COMPARATIVE ANALYSIS OF C99 AND TOPICTILING TEXT SEGMENTATION ALGORITHMS

Sukhpreet Kaur, Kamaljeet Kaur Mangat

M.TECH (CSE) Student, Punjabi University Regional Centre of IT and Management, Mohali, Punjab, India,
sukhpreetkaur125@gmail.com

Assistant Professor (CSE), Punjabi University Regional Centre of IT and Management, Mohali, Punjab, India,
kamalmangat@yahoo.co.in

Abstract

In this paper, the work done includes the extraction of information from image datasets which contain natural text. The difficulty level of segmenting natural text from an image is too high and so precision is the most important factor to be kept in mind. To minimize the error rates, error filtration technique is provided, as filtration is adopted while doing image segmentation basically text segmentation present in images. Furthermore, a comparative analysis of two different text segmentation algorithms namely C99 and TopicTiling on image documents is presented. To assess how well each algorithm works, each was applied on different datasets and results were compared. The work done also proves the efficiency of TopicTiling over C99.

Index Terms: *Text Segmentation, text extraction, image documents,C99 and TopicTiling.*

--***--

1. INTRODUCTION

In computational linguistics (CL) and natural language processing (NLP), text segmentation is the task of splitting text into a set of segments which are coherent about a specific topic. The coherently segmented text enables effective querying, analysis and usage. There are two main approaches for Text segmentation namely, lexical cohesion based approaches and feature based approaches. Lexical cohesion based approaches depend on the tendency of topic units to hang together. In feature based approaches, features like cue phrases, full proper nouns and named entities are used to detect boundaries between topics [1].Segmentation is used as a pre-processing step by NLP systems that perform various tasks such as video and audio retrieval, question answering, subjectivity analysis, automatic summarization, etc. Apart from these tasks, there are applications where text has to be segmented from the image data. The image documents are segmented to extract natural text [2]. The algorithms from lexical cohesion based approach are studied for segmentation. The algorithms undertaken for this study are C99 and TopicTiling to segment text from images. C99 segments a document horizontally in blocks and segmentation is done sequentially. The basic unit for C99 is a block which is a group of words. In TopicTiling, a document is treated as a hierarchical distribution of topics and thus performs segmentation in a hierarchical manner. The basic unit in this algorithm is a topic which can be a word or a group of words. Each unit is assigned topic id, based upon which text segmentation is done. Extensive research has been done on English literatures. Basically C99(presented by choi) and Topic Tiling are extensively used to segment text present in long literature documents but very few have tried to segment text from document of images. Presented systems are based on measuring the lexical cohesion between textual units. This research paper is organized as follows: section 2 presents related work, section 3 presents an overview of the analysed systems; results and discussion are reported in section 4 and finally section 5 concludes the paper.

2. RELATED WORK

Lots of work has been done on text segmentation techniques implemented on different literature corpus. Hearst [3] introduced the first topic based text segmentation algorithm named as: TextTiling which 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. Galley [4] presented a TextTiling based algorithm known as LcSeg in comparison to TextTiling, it uses tf-idf term weights, which improves TS results. Choi [5] introduced an algorithm called C99 that uses a matrix-based ranking and a clustering approach in order to relate the most similar textual units. Similar to the previous introduced algorithms, C99 uses words. He again improves C99 related to work undertaken , modified C99 algorithm, in 2001 that uses the term-representation matrix in latent space of LSA in combination with a term frequency matrix to calculate the similarity between sentences. Utiyama and Isahara [6] introduced one of the first probabilistic approaches using Dynamic Programming (DP) called U00. DP is a paradigm that can be used to efficiently find paths of minimum cost in a graph. Text Segmentation algorithms using DP, represent each possible segment (e.g.

every sentence boundary) as an edge. Providing a cost function that penalizes common vocabulary across segment boundaries, DP can be applied to find the segments with minimal cost. Misra et al. and Sun et al. [7] presented two new DP approaches: here, topic modeling is used to alleviate the sparsity of word vectors. The algorithm of Sun et al. follows the approach described in Fragkou et al., but uses a combination of topic distributions and term frequencies. Misra et al. [7] extended the DP algorithm from Utiyama and Isahara (2001) using topic models. Instead of using the probability of word co-occurrences, they use the probability of co-occurring topics. Segments with many different topics have a low topic-probability, which is used as a cost function in their DP approach. (Misra et al., 2009) trained the topic model on a collection of the Reuters corpus and a subset of the Choi dataset, and tested on the remaining Choi dataset. The topics for this test set have to be generated for each possible segment using Bayesian inference methods, resulting in high computational cost.

Mulbregt et al. [8] introduced a further approach for text segmentation i.e. the usage of Hidden Markov Model (HMM). Blei and Moreno introduced an Aspect Hidden Markov Model (AHMM) which combines an aspect model (Hofmann) with a HMM. The limiting factor of both approaches is that a segment is assumed to have only one topic. This problem has been solved by Gruber et al. who extends LDA to consider the word and topic ordering using a Markov Chain. In contrast to LDA, not a word is assigned to a topic, but a sentence, so the segmentation can be performed sentence-wise. Yaari [9] proposed the first hierarchical algorithm using the cosine similarity and agglomerative clustering approaches. A hierarchical Bayesian algorithm based on LDA is introduced by Eisenstein.

The current paper also includes image processing techniques by which natural text gets extracted from images. Lots of work has been done on different domains of image processing. Ohya et al. [10] presented a method for the extraction and recognition of characters in real scene images. They work with grey-level images, and proposed binarising image using threshold of sub-blocks, then those thresholds interpolate whole image, but it generates lot of noise. Wu et al. [11] proposed a two-step method to segment an image into text regions, and a robust (but not very generic) way to binarize the extracted text regions. First, a texture segmentation algorithm takes advantage of certain characteristics of text to segment areas possibly containing text lines. The second phase focuses on the previously detected areas and constructs “chips” from strokes taking into account text characteristics. Messelodi and Modena [12] proposed a method which better addresses problems of different orientation angles of text lines in real scene images. The method proposed, does address skew problems, but it is only described and tested on images of book covers. Clark and Mirmehdi [13] go one step further, and suggest two approaches to the location and recovery of text in real scenes. The first method is focused on situations where text lies on the surface of an object and the edges of the object are rectangular and aligned with the text. This is the case for paper documents, posters, signs, stickers etc. Lopresti and Zhou [14] proposed two methods to locate text in images, as well as methods to recognize text. It was the main contribution to the specific problem of Web Image text extraction. And although they make a number of assumptions that do not always hold, their approaches can produce satisfactory results to a significant sub-group of Web Images, namely images stored as GIF files (8-bit palettized colour).

3. IMPLEMENTATION

Design of this research work is basically implemented into two phases – Text Segmentation from Images and Segmentation of texts based on lexical cohesion which is received from phase-I.

3.1 Phase I-Text segmentation from images

The first phase concerns with extracting natural text from images which is proposed and implemented, passing through stages shown in Figure 1.

Block Diagram:

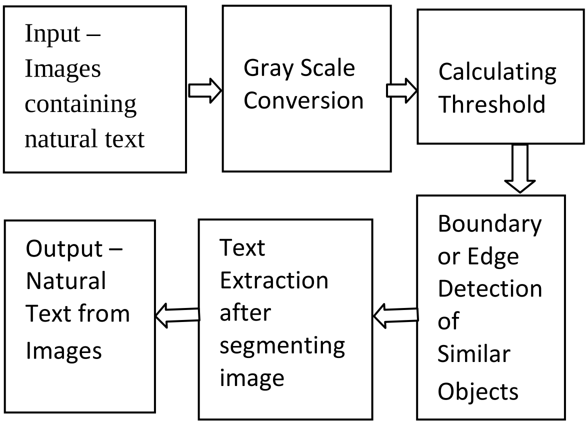
Input – Images containing natural text
Gray Scale Conversion
Calculating Threshold
Boundary or Edge Detection of Similar Objects
Text Extraction after segmenting image
Output – Natural Text from Images

Fig- 1: Extraction of natural text from image

Input – Images containing natural text can be black-white or colored image, but it is important that it must have natural text so that natural text gets segmented.

Gray Scale Conversion – It is basically a preprocessing stage which takes images as input which are usually contaminated with noise and are often in the Red-Green-Blue (RGB) color space, and convert the 3 color components (i.e. R, G and B) to the intensity component by [15]:

$Y=0.299R+0.587G+0.114B$

This is the actual Value component of the Hue-Saturation-Value (HSV) color space and the noise of the images is reduced using a weighted median filter that is applied on this component using the mask of:

$$\begin{matrix} 1 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 1 \end{matrix}$$

After this filtering step, a great part of noise will be removed while the edges in the image are still preserved.

Calculating Threshold - “Thresholding is an image point operation which produces a binary image from a gray scale image. A binary one is produced on the output image whenever a pixel value on the input image is above a specified minimum threshold level. A binary zero is produced otherwise[16].” Herein Multi - Thresholding is used which is defined as a point operator employing two or more thresholds. Pixel values which are in the interval between two successive threshold values are assigned an index associated with the interval”. Multi-thresholding is described mathematically as [16]:

$S(x, y) = k$, if $T_{k-1} \leq f(x, y) < T_k$, $k=0, 1, 2, \ldots, m$

where (x, y) is the x and y co-ordinate of a pixel, S(x, y), f(x, y) are the segmented and the characteristic feature functions of (x, y) respectively, T0, …, Tm are threshold values with T0 equal to the minimum and Tm the maximum and m is the number of distinct labels assigned to the segmented image [17].

Boundary Detection - Edge detection based segmentation techniques are based on the detection of discontinuities in the image. Discontinuities in a grey-level image are defined as abrupt changes in grey-level intensity values. Edges are therefore defined as the points where significant discontinuities occur. The importance of edge detection lies to the fact that most of the information of an image lies on the boundaries between different regions, and that biological visual systems seem to make use of edge detection rather than thresholding [18]. Ideally, edge detection should yield pixels lying only on the boundary between the regions one tries to segment. In practice, the identified pixels alone rarely define a correct boundary sufficiently because of noise, breaks in the boundary and other effects. More often than not, the edge detection stage is followed by an edge linking and boundary detection stage, intending to combine pixels into meaningful boundaries.

Edge detection methods can be classified as parallel and sequential [19].In parallel techniques which is herein used, the decision whether a pixel is an edge pixel or not is not dependent on the result of the operator on any previously examined pixels, whereas in sequential techniques it is. The edge detection operator in parallel techniques can therefore be applied simultaneously everywhere in the image.

Text Extraction after segmenting image - The procedure of extracting natural text from images done in this research paper is based on sobel edge detection of boundaries and then comparing it with probable values or it can be understood as aspect ratio of alphabets and digits as mentioned standardly in matlab. Simply for sobel detection matlab provides fx(imread) which reads binary image and detected boundaries are compared with probable values. Example for identifying ‘A’ probable value is 36. After comparing if probabilistic chance or aspect ratio suggests that the text is ‘A’ (for example) then output is declared as ‘A’.

	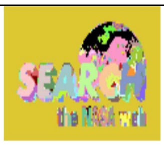	
Original	Segmented	Segmented Natural Text Characters

Fig-2: Example of text segmentation

Output – Natural text including all alphabets and digits which will act as an input for phase two where comparison between text segmentation algorithms will be done.

3.2 Phase II-Text Segmentation

The first phase concerns with extracting natural text from images which is proposed and implemented, passing through stages.

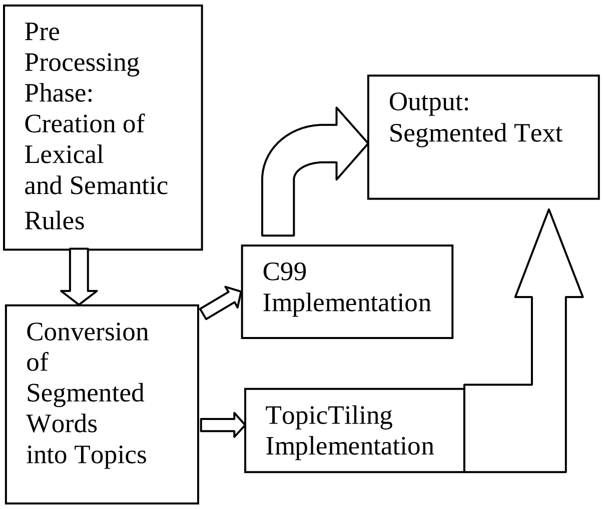
Pre Processing Phase: Creation of Lexical and Semantic Rules
Output: Segmented Text
C99 Implementation
Conversion of Segmented Words into Topics
TopicTiling Implementation

Fig- 3: Block diagram of text segmentation

Pre Processing Phase - TS algorithms which are evaluated in this research have basic block units as “words”, so there is a need to understand concept of breaking words from sentences. But before segmenting sentences into words, rules defining lexical cohesion must be made. In a sentence apart from words there are symbols like punctuation marks (“” , . ! & ; : ‘ ?) etc.and abbreviations. Words are also of different categories based on parts of speech like noun, pronoun, adjectives, verbs and conjunctions. A rule repository having all these semantic rules has been created prior which will further help in assigning topic ids to words and dividing sentences into topics. To segment the text line under consideration into words, a threshold has to be determined to distinguish pairs of connected components that belong to one word from these belonging to two separate words. For this purpose, the line width ‘w1’, the median stroke width ‘ws’ and the median distance between two vertical strokes ‘ds’ are considered. The line width ‘w1’ is defined as follows [20]:

$$w1=max(p(x,y)=black) – min(p(x,y) = black)$$

It represents the horizontal distance between the leftmost and rightmost black pixel in the considered line of text. The median stroke width ‘ws’ is defined according to the following formula:

$$Ws= median (h[y])$$

Using the horizontal projection histogram h[y] it represents the median of the number of black pixels, taken over all rows of the image of the considered text line. Finally the threshold t_{seg} for line segmentation is given by the following formula:

$$t_{seg} = alpha * (w1 – ws) /t$$

where, alpha is scaling constant that needs to be experimentally determined. By deleting connections in the minimal spanning tree which are longer than the threshold t_{seg} the text line is segmented into single words. For each text line, the threshold t_{seg} is computed separately.

From Words to Topics – This phase instead of using words directly as features to characterize textual units, uses their topic IDs as assigned by Bayesian inference. LDA inference assigns a topic ID to each word in the text document in each inference iteration step, based on lexical rules created in preprocessing phase.

Implementation of C99 - The topic based version of the C99 algorithm [21], called C99LDA, divides the input text into minimal units on sentence boundaries. A similarity matrix $S_{m \times m}$ is computed, where m denotes the number of units (sentences). Every element s_{ij} is calculated using the cosine similarity [22] between unit i and j. For these calculations, each unit i is represented as a T-dimensional vector, where T denotes the number of topics selected for the topic model. Each element t_k of this vector contains the number of times topic ID k occurs in unit i. Next, a rank matrix R is computed to improve the contrast of S: Each element r_{ij} contains the number of neighbors of s_{ij} that have lower similarity scores then s_{ij} itself. This step increases the contrast between regions in comparison to matrix S. In a final step, a top-down hierarchical clustering algorithm is performed to split the document into m segments. This algorithm starts with the whole document considered as one segment and splits off segments until the stop criteria are met, e.g. the number of segments or a similarity threshold. At this, the ranking matrix is split at indices i, j that maximize the inside density function D [20].

$$D= \sum_{k=1}^{m} \textit{sum of ranks /area within segment k}$$

TopicTiling Implementation - This section introduces Text Segmentation algorithm called TopicTiling which is based on TextTiling, but conceptually simpler. TopicTiling assumes a sentence s_i as the smallest basic unit. Between each position p between two adjacent sentences, a coherence score c_p is calculated. 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 respective block. The coherence score is calculated by cosine similarity for each adjacent “topic vector”. Values 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 dp is calculated for each minimum [3]. In comparison to TopicTiling, TextTiling calculates the depth score for each position and then 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 the 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. This threshold predicts a segmentation 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.

4. RESULTS AND DISCUSSION

4.1 TopicTiling Results on different datasets (images)

Table 1 shows the outcomes of 5 experiments performed over 5 different datasets and they are evaluated on three performance parameters: Precision, Recall and WindowDiff.

Table -1: Results of TopicTiling Algorithm

Precision	**Recall**	**WindowDiff**
92.4242	13.2461	0.21639
96.2963	17.4623	0.20769
94.3662	12.7276	0.21194
93.75	11.0938	0.21333
95.1807	10.9856	0.21013

4.2 C99 Results on different datasets (images)

Table 2 shows the outcomes of 5 experiments performed over 5 different datasets and they are evaluated on three performance parameters: Precision, Recall and WindowDiff.

Table -2: Results of C99 Algorithm

Precision	Recall	WindowDiff
72.549	11.4802	0.27568
88.0952	19.7846	0.22703
80.3571	12.3852	0.24889
67.6923	7.1834	0.29545
79.4118	9.6194	0.25185

4.3 Comparison of TopicTiling and C99 algorithms

Table 3 shows the comparison of the two algorithms TopicTiling and C99 based on the results obtained and shown in Table 1 and Table 2.

Table -3: Comparison of TopicTiling and C99 Algorithms

Parameters	TopicTiling	C99
Average Precision	94.397	77.62
Average Recall	12.92	12.092
Average WindowDiff	0.21	0.259

The experiments show that the average precision of TopicTiling algorithm is 94.397 as compared to that in C99 which is 77.62. High precision value in TopicTiling clearly shows improvement in text segmentation when segmentation is done topic based. There is one more trend need to be observed that is, precision is directly proportional to Recall and inversely proportional to WindowDiff. More the precision more recall and less is the window diff. It means less is the value for WindowDiff , better is the algorithm. So in this case TopicTiling algorithm performs better since the value for precision in case of TopicTiling is 94.397% which is more than that in case of C99 with 77.62% as precision. The value for recall in TopicTiling is 12.92 which again is more than the recall value 12.092 in case of C99. The value for WindowDiff in case of TopicTiling is 0.21 and is marginally less than in case of C99 with 0.259 as value for WindowDiff. Although the difference in values for Recall and WindowDiff is marginal, still they contribute in proving TopicTiling as a better algorithm than C99. As C99 does not assign topic ids and works on matrix rank calculation system, so precision in extracting text from sentence drops down marginally whereas TopicTiling uses topic ids to identify topics and calculate coherence factor for each sentence and so the precision in extracting text from sentence rises up marginally.

CONCLUSIONS

The Research was started with the objective of comparing and evaluating two text segmentation algorithms: C99 and TopicTiling and has gradually reached the conclusive phase. Both the text segmentation algorithms were performed and implemented on image datasets which contains text. Till now this area was scarce. The motive of this research, to find a better solution to segment text which minimizes computational cost and windowDiff and improve precision and recall and thus performance, has been accomplished. The work performed and implemented showed that replacing words in documents by topic IDs, leads to better results in the Text Segmentation task. This technique is applied in the TopicTiling and C99 algorithms. In comparison to other TS algorithms using topic models the runtime of TopicTiling is linear in the number of sentences. During implementation of TopicTiling, repeating the inference several times and using the most frequently assigned topic IDs in the last iteration not only reduces the variance, but also improves overall results. The important part of implementation is that performance is equal and at the same time minimize the computational cost. This method is not only applicable to Text Segmentation, but in all applications where performance crucially depends on stable topic ID assignments per token as this research shows with datasets having images. The higher precision and less windowDiff justifies the objective of research but in future there is still scope of improvement. More granual testing in different scenarios is needed and filling pitfalls of C99 or TopicTiling will always be welcomed. The concern is to devise a method to detect the optimal setting for the window parameter w automatically. Moreover, research can be done to extend the usage of the algorithms undertaken to more realistic corpora. Equipped with a highly reliable segmentation mechanism, application of text segmentation as a writing aid to assist authors with feasible segmentation boundaries and work in future would also assist in segmenting and classifying web documents with precision as getting consistent and relevant information is need of hour. This could be applied in an interactive manner by giving feedback about the coherence during the writing process.

REFERENCES

[1] Michael A. El-Shayeb, Samhaa R. El-Beltagy and Ahmed Rafea “Comparative Analysis of Different text Segmentation Algorithms on Arabic News Stories,” in proceedings of *Information Reuse and Integration-IRI*, pp.441-446, 2007

[2] Shohreh Kasaei, Roshanak Farhoodi, “Text Segmentation from Images with Textured and Colored

Background”, *13th Iranian conference on Electirc Engineering*, 1384.

[3] Hearst, M. A., “TextTiling: Segmenting Text into Multiparagraph Subtopic Passages”, *Computational Linguistics*, 1997.

[4] Galley, M., McKeown, K., Fosler-Lussier, E., Jing, H.Discourse, “Segmentation of multi-party conversation.” In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, volume 1, pages 562–569, Sapporo, Japan.

[5] Choi, F. Y. Y., “Advances in domain independent linear text segmentation”, *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference, Seattle*, WA, USA, 2000.

[6] Utiyama, M. Isahara H., “A statistical model for domain-independent text segmentation” *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics, Toulouse, France*, 2001.

[7] Misra H., Yvon F., Jose, J. M., Cappe, O., “Text Segmentation via Topic Modeling: An Analytical Study”, *Proceedings of the 18th ACM Conference on Information and Knowledge Management, Hong Kong*, 2011.

[8] Mulbregt, P. v., Carp, I., Gillick, L., Lowe, S., Yamron, J., “Text segmentation and topic tracking on broadcast news via a hidden markov model approach.” In *Proceedings of 5th International Conference on Spoken Language Processing*, Sydney, Australia.

[9] Yaari, Y.,” Segmentation of expository texts by hierarchical agglomerative clustering.” In *Proceedings of the Conference on Recent Advances in Natural Language Processing*, Tzigov Chark, Bulgaria.

[10] J. Ohya, A. Shio, S. Akamatsu, "Recognizing Characters in Scene Images", *Proceedings of IEEE Transactions on Pattern Analysis and Machine Intelligence.*

[11] V. Wu, R. Manmatha, E. M. Riseman, "Finding Text in Images", *Proceedings of 2nd ACM International Conference on Digital Libraries*, *Philadephia.*

[12] S. Messelodi, C. M. Modena, "Automatic Identification and Skew Estimation of Text Lines in Real Scene Images," *Pattern Recognition*, vol. 32, pp. 791-810, 1999.

[13] P. Clark, M. Mirmehdi, "Recognising Text in Real Scenes", *Proceedings of International Journal on Document Analysis and Recognition.*

[14] D. Lopresti, J. Zhou, "Document Analysis and the World Wide Web," Proc. of the Workshop on Document Analysis Systems, Marven, Pennsylvania, October 1996, pp. 417-424.

[15] Shohreh Kasaei, Roshanak Farhoodi, “Text Segmentation from Images with Textured and Colored Background”, *13th Iranian conference on Electirc Engineering*, 1384.

[16] R. M. Haralick, "Glossary of Computer Vision Terms," *Pattern Recognition*, Vol. 24, pp.69-93, 1991.

[17] K. S. Fu, J. K. Mui, "A Survey on Image Segmentation", *Pattern Recognition*,
Vol. 13, Issue 1, pp. 3-16.

[18] A. Rosenfeld, A. C. Kak, “Digital Picture Processing”, *Proceedings of New York*
Academic Press, 1969.

[19] L. S. Davis, "A Survey of Edge Detection Techniques", *Computer Graphics*
and Image Processing, Vol. 4, pp. 248-270, 1975.

[20] Riedl, Martin, and Chris Biemann. "Text Segmentation with Topic Models." *Herausgegeben von*: 47.