UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Kaur Karus

Using Embeddings to Improve Text Segmentation

Master's Thesis (30 ECTS)

Supervisor(s): Assoc. Prof. Mark Fišel, PhD

Tartu 2019

Using Embeddings to Improve Text Segmentation

Abstract:

Textual data is often an unstructured collection of sentences and thus difficult to use for many purposes. Creating structure in the text according to topics or concepts can aid in text summarization, neural machine translation and other fields where a single sentence can provide too little context. There have been methods of text segmentation that are either unsupervised and based on word occurrences or supervised and based on vector representations of words and sentences. The purpose of this Master's Thesis is to develop a general unsupervised method of text segmentation using word vector. The created approach is implemented and compared to a naïve baseline to assess the viability of this method. An implemented model is used as part of extractive text summarization to assess the benefit of the proposed approach. The results show that while the approach outperforms the baseline, further research can greatly improve its efficacy.

Keywords:

Text segmentation, word vectors

CERCS: P176 – Artificial Intelligence

Teksti vektorkujul esitamise kasutamine teksti segmenteerimiseks

Lühikokkuvõte:

Tekstipõhised andmestikud on tihti struktuuritud lausete kogumid ning seega raskesti kasutatavad paljude eesmärkide täitmiseks. Tekstis struktuuri loomine teemade või mõtete kaupa aitab teksti kokkuvõtmisel, tehisnärvivõrkudega masintõlkel ning teistel rakendustel, kus üksik lause võib pakkuda liiga vähe konteksti. Teksti segmenteerimiseks loodud meetodid on olnud kas juhendamata ning põhinevad sõnade koosesinemise vaatlemisel või juhendatud ning põhinevad sõnade ja lausete vektoresitustel. Selle magistritöö eesmärgiks on üldise tekstisegmenteerimise meetodi arendamine, mis kasutab sõnavektoreid ning koosinuskaugust. Loodud meetodi implementatsioone võrreldakse naiivse tõenäosusliku lahendusega, et hinnata loodud lahenduse efektiivsust. Ühte mudelit kasutati ka osana teksti kokkuvõtmise algoritmi osana, et hinnata lähenemise praktilist kasu. Tulemuste põhjal võib öelda, et kuigi loodud lahendus töötab paremini kui võrdlusalus, edasise uurimistööga on võimalik lähenemise võimekust märkimisväärselt tõsta.

Võtmesõnad:

Teksti segmenteerimine, sõnavektorid

CERCS: P176 – Tehisintellekt

Table of Contents

1 Introduction

Text segmentation is the process of dividing textual content to subgroups. Depending on the context and granularity, the subgroups may be thought of as distinct sections of an article or paragraphs of a section. The important thing about the subgroups is that they share context and can be thought of as entire thoughts or concepts. This means they have more context and content than individual sentences, but ideally deal with a single concept as opposed to entire texts that can look at a topic from multiple views or include many distinct event descriptions. Segmenting text can be used as a pre-processing step in machine translation, text summarization, or as part of text editing for improved readability. As a result, text segmentation is can be categorized as linear, where the text is completely divided into distinct segments, or hierarchical, where the text is divided into segments, and each of the segments are further divided into subsegments. Which type of segmentation is used depends on the end-use purpose of the segmentation.

While numerous different approaches have been used for text segmentation, they often rely on statistical formulae based on word occurrence frequencies [1, 2]. These approaches have shown great efficacy in synthetic datasets, but generally underperform on real life data. Furthermore, they are relatively unsuitable for use with noisy datasets that are created from transcribing spoken word or handwritten text. Word vectors are used in conjunction with neural transformer models, which require dedicated training data for the segmentation [3, 4], or in semantic relatedness graphs [5]. Since large corpora of training data are not always available for developing practical solutions, neural network models can be difficult to use in real-life applications. Depending on the specific graph-based algorithm, the problems can range from those of statistical models to neural network model or be a matter of long runtime due to graph construction. A compromise solution could use the benefit of pre-computed embeddings over word occurrences, and the perks of using a more statistical approach over those of graph-based or neural network based models. The proposed solution that uses distances between pre-trained word vectors for the basis of segmentation has not been reported.

Goal and Problem Statement

In this Thesis, a solution was constructed to explore the viability of using word or sentence embeddings as the main characteristic for text segmentation. Our proposed method constitutes a divisive clustering approach on textual data. The approach consists of the following steps:

First, the text is pre-processed and transformed into word and sentence vectors using a pre-trained model. The sentence vectors may be processed further processed to improve the overall quality of segmentation. Secondly, these vectors are used to decide where the text should be divided into subgroups using a general-purpose distance measure. The process is halted upon meeting a stopping criterium.

The research questions were defined as follows:

RQ1: How does the proposed approach compare to a naïve baseline?

This question focuses on the viability of the proposed approach. To prove that our method can be used for text segmentation, it has to consistently outperform a naïve baseline on different types of data. Whether it does and by how much, affects the assessment of the efficacy of the proposed approach.

RQ2: How does the choice of embedding model affect the performance of the model?

This question focuses on the various ways of processing the raw text to produce sentence embeddings and their effect on the model. While all text embeddings are designed to represent parts of text according to how each part relates to all other parts, grouping together elements that occur in similar contexts, the specific technique how this is achieved can have significant influence on how the distance measure relates to the sentence embeddings. By using embedding models that are based on different approaches to embedding calculation, and a potential post-processing step, we can assess how the choice of the embedding model changes how well the distance measure describes the shift in context.

RQ3: How does the proposed approach affect the performance of existing algorithms?

This question is aimed to assess whether using the proposed method of text segmentation is an advantage or a disadvantage to existing algorithms of text processing. Due to the nature of text processing, this can include a variety of algorithms for different purposes that can be based on word occurrences or embeddings as appropriate. If the proposed approach proves to be beneficial to algorithms in some contexts, then it is a valuable contribution to solving the appropriate problems in at least these contexts.

The Thesis consists of multiple parts. Chapter 2 describes the previously published models used for text segmentation. Chapter 3 discusses the elements of the model and describe it in detail. The fourth Chapter describes the experiments that were carried out, followed by the results and comparison of the proposed approach to the respective baselines. The final Chapter summarizes the Thesis.

2 Overview

2.1 Related work

Statistical models

Because different text segments are likely to hold different words, using the statistical properties of words within texts has been a common approach to text segmentation. TextTiling [6] checks every potential candidate sentence and compares the similarity of the word occurrences on either side of the candidate. It assumes that the algorithm will not detect every paragraph, but will detect sets of paragraphs pertaining to the same topic, split apart for the purposes of readability rather than as containers of entire concepts. Choi [2] improved upon this model by using a ranking algorithm on the similarity matrix and clustering the ranking matrices. These approaches, however, rely purely on term occurrences. These approaches are also difficult to use in a hierarchical setting for topic detection.

TopicTiling [7] is an approach to text segmentation that uses Latent Dirichlet Association (LDA) topic modelling to detect potential segment boundaries. Instead of using bags of words for a similarity matrix, it uses the topic ID-s returned by LDA over multiple runs. This requires data to train an LDA model on to properly detect the word-topic relationships. The approach does use word occurrences, which may cause a problem in real-life scenarios where lexical variety can greatly differ in documents.

Graphs-based models

Word occurrence can also be used in graphs [8], considering each sentence as a node and drawing edges between the sentences if they contain the same word or words. The weight of the edge would depend on how common the shared words are in the document and on the Google search engine. The resulted graph is then split into pieces that maximize the similarity measure within each subgraph.

Semantic relatedness graphs [5] have been used to divide a text into segments by attempting to estimate how much each word in a sentence relates to each other word in the sentence preceding it using word embeddings. The strongest related sentences would be connected into segments while the boundaries are determined by the weakest relations. The described model employed two parameters and did not require any training data.

Affinity Propagation for Segmentation [9] is an approach that uses a similar approach, comparing the similarities between sentences is used to construct a factor graph. Using a user-defined *a priori* preference for each sentence to start a paragraph and the factor graph, a result in which the segments are defined by a single sentence in the middle of segment that explain the surrounding sentences the most is created. This approach also requires training data and pre-existing knowledge of the expected content of the text.

Neural network models

Long Short-Term Memory (LSTM) models have been shown to be usable to detect segmentation boundaries [3]. In such case, the architecture relies on two sub-networks where the first one processes the sentences as matrices containing their word vectors and returns corresponding sentence embeddings, and the other makes the predictions concerning sentence boundaries. This approach can be faster to use than the graph-based algorithms, but requires significantly more training data. It is also inconvenient to use if a hierarchical segmentation is preferred.

3 Methodology

3.1 Embeddings

Word vectors or embeddings are a method of representing words in a text as numerical data points inside a large vector space. The most common way of learning these embeddings is with the use of Recurrent Neural Networks, such as Word2Vec [10] , GloVe [11] and FastText [12]. The aim of embeddings is to encode the semantic relationships between words so that the meaning or context of a word could be calculated from other words. For example, the calculation *Paris - France + Italy* should result in *Rome* [10]. As a corollary, words used in similar contexts, referencing similar concepts or topics are likely to be closer to each other. While the exact implementations of the methods differ, the embeddings are learned from text corpora, where the position of a word in relation to other words in the sentence or text help define its relationships to the other words. Due to the differences in implementation, the different methods for learning word vectors have different performance in regards to various tasks.

The Google's Word2Vec model is designed to aid in word prediction and uses a lemmatized text corpus to train the model. Common words as well as extremely rare word are discarded, and the context of the remaining words is analysed with a skip-gram model. Negative sampling is used to reduce noise. The result is a model where each non-discarded word in the training text corpus receives a fixed set of coordinates that can be used for processing any new texts. A model created this way is relatively small, depending on the number of words in its vocabulary, and fast to use. However, coordinates cannot be calculated for any words not belonging to the vocabulary. In addition, since it uses lemmatized words, it ignores a lot of grammar that can affect context.

FastText, developed by Facebook, attempts to remedy these problems by using character n-grams instead of lemmatized words as the basic elements of a text. This means that grammar is preserved, with different grammatical cases of the same word being located very close to each other, and the coordinates of previously unseen words can be inferred through the order of letters used in those words.

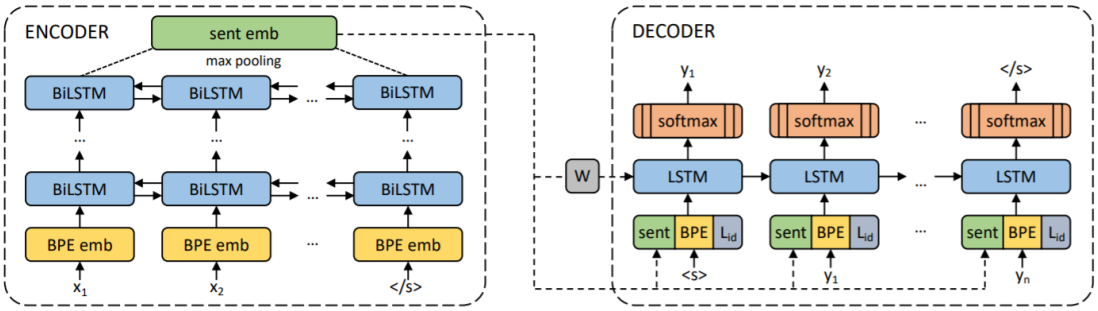

Figure 1. Structure of LASER encoder-decoder training system [14].

Embeddings can also be applied to entire sentences. Sentence embeddings can be calculated as an average or sum of the embeddings of all words it contains or using novel approaches such as sent2vec [13], which uses a bag-of-n-grams approach to estimate the sentence embedding, or LASER [14], which uses a BiLSTM encoder on texts that are pre-processed using Byte-Pair Encoding (BPE) [15] to calculate more representative sentence embeddings. BPE is used to find common character n-grams that words consist of, similarly to FastText's approach. A LASER model is trained on parallel corpora using an encoder-decoder model

where the encoder calculates embeddings for each BPE-processed element and combines them into a sentence embedding while the decoder calculates the respective sentence for another language. This allows the model to be trained on any number of languages, as long as there exists a parallel corpus for that language and another. At the end of training, the decoder is discarded, and the encoder can be used to calculate sentence embeddings of any BPE-processed sentences.

3.2 Text Segmentation with Embeddings

To evaluate the proposed approach, three implementations of the algorithm were created using four different methods of embedding pre-processing. The used embeddings are based on either the Word2Vec model or the LASER model, with an optional further step using PCA.

Word2Vec model

Each sentence gets a temporary matrix for storing all of the vectors of its words. Each word is tokenized and the vector of the token in the pre-trained model is added to the matrix. The vector values for the words in a sentence are averaged to result in a single 300-dimension sentence embedding. Experiments show that using a sum or weighted sum dependent on inverse word frequency instead of the average does not change the performance of the model in any significant way.

LASER model

Each sentence is tokenized and processed using Byte-Pair Encoding. This results in most words being divided into common groups of characters. The result is then processer by the LASER encoder that results in a single 1000-dimension embedding for each sentence.

Depending on the experiment, this can be the end of pre-processing, or the sentence vectors can be processed using Principal Component Analysis to reduce the dimensionality of data in order to make all sentences easier to separate.

3.3 Boundary estimation

The nature of text segmentation defines two scenarios. Firstly, there is hierarchical segmentation, wherein a document is divided into topics and each topic into subtopics recursively in order to create a tree that describes the overall structure of the document. Secondly, linear segmentation only requires the segment boundaries with no hierarchical structures. As the proposed approach does not discriminate what type of text segmentation is required, a model of either type was implemented.

Recursive model

The recursive model looks at all of the text they are required to split in two. For each candidate sentence they check the sum vector for all sentences preceding the candidate and the sum vector of all sentences following the candidate sentence, including the sentence itself. When the cosine distance is the highest, meaning the greatest separation between the two parts of the text, a split is marked at the candidate sentence index. The process is continued

Equation 1. Distance between two sets of sentences at candidate index *m*.

$$distance = 1 - \cos(\sum_{i=0}^{m} vec_i\,,\ \sum_{i=m+1}^{n} vec_i)$$

recursively for all the sentences before the split, and all the sentences following it. Every time, the previous best distance is carried down the recursive tree. If a level of recursion does not find a cosine distance that would be greater than a scalar multiplied the largest distance found in its parent level, the splitting is cancelled as the level of separation that would be resulting from the split would probably be inside an entire segment and would thus be superfluous.

As a rule, a number of first and last sentences under view are not considered as splitting candidates. The specific limit of sentences not considered at either end is defined by a *gap* parameter. This is because otherwise the most likely splitting candidate would be near the limits of the index space, resulting in very imbalanced segmentation that is not common in human texts. The recursive model therefore has two adjustable parameters: the number of sentences not considered as splitting candidates at the index limits and the scalar that defines how much smaller a recursively decided split can be in relation to the largest detected distance in its parent split.

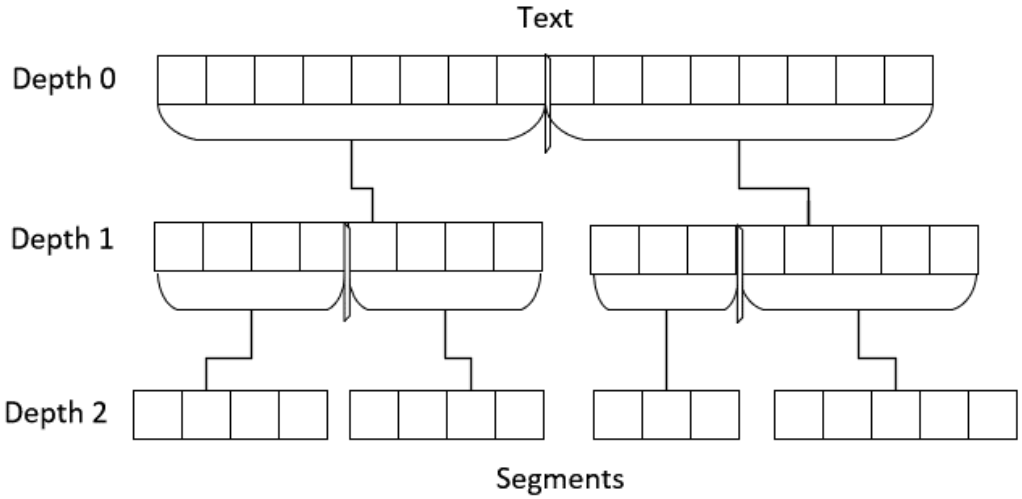

Figure 2. Diagram illustrating the process of splitting a list of sentences into segments by the recursive model, `gap=3`.

Sequential model

While the recursive model is using a top-down approach, a sequential model using a sliding window over the sentences to detect segmentation points. It observes the embeddings of k first sentences, finds the greatest cosine distance between sentence sequences, and splits the sentences along that distance index. Then it processes the k embeddings after the decided split and repeats the process until it has reached the end of the text. In comparison to the recursive model, the segments are more evenly distributed and the number and length of segments detected is more consistent across different texts, but it is also more likely to split long segments that should not be split. The sequential model does not have a scalar to compare the splitting distances between child and parent, but does use a parameter that defines how many sentences are within view of the model.

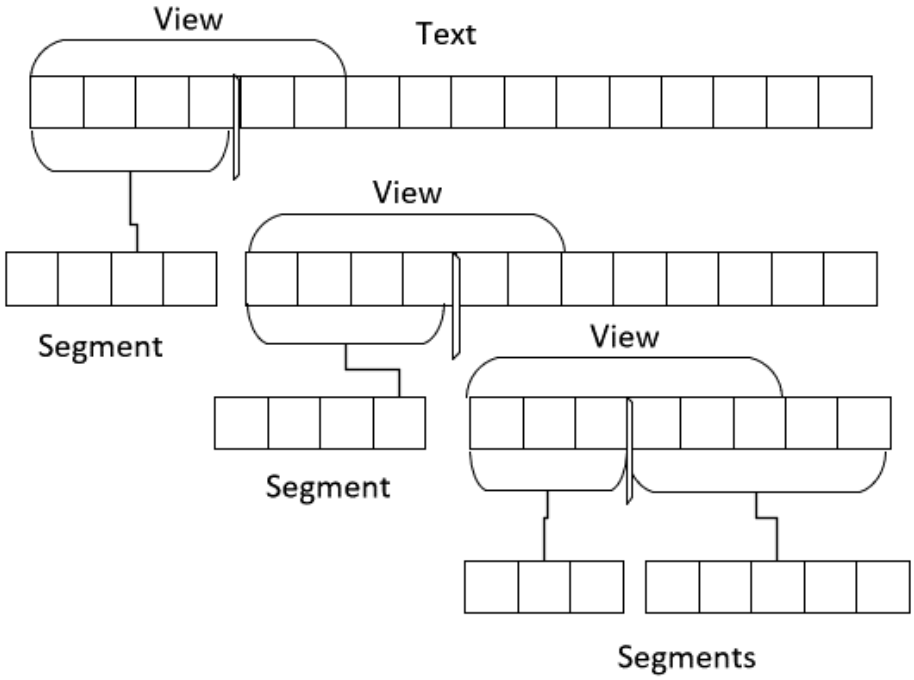

Figure 2. Diagram illustrating the process of splitting a list of sentences into segments by the sequential model, the view size is 6, as defined by k, `gap=3`.

3.4 Text summarization

To test the effect of using the proposed approach for text segmentation in a real-life scenario, an experiment using one of the implemented models as a pre-processing step as part of a pre-existing text summarization algorithm was carried out. This would be compared to the performance of the same algorithm without using text segmentation.

Centroid-based summarization

Centroids are the centre coordinates of elements that consist of sub-elements, such as sentences that consist of words or documents that consist of sentences. The manner in which the centroid is calculated can differ depending on the specific approach taken. Traditionally, sentence centroids have been defined by a $tf \times idf$ weighted sum of the embeddings of its words [16, 17]. $tf \times idf$ (term frequency multiplied by inverse document frequency) increases the weight of words that occur in a sentence more often than in other sentences and

marginalizes the effect of words that occur in many sentences, such as articles “a”, “an” and “the” in English. It has the desired effect of giving more weight to words that are more text- or topic-specific. If possible, *idf* is calculated over a larger dataset than the single document. The same process is repeated over all words in the document to calculate the document centroid. A distance measure, such as cosine distance, is used to compare the distances between each sentence and the document itself, and the closest sentences are chosen for the summary. The chosen sentences are then sorted in the order of their appearance in the text.

To compare the naïve approach of using $tf \times idf$ to calculate sentence centroids and summarizing by choosing single sentences, the same text is segmented and the centroids of the segments are used as candidates for the summary. The centroids of the segments are defined as the average of the centroids of the sentences within the segments.

4 Experiments

4.1 Text segmentation

Datasets

To evaluate the efficacy of the approach, three datasets were used. Clinical dataset [9] contains 227 chapters from medical textbooks with chapter sections used as segmentation borders. Fiction dataset [9] contains 85 books in the fiction genre sourced from Project Gutenber. Book chapters are used as segmentation borders. Wikipedia dataset [4] contains 300 randomly selected Wikipedia articles with article sections used as segmentation borders.

Metrics

For evaluation purposes, the average Pk value [1] over all documents in a dataset was used. Pk uses a sliding window approach to evaluate whether the sentences within the window are correctly identified to belong to different segments or the same segment. Pk is a loss measure, the lower the value, the better the model performs. The SegEval[1] package implementation of Pk measurement was used.

Baseline

A randomized model based on the ground truth values was created to act as a baseline model. At each sentence index, a segment boundary would be placed with a probability equal to the ratio of the number of segment boundaries in the dataset to the number of all sentences in the dataset. The Pk values for the baselines, averaged over 5 runs, are described in Table 1.

Table 1. Baseline model Pk values.

Model	Clinical	Fiction	Wikipedia
Baseline	0.5658	0.7465	0.9567

The difference in results is due to difference on homogeneity in the datasets. Clinical textbook chapter sections are of similar length throughout the books, resulting in a relatively uniform distribution of segmentation borders throughout the dataset. Wikipedia sections, however, differ significantly in length, causing a poor score for a randomized algorithm.

Models

Three models were used for evaluation. Firstly, the recursive model with pre-set parameters. The `gap` parameter was initially set to 50, and decreasing by 5 at each recursive call. This would allow for a broad segmentation at a high level while allowing the same model to segment the resulting divide with increased granularity. Secondly, the sequential model with pre-set parameters. This model would look at 100 sequential sentences, but only place the segmentation border in the middle 50 sentences, effectively stating that the size of a segment is always between 25 and 75 sentences. Thirdly, the recursive model with parameters tuned by a grid search algorithm over randomly selected 10% of the dataset. The motivation behind the approach is that it intended to be used without any training process taking place at all, but it can indicate whether the model can benefit from having small training set. Since this is the only model that employed a training set, it is also the only model to use

[1] `https://segeval.readthedocs.io/en/latest/`

crossvalidation. All models used a pre-trained Word2Vec model[2] and all except the tuned model used a pre-trained LASER[3] model, as well as embeddings processed with PCA to reduce the data to 30 dimensions. The pre-set parameters are shown in Table 2.

Table 2. Used pre-set parameters.

Model	Gap	Threshold scalar	Window width
Recursive	50 *	0.85	
Sequential	25		100

4.2 Text summarization

Datasets

Two datasets containing news articles from CNN and Daily Mail [18] were chosen for this real-life scenario. Each article has human-annotated abstractive highlights that are used on the respective websites to give a fast overview of the most important facts in the articles. These highlights were used as the reference summaries for the evaluation of the extractive summarization method. Python's `nltk` package was used to split the article into sentences. The lengths of the articles ranged from 4 sentences to 93 sentences, with the average summary length being four sentences. Articles of length less than 10 sentences were removed from the datasets; the resulting dataset sizes were 88719 individual articles for the CNN dataset and 149537 articles for the Daily Mail dataset. As the datasets consisted of texts of various sizes and topics, but had to be summarized with very few sentences, these datasets pose great difficulty for any summarization algorithm.

Metrics

For the evaluation of summarization, ROUGE-N [19] metrics were used. ROUGE-N calculates the overlap between n-grams that occur in both the proposed summary and the reference summary. In essence, the metric checks how many n-grams are present in both summary and reference and divides them with the number of n-grams present in either the summary or the reference. This gives an assessment on how well the summary matches with the reference on a word occurrence basis. ROUGE is not an error metric, higher value is better. ROUGE-N prediction has also been used for extractive text summarization algorithms [20].

Models

For the purposes of the experiment, the sequential segmentation model was used. The model employed the same pre-trained LASER model to calculate sentence embeddings and used PCA. As the target summaries were significantly shorter than book chapters, the parameters for the sequential model were accordingly set to be lower. The `window` parameter was set to 20 and the `gap` parameter was set to 3, meaning segment lengths would range from 3 to 17 sentences. Two segments closest to the centroid would be chosen. As many sentences as

[2] `https://code.google.com/archive/p/word2vec/`
[3] `https://github.com/facebookresearch/LASER`

the two segments contained would be chosen by the non-segmented method for an equal comparison.

5 Results

5.1 Text segmentation

The results of the models using Word2Vec embeddings are described in Table 3. While the improvement over the baseline in the Clinical dataset is relatively small, larger benefits are seen on the other two datasets. In particular, the difference between the baseline and the PCA-using models proves that the approach is viable. The best results were achieved by the sequential model using PCA-processed embeddings or the recursive model using PCA-processed embeddings and a small training set. This demonstrates the importance of the choice of which implementation of the algorithm is currently used, as well as the benefit of being able to tune the model parameters to suit the domain more efficiently.

Table 3. Pk scores of models using Word2Vec embeddings.

Model	Clinical	Fiction	Wikipedia
Baseline	0.5658	0.7465	0.9567
Recursive	0.6784	0.6331	0.6550
Recursive-PCA	0.6543	0.6464	0.7312
Sequential	0.5424	0.5319	0.5919
Sequential PCA	**0.5239**	0.5242	**0.5542**
Recursive tuned	0.5816	0.5400	0.6132
Recursive tuned PCA	0.5834	**0.4620**	0.5898

The parameter tuning mostly affected the threshold scalar, raising it to 0.9 for Clinical and Fiction datasets and lowering it to 0.75 for the Wikipedia dataset. This would result in a more conservative approach than the pre-set parameters proposed for the first two datasets, creating fewer segments, and a significantly more aggressive approach in the Wikipedia dataset, resulting in more segments. This demonstrates that the proposed approach requires some tuning depending on the text domain. In particular, a lower threshold scalar can delay the point at which the segmentation is stopped, creating a deeper segment hierarchy. This can be especially useful if the segments are relating to very similar topics where the inter-segment embedding distances are small.

The results of the models that used LASER embeddings in Table 4 demonstrate the effect of using a model specifically created for sentence embeddings on the approach that uses them over sentence embeddings calculated by averaging word embeddings. As visible, there is no consistent positive or negative effect of embedding choice on the approach. There is, however, a notable benefit in using PCA on almost every model.

Table 4. Pk scores of models using LASER embeddings.

Model	Clinical	Fiction	Wikipedia
Baseline	0.5658	0.7465	0.9567
Recursive	0.6818	0.6364	0.6556
Recursive-PCA	0.6631	0.6510	0.7313
Sequential	0.5452	0.5347	0.5916
Sequential PCA	**0.5277**	**0.5269**	**0.5554**

The analysis of the resulted segmentation borders shows that almost every document was split too conservatively, creating too few segments. This indicates that while the general number of segments did not match the expected number of segments, the placement of the segment boundaries was accurate. This is likely due to the `gap` parameter being a very strict limiter for segmentation as the segment length could range from 10 sentences to 162 sentences even in a single document. While decreasing the `gap` parameter for the recursive model that can also stop dependent on segment distances may offer a great benefit, it may cause complications for the sequential model, allowing it to become too aggressive.

The positive results attained by the sequential model were most likely caused by the sliding window approach forcing the algorithm to place the segmentation borders at relatively regular intervals. This has the effect that an incorrectly placed border among the first decisions can be remedied by the next placements. In the recursive case, if a border was placed incorrectly, it is more likely to affect the locations of the last placements made. In essence, the recursive model is more vulnerable to early errors.

5.2 Text summarization

Table 5 shows that in the CNN dataset, the model that chose segments with centroids closer to the document consistently outperformed the initial algorithm that added individual sentences. The average length of a proposed summary consisting of two segments was 8.63 sentences, which is more than twice the length of the average reference summary. As the length of both proposed summaries was always kept identical, this explains the low precision score for both models.

Table 5. ROUGE-1 and ROUGE-2 scores for text summarization on the CNN dataset.

Model	ROUGE-1 Recall	ROUGE-1 Precision	ROUGE-1 F1-score	ROUGE-2 Recall	ROUGE-2 Precision	ROUGE-2 F1-score
Segmented	0.4768	0.2007	0.2788	0.1577	0.0641	0.090
Sentence-based	0.4288	0.1823	0.2522	0.1200	0.0494	0.0689

Table 6. ROUGE-1 and ROUGE-2 scores for text summarization on the Daily Mail dataset.

Model	ROUGE-1 Recall	ROUGE-1 Precision	ROUGE-1 F1-score	ROUGE-2 Recall	ROUGE-2 Precision	ROUGE-2 F1-score
Segmented	0.4494	0.2174	0.2818	0.1491	0.0709	0.0928
Sentence-based	0.4234	0.2041	0.2650	0.1260	0.0584	0.0770

The results on the Daily Mail dataset, as shown in Table 6, are slightly higher than in the CNN dataset. The other statistics remain the same, confirming that the sequential segmentation model produces segments of relatively consistent length. It also supports the intuition that sentences that give a document, such as a news article, meaningful context are generally close together. A single sentence often carries very little meaningful context about the entire text, a segment consisting of multiple sequential sentences can hold concepts important to the text. This is reflected in the fact that the man-made highlights never directly quoted sentences from the article, but often consisted of parts of multiple sentences to add brevity.

From the results (see Appendix I and II) we can also see that some assistance and difficulties for the models stemmed from the nltk sentence tokenizer also splitting some sentences into sentence parts. This may have been beneficial as the reference summary often used only parts of the existing sentences, or disadvantageous as choosing only a part of the sentence for the summary without properly completing it can produce aesthetically unpleasant results,

calling into question the practicality of this approach. Since the segmentation model uses sentences that are close together, it generally connected the sentences that were broken apart by `nltk`, which demonstrates the flexibility of the approach.

It is also important to note that the two methods choose similar short sentences that contain terms that occur very rarely in general texts. The main difference being that while the sentence-based model can then choose another significant sentence, the segmenter-modified model has to accept sentences immediately preceding or following the most significant sentence that are part of the same segment. The latter has significantly fewer choices, while the former is more flexible. But as the results show, a collection of independently significant sentences does not carry as much relevant information as significant sentences that are surrounded by context.

6 Conclusions

The proposed and described approach of using the cosine distances between sentence embeddings to segment texts is a novel approach that shows great promise. The initial results from testing with different types of textual data with varying segment sizes shows that the system works relatively well when using a fixed set of parameters, but can be greatly improved by parameter tuning even on very little example data. In a practical setting, the model demonstrates a benefit to text summarization on real world data.

RQ1: How does the proposed approach compare to a naïve baseline?

Depending on the type of data and the specific implementation of the approach, the improvement gained with this approach can be significant. The approach clearly works best with unbalanced data where the length of segments varies greatly between and in documents. The specific implementation of the approach also has a significant effect to the quality of segmentation. Additionally, the parameter values used for segmentation are critical to the performance of the models, as shown by the third tested model.

RQ2: How does the choice of embedding model affect the performance of the model?

In the experiments involving a pre-trained Word2Vec model and a pre-trained LASER model, it is apparent that the choice of which embeddings model to use made relatively little difference for this approach. Using Principal Component Analysis, however, significantly improved the quality of segmentation. This means that while initial tests using different embedding models failed to show any notable benefits to any, it cannot be ruled out that an embedding model exists or can be created that suits the purposes of the algorithm significantly better.

RQ3: How does the proposed approach affect the performance of existing algorithms?

The extractive summarization experiments show that summarizing by segments instead of single sentences can have a beneficial effect on extractive summarization on real life data. With the benefit being apparent with a relatively simple algorithm, the proposed approach shows great promise to be an advantageous pre-processing step in other algorithms pertaining to text analysis.

In conclusion, we have presented a novel approach to text segmentation and described two structurally different ways of implementing it. The methods of implementation were compared against each other and a naïve baseline. Based on these comparisons it can be claimed that the proposed approach is a functional method for text segmentation. Further analysis to the effect different forms of pre-processing, including the details of way embeddings are calculated for the sentences and the processing of the sentence embeddings prior to applying the algorithm, parameter tuning as well as model implementation can have to the efficacy of the approach would be critical in order to improve the quality of segmentation. By performing another experiment with an existing method of text summarization to assess the effect of using the proposed method as a part of pre-processing. As the results showed a consistent improvement over the baseline performance of the same algorithm, it is evident that the proposed approach has practical applications in real-life scenarios. It is also apparent that the proposed approach can improve the performance of other natural language processing algorithms when used as a pre-processing step. Whether the approach can perform at an equal level to the cutting-edge approaches using recurrent neural networks is doubtful,

but it has the significant benefit of requiring little or no training data at all to have consistent performance over different domains.

7 References

[1] D. B. A. L. J. Beefermann, "Statistical Models for Text Segmentation," in *Machine Learning*, 1999, pp. 177-210.

[2] F. Choi, "Advances in domain independent linear text segmentation," in *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, 2000.

[3] O. C. A. R. M. B. J. Koshorek, "Text Segmentation as a Supervised Task," in *NAACL*, 2018.

[4] P. K. L. G. M. V. V. Badjatiya, "Attention-based Neural Text Segmentation," 2018.

[5] G. N. F. P. S. Glavaš, "Unsupervised text segmentation using semantic relatedness graphs," *The Fifth Joint Conference on Lexical and Computational Semantics*, pp. 125-130, 2016.

[6] M. Hearst, "Multi-paragraph segmentation of expository text," in *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, 1994.

[7] M. Rield and C. Biemann, "TopicTiling: A Text Segmentation Algorithm based on LDA," in *Proceedings of the 2012 Student Research Workshop*, 2012.

[8] M. A. M. Pourvali, "A new graph based text segmentation using Wikiperia for automatic text summarization," *International Journal of Advanced Computer Science and Applications*, 2012.

[9] A. Kazantseva and S. Szpakowicz, "Linear Text Segmentation Using Affinity Propagation," *EMNLP*, pp. 284-293, 2011.

[10] T. C. G. C. K. D. J. Mikolov, "Efficient Estimation of Word Representatives in Vector Space," in *International Conference on Learning Representations (ICLR)*, 2013.

[11] J. Pennington, R. Socher and C. D. Manning, "GloVe: Global Vectors for Word Representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532-1543.

[12] P. Bojanovwski, E. Grave, A. Joulin and T. Mikolov, "Enriching Word Vectors with Subword Information," 2016.

[13] M. Pagliardini and P. J. M. Gupta, "Unsupervised Learning of Sentence Embeddings using Compositional n-Gram features," *NAACL 2018*, pp. 528-540, 2017.

[14] M. Artetxe and H. Schwenk, "Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond," 2018.

[15] Y. Shibata, T. Kida, S. Fukamachi, M. Takeda, A. Schinohara, T. Shinohara and S. Arikawa, "Byte Pair Encoding: A Text Compression Scheme ThatAccelerates Pattern Matching," 1999.

[16] G. Erkan and D. Radev, "LexRank: Graph-based Lexical Centrality as Salience in Text Summarization," *Journal of Artificial Intelligence Research*, pp. 457-479, 2004.

[17] G. Rossiello, P. Basile and G. Semararo, "Centroid-based Text Summarization through Compositionality of Word Embeddings," 2017.

[18] K. Hermann, T. Kocisky, E. Grefenstette, L. Espehold, W. Kay, M. Suleyman and P. Blunsom, "Teaching machines to read and comprehend," *Advances in Neural Information Processing Systems*, pp. 1684-1692, 2015.

[19] L. Chin-Yew, “ROUGE: a Package for Automatic Evaluation of Summaries,” in *Proceedings of the Workshop on Text Summarization Branches Out*, Spain, 2004.

[20] M. Zopf, E. Mencia and J. Fürnkranz, “Which Scores to Predict in Sentence Regression for Text Summarization?,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2018.

Appendix

I. Sample result of text summarization, Daily Mail dataset

Reference summary

"Eric Craggs, 68, accused of asking for 'Laserstar' device to be fitted to car", 'Device interferes with lasers in speed guns and stops reading being taken', 'Police officer found device attached to front of his Aston Martin last year', 'Prosecution claim Craggs has been driving around with it fitted since 2009', 'Craggs denies asking for it to be fitted and perverting the course of justice'

Candidate of the segmented model

'Emma Glanfield', "Eric Craggs, 68, of Stockton, County Durham, pictured arriving at court, is accused of asking for the 'Laserstar' device to be fitted to his car in 2009", 'A driver attempted to evade the law by having a piece of equipment knowingly fitted to his Aston Martin which stops police speed guns from taking an accurate reading, a court heard.', "Eric Craggs, 68, is accused of asking for the device, known as a ‘laser jammer', to be fitted to the front of his car during a service in 2009.", 'It is alleged that Craggs, of Stockton, County Durham, has been driving the car with the device activated for the last four years and has managed to evade police detection on two occasions.', 'The device interferes with the lasers used in police speed camera devices and stops officers from being able to take a reading - instead issuing an error message.', 'Prosecutor Andrew Walker told the jury at Teesside Crown Court today how Craggs was caught by PC Lorraine Williams after she failed to get a reading from his Aston Martin during a routine speed check in August last year.'

Candidate of the sentence-based model

'Emma Glanfield', 'She tried again but again an error code showed.’', 'Mr Walker said PC Williams also remembered a similar instance with the same car in April 2011.', 'Mr Walker', 'said: ‘Its purpose is to alert the driver of the vehicle that is being', 'the vehicle.', 'Mr Walker said Craggs denies asking anyone to fit the device and says he had no knowledge of it being there.'

II. Sample result of text summarization, CNN dataset

Reference summary

"Hamed Haddadi began playing for Tennessee's Memphis Grizzlies in August 2008", 'Despite U.S.-Iran tensions, strains appear absent with teammates, fans alike', 'NBA had to apply to the U.S. government for a license to let Haddadi play', "He's been trying to bridge gap between Iranian-Americans and basketball"

Candidate of the segmented model

'East Rutherford, New Jersey (CNN) -- From nuclear weapons to human rights, the image of Iran is quite negative in America.', 'But with little fanfare, one Iranian man has won hearts and cheers battling Americans on the court in basketball arenas around the country.', "Hamed Haddadi is the NBA's first Iranian basketball player.", '"Iranian playing basketball in America ... that's rare.', 'There aren't many Iranians doing anything in bona fide sports arenas in the U.S."', 'Haddadi faces big challenges.', 'One is speaking and learning English.'

Candidate of the sentence-based model

'East Rutherford, New Jersey (CNN) -- From nuclear weapons to human rights, the image of Iran is quite negative in America.', "It wasn't as easy getting permission to play in the United States.", '"You're sure it's not Borat's older brother?"', '"He's gotten more press then any of his teammates this year and the past couple of years just for the sole reason that he's Iranian-American," said Zokaei.', 'Furthermore, his family is almost 7,000 miles away in Iran.', 'The foundation has not been his only initiative.', "Haddadi's team did not make the NBA playoffs, which start within the week.

III. License

Non-exclusive licence to reproduce thesis and make thesis public

I, **Kaur Karus**,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

Using Embeddings to Improve Text Segmentation,

supervised by Mark Fišel, PhD.

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Kaur Karus

20/05/2019