Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 3375–3381
Marseille, 20-25 June 2022

Detecting Multiple Transitions in Literary Texts

Nuette Heyns, Menno van Zaanen
North-West University, Southern African Centre for Digital Language Resources
Potchefstroom 2520,
South Africa
nuette.heyns@gmail.com, menno.vanzaanen@nwu.ac.za

Abstract

Identifying the high level structure of texts provides important information when performing distant reading analysis. The structure of texts is not necessarily linear, as transitions, such as changes in the scenery or flashbacks, can be present. As a first step in identifying this structure, we aim to identify transitions in texts. Previous work (Heyns and van Zaanen, 2021) proposed a system that can successfully identify one transition in literary texts. The text is split in snippets and LDA is applied, resulting in a sequence of topics. A transition is introduced at the point that separates the topics (before and after the point) best. In this article, we extend the existing system such that it can detect multiple transitions. Additionally, we introduce a new system that inherently handles multiple transitions in texts. The new system also relies on LDA information, but is more robust than the previous system. We apply these systems to texts with known transitions (as they are constructed by concatenating text snippets stemming from different source texts) and evaluation both systems on texts with one transition and texts with two transitions. As both systems rely on LDA to identify transitions between snippets, we also show the impact of varying the number of LDA topics on the results as well. The new system consistently outperforms the previous system, not only on texts with multiple transitions, but also on single boundary texts.

Keywords: topic modelling, LDA, transition identification

1. Introduction

The digitization and annotation of texts is one of the main focus area in the field of digital humanities. The large number of digitization projects boost the amount of digitally available texts and this, in turn, allows humanities scholars to access and analyze texts that previously were difficult to access or process.

With the huge amounts of texts available, one may try to identify patterns that cross (large amounts of) texts. However, given the size of the number of texts, close reading approaches (consisting of in depth literary analyses of texts) are practically infeasible. Instead, distant reading approaches (Moretti, 2013), which rely on the automatic analysis of a text, can be considered instead. Distant reading can identify global properties of one or more texts, in contrast to close reading, which focuses on more fine-grained properties (Franzini et al., 2015). One advantage of distant reading is that the computer can perform large scale objective analyses of texts, as opposed to the time consuming, subjective analyses of close reading.[1]

One task that is particularly useful in a distant reading setting is that of identifying the high level structure of a text. This requires an overview of the entire text (which might be very long, making it difficult for humans to retain a full overview). Genette et al. (1980) identified different levels within literary texts. On one level, the sequence of events is viewed in relation to the ordering of the narration. This text structure is not necessarily linear, but transitions, such as changes in the scenery and flashbacks, may be present.

In previous work, Heyns and van Zaanen (2021) proposed a system that identifies a boundary describing a high level transition in a literary text. This method assumes that transitions occur when there is a shift in topics. By identifying the topics that occur in the text, they showed that it is possible to identify a transition in a text by finding the position in the text that shows a relatively large contrast in topics between those that occur before and those that occur after the position in the text. This position can then be expected to be a high level textual transition.

Practically, the system subdivides the literary text into smaller snippets. LDA (Blei et al., 2003) is then used to identify the main topic for each snippet. Next, each boundary between the snippets is considered a potential textual transition. The boundary that shows the largest difference between LDA topics occurring before and those after the specific boundary, is assigned to be the high level textual transition. This occurs when there is a minimum of overlap between the LDA topics before and after the boundary. The system is evaluated by applying it to a text which consists of the concatenation of two texts. As such, a real transition is known. The random mean squared error (RMSE) is used to measure how well the proposed transition matches the real transition.

The system works relatively well, with low RMSE (even working perfectly in some cases) which shows its practical feasibility. However, the system also shows some shortcomings. First, it is designed to only identify one transition in a text. In real texts, one may ex-

[1] A discussion on the advantages and disadvantages of close and distant reading approaches is beyond the scope of this article.

pect more transitions, limiting the practical applicability of the system. Second, the experiments were only performed on one text (that was created by the concatenation of two texts). This limited evaluation setting may mean that the results cannot be generalized to other texts.

In this article, we aim to extend the previous work in several ways. First, we extend the system proposed by Heyns and van Zaanen (2021), allowing it to identify multiple transitions in a text.

Second, we introduce a new method to identify transitions in a text. This new method is designed from scratch to be able to identify multiple transitions in a text and as such, it will be closer to a real world scenario where multiple transitions may occur in a text.

Third, we follow the same evaluation methodology as Heyns and van Zaanen (2021), but extend the evaluation to multiple transitions in the text. In this situation, the correct transitions are known (like the previous evaluation approach which only has one transition), but the task is much more complex.

Finally, Heyns and van Zaanen (2021) performed experiments on snippets from only one pair of texts. Some results may be attributed to the specific texts that were used. Here, we run experiments on multiple texts. This allows us to experiment with text pairs that are semantically more closely related to each other, further complicating the identification of text transitions.

2. Background

To our knowledge, only a limited amount of research on the automatic detection of transitions in language has been conducted.

Previous research by Grosz and Sidner (1986) and Hirschberg and Litman (1993) focused on specific properties in the linguistic signal. This is illustrated by the identification of transitions in the area of spoken dialog. In particular, characteristic features of transitions that can be found in the language signal are used, e.g., phrases used to signal a topic change, significant pauses in the speech signal, changes in intonation, or domain specific cue phrases. A similar approach has been applied to textual data where the text structure, e.g., headings, chapter divisions, paragraphs, etc., is used to detect transitions.

These techniques that rely on specific aspects of the linguistic signal, however, are specialized, are difficult to use for longer, unstructured texts, and can be expected to lead to low quality results in situations where the structure of the text does not directly follow the layout (e.g., when a topic crosses multiple paragraphs). In particular, in literary text, where, for instance, we may be interested in the structure that shows a (non-linear) story line, the relationship between properties such as paragraph transitions and story line transitions becomes unclear.

Reynar (1994) proposed a method based on lexical cohesion and a graphical technique called Dot-plotting to identify transitions in text. Dot plotting was first proposed by Church (1993) to align bilingual corpora. Reynar (1994) adjusted the Dot-plotting method so that it enumerates the lexical items in a text. If a particular word appears at positions x and y in a text, the four points corresponding to the Cartesian product of the set would be plotted, i.e., the area indicated by (x, x), (x, y), (y, x), and (y, y) is plotted. The repetition of lexical items occurs more frequently within regions of a text discussing the same topic. The density of the areas outside a region is calculated and the boundary is identified at the lowest density point. Choi et al. (2001) have extended and improved upon this method by introducing LDA as a classification method in the popular c99 algorithm.

Aurnhammer et al. (2019) describe results from a study that compares results from a close reading analysis with those of a distant reading analysis. The close reading approach manually annotated Reddit posts and these annotations are compared against a distant reading approach that relies on the identification of topics using LDA. The results showed that there is a relationship between manually annotated topics and LDA topics in a text, although some types of annotations cannot easily be identified automatically.

To evaluate the performance of a system that identifies transitions in texts, the system has to be applied to a text in which the transitions are known. The output of the system can then be compared against the true transitions in the text. However, (manually) identifying text transitions is a subjective task. Texts often have a hierarchical structure (Grosz and Sidner, 1986), where text parts can consist of multiple (sub) text parts as apposed to a simple linear structure (Skorochod'ko, 1971). This means that textual transitions can take place on multiple levels in the hierarchy. Two separate studies by Galley et al. (2003) and Gruenstein et al. (2008) found that human annotators did not always agree on the transition positions in texts they were asked to annotate.

3. Methodology

In this article, we will introduce, evaluate, and compare two systems. These systems are applied to texts with known transitions. Information on the data that is used to evaluate these systems is provided first, followed by a description of the systems and their experimental settings.

3.1. Data sets

To evaluate the text transition identification systems, we require texts in which the transitions are known. Previous work already indicated that manual annotation may prove difficult. As such, we create evaluation data by concatenating text snippets from different texts. This way, we have control over the position of the transitions.

The creation of a text used for evaluation is done by taking snippets from two texts (say texts *A* and *B*).

These snippets are then concatenated into a new text in such a way that the transition between snippets from text *A* and *B* is still known.

Our first data set (called *ST* for single transition) contains concatenated texts with only one boundary (which means that the text is created by concatenating snippets from *A* and *B* such that it follows the sequence *AB*). The second data set (which is called *MT* for multiple transitions) contains texts with multiple boundaries and these are created by concatenating snippets in the format *ABA*.

In contrast to the evaluation of Heyns and van Zaanen (2021), which only used one pair of texts, here we use ten pairs of texts (extracted from a total of twenty books). In particular, we used the following text pairs:

1. Utilitarianism by John Mill, and Hide and Seek by Wilkie Collins,
2. Crime and Punishment by Fyodor Dostoevsky, and Great Expectations by Charles Dickens,
3. Eureka by Edgar Allan Poe, and A study in scarlet by Sir Arthur Conan Doyle,
4. And Then There Were None by Agatha Christie, and In the woods by Tana French.
5. The Count of Monte Cristo by Alexandre Dumas, and Our Mutual Friend by Charles Dickens
6. Middlemarch by George Eliot, and Jude the Obscure by Thomas Hardy
7. Through the looking glass by Lewis Carroll, and Anne of green Gables by Lucy Montgomery
8. Jane Eyre by Charlotte Brontë and Little Dorrit by Charles Dickens
9. The Moonstone by Wilkie Collins, and Frankenstein by Mary Shelley
10. Barchester Towers by Anthony Trollope, and Cranford by Elizabeth Gaskell

For the ST data set, we selected 25 snippets of 500 words each from both texts *A* and *B*, which we concatenated. We did this for each of the ten pairs of texts. For the MT data set (which contains *ABA* created texts), we used twelve snippets of text *A* for the first part and thirteen snippets from text *A* for the last part. The *B* part still consisted of 25 snippets from text *B*. This means that all concatenated texts in both data sets consist of 50 snippets each, where for the ST texts the true transition occurs after 25 snippets and for the MT texts, the transitions occur after snippet twelve and after snippet 37. All concatenated texts contain 25,000 words in total.

Before creating the snippets, straightforward preprocessing is applied to the texts. Stop words are removed using NLTK[2] as stop words occur frequently in the text and do not aid in assigning LDA topics to the snippets. The texts are also lower cased, lemmatized, and the punctuation is removed using spaCy[3].

3.2. Systems

In previous work, Heyns and van Zaanen (2021) described a transition identification system that can identify a single transition in a text. The system accepts an input text consisting of a sequence of snippets (S). Using LDA a topic is assigned to each snippet, resulting in a sequence of LDA topics: $LDA(S) = \langle LDA(s_1), LDA(s_2), \ldots, LDA(s_n)\rangle$. Each position between two snippets, (s_x, s_{x+1}) in the sequence (with $x = 1 \ldots n - 1$) is considered as a potential transition. At each potential transition, the entropy at that position is computed. The potential transition with the minimum entropy indicates the best potential transition.[4] If there is multiple positions that have the same minimum entropy, the system selects one of the potential transitions at random.

To allow the system to identify multiple boundaries, we extended the existing system. We will call this extended system the STI (Single Transition Identification) system as it is based on the original system that only identifies a single transition. After the system identifies the first transition (as explained in the previously proposed system), the sequence is divided in two at the transition point. The algorithm is repeated on the first and second part of the sequence. A potential transition for both parts of the sequence is then identified and the position with the minimum entropy is selected as the second intersection.

In this article, we also propose a new system, which can identify multiple transitions in a text. We will call this the MTI (Multiple Transition Identification) system. We start with the same input as the STI system, i.e., a text consisting of a sequence of snippets (S), which is handed to the LDA system, again resulting in a sequence of LDA topics: $LDA(S) = \langle LDA(s_1), LDA(s_2), \ldots, LDA(s_n)\rangle$. We then use a collation algorithm to search for the boundary using the following steps:

1. Number snippets according to their position in the text.
2. For each of the LDA topics, identify all snippets (represented by their number) that have that LDA topic assigned to them.
3. For each LDA topic, identify all sequences of consecutive snippet numbers. Calculate the length of all of these consecutive snippet numbers and add the lengths, which forms the value for that LDA

[2] `https://www.nltk.org/`

[3] `https://spacy.io/`

[4] The original system selected the transition that has the smallest intersection between the LDA topics on both sides of the boundary using a bag for one side and a set for the other side.

LDA	Snippet numbers	Value
1	[13, 14, 15], [17, 18], [22, 23, 24, 25], [33, 34, 35, 36, 37]	$A =$**14**
2	[1, 2, 3], 39, [44, 45, 46, 47, 48]	8
3	27, 50	0
4	49	0
5	6, 19, 21, 26, [28, 29, 30, 31, 32]	5
6	[4, 5, 6, 7, 8, 9, 10, 11, 12], 38, [40, 41, 42, 43]	$B =$**13**
7	16, 20	0

Table 1: For each LDA topic, corresponding snippet numbers are ordered. Groups of consecutive snippet numbers are identified, indicated by square brackets. The value belonging to an LDA topic is the sum of the lengths of all groups of consecutive snippets in a topic. The LDA topics with the largest values (which will be list A and B) are indicated in bold in the value column.

Group	[1	3]	39	[44	48]	Sum
A	12	10	2	7	11	42
B	3	1	1	1	5	11

Table 2: Calculate the minimum distance for the start and end snippets (indicated using open and close square brackets) of each group to the closest snippet value in list A and list B. Here this is illustrated for LDA topic 2.

class. Note that a snippet with number x forms its own group (which then has value zero) if no snippet with number $x-1$ and $x+1$ can be found with the same LDA topic.

4. Identify the two LDA topics with the highest value and call these list A and list B. If two LDA topics have the same value, pick the topic with the most snippets. If more than one topic also have the same number of elements, pick the first topic.

Steps 1–4 are demonstrated in Table 1.

5. Identify the LDA topic X with the next highest group value.

6. For the snippets that can be found at the start or end of a group in topic X or are not present in a group, identify the snippet that is closest to the snippet under consideration in both lists A and B. Store the minimum distances for each of these snippets. Add the distances. Table 2 illustrates this process for LDA topic 2. E.g., [1 is the beginning of a group from LDA topic 2. The closet snippet for this in list A is snippet 13, so the distance is 12. For list B, this is snippet 4, so the distance for list B is 3.

List	Snippet numbers
A	13, 14, 15, 17, 18, 22, 23, 24, 25, 33, 34, 35, 36, 37
B	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48

Table 3: Lists A and B after the snippets of LDA topic 2 are added to the list with the minimum total values, i.e., list B.

List	Snippet numbers
A	**13**, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, **37**
B	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, **12**, **38**, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50

Table 4: The final versions of list A and list B. All possible boundary positions are printed in bold.

7. Add the snippet numbers of the LDA topic under consideration to the list with the smallest sum of distances. Table 3 shows the altered list A and B.

8. Steps 5–7 are repeated for all the remaining LDA topics. If an LDA topic has the same sum of minimum distances to list A and list B, the length of the group in list A and list B that follows the start and end of each group in the LDA topic is used to calculate the minimum distance instead. If the minimum distance to list A and B is still the same, the segment is added to list A.

9. After all LDA topics are added to either list A or B, we identify and remove the outliers in both lists. An outlier is defined as a single snippet number x for which $x-1$ and $x+1$ are not found in the same list. The outliers are added to an outlier list.

10. For every outlier in the outlier list, we check if can be added to a group in either list A or B and the outlier is added to the appropriate list. Table 4 shows the final division of the snippets in lists A and B.

11. The last step is to identify the start and end of each group in list A and B. One group will always start with snippet number 1 and one group will always end with snippet number 50. We can delete these values from the possible boundaries as a boundary cannot occur in these positions. The remaining list of possible boundaries will then contain groups of two consecutive numbers. The boundaries will be between the consecutive numbers found in the different lists. In this example (see Table 4) the possible boundaries are between snippet numbers 12, 13 and between snippet numbers 37, 38.

3.3. Experimental Settings

To evaluate the performance of the new MTI system against that of the STI system which is an extension of the system proposed by Heyns and van Zaanen (2021), we first perform an experiment that requires the identification of one transition in texts. Note that the extended STI system behaves like the system proposed by Heyns and van Zaanen (2021) when identifying only one transition although this decides on a transition using entropy. From previous work we know that the STI system performs well on an ST data set. However, the STI system so far has only been tested on one text. Here we will extend the evaluation to multiple texts (as described in Section 3.1). Additionally, we can compare the performance of the MTI system against that of the STI system when assigning one transition.

The second experiment focuses on the identification of multiple transitions. Both STI and MTI systems will be applied to the MT data set, which illustrates how well both systems can identify multiple boundaries. We expect the MTI system to clearly outperform the STI system as this system has been specifically designed to deal with multiple boundaries.

As seen in previous work (Heyns and van Zaanen, 2021), the number of LDA topics assigned to the text can have a drastic influence on the performance of the system. Previous results showed that in cases where only one transition is identified, the (STI) system performs well. To provide a good overview, we will evaluate both the STI and the MTI system with two to 30 LDA topics (in steps of two). For each number of LDA topics, each system is run 100 times as, due to the random factor inherent in LDA, the LDA topics might be slightly different in each run. We provide the median, mean, and standard deviation results for each of these settings.

3.4. Evaluation

Standard evaluation metrics used in classification tasks, e.g., precision, recall, and F-score, are not directly suitable when trying to evaluate this particular problem even though the metrics could be used. True positives can be defined as when the system identified a transition that is the same as the real transition. In the same line, if the identified transition is not the same as the true transition, it is a false positive. False negatives occur when a true transition is not identified by the system. True negatives are cases where the system (correctly) identified a non-transition. From these, the standard metrics can be calculated. However, the problem with these evaluation metrics is that they do not take distance into account. It is better for the system to propose a transition close to the real transition than to propose one that is far.

Reynar (1994) proposed a metric where a window of three sentences is considered, which allows for a bit of leniency with respect to the transition location. However, this window length is arbitrary and will still rank a terrible system on par with an OK system that proposes transitions just outside the window. Alternative measures have been proposed by Beeferman et al. (1999), Pevzner and Hearst (2002) and Georgescul et al. (2008), each building upon one another. The advantages and shortcomings of each is discussed in detail in Purver et al. (2011).

To take into account the distance between the proposed transition and the true transition, we follow Heyns and van Zaanen (2021) and use the Root Mean Square Error (RMSE). RMSE takes distance between proposed and true transitions into account, allowing for a fine-grained comparison of systems. RMSE is defined as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (p_i - r)^2}{n}}$$

where n is the number of times the experiment is run, p_i is the position of the proposed transition position (which can range from one to 49) in run i (which ranges from one to 100) and r is the position of the real transition (at position 25 for the ST data set, and positions 13 and 38 for the MT data set). To calculate the RMSE for multiple transitions, the RMSE is computed for each transition and these values are combined using the average. The scikit-learn Python package[5] was used to calculate the RMSE.

4. Results

The results of all the experiments is provided in Table 5. We first consider the performance of the existing STI system on the ST data set that requires identification of a single transition. These results are comparable to the previous results as provided in Heyns and van Zaanen (2021), although the use of entropy to identify boundaries has improved performance. The differences between the texts are marginal, indicating that the STI system is robust.

Next, we can compare the performance of both STI and MTI systems on the ST data set. Here we see that the new MTI system consistently outperforms the STI system (with the exception when only two LDA topics are used where both systems have the same perfect performance).

When increasing the number of LDA topics the performance starts to decrease as can be seen by the increase of the mean and median of RMSE values. This holds for both STI and MTI systems. However, the results of the MTI system degrade more slowly.

Next, we compare the performance of the STI and MTI systems on the MT data set, which requires the identification of multiple boundaries. Here, the performance of the STI system is worse than that of the MTI system. Similarly to the results on the ST data set, the MTI system shows a perfect performance when identifying the intersections using two LDA topics. The performance starts to decrease as the number of topics increase, although the mean for each number of LDA topics is not much higher than the mean of the MTI system when a single boundary is identified.

[5] `https://scikit-learn.org`

		ST data set			MT data set		
System	# topics	MED	M	SD	MED	M	SD
STI	2	0.000	0.000	0.000	1.000	2.250	3.284
	4	0.000	0.000	0.000	2.000	2.875	2.900
	6	0.000	0.000	0.000	4.500	5.125	4.291
	8	0.000	0.000	0.000	5.000	5.250	2.435
	10	0.000	0.000	0.000	1.000	3.375	4.779
	12	0.000	0.000	0.000	5.000	5.000	4.408
	14	0.000	0.000	0.000	4.500	5.125	2.100
	16	0.000	0.000	0.000	6.000	6.375	3.583
	18	0.000	0.000	0.000	6.500	6.875	4.324
	20	0.500	0.000	0.000	4.000	6.125	4.643
	22	0.000	0.000	0.000	8.500	8.000	4.660
	24	0.000	0.250	0.000	8.000	8.500	3.464
	26	0.000	0.125	0.463	12.000	10.625	4.534
	28	0.000	0.375	0.354	12.000	11.875	1.853
	30	0.050	0.050	1.061	12.000	11.750	1.581
MTI	2	0.000	0.000	0.000	0.000	0.000	0.000
	4	0.000	0.000	0.000	0.000	0.000	0.691
	6	0.000	0.000	0.000	0.000	0.000	0.000
	8	0.000	0.000	0.000	0.000	0.000	0.000
	10	0.000	0.230	0.332	1.000	0.333	0.873
	12	0.000	0.000	0.000	0.000	0.000	0.000
	14	0.000	0.641	0.732	1.000	0.610	1.002
	16	0.000	0.000	0.000	0.000	0.000	0.000
	18	0.000	1.031	1.732	0.500	0.833	1.102
	20	0.000	0.750	1.500	1.200	1.050	1.321
	22	0.000	1.000	1.732	2.000	1.267	1.845
	24	0.000	1.000	1.732	2.300	1.714	1.500
	26	3.000	2.333	2.082	2.000	1.667	1.800
	28	0.500	1.000	1.414	1.650	1.200	2.004
	30	0.000	0.667	1.155	2.000	1.004	2.679

Table 5: RMSE results (MED: median, M: mean, SD: standard deviation) of the STI and MTI systems on both single (ST) and multiple (MT) transitions data sets, for the range of LDA topics.

5. Discussion

Being able to identifying the high level structure of a text is an important and useful aspect of distant reading analyses. With the aim of identifying the structure of a text, we start by identifying transitions within a text. Previous work (Heyns and van Zaanen, 2021) proposed a system that can identify a transition using topic information extracted using LDA.

In this article we extend the system, so that it can identify multiple transitions. We also introduce a new system specifically designed to identify multiple transitions in a text.

We compare both systems on data sets requiring single transitions as well as multiple transitions. This showed that the MTI system consistently outperformed the STI system when multiple transitions occur in the text. Both systems rely on LDA topic information assigned to snippets of the text under consideration. However, the MTI system performs a more complex, and as a result, more robust analysis of the behavior of the LDA topics over the snippets. The variation of the assigned LDA topics to the snippets in the text has a larger impact on the performance of the STI system as a result. What could be seen as noise in the assignment of LDA topics has a larger effect on the STI system compared to the MTI system. This is emphasized by the fact that this influence becomes larger when a larger number of LDA topics is assigned.

In contrast to previous work, the systems are evaluated on more than one text (created by concatenating snippets from two source texts). The fact that the results are highly comparable means that (even) the STI system is robust with respect to different texts.

Finally, as could be expected, the task of identification of multiple boundaries in a text is more complex than that of the identification of a single transition. However, the performance of the MTI system shows that multiple transitions can be identified with a reasonable performance (in particular if a low number of LDA topics is assigned).

6. Conclusion

In this article, we tackle the task of automatically identifying high level transitions in texts, which, for instance, indicates locations of changes in scenery, flashbacks, etc. This is an essential task in the context of distant reading as it provides information on the high level structure of texts.

We build on an existing system that has been shown to be able to identify one transition in a text (Heyns and van Zaanen, 2021) and extend it to handle multiple transitions. Additionally, we proposed a novel system that is specifically designed to identify multiple transitions. Both systems build on information provided by LDA, a topic modeling system.

The systems are evaluated on multiple texts which are created by concatenating snippets from two source texts (in contrast to the existing system which had only been evaluated on one text in previous work), showing that they lead to robust results.

The novel MTI system consistently outperforms the STI system for the identification of two transitions in the text. Also, the performance of the MTI system is comparable in the experiments requiring the assignment of one or two transitions.

With respect to future work, Heyns and van Zaanen (2021) indicated that further investigation is needed regarding the influence of the length of the snippets used to assign the LDA topics. This particular question has not been addressed in this article and still remains an open question. Given the robust results which are based on the LDA topics, we believe that the snippets may be made shorter allowing for a more fine-grained assignment of transitions.

We realize that concatenating snippets into a text in order to evaluate the systems' performance is artificial. In particular, when the source texts are very different semantically, LDA might provide exaggerated differences. Transitions in texts identified by human judgement should lead to a more natural means to evaluate the performance. However, one has to keep in mind that previous work indicated low agreement between annotators on such a task.

7. Bibliographical References

Aurnhammer, C., Cuppen, I., van de Ven, I., and van Zaanen, M. (2019). Manual annotation of unsupervised models: Close and distant reading of politics on reddit. *DHQ: Digital Humanities Quarterly*, 13(3).

Beeferman, D., Berger, A., and Lafferty, J. (1999). Statistical models for text segmentation. In *Machine Learning*, pages 177–210.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Choi, F. Y., Wiemer-Hastings, P., and Moore, J. (2001). Latent semantic analysis for text segmentation. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*.

Church, K. W. (1993). Char_align: A program for aligning parallel texts at the character level. In *31st Annual Meeting of the Association for Computational Linguistics*, pages 1–8, Columbus, Ohio, USA, June. Association for Computational Linguistics.

Franzini, G., Jänicke, S., Scheuermann, G., and Cheema, M. (2015). On close and distant reading in digital humanities: A survey and future challenges. a state-of-the-art (star) report. In *EuroVis*, 05.

Galley, M., McKeown, K. R., Fosler-Lussier, E., and Jing, H. (2003). Discourse segmentation of multi-party conversation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 562–569, Sapporo, Japan, July. Association for Computational Linguistics.

Genette, G., Lewin, J. E., and Culler, J. D. (1980). Narrative discourse : an essay in method. *Comparative Literature*, 32:413.

Georgescul, M., Clark, A., and Armstrong, S. (2008). A comparative study of mixture models for automatic topic segmentation of multiparty dialogues. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*.

Grosz, B. J. and Sidner, C. L. (1986). Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204, July.

Gruenstein, A., Niekrasz, J., and Purver, M., (2008). *Meeting Structure Annotation: Annotations collected with a general purpose toolkit*, pages 247–274. Springer, 02.

Heyns, N. and van Zaanen, M. (2021). Finding topic boundaries in literary text. In *Proceedings of the International Conference of the Digital Humanities Association of Southern Africa 2021*.

Hirschberg, J. and Litman, D. (1993). Empirical studies on the disambiguation of cue phrases. *Comput. Linguist.*, 19(3):501–530, sep.

Moretti, F. (2013). *Distant Reading*. Verso, London.

Pevzner, L. and Hearst, M. A. (2002). A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, 28:19–36.

Purver, M., Körding, K. P., Griffiths, T. L., and Tenenbaum, J. B. (2011). Unsupervised topic modelling for multi-party spoken discourse. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 17–24, Sydney, Australia, July. Association for Computational Linguistics.

Reynar, J. C. (1994). An automatic method of finding topic boundaries. *ArXiv*, abs/cmp-lg/9406017.

Skorochod'ko, E. F. (1971). Adaptive method of automatic abstracting and indexing. In *IFIP Congress*.