Text Segmentation With Topic Modeling And Entity Coherence

Adebayo Kolawole John, Luigi Di Caro, and Guido Boella

Dipartimento di Informatica, Universita Di Torino
Corso Svizzera 185, Torino, 10149, Italy
{kolawolejohn.adebayo}@unibo.it,
{dicaro,guido.boella}@di.unito.it,

Abstract. This paper describes a system which uses entity and topic coherence for improved Text Segmentation (TS) accuracy. First, Linear Dirichlet Allocation (LDA) algorithm was used to obtain topics for sentences in the document. We then performed entity mapping across a window in order to discover the transition of entities within sentences. We used the information obtained to support our LDA-based boundary detection for proper boundary adjustment. We report the significance of the entity coherence approach as well as the superiority of our algorithm over existing works.

Keywords: Text Segmentation, Entity coherence, Linear Dirichlet Allocation, Topic Modeling

1 Introduction

The goal of Text Segmentation (TS) is to identify boundaries of topic shift in a document. Discourse structure studies have shown that a document is usually a mixture of topics and sub-topics. A shift in topics could be noticed with changes in patterns of vocabulary usage [14]. The process of dividing text into portions of different topical themes is called Text Segmentation[16]. The text units (*sentences* or *paragraphs*) making up a segment have to be coherent, i.e., exhibiting strong grammatical, lexical and semantic cohesion [18]. Applications of TS includes Information Retrieval (IR), passage retrieval and document summarization [1].

Our approach is an unsupervised method which also incorporates the use of topics obtained from LDA topic modeling of some documents. Furthermore, we incorporate entity coherence [2], that allows the introduction of some heuristic rules for boundary decision. The remaining parts of the paper describes the proposed system. In section 2, we describe the general text segmentation task and related works. Section 3 details the proposed system followed by evaluation and results on choi's TS dataset.

2 Background and Related Works

A document is a mixture of topics spread across its constituent words, sentences and paragraphs. The dimension of shift in topics is thus a function of the semantic bond and relationships within these units. Observingly, this bond tends to be higher among units with common topics. This notion is what is termed *cohesion* or *coherence* within a document. Cohesion is a function of grammatical factors, e.g., co-reference and sentential connectives as well as lexical factors like collocation [18]. Coherence is higher within units that share several topics. The goal of TS is to identify points of weak or no coherence in a text.

Text Segmentation could be Linear or Hierarchical. Unlike hierarchical Text Segmentation [12] which is more fine-grained, Linear TS algorithms [8, 16, 3] observes sequence of topic shifts without considering the sub-topic structures within segments. Past works have relied on the use of similarity in vocabulary usage in sentences in order to detect potential topic shift [16, 8]. This idea, otherwise known as *lexical cohesion* could be tricky as it suffers from *lexical ambiguity*. This is because there are usually more than one words available to express an idea, i.e., *synonyms* while some words have multiple meanings, i.e., *polysemy*. The use of topics has recently been proposed [9, 28, 11, 10], inspired by distributional semantics based approaches such as Latent Sementic Analysis (LSA) [19, 9] and LDA topic models[28, 22]. Previous works on Text Segmentation basically adopt two approaches, e.g., lexical cohesion and discourse based techniques [9]. In the former, lexical relationships that exist between contiguous text units are used as a measure of coherence. These lexical relationships include vocabulary overlap which could be identified by word stem repetition, context vectors, entity repetition, word frequency model and word similarity [15, 18, 3, 26, 29]. High vocabulary intersection between two compared units is taken to mean high coherence and vice versa. The TextTiling algorithm [16] excels in this category. It assigns a score to each topic boundary candidate within *k* chosen window. Topic boundaries are placed at the locations of valleys in this measure, and are then adjusted to coincide with known paragraph boundaries. The authors in [9] builds on this ideas with the introduction of a similarity matrix neighborhood ranking, where the rank of an element corresponds to the number of neighours with lower values.

The discourse-based techniques rely on the use of cue phrases and Prosodic features, e.g., pause duration that are most probable to occur close to a segment boundary. These features are combined using a machine learning model [3, 24, 26]. This approach however is domain independent and can only perform well if the system is evaluated on documents which uses the same cue words.

Recent works [11, 22, 28] employed topic modeling with LDA [4]. The idea is to induce the semantic relationship between words and to use frequency of topic assigned to words by LDA instead of the word itself to build sentence vector. This makes sense since a word could appear under different topics thus partially overcoming lexical ambiguity.

Similarly to these works, our implementation uses topics obtained with the LDA topic model. However, we introduced two heuristics (*lexical* and *semantic*) strictly for boundary adjustment. For instance, a position *m+1* after a sentence S$_m$ is a valid boundary only if sentences within the region S$_{m-k}$ and S$_{m+k}$ have no common entities, where *k* is chosen window. Also, coherent sentences tend to have similar semantics. This is the main idea in TextTiling and Choi's work [15, 8] with the exception that they rely on term frequency to build sentence vector used for similarity calculation. Since this approach suffers from lexical ambiguity, e.g. the word *dog* appearing in one sentence followed by *puppy* in another are not deemed to be similar, we incorporate a semantic-net based similarity using WordNet. This typically overcomes the *synonymy* problem for a more efficient similarity calculation. The two heuristics were combined in a way to help in boundary decision making with topics-based sentence similarity. The experiment conducted on Choi's text segmentation evaluation dataset has shown the competitiveness of our approach.

3 Approach Description

Given an input document **W**, our algorithm divides the document into a set of minimal text units (s_1, s_2, s_3, ..., s_T), where T is the number of sentences in the document, each s_i can be viewed as a pseudo-document that contains a list of tokens v $\in$ V, where V is the set of vocabulary of *W*. In practice, the goal is to identify sets of contiguous s_i that are mono-thematic, each member of the set being a segment.

Following similar works [11, 22], we employ LDA topic modeling algorithm[5, 4] to obtain topics for each word. Topic models are a suite of unsupervised algorithm that uncovers the hidden thematic structures in document collection. Modeling documents based on topics provides a simple way to analyze large volumes of unlabelled text while exposing the hidden semantic relationships between them.

3.1 LDA Basics

LDA is a generative probabilistic model of a corpus with the intuition that a document is a random distribution over latent topics, where each topic is characterized by a distribution over words in the vocabulary. Say for instance that a document is perceived as a bag of words where the order does not matter, suppose that the fixed number of topics (say for instance n_T) is known. Considering there could be many of such documents in a bag, then each word in the bag is randomly assigned a topic *t* drawn from the Dirichlet distribution. This gives a topic representations of the documents and word distributions of all the topics. The goal is then to find the proportion of the words in document **W** that are currently assigned to each topic *t* as well as the proportion of assignments to topic *t* over all documents that come from this word *w*. In other words, a Dirichlet distribution of each word over each topic is obtained. The model has shown capability to capture semantic information from documents in a way similar to probabilistic latent semantic analysis [17] such that a low dimensionality representation of texts is produced in the semantic space while preserving their latent statistical features.

More formally, Given a document **w** of N words such that **w** = (w_1,w_2,w_3...w_N) and a corpus D of M documents denoted by D = ($\mathbf{w}_1$,$\mathbf{w}_2$,$\mathbf{w}_3$.....$\mathbf{w}_M$). For each of the words w_n in the document, a topic z_n is drawn from the topic distribution θ, and a word w_n is randomly chosen from P(w_n $|$ z_n, β) conditioned on z_n. Given α, a k-vector with components with $\alpha_i > 0$ and the Gamma function Γ(x). The probability density of the Dirichlet is given as

$$P(\Theta|\alpha) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\Pi_{i=1}^{k} \Gamma(\alpha_i)} \Theta_1^{\alpha_1 - 1} .... \Theta_k^{\alpha_k - 1} \tag{1}$$

Given the parameters α and β, the joint distribution of a topic mixture θ, a set of N topics **z**, and a set of N words **w** is thus given by

$$P(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta) = P(\theta|\alpha) \mathit{\Pi}_{n=1}^{N} P(z_n|\theta) P(w_n|z_n, \beta) \tag{2}$$

Integrating over θ and summing of z, the set of topic assignments, the distribution of a document can be obtained as below

$$P(\mathbf{w}|\alpha, \beta) = \int P(\theta|\alpha) \left(\mathit{\Pi}_{n=1}^{N} \sum_{z_n} P(z_n|\theta) P(w_n|z_n, \beta) \right) \mathrm{d}\theta \tag{3}$$

where P(z$_n$ | θ) is θ_i for the unique i such that $z_n^i = 1$ The probability of a corpus is obtained through the product of marginal probability above for each $\mathbf{w}_n$ in D as given below:

$$P(\mathbf{w}|\alpha, \beta) = \left\{ \mathit{\Pi}_{d=1}^{M} \int P(\theta_d|\alpha) \left(\mathit{\Pi}_{n=1}^{N_d} \sum_{z_d n} P(z_d n|\theta_d) P(w_d n|z_d n, \beta) \right) \mathrm{d}\theta_d \right\} \tag{4}$$

Training the LDA model on a corpus requires feeding the model with sets of tokens from the document. The model statistically estimate the topic distribution θ_d for each document as well as the word distribution in each topic. A model can also be used to predict topic classes for a previously unseen document. We trained the LDA algorithm with a mixture of the a subset of the wikipedia data, Brown corpus and Choi's dataset [8].

3.2 Topics-based Sentence Similarity

The authors in [27] used the most frequent topics assigned to a word after the gibbs inference to avoid instability associated with a generative algorithm like the LDA. Contrarily, for each sentence, we obtain the distribution of topics for each word[1], together with their probability score and simply choose the topic with highest probability for each word. For each sentence, this results into a bag of topics where order does not matter. We obtain a matrix G = $L \times T$ where $l \in L$ is a vector of length *k*, the chosen number of topics. Each vector l contains the frequency of each topic ID assigned by the LDA to the words in a sentence, where by topic ID, we denote the topic group or cluster that a word belongs, i.e., a number in the range [0, $T - 1$]. As an example, assuming the number of topics n = 10 and the bag of topics for a sentence is $\{0, 0, 5, 2, 3, 3, 7, 7, 1, 6, 5\}$, then the vector for such a sentence will be [2,1,1,2,0,2,1,2,0,0], each element representing the frequency of occurrence of topics 0 to 9. A generally accepted assumption is that sentences with similar topics have some semantic relationship. Furthermore, the LDA is able to unravel the latent relationship between words through its probabilistic clustering.

We introduce a lookahead window w_n which has a value of 3 by default. This is similar to the *k-block* of sentences employed in [28] but with different objective. The

[1] our system is being developed in the context of our bigger project Eunomos [7, 6]

previous works compares the vector of a sentence to the *k-block* of sentences on the left and the right of the sentence in order to get the similarity score [2] for that sentence. The process is then carried out for all sentences in the document in order to yield the measure of closeness of a sentence to its surrounding sentences. In our implementation, for each pass over the list of sentences, using the lookahead window, we sum up the vectors of sentences within the window and use it as a *reference* vector for sentences within that window. The intuition is that we can treat the set of sentences within a window as a *mini* document, summing up the vectors gives the overall meaning of the *mini* document. Thus, we can estimate the semantic distance between the *mini* document and each neighour sentence. Sentences with high topic correlation will have high similarity to the reference. Figure 1 shows the process of summing over vector for a sample document of 10 sentences. Once the reference values have been obtained, the next step is to obtain sentence similarity, otherwise called the coherence score. To do this, for each window, we use the cosine similarity between each sentence and the reference vectors. Repeating this over all sentences results into a time series, e.g., a one dimensional vector of similarity values over all the sentences.

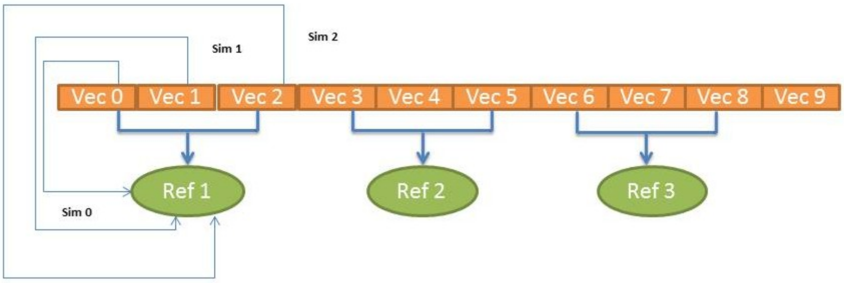

Fig. 1. Summing over window vector

3.3 Inter-Sentence Semantic Similarity

To further alleviate the language variability problem, we introduce another similarity vector. First, we perform parts of speech (POS) tagging [3] on the sentences in order to select the verbs, nouns and the adjectives. We call this the *POS profile* of each sentence. Here, we also rely on the use of the lookahead window. For instance, using the WordNet concept hierarchy, we calculate the similarity of the POS profile of a sentence with available sentences within a shifting window of 3. As an example, given the verbs, nouns and adjectives in a sentence S_1, instead of comparing these POS entries directly with those in sentence S_2 only, it is compared with those sentences that falls into the set $\{S_2, S_3, S_4\}$. To derive similarity from WordNet, we used both the path length between

[2] Otherwise called coherence score

[3] We used the Stanford POStagger. It is available at http://nlp.stanford.edu/software/tagger.shtml

each word as well as the depth function. Our similarity implementation is similar to the approach in [20] and produces a score within the range 0 and 1 for each compared POS filtered sentences. Similarly, we obtain a 1-D similarity vector with length equal to the number of sentences.

3.4 Entity-Based Coherence

An observation well established in grounded theories of coherence [21, 13] in discourse analysis is that entity distribution and transition signals coherence. The works in [2] is based on the centering theory, where the authors represents a document as a grid of entities in the document with their roles (subject, object, neither subject nor object and absence) specified as the actions of these entities. The rows of the grid correspond to sentences, while the columns correspond to discourse entities. We follow this ideas by observing the spread of entities across the sentences in the document to be segmented. Contrary to the grid-based entity ranking [2], our goal is to observe the entity overlaps that exist between sentences within a chosen shift window[4]. Succinctly, we only use the information about entity coherence for necessary boundary adjustment and not boundary detection to be specific. To achieve this, we use a grammar-based Regex parser to extract all the noun phrases in each sentence. To determine the overlap for a sentence S_i, we compute the ratio of its common noun-phrases to its right neighours within a specified window, e.g., $\{ S_{i+1}, S_{i+2}, S_{i+3}\}$. The entity overlap is obtained as follows:

$$\text{EOV} = \frac{|A \,\tilde{\cap}\, B^*|}{|A \cup B^*|} \tag{5}$$

Where A and B^* represent the set of entities in the sentence being considered and right neighors within a specified window, respectively. The intersection, $\tilde{\cap}$, allows partial matches since the entities are considered equivalent if there is an exact match or an entity is a substring of the other. Instead of using the overlap score, we record the last sentence from within the B^* that has shared entities with A if the overlap score actually exceeds a threshold. As an example, if a sentence S_1 is compared to $\{S_2, S_3, S_4\}$ with the entity overlap score between them exceeding the threshold, then, one by one, we check if it actually has an overlap with each of S_2, S_3 and S_4 independently. If say for instance, we discover that S_1 and S_4 do not have any common entities but it has with S_2 and S_3, then the index of sentence S_3[5] is used as its last sentence collocation. It becomes plain whether a sentence share entities with immediate neighbors in which case the assumption is that such a sentence is not likely to be a boundary. As an example, the text below shows how entity coherence may support boundary adjustment. The entities detected by our custom parser are in bold.

S_1: ***Cook*** *had discovered a* ***beef*** *in his possession a few days earlier and , when he could not show the* ***hide****, arrested him.*

S_2: *Thinking the evidence insufficient to get a conviction, he later released him.*

S_3: *Even while suffering the trip to his* ***home, Cook*** *swore to* ***Moore*** *and* ***Lane*** *that he would kill*

[4] Following our previous parameter w_n, we use a window of 3 sentences as default.

[5] We use index here to mean the unique ID of a sentence, e.g., sentence 1 will have index 0, sentence 2 will have index 1 etc..

the ***Indian****.*
S_4: *Three weeks later, following his recovery, armed with a* ***writ*** *issued by the* ***Catskill justice*** *on* ***affidavits*** *prepared by the* ***district attorney****,* ***Cook*** *and* ***Russell*** *rode to arrest* ***Martinez****.*
S_5: *Arriving at daybreak, they found* ***Julio*** *in his* ***corral*** *and demanded that he surrender.*
S_6: *Instead, he whirled and ran to his* ***house*** *for a* ***gun****, forcing them to kill him,* ***Cook*** *reported.*

In the example above, the entity *Cook* appears in S_1, S_3,S_4 and S_6. Considering S_1, we conclude that no boundary exist until S_4 since there is significant entity overlap with S_3 and S_4 when moving over the sentence window. Even though there appears to be no overlap with S_2 and S_1, it is safe to assume that S_2 is not a boundary since it falls within a coherent window, same goes for S_5 which falls within sentences S_3 and S_6. In our implementation, we create a vector whose elements holds the index of the last sentence it has an overlap with. In case of no overlap, the entry for a sentence is set at 0. Identifying the entity distribution in this way is useful for boundary adjustment for the suggested boundary from our topic based segmentation.

3.5 Boundary Detection and Segmentation

To obtain the sets of possible segmentation from the coherence score vectors, we obtained the local minima (valleys) and the local maxima (peaks). The valleys are the smallest values within a local range of the coherence scores vector. Since coherence scores are higher within sentences sharing many topics, we assume that these points of minimum values signals the points where least topic cohesion occurs, hence a segment boundary. The indices of the valleys [6] are collected in a vector as potential points of topic shift. We use the entries from our entity based coherence for necessary boundary adjustment. A mapping between the topic-based vector and the entity-coherence vector is created. For each sentence in a document, each column of the entity coherence vector references the index of the last sentence it has an overlap with. If there is a boundary after a sentence but there is an overlap reference to a sentence index higher than the boundary point then we *left-shift* the boundary as an adjustment task. Figure 2 shows the process of boundary adjustment over a sample sentence. The idea is based on centering theory [2], sentences with overlapping entities above a threshold have some level of coherence.

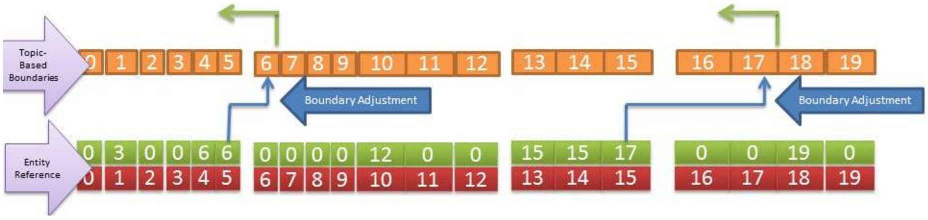

Fig. 2. Entity Coherence-Based Boundary Adjustment

[6] i.e., the vector index which corresponds to the index of each sentence in the local minima

4 Evaluations

For all evaluations, we used the Choi's dataset since it allows easy comparison with our baseline systems [8, 28, 16] In order to evaluate the accuracy of our system, we used the P_k error [3] and WindDiff [25] evaluation metrics which are commonly used. These two metrics measures the rate of error in segmentation with a lower value signifying better segmentation accuracy. Other common metrics are the IR based precision, recall and accuracy. However, these IR based metrics over-penalizes the *near miss* scenarios, e.g., when an actual segment is wrongfully partitioned into two different segments by an algorithm.

We trained the LDA model on the Brown corpus and a trimmed version of Wikipedia dump [7]. We used the Gensim version of the LDA algorithm. Gensim is a python library for an array of NLP tasks [8]. The number of Topics specified for training is 50 with 20 inference iterations.

We compared the result of our algorithm with the TopicTiling system [28], a TextTiling based system which solely rely on topics assignment to document from LDA. We also compared the result with TextTiling and Choi's system as reported by Rield and Bielmann [27]. For all the reported results from other systems, we did not reproduce the experiments, relying on the results reported in [27]. Tables 1 and 2 shows the results of our algorithm on Choi's Text Segmentation dataset using the P_k and WinDiff error metrics, respectively. Table 3 gives the comparison of our system against some state-of-the-art systems. Specifically, we selected TopicTiling [27] algorithm as it is the most similar to our work. Our intention is to show that our boundary-adjustment ideas really improves the performance of the system. The TextTiling and Choi's work have been severally outclassed by other systems [11, 23, 22] but were selected based on their

Table 1. P_k Error metrics on Choi's dataset

Window	3 - 5	6 - 8	9 - 11	3 - 11
1	1.76	2.90	4.0	2.64
3	0.89	1.18	0.49	0.67
5	1.30	1.53	3.80	1.80

Table 2. WinDiff Error metrics on Choi's dataset

Window	3 - 5	6 - 8	9 - 11	3 - 11
1	1.82	2.94	4.21	2.68
3	0.93	1.41	0.49	0.71
5	1.29	1.48	3.87	1.82

Table 3. Comparison of our systems's performance with selected state of the arts algorithm

Algorithm	3 - 5	6 - 8	9 - 11	3 - 11
TextTiling	44	43	48	46
Choi LSA	12	9	9	12
Topic Tiling	1.24	0.76	0.56	0.95
Our System	0.89	1.18	0.49	0.67

Table 4. P_k Error metrics on Choi's dataset without Boundary Adjustment

Window	3 - 5	6 - 8	9 - 11	3 - 11
1	1.92	3.30	4.1	2.98
3	1.19	2.23	0.82	0.91
5	1.70	2.36	3.89	2.20

[7] The wikipedia dump was downloaded on July 30, 2015. It is accessible at https://dumps.wikimedia.org/enwiki/.

[8] It is available at https://radimrehurek.com/gensim/.

popularity. The TopicTiling algorithm has also shown slight superiority over the latter algorithms. To show the importance of the boundary adjustment, we repeated the experiment without adjusting the boundary. Table 4 shows the effect of the boundary adjustment. Note the decrease in performance when boundary adjustment is not used.

5 Conclusion

We presented a TS approach that outperforms famous state-of-the-art systems on the Choi's TS dataset. Our approach combines the use of topics for segmentation with Entity Coherence-based heuristics for an improved performance. For the topic-based segmentation, we used the popular topic modeling algorithm, LDA. We described the approach of obtaining the coherence scores of the sentences. The reported results confirm the competitiveness of our approach.

Acknowledgments. Kolawole J. Adebayo has received funding from the Erasmus Mundus Joint International Doctoral (Ph.D.) programme in Law, Science and Technology. Luigi Di Caro and Guido Boella have received funding from the European Union's H2020 research and innovation programme under the grant agreement No 690974 for the project "MIREL: MIning and REasoning with Legal texts".

References

1. Regina Barzilay and Michael Elhadad. Using lexical chains for text summarization. *Advances in automatic text summarization*, pages 111–121, 1999.
2. Regina Barzilay and Mirella Lapata. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34, 2008.
3. Doug Beeferman, Adam Berger, and John Lafferty. Statistical models for text segmentation. *Machine learning*, 34(1-3):177–210, 1999.
4. David M Blei and John D Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120. ACM, 2006.
5. David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.
6. G. Boella, L. Di Caro, L. Humphreys, L. Robaldo, R. Rossi, and L. van der Torre. Eunomos, a legal document and knowledge management system for the web to provide relevant, reliable and up-to-date information on the law. *Artificial Intelligence and Law*, to appear, 2016.
7. Guido Boella, Luigi Di Caro, Alice Ruggeri, and Livio Robaldo. Learning from syntax generalizations for automatic semantic annotation. *The Journal of Intelligent Information Systems*, 43(2):231–246, 2014.
8. Freddy YY Choi. Advances in domain independent linear text segmentation. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 26–33. Association for Computational Linguistics, 2000.
9. Freddy YY Choi, Peter Wiemer-Hastings, and Johanna Moore. Latent semantic analysis for text segmentation. In *In Proceedings of EMNLP*. Citeseer, 2001.
10. Gaël Dias, Elsa Alves, and José Gabriel Pereira Lopes. Topic segmentation algorithms for text summarization and passage retrieval: An exhaustive evaluation. In *AAAI*, volume 7, pages 1334–1339, 2007.

11. Lan Du, John K Pate, and Mark Johnson. Topic segmentation in an ordering-based topic model. 2015.
12. Jacob Eisenstein. Hierarchical text segmentation from multi-scale lexical cohesion. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 353–361. Association for Computational Linguistics, 2009.
13. Barbara J Grosz, Scott Weinstein, and Aravind K Joshi. Centering: A framework for modeling the local coherence of discourse. *Computational linguistics*, 21(2):203–225, 1995.
14. Michael Alexander Kirkwood Halliday and Ruqaiya Hasan. *Cohesion in english*. Routledge, 2014.
15. Marti A Hearst. Texttiling: A quantitative approach to discourse segmentation. Technical report, Citeseer, 1993.
16. Marti A Hearst. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Computational linguistics*, 23(1):33–64, 1997.
17. Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999.
18. Stefan Kaufmann. Cohesion and collocation: Using context vectors in text segmentation. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 591–595. Association for Computational Linguistics, 1999.
19. Thomas K Landauer, Peter W Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284, 1998.
20. Yuhua Li, David McLean, Zuhair A Bandar, James D O'shea, and Keeley Crockett. Sentence similarity based on semantic nets and corpus statistics. *Knowledge and Data Engineering, IEEE Transactions on*, 18(8):1138–1150, 2006.
21. William C Mann and Sandra A Thompson. Rhetorical structure theory: Toward a functional theory of text organization. *Text-Interdisciplinary Journal for the Study of Discourse*, 8(3):243–281, 1988.
22. Hemant Misra, François Yvon, Olivier Cappé, and Joemon Jose. Text segmentation: A topic modeling perspective. *Information Processing & Management*, 47(4):528–544, 2011.
23. Hemant Misra, François Yvon, Joemon M Jose, and Olivier Cappe. Text segmentation via topic modeling: an analytical study. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1553–1556. ACM, 2009.
24. Rebecca J Passonneau and Diane J Litman. Discourse segmentation by human and automated means. *Computational Linguistics*, 23(1):103–139, 1997.
25. Lev Pevzner and Marti A Hearst. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, 28(1):19–36, 2002.
26. Jeffrey C Reynar. Statistical models for topic segmentation. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 357–364. Association for Computational Linguistics, 1999.
27. Martin Riedl and Chris Biemann. Text segmentation with topic models. *Journal for Language Technology and Computational Linguistics*, 27(1):47–69, 2012.
28. Martin Riedl and Chris Biemann. Topictiling: A text segmentation algorithm based on lda. In *Proceedings of ACL 2012 Student Research Workshop*, pages 37–42. Association for Computational Linguistics, 2012.
29. Masao Utiyama and Hitoshi Isahara. A statistical model for domain-independent text segmentation. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 499–506. Association for Computational Linguistics, 2001.