and TT then tested on the remaining 10% of the samples. The 700 samples in the Choi-Corpus are only concatenations of 1111 different excerpts from the Brown Corpus and each sample contains 10 of these excepts it is clear that there are just not enough excerpts to make sure that the samples in the training set do not contain any excerpt that is also part of some samples in the testing set.

That is one reason why we do not use the same approach since we want to make sure that training and testing sets are truly disjoint to evaluate TT on the macro dataset. The other reason is that the topic structure generated by TT should be based on an LDA topic model with topics extracted from documents which are thematically related to certain parts of the course that is to be created without using its text source.

[3] https://github.com/cfournie/segmentation.evaluation

We train the LDA topic model to extract topics from the real Wikipedia articles. This model is then used to evaluate TT on the macro dataset and not the Wikipedia articles. This approach has consequences for the LDA topic model training and respective TT testing sets, since the LDA training set contains real articles and the TT test set contains the samples from the macro dataset. Because training and testing set should truly be disjoint we cannot train with any article that is part of a sample from the test set. Because each test sample from the macro dataset contains parts of 6 to 8 articles, the training set is reduced by a large factor, even with little test set size, which is shown for different number of folds (k) for cross validation in Table 3.

k	Test Set Size	Training Set Size
10	120±0 Samples (10% of the macro dataset)	139±7 featured Articles (26% of all articles)
20	60±0 Samples (5% of the macro dataset)	267±8 featured Articles (51% of all articles)
30	40±0 Samples (3% of the macro dataset)	338±7 featured Articles (64% of all articles)

Table 3: Mean size and standard deviation of truly disjunctive LDA training and respective TT testing set.

If we truly separate training and testing sets and train the LDA topic model with real articles a 10-fold cross validation leads to very small training sets (only 26% of all articles are used), which is why we also used higher folds to evaluate the results of TT on the macro dataset.

4 Evaluation Results

We evaluated TT on the macro dataset without providing the number of boundaries. On the micro dataset we evaluated BS with the expected number of boundaries provided. We also implemented a scalable random segmenter (RS) to compare TT and BS against some algorithm with interpretable performance. The interpretation of the values in any metric even with respect to different metrics is very difficult without comparison to another segmenter. For every true boundary in a document, RS predicts a boundary drawn from a normally distributed set around the true boundary with scalable standard deviation σ. Thus smaller values for σ result in better segmentations because the probability of selecting the true boundary increases, e.g., for σ = 2, more than 68% of all predicted boundaries are at most 2 sentences away from the true boundary and more than 99% of all predicted boundaries are located within a range of 6 sentences from it. But whether 6 sentences is a large or small distance should depend on the average topic size. We therefore relate the performance of RS to the mean number of sentence per topic by defining σ in percentages of that number as shown in the table below.

Distance from True Boundary:	Standard Deviation
very close	σ = 0% - 5%
close	σ = 5% -15%
large	σ = 15% - 30%

Table 4: Defined performance of RS for different standard deviations σ, given in percentage of mean sentences per topic.

To give an example, the subset *7_6* of the macro dataset has an average of 40 sentences per topic, therefore RS with σ=15% means that it is set to 6 which is 15% of 40. This is defined as a medium performance in Table 4 because 68% of the boundaries predicted are within a range of 6 sentences from the true boundaries and 99% within 18 sentences.

One important difference between the macro and micro dataset is that all subsets of the macro dataset have 7 topics, differing in length, except for subset *R_R* where this number is only slightly varied (Table 1). In contrast, all topics subsets of the micro dataset have roughly the same number of sentences but highly differ in the number of topics (Table 2). We therefore do not compare the performance of BS and TT since they are evaluated on quite different datasets designed for testing different types of segmentation tasks relevant to course generation, as explained earlier. We compare both to RS for different standard deviations σ.

4.1 Results for TopicTiling on the Macro Dataset

For the LDA topic model training we used the following default parameters: `alpha=0.5, beta=0.1,ntopics=100,niters=1000,`

`twords=20,savestep=100,` for details we refer to (Griffiths & Steyvers, 2004). To compare TT's performance for different folds of the macro dataset we optimized the window parameter which has to be set for TT, it specifies the number of sentences to the left and to the right of the current position *p* between two sentences that are used to calculate the coherence score between these sentences (Riedl & Biemann, 2012). The performance for TT has been best with window sizes between 9 and 11 for all metrics as shown in Figure 2. As expected, higher folds increase TT's overall performance especially with respect to metric *b* (Figure 3). This is due to the larger training set sizes of the LDA topic model.

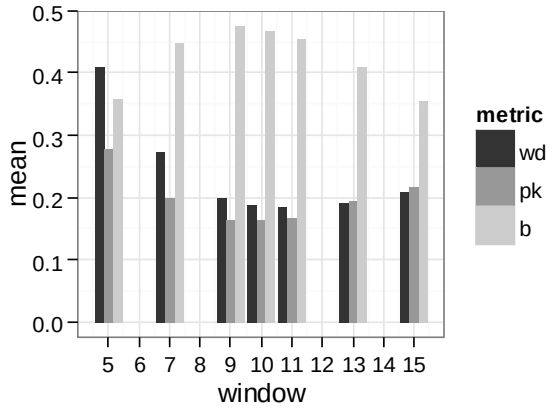


Figure 2: TT performance for different window sizes with 30-fold cross validation.

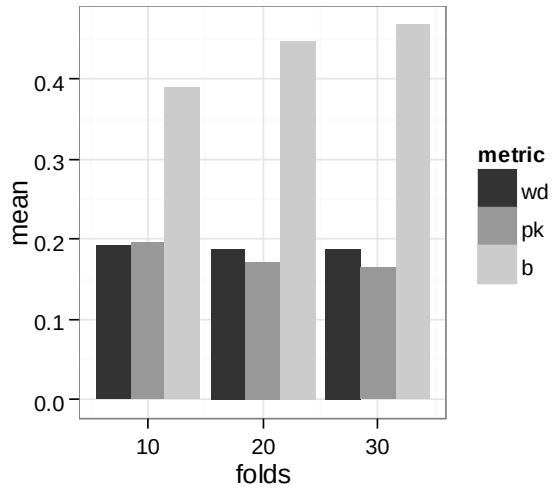


Figure 3: TT performance for different folds and window size set to 9.

In general smaller window sizes increase the number of predicted boundaries. The optimal window size is between 9 and 11 and we would expect the measures for 5 and 15 to be similar (Figure 2). This is only the case for metric *b*, the metrics *wd* and *pk* seem to penalize false positives more than false negatives. This would be a contradiction to the findings of Lamprier et al. (2007) since they actually found the opposite to be true. This behaviour is explained by the non-linear relation between the window parameter and number of predicted boundaries by TT as shown in Figure 4.

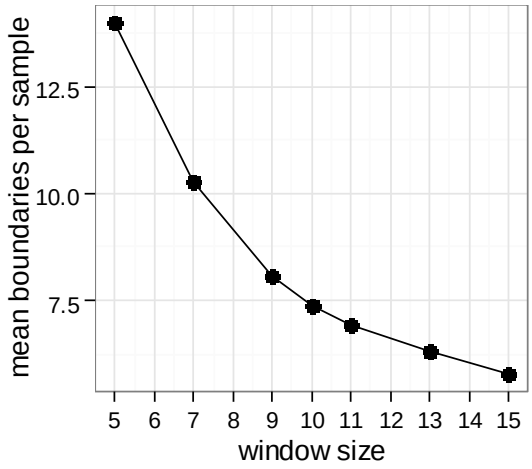
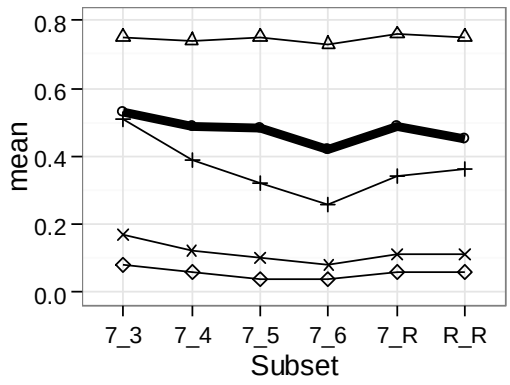


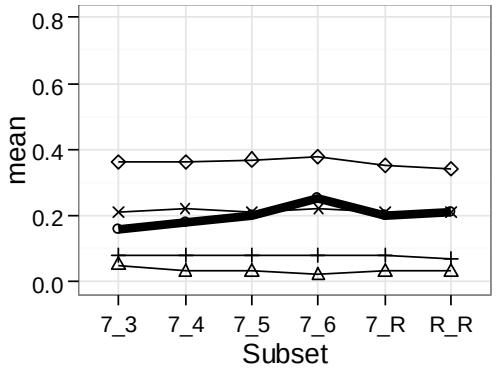
Figure 4: Mean number of predicted boundaries by TT for different window sizes and an LDA topic model trained with 30 folds.

Another important finding is the stability of TT's performance over different window sizes (from 9 to 11). This is important since a very sensitive behaviour would be very difficult to handle for course creators because they would have to estimate this parameter in advance.

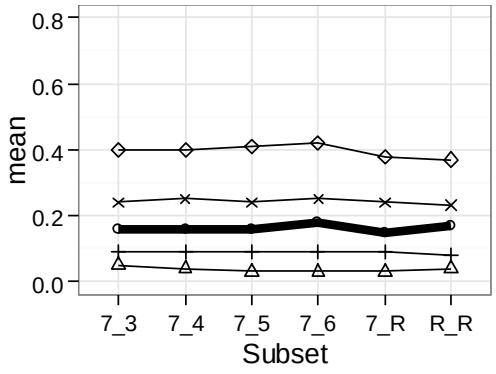
For the following detailed evaluation TT window size is set to 9 because of the best overall results with respect to metric *b* and 30-fold cross validation. The detailed performance with respect to metric *wd*, *pk* and *b* of TT compared to RS with different standard deviations σ is shown in Figure 5 i), ii) and iii).



i. TT measured with metric *b*.



ii. TT measured with metric *wd*.



iii. TT measured with metric *pk*.

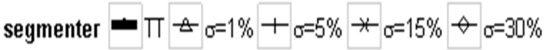


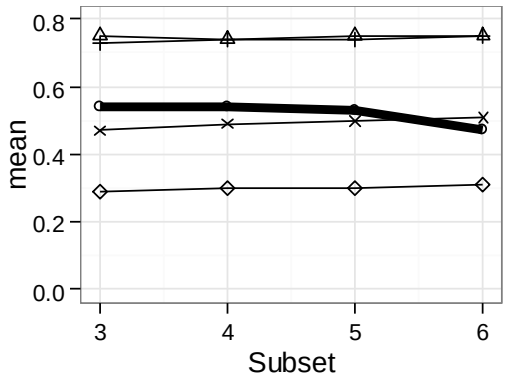
Figure 5: Performance of TT on the macro dataset.

First of all we want to point out that the graphs of RS for different values of σ are ordered as expected by all metrics. Lower percentages indicate better results. And with respect to metric *wd* and *pk* the performance for each σ is nearly constant over all subsets, which indicates that the metrics correctly consider the relative distance of a predicted boundary from the true boundary by using the mean number of sentences per topic. In metric *b* only the RS with σ=30%, 15% and 5% are constant. For σ=5% there is a strong decrease in performance for subsets with more sentences per topic.

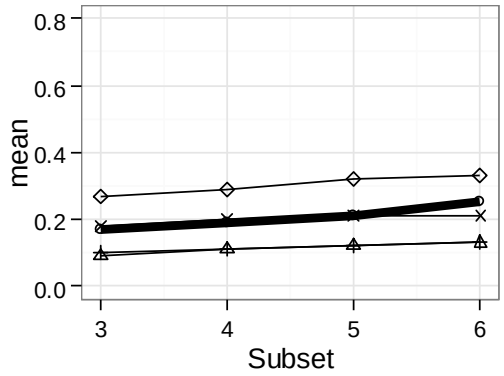
The overall performance of TT is between that of RS for σ=1% and σ=15%, except for subset *7_6* with respect to metric *wd*. With respect to metric *b* TT even predicts *very close* boundaries. In all metrics TT has the worst results on subset *7_6*, which has the largest number of sentences per topic (see Table 1). This is due to TT's window parameter which influences the number of predicted boundaries as shown in Figure 4.

4.2 Results for BayesSeg on the Micro Dataset

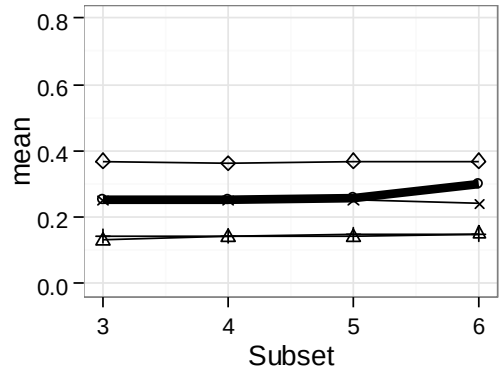
BS does not need any training or parameter fitting, since it is provided with the number of expected segments. We therefore used the default parameter settings.



i. BS measured with metric *b*.



ii. BS measured with metric *wd*.



iii. BS measured with metric *pk*.

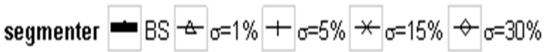


Figure 6: Performance of BS on the micro dataset.

As expected, the performance of RS is decreasing for higher values of σ in all metrics (Figure 6 i), ii), iii)). For metric *wd* and *pk* the increasing

number of topics leads to slightly increasing penalties for constant values of σ, which clearly indicates that the metrics do not treat all errors equally, as repeatedly pointed out. Metric *b* treats errors equally over increasing number of topics for RS. BS predicts with respect to all metrics *close* boundaries since it is better than RS with σ=15% except on subset *6* (Table 4). With an increasing number of topics BS is getting worse in all metrics.

Comparing the measures of metric *b* for macro and micro dataset it seems that it handles increasing numbers of topics better than increasing size of topics. On the micro dataset the results with respect to all metrics are far more similar than the once on the macro dataset, where the differences are very large. Since we are only interested in comparative measures of the performance of the segmenters and RS, which has shown to be a very useful approach to interpret segmentation results, we leave detailed explanations of the metrics behaviours itself to further research.

5 Conclusion

We demonstrated that text segmentation algorithms can be applied to the generation of e-learning courses. We use a web-didactic approach that is based on a flat two-level hierarchical structure. A new corpus has been compiled based on featured articles from the English Wikipedia that reflects this kind of course structure. On the broader macro level we applied the linear LDA-based text segmentation algorithm TopicTiling without providing the expected number of boundaries. The LDA topic model is usually trained with concatenated texts from the very same dataset TopicTiling is tested on. We showed that it is very difficult to ensure that the two sets are always truly disjoint. The reason is that concatenated texts normally always have identical parts. This problem is solved by applying a different training and testing method.

The more fine grained micro level was segmented using BayesSeg, a hierarchical algorithm which we provided with the expected number of boundaries.

We used three different evaluation metrics and presented a scalable random segmentation algorithm to establish upper and lower bounds for baseline comparison. The results, especially on the macro level, demonstrate that text segmentation algorithms have evolved enough to be used for the automatic generation of e-learning courses.

An interesting aspect of future research would be the application and creation of real e-learning content. Based on the textual segments, summarization and question generation algorithms as well as automatic replacement with appropriate pictures and videos instead of text could be used to finally evaluate an automatically generated e-learning course with real learners.

Regarding text segmentation in general, future research especially needs to address the difficult task of transparently and equally measuring the performance of segmentation algorithms. Our results, i.e., the ones from the random segmentation algorithm, indicate that there are still unsolved issues regarding the penalization of false positives and false negatives when the number of topics or sentences per topic is changed.

Reference

Beeferman, D., Berger, A. & Lafferty, J., 1999. Statistical Models for Text Segmentation. *Mach. Learn.*, #feb#, 34(1-3), pp. 177-210.

Bird, S., Klein, E. & Loper, E., 2009. *Natural Language Processing with Python.* s.l.:O'Reilly Media.

Capuano, N. et al., 2009. LIA: an intelligent advisor for e-learning. *Interactive Learning Environments*, 17(3), pp. 221-239.

Choi, F. Y. Y., 2000. *Advances in Domain Independent Linear Text Segmentation.* Stroudsburg, PA, USA, Association for Computational Linguistics, pp. 26-33.

Eisenstein, J. & Barzilay, R., 2008. *Bayesian Unsupervised Topic Segmentation.* Honolulu, Hawaii, Association for Computational Linguistics, pp. 334-343.

Fournier, C., 2013. *Evaluating Text Segmentation using Boundary Edit Distance.* Stroudsburg, PA, USA, Association for Computational Linguistics, p. To appear.

Fournier, C. & Inkpen, D., 2012. *Segmentation Similarity and Agreement.* Montreal, Canada, Association for Computational Linguistics, pp. 152-161.

Galley, M., McKeown, K., Fosler-Lussier, E. & Jing, H., 2003. *Discourse Segmentation of Multi-party Conversation.* Stroudsburg, PA, USA, Association for Computational Linguistics, pp. 562-569.

Griffiths, T. L. & Steyvers, M., 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences*, April, 101(Suppl. 1), pp. 5228-5235.

Hearst, M. A., 1997. TextTiling: Segmenting Text into Multi-paragraph Subtopic Passages. *Comput. Linguist.*, #mar#, 23(1), pp. 33-64.

Huang, X. et al., 2002. *Applying Machine Learning to Text Segmentation for Information Retrieval.* s.l.:s.n.

Janin, A. et al., 2003. *The ICSI Meeting Corpus.* s.l., s.n., pp. I-364--I-367 vol.1.

Kan, M.-Y., Klavans, J. L. & McKeown, K. R., 1998. *Linear Segmentation and Segment Significance.* s.l., s.n., pp. 197-205.

Kim, S., Medelyan, O., Kan, M.-Y. & Baldwin, T., 2013. Automatic keyphrase extraction from scientific articles. *Language Resources and Evaluation*, 47(3), pp. 723-742.

Lamprier, S., Amghar, T., Levrat, B. & Saubion, F., 2007. *On Evaluation Methodologies for Text Segmentation Algorithms.* s.l., s.n., pp. 19-26.

Lin, Y.-T., Cheng, S.-C., Yang, J.-T. & Huang, Y.-M., 2009. An Automatic Course Generation System for Organizing Existent Learning Objects Using Particle Swarm Optimization. In: M. Chang, et al. Hrsg. *Learning by Playing. Game-based Education System Design and Development.* s.l.:Springer Berlin Heidelberg, pp. 565-570.

Misra, H., Yvon, F., Jose, J. M. & Cappe, O., 2009. *Text Segmentation via Topic Modeling: An Analytical Study.* New York, NY, USA, ACM, pp. 1553-1556.

Pevzner, L. & Hearst, M. A., 2002. A Critique and Improvement of an Evaluation Metric for Text Segmentation. *Comput. Linguist.*, #mar#, 28(1), pp. 19-36.

Riedl, M. & Biemann, C., 2012. *TopicTiling: A Text Segmentation Algorithm Based on LDA.* Stroudsburg, PA, USA, Association for Computational Linguistics, pp. 37-42.

Strassel, S., Graff, D., Martey, N. & Cieri, C., 2000. *Quality Control in Large Annotation Projects Involving Multiple Judges: The Case of the TDT Corpora.* s.l., s.n.

Sun, Q., Li, R., Luo, D. & Wu, X., 2008. *Text Segmentation with LDA-based Fisher Kernel.* Stroudsburg, PA, USA, Association for Computational Linguistics, pp. 269-272.

Swertz, C. et al., 2013. *A Pedagogical Ontology as a Playground in Adaptive Elearning Environments..* s.l., GI, pp. 1955-1960.

Tan, X., Ullrich, C., Wang, Y. & Shen, R., 2010. *The Design and Application of an Automatic Course Generation System for Large-Scale Education.* s.l., s.n., pp. 607-609.

Utiyama, M. & Isahara, H., 2001. *A Statistical Model for Domain-independent Text Segmentation.* Stroudsburg, PA, USA, Association for Computational Linguistics, pp. 499-506.

Yaari, Y., 1997. *Segmentation of Expository Texts by Hierarchical Agglomerative Clustering.* s.l.:s.n.