2018 IEEE International Conference on Data Mining Workshops (ICDMW)

A Domain-Independent Text Segmentation Method for Educational Course Content

Yuwei Tu
Advanced Research
Zoomi Inc.
Wayne, PA
yuwei.tu@zoomi.ai

Ying Xiong
Department of Computer Science
Rice University
Houston, TX
yx47@rice.edu

Weiyu Chen
Advanced Research
Zoomi Inc.
Shenzhen, China
weiyu.chen@zoomi.ai

Christopher Brinton
Advanced Research
Zoomi Inc.
Wayne, PA
christopher.brinton@zoomi.ai

***Abstract*—In this study, we have proposed a domain-independent text segmentation algorithm which is particularly useful in online educational courses. Text segmentation is proven to be helpful in improving the readability of large corpora of documents, which is essential in education scenarios. While existing domain-dependent text segmentation methods have much better performance than domain-independent methods in most cases, only domain-independent methods are applicable to sparse training content in education scenarios. Our method, unlike other domain-dependent text segmentation methods, doesn't require heavy training on prior documents, but only need to train on the current corpus of documents with topic distributions and word vector representations. Our proposed method develops text boundaries between small text units in three steps. We first calculate input text features via topical distributions (latent Dirichlet allocation) and word embeddings (GloVe). We then calculate similarity values between such textual features and detect distribution changes between the similarities. We finally perform clustering on the similarities and detect sub-topic boundaries via cluster differences. We test our method on two datasets, one from an online education course and one from a popular public dataset - Choi Dataset. The results demonstrate that our method outperforms other state-of-the-art domain-independent text segmentation approaches while achieving performance comparable to a few domain-dependent algorithms.**

***Keywords*—Text Segmentation, Topic Modeling, Latent Dirichlet Allocation, Semantic Information, Word Embedding**

I. Introduction

Text segmentation – the task of automatically categorizing text into coherent parts – has become essential to document summarization and natural language processing (NLP) that aim to improve the readability of large corpuses of documents [1]. Even well-structured texts, such as those that the author has organized logically by topic or by paragraph, can benefit from such methods in at least two ways. On the one hand, by extracting relevant parts of a document, infobesity (information overload) for readers can be alleviated. On the other, automated text segmentation can provide insight to the author on the latent structure of their content and how it may be improved.

The potential for benefit from intelligent text segmentation is particularly salient in online electronic learning (e-learning). Personalized instruction – the provision of tailored course materials to accommodate specific user needs and interests – is imperative to the learning process but difficult to provide in a distance learning scenario [2]. Several studies, for example, have pointed to lower learning outcomes in online courses as opposed to traditional instructor-led training [3]. Fine-granular text segmentation, e.g., grouping segments by sentences instead of larger sections of text, can facilitate personalization in many ways. For one, it can provide useful information on the prerequisite relationships between content in a course [3], which can then be used either by learners themselves when they feel they are struggling with material or by automated individualization algorithms that modify content based on signals of learning inefficacy. Further, in conjunction to machine learning-driven user models [4], text segmentation can detect a learner's weak topic areas and customize course materials to address these areas. It can also detect lack of user interest and arrange other topics for higher user engagements.

Several challenges exist to text segmentation for online courses, though. First is *sparsity*: relative to other corpuses, educational text can be rather short, especially in corporate training short-course scenarios [3]. This requires the use of domain-independent methods, i.e., those pre-trained on large libraries of text but optimized for the particular scenario. Existing domain-independent methods, however, generally perform worse their domain-dependent counterparts. The second challenge is *interpretability*: ideally, instructional designers would be able to use the results to gain insight into their courses and ways of optimizing their content [5]. Our objective in this paper is to propose a text segmentation method that overcomes these challenges.

A. Summary of Contribution

A text segmentation method should detect changes in sub-topical relationships and use these to create a mapping between topics and segments. In this work, we propose a domain-independent text segmentation method based on topic distribution and pre-trained embeddings (developed in Section. III). This method, unlike domain-dependent algorithms, does not require heavy training on prior documents, but only need to train on the current corpus of documents, instead relying on unsupervised clustering of segments based on their inferred topic compositions.

Through evaluation (presented in Section. V) on two datasets, one from an online education course and one from a popular open source corpus (described in Section. IV), we find that our method outperforms several domain-independent baselines while achieving performance comparable to a few domain-dependent algorithms. We also show that the resulting

2375-9259/18/$31.00 ©2018 IEEE
DOI 10.1109/ICDMW.2018.00053

topic distributions extracted from our method lead to superior coherence and content analytics compared with more standard NLP algorithms.

II. Related Works

A. Domain-Dependent Text Segmentation

Domain-dependent methods can be divided into three categories: word-based, topic-based, and neural network-based.

1) Word-based Text Segmentation: Several existing methods make use of lexical co-occurrence and distribution patterns [6] to divide a set of paragraphs to multi-paragrah subtopics. In [7], the algorithm consists of three steps to achieve text segmentation: tokenize paragraphs into smaller sentenced-size units, assign similarity scores (between either neighbor units or token-sequence gap) to such units, and detect subtopic boundaries based on changes in lexical patterns. However, methods based on word co-occurence and distribution may omit thesaurus relationships between terms. Thus, in this paper, we employ word embeddings, which have demonstrable improvement in representing thesaurus relationships via Euclidean distances [8].

2) Topic-based Text Segmentation: Some algorithms that segment based on identified topics, such as [9], [10], are similar to word-based segmentation methods. On the other hand, [11] calculates topic distribution on each sentence based on latent dirichlet allocation (LDA), while [12] employs latent semantic analysis to map from a high-dimensional word-document count matrix to a lower dimensional latent 'semantic' space via singular value decomposition. With topics identified, the next step is typically calculating similarities between topical distributions and finally detecting subtopic boundaries between sentences. However, topic modeling based on smaller datasets, including educational content, has proven problematic [13]. Moreover, both topic-based and word-based text segmentation algorithms, require significant amount of training data in the corresponding domain, which may not be always available.

3) Neural Network-based: Text segmentation algorithms based on neural networks have also emerged over the past decade, as neural networks do not require manual feature engineering. [14] proposed a generic end-to-end text segmentation method based on bi-directional recurrent neural networks. It significantly outperformed Domain-dependent Text Segmentation methods described above; yet, such model requires massive computation and intensive parameter training.

B. Domain-Independent Text Segmentation

While domain-dependent text segmentation methods have shown high performance on open source datasets, there are occasions when domain knowledge and training data are insufficient, resulting difficulty in calculating semantic similarity. To address this issue, [15] employs the word2vec model to represent the semantic relationships between words in vector space. Then, segmentation boundaries are automatically detected via dynamic programing, similar to [16], [17]. Also, [18] demonstrates the use of similarity measures and divisive clustering for linear text segmentation. While such methods require minimal amounts of training, performance is generally poorer when compared with domain-dependent methods in the presence of sufficient training data.

III. Text Segmentation Methodology

In this section, we specify our domain-independent text segmentation method. We consider a document $D = d_1 \cup d_2 \cup \cdots d_I$ where the d_i are *minimal content pieces* comprising the the document. The granularity at which d_i is defined can vary depending on the particular type of document: for a PDF document, each d_i may be a separate sentence, while for a slideshow they may be separate slides [3], and for a multimedia video they may be different chunks [5]. Our goal is to find subtopical boundaries between the d_i that partition the document D into a set of smaller sequences, with each small sequence representing one topic, as shown in Figure. 1.

We first describe the necessary steps to pre-process all the documents and perform feature engineering in Section. III-A. We then develop merged candidate blocks from the pre-processed inputs via distribution similarity. We finally develop sub-topical boundaries between d_i via unsupervised clustering in Section. III-B.

A. Content Representation

In this section, we discuss the feature engineering steps required for our text data d_i. Specifically, we seek vector representation v_i for each content piece that quantifies the materials covered in d_i. Our specified GloVe embeddings [1] (Global Vectors for Word Representation) and topic models produce vector representations g_i and l_i for each d_i, respectively. Afterwards, both outputted vectors are concatenated together as one long vector $v_i = [g_i^T, l_i^T]^T$ for the next step, as shown in Figure. 1.

1) Content Preprocessing: We first seek to obtain a bag-of-words representation of each d_i. Depending on the source content format, speech-to-text conversion and optical character recognition (OCR) may be required to extract a textual representation. With this in hand, we remove all punctuations and aggressively remove stop-words from the bag-of-words in d_i to focus on the important terms.

2) Word Embedding (g_i): Word embeddings capture the meaning and the semantic relationships across different contexts for words using numeric vector representations. Processing strings into numeric form promotes the analysis of textual information, as many algorithms are built based on Euclidean distances. Embedding types include frequency-based embeddings, such as TF-IDF and N-Gram, and prediction-based embeddings, our focus in this paper. Compared to frequency-based embeddings, which are usually sparse and high-dimensional, prediction-based embeddings tend to be dense and lower-dimensional, which is desirable especially in the context of educational documents where content documents can be sparse [5].

In particular, we consider GloVe embeddings. For each document d_i, we convert each pre-processed words in content

[1] https://nlp.stanford.edu/projects/glove

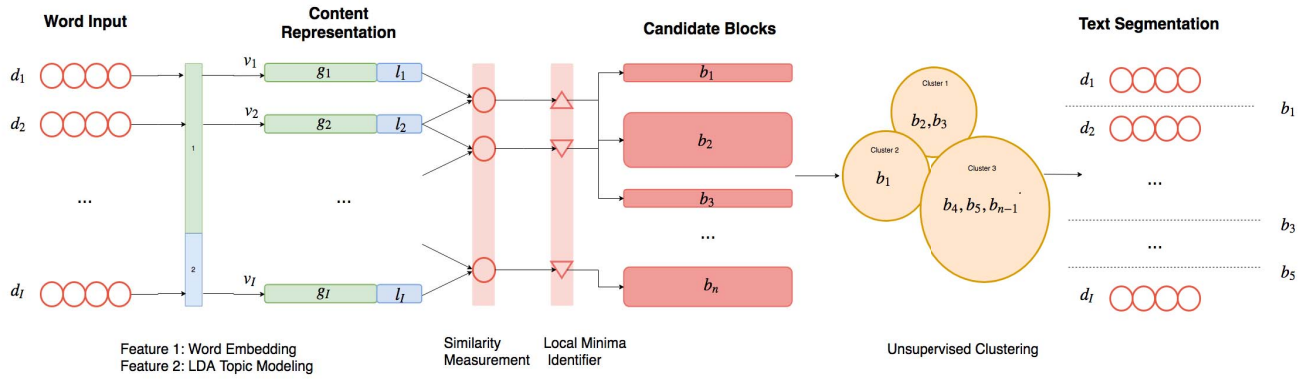

Fig. 1. Visualization of the workflow of our proposed model structure. It mainly consists of 5 main parts: feature engineering for input data, similarity measurement, local minima identifier, unsupervised clustering, and finally text boundary detection.

piece i to one-dimensional vectors of dimension K, and sum these word vectors into into one-dimensional g_i with

$$g_i = \frac{1}{N}\sum_{n=1}^{N} w_{in}, \tag{1}$$

where N is the length of content piece d_i, and w_{in} is the vector representation of n^{th} word in d_i. Previous research [19] has demonstrated that linear summed substructures well resemble the semantic relationships between words. Because of this, GloVe embeddings are capable of providing individual word meaning to our model without intensive training. However, these pre-trained embeddings cannot capture potential variation in the meaning of a word across different contexts thus using only pre-trained embeddings may not be sufficient.

3) LDA Topic Modeling (l_i): To complement pre-trained embeddings, we also consider the LDA topic distributions. LDA operates on a collection of documents via probabilistic graphical models to explain shared similarity between different parts of the documents. A document is modeled as a mixture of small number of topics with each word belongs to one of these topics. Such topic distributions help us to identify the amount of topics covered by a set of documents, as well as which topics are covered by certain documents.

In our methodology, the LDA topic distributions are inferred on a case-by-case basis, i.e., for separate documents. This technique is known to demonstrate high performance when the text corpus is large and the true segmentation is known [12]. The number of topics generated by LDA can be treated as a hyper-parameter and tuned to get the best performance in training data. Nonetheless, in real word scenarios, the amount of data required for robust modeling may not be always available. In educational training settings, for instance, short online training courses may contain only one single document or a set of documents of limited length. Hence, when faced with a small corpus or the true segmentation being unknown, we train our LDA model to optimize topic coherence value [20]. Topic coherence evaluates topic models in four main stages: segmentation, probability estimation, confirmation measure, and aggregation. Maximizing topic coherence value allows us to make optimal choices for the parameters used in LDA model, specifically, the number of topics presented.

To motivate the use of LDA and topic coherence as part of our methodology, we apply them to a "short" educational course dataset consisting of a relatively small number of documents. We will consider this dataset further for experimentation in Section IV. Figure. 2 plots the evolution of topic coherence as the number of topics is varied: We can see that the model is optimized, i.e., the coherence value is locally maximized, when the choice of number of topics is 8. With the number of topics larger than 8, the coherence score decreases significantly. This measure thus allows us to cautiously select a reasonably small number of topics, as our course consists a relatively limited set of documents, and would likely to have only a smaller number of topics.

With the choice of the number of topics set at 9, Figure. 3 visualizes the output of LDA in terms of its document-topic matrix, i.e., $p_{i,j}$ is the proportion of document i made up of topic j. These topics contain the contextual information of each document, complementary to GloVe which encapsulates semantic meanings and relationships between individual words. The next steps of our method seek to aggregate segments and topics together to promote interpretability of heatmaps like Figure. 3.

Given the choice of number of topics used in LDA model, the mixture of topics for each document, i.e., topic distributions, can be now represented by a matrix, demonstrated in Figure. 3. While GloVe covers semantic meanings and relationship for individual words, topic distribution from LDA should present the context information for each document.

4) Feature Combination: The NLP outputs g_i and l_i for each segment are concatenated together as one long vector $v_i = [g_i^T, l_i^T]^T$.

B. Segment Aggregation

The next step is to detect sub-topical boundaries between documents d_i. To do this, we first calculate similarities between v_i in order to identify variation across the text distributions. We then identify candidate blocks by locating local minima within the text distributions. The identified candidate blocks are subsequently used in unsupervised clustering. Finally, the cluster boundaries between candidate blocks determines the text segmentation between documents d_i.

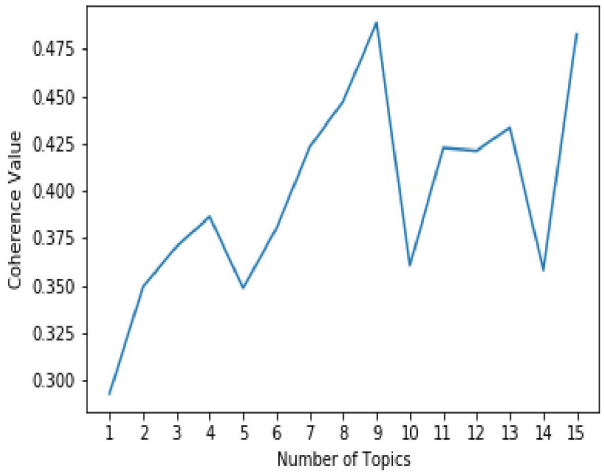

Fig. 2. Visualization of the topic coherence value across the number of topics used in LDA model. Using this metric allows us to avoid selecting a number of topics that is too high (causing overfitting) or too low (losing important variation between topics).

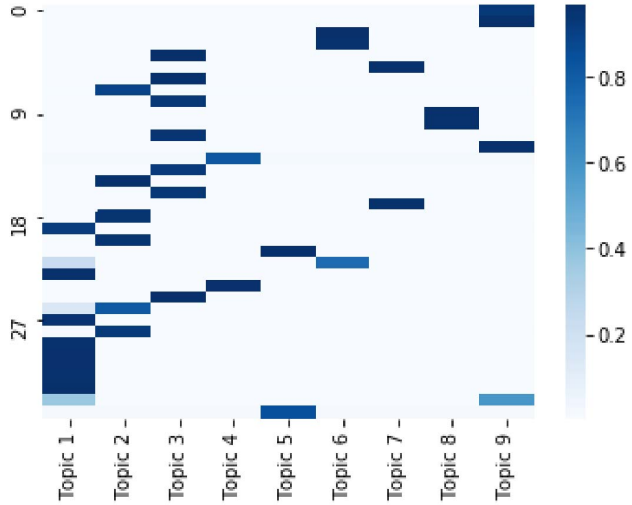

Fig. 3. Visualization of the topic distributions across video segments in the course, as inferred by LDA.We see that videos tend to cover disparate sets of topics.

1) Similarity between segments ($c_{i,i+1}$): In order to locate potential breakpoints between segments, we calculate the Wasserstein distance(WAS), also known as Earth Mover's Distance (EMD) [15], as our similarity score for each adjacent segments based on v_i. Wasserstein distance has been used in previous domain-independent text segmentation algorithms and achieved comparable performance [15]. The Wasserstein distance $c_{i,i+1}$ between segment i and segment $i+1$ is calculated as:

$$c_{i,i+1} = \inf_{\pi \in \Gamma(v_i, v_{i+1})} \int_{R \times R} |v_i - v_{i+1}| d\pi(v_i, v_{i+1}) \quad (2)$$

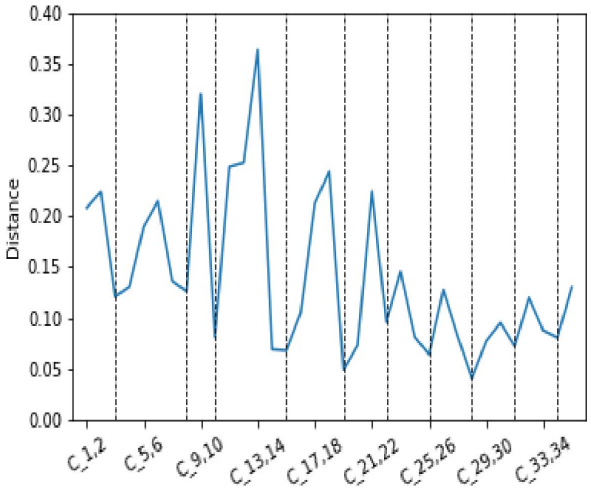

Fig. 4. Cosine Similarity calculated between adjacent segments. A large value indicates that the two adjacent segments are strongly similar, such as $c_{13,14}$. A local minimal, on the contrary, demonstrates that the two adjacent segments are significantly different in textual relationships, such as $c_{10,11}$.

where $\Gamma(v_i, v_{i+1})$ is the set of distributions on $R \times R$ whose marginals are v_i and v_{i+1} on the first and second factors respectively. In Figure. 4, we show the Wasserstein distance varies between adjacent segments the Wasserstein distance varies between. Since a local minima suggests a significant decrease of the similarity score, and consequently a substantial change of topic distribution, we use these to identify the sub-topical boundaries.

2) Candidate blocks (b_k): More specifically, we use the similarity scores $c_{i,i+1}$ to group d_i into candidate blocks $b_1, b_2, ...$, where each block b_k contains at least one segment i. We decide a breakpoint exists if $c_{i,i+1}$ is a local minima, as indicated in Figure. 1. Those local minima are utilized as possible segmentation boundaries because they can identify the most obvious semantic shift. For instance, in Figure. 4, $c_{10,11}$ reaches a global minimum of 0.06, indicating a strong change of topic information. Consequently, we insert a topical boundary between d_{10} and d_{11}. The same process is repeated for all local minima, until all segments d_i are grouped into candidate blocks: in this case, there are 10 such minima, highlighted in the plot.

3) Block Clustering: Finally, with all the candidate blocks b_k determined, we employ unsupervised clustering to refine the final boundaries for text segmentation. The groups of segments $b_1, b_2, ...$ are themselves partitioned into clusters to identify groups of segments with similar meanings.

In particular, we propose affinity propagation [21], which doesn't require the number of clusters to be determined or estimated before running the algorithm. Affinity propagation takes as input measures of similarity between pairs of data points. Real-valued messages are exchanged between data points until a final set of exemplars and corresponding clusters emerges.

	3-5	3-11	3-15	6-8	9-11	12-15
#document	130	400	100	130	130	30
avg. segment length	3.87	6.92	8.60	6.88	9.75	12.99

TABLE I. CHOI DATASET STATISTICS

IV. IMPLEMENTATION AND EXPERIMENTAL SETUP

We now turn to evaluate our text segmentation method proposed in Section. III. After specifying our datasets, evaluation metrics, and experimental setting in this section, we present and discuss our evaluation results in Section. V.

A. Datasets

1) Educational Content: We are interested in evaluating our method on educational content in online courses, as such content tends to be small with minimal prior training data. As a result, we first apply our algorithm in an online course on the topic of product development that we hosted on the ZOOMI platform[2]. This course consists of 4 sequential videos that we divide into a total of 36 segments, with each segment spanning 20 seconds; totaling less than 15 minutes, this qualifies as a short-course [3]. The video script for this course is around 700 words in length, containing 294 unique words. We apply our text segmentation algorithm to group segments into larger clusters based on semantic meaning in the video script.

2) Open source corpus: We also apply our method to rich open-source text data. Choi dataset [18] is one of the most popular public datasets for the text segmentation task [22]. It is artificially generated from the Brown corpus and consists of 920 documents. Each document is composed of ten segments, with each segment containing a certain number of sentences from different articles. These 920 documents are further divided into six groups, with each group possessing a different range of segment sizes. There are 400 documents with segment lengths of 3-11 sentences, 130 documents with sentence lengths of 3-5, 6-8, and 9-11. Additionally, the updated version also provides 100 documents with sentence lengths of 3-15, and other 30 documents with sentence lengths of 12-15.

The Choi dataset groupings are summarized in Table I, with the average segment length given in each case. The variation in segment length and document size between the six groups will be useful in evaluating the performance of our text segmentation method in sparse versus dense document environments.

B. Evaluation Metrics

Since the Choi dataset has a pre-specified segmentation of sentences, we employ P_k and WindowDiff(WD) evaluation metrics, each of which compares the true segmentation to that segmentation extracted by our algorithm. We define the true segmentation as hypothesized segmentation (Hyp), and our estimated segmentation as reference segmentation(Ref). P_k is calculated by setting a window size k to half of the average true segment size, and then accumulating penalties across a moving window where the two ends of the probe are not in the same segment [9]. Formally, P_k is determined as:

$$P_k(Hyp, Ref) = \Sigma_{1 \leq i \leq j \leq n} D_\mu(i,j)(\delta_{Ref}(i,j) \oplus \delta_{Hyp}(i,j)) \tag{3}$$

Where k is half of the average true segment size, both $\delta_{Ref}(i,j)$ and $\delta_{Hyp}(i,j)$ are indicator functions for whether sentences i and j are part of the same segment in the true segmentation. The operator between $\delta_{Ref}(i,j)$ and $\delta_{Hyp}(i,j)$ in the above formula is the XOR function. The function $D_\mu(i,j)$ is a distance probability distribution over the set of possible distances between sentences chosen randomly from the document; it generally depends on a set of parameters μ such as the average spacing between sentences.

WindowDiff(WD), on the other hand, moves a fixed-sized window across the text, and penalizes when the number of boundaries assigned by the hypothesized segmentation (Hyp) within the window does not match the true number of boundaries (Ref) for that window of text [23]. Formally, it is determined as:

$$\text{WindowDiff}_k(Hyp, Ref) = \frac{\Sigma_{i=1}^{N-k} |r(i,k) - h(i,k)|}{N-k} \tag{4}$$

Here, $r(i,k)$ represents the number of boundaries of the reference segmentation (Ref) contained between sentences i and $i+k$, whereas $h(i,k)$ represents the number of boundaries of the hypothesized segmentation (Hyp) contained between sentences i and $i+k$ (N is the number of sentences, k is the window size).

Lower P_k or lower WD_k both correspond to a higher match between the reference and hypothesized segmentations. While the results of P_k and WD_k will be highly positively correlated in general, they still convey some distinct information. The main difference between the two metrics is that P_k penalizes "false negatives" more heavily than "false positives" while overpenalizing "near misses" compared to WD_k [23].

For the educational dataset, on the other hand, the real text segmentation is unknown. Therefore, we resort to the topic coherence metric (see Figure 2) for evaluation here, to assess whether our method would output a more coherent set of topics than LDA.

C. Parameter Variation and Baselines

1) Dimension of Word Embeddings (Sec. V-C-1): We are interested in quantifying how informative the word embeddings g_i are for inferring semantic relationships between sentences and identifying segment boundaries. To evaluate the algorithm performance given varying dimension of word embeddings, we use 50d, 100d and 300d Glove word embeddings pre-trained on Wikipedia with 400K vocabularies, and 300d Glove word embeddings trained on common crawl with 2.2M vocabularies [8], to evaluate the trade-off between higher dimension embeddings and longer training time. We also apply Principal Component Analysis (PCA) to both GloVe 300d word embeddings, creating new 50d embedding matrices.

[2]https://zoomiinc.com/

2) Number of Topics (Sec. V-C-2): A main disadvantage of LDA topic modeling (and related algorithms) is that the number of topics for l_i needs to be set manually, and it is difficult to obtain an optimal value without exhaustive parameter search. To investigate the robustness of our method to the number of topics, we vary this latent dimension over 0, 5, 10, 15, 20 and 100 to evaluate the text segmentation performance on the Choi dataset (since it has a "true" segmentation)

3) Similarity measure (Sec. V-C-3): A more conventional measurement of similarity, cosine similarity, which has been used in many natural language processing tasks to represent the distance between words/sentences/segments [12]. To compare with the Wasserstein distance that we employ in our algorithm, we base the comparison on evaluation results from the Choi dataset.

4) Clustering methods (Sec. V-C-3): To merge candidate blocks into final segments, recall that we proposed affinity propagation. A more traditional way is to calculate the depth score [7], which measures the "deepness" of a minimum by looking at the highest coherence scores on the left and on the right, and then search for maxima. The depth score of a pair of segments $i, i+1$ is calculated as:

$$d_{i,i+1} = \frac{1}{2}(hl_{i,i+1} - c_{i,i+1} + hr_{i,i+1} - c_{i,i+1}) \qquad (5)$$

where $c_{i,i+1}$ is the similarity score calculated above, $hl_{i,i+1}$ and $hr_{i,i+1}$ are the highest similarity scores on the left and on the right. If the number of true segments N is given, then the N highest depth score are used as segment boundaries; otherwise, all segment pairs with $d_{i,i+1} > \mu - \sigma/2$ are used, where μ is the mean and σ is the standard variation calculated on the depth scores.

In this experiment, we compare the final performance between depth score and affinity propagation with other parameters fixed.

D. Text segmentation baselines (Sec. V-D)

To demonstrate the performance of our method, we compare it to several text segmentation baselines. For domain-independent text segmentation algorithms, we choose TextTilling [7], C99 [18], LCseg [24] and U00 [25]. And we also choose several domain-dependent algorithms: F04 [26], M09 [22] and TopicTilling [12].

V. Evaluation

A. Educational Dataset

We first present the results for the educational dataset. Figure. 5 visualizes the topic distributions for the course extracted by our text segmentation algorithm. Compared with the results from LDA presented previously in Figure. 3, the topic coherence has been increased from 0.47 to 0.61, showing a substantial improvement of about 30% in topic coherence.

Comparing the visualizations in Figure. 3 and Figure. 5, we see that our text segmentation method reduced the size of the latent dimension substantially from LDA (9 to 3). Pairs of topics in Figure 3, then, were likely similar: segments 6 through 18, for example, tended to vary between Topics 2 and 3. In Figure 5, on the other hand, most of these segments are assigned to Topic 2. Such a visualization may be more useful for content designers analyzing the progression of material through a course.

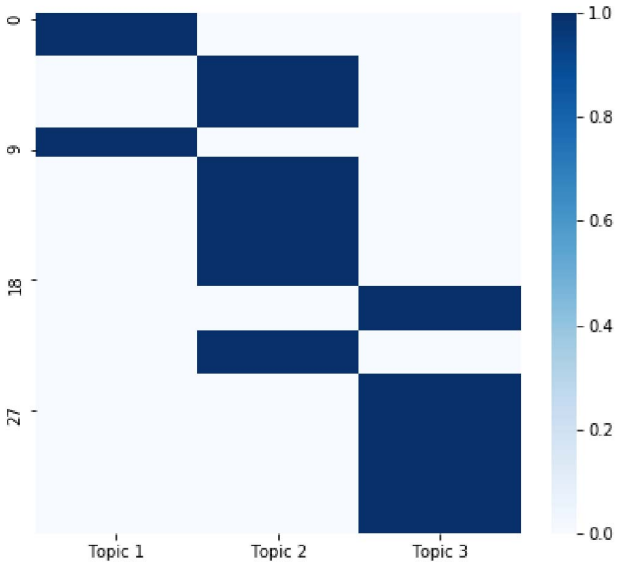

Fig. 5. Visualization of the topic distributions across video segments after inputted into algorithm. Compared to Figure. 3, We see that videos are now clustered into fewer number of topics.

B. Choi dataset

We now present our results on the open source Choi dataset. Table. II gives the obtained WD_k and P_k values for each partition of the dataset (by document segment size).

Overall, we see that the performance of our text segmentation method is reasonably robust across the different segment sizes. Across the two metrics, our method achieves optimal result when the size of segments is in range 3-11, and lowest when segment size is 3-5. This is consistent with the intuition that, a greater range should allow better segment detection, whereas a small range combined with fewer sentences should lead to worse detection rate. Interestingly, however, the error rates on the largest range 3-15 are worse than the 3-11 range. When a document has extremely small and extremely large segments, then our algorithm may lose its sensitivity to the segment boundaries. This can be further explored in future works, i.e., eliminating the impact of content length in each segment.

C. Model and Parameter Variation

1) Dimension of word embeddings: Table III shows that the influence of the dimension of word embeddings on the performance of our method for the Choi dataset. The entries indicate embedding model, and the training time is given in minutes for each. Overall, we see that embeddings with higher dimension, or embeddings trained on a larger corpus, can increase our prediction accuracy significantly: A dimension

segment size	Performance	
	WD(%)(Lowest)	*P_k(%)(Lowest)*
3-5	12.5	15.4
3-11	**4.9**	**5.1**
3-15	6.7	11.8
6-8	8.4	13.2
9-11	6.2	16.5
12-15	7.2	15.0

TABLE II. EVALUATION ON THE OPEN-SOURCE CHOI DATASET FOR DIFFERENT SEGMENT RANGES. OVERALL, OUR METHOD IS REASONABLY ROBUST TO VARYING DOCUMENT SIZES

Word Embedding Model	Performance		Runtime(m)
	WD(%) *(Lowest)*	P_k(%) *(Lowest)*	
glove-400K-50d	7.4	14.7	2.27
glove-400K-100d	7.8	14.1	2.30
glove-400K-300d	8.1	12.9	2.57
glove-400K-300d - 50d	6.2	10.8	2.39
glove-2.2M-300d	**4.9**	**5.1**	2.57
glove-2.2M-300d - 50d	7.0	11.8	2.39

TABLE III. VARYING THE DIMENSION OF WORD EMBEDDINGS ON THE CHOI DATASET. HIGHER DIMENSIONS LEAD TO SIGNIFICANT GAINS IN PERFORMANCE WITH ONLY SMALL INCREASES IN RUNTIME.

of 50 only obtains an error rate as high as 14.7%, whereas a dimension of 300 can obtain an error as low as 4.9%. Also, as the training time grows slowly with the dimension of word embeddings, enhancing the dimension from 50d to 300d will lower the error rates without sacrificing many training sources.

2) Number of topics: Table IV shows the effect of varying the number of topics in LDA on the Choi dataset. Overall, we see that varying the number of topics has minimal influence on algorithm accuracy. However, a large number of topics doubles the training time on 920 documents from around 2 minutes to 4 minutes. Hence, our text segmentation is better off when the number of topic in LDA is chosen to be smaller.

3) Similarity measurements and clustering methods: Table V shows the results for varying the similarity metric (cosine and Wasserstein) and the clustering method (depth, affinity, and spectral). Overall, we see that Wasserstein distance outperforms cosine distance for all clustering methods using the Choi dataset, whereas affinity propagation outperforms depth score method. Combining Wasserstein distance and affinity propagation decreases the error rate by roughly 81%, relative to the high of 27.1.

Even though depth score is a widely used method in text segmentation [12], but it can only merge existing candidate blocks in sequence, and it is proved that without providing the true number of segments, depth score performs much worse than providing the true number of segments. However,in most of text segmentation tasks, the number of segments in true label is always keep unknown, which constrains the application of depth score. Our unsupervised clustering methods (affinity propagation) can significantly enhance the flexibility of grouping candidate blocks based on their correlation with each other.

Number of Topics	Performance		Runtime(m)
	WD(%) (Lowest)	P_k(%) (Lowest)	
k = 0	**4.9**	**5.1**	2.39
k = 5	5.8	10.0	4.38
k = 10	4.9	5.1	3.85
k = 15	5.8	11.8	4.01
k = 20	5.8	14.1	4.27
k = 50	6.1	11.8	4.83
k = 100	5.8	14.1	5.35

TABLE IV. VARYING THE NUMBER OF TOPICS IN THE LDA MODEL. MORE TOPICS HAS NEGLIGIBLE (AND EVEN POOR) IMPACT IN PERFORMANCE WHILE ROUGHLY DOUBLING THE RUNTIME.

Distance	Clustering	
	Depth Score	Affinity Propagation
COS	27.1%	19.7%
WAS	25.9%	**5.1%**

TABLE V. VARYING DISTANCE MEASUREMENTS AND CLUSTERING METHODS ON THE CHOI DATASET FOR THE P_k METRIC. THE COMBINATION OF WASSERSTEIN AND AFFINITY PROPAGATION PERFORMS THE BEST.

D. Comparison with Baselines

Table VI compares the performance of our method against other state-of-the-art methods on the Choi dataset for the P_k metric. The methods in Group A, domain-independent methods, involve no training set, but still require specification of some hyper-parameters. Methods in Group B, on the contrary, require a full training procedure. Overall, we see that our proposed method outperforms other domain-independent approaches, while achieving performance comparable to some of the domain-dependent approaches that require training. Our domain-independent text segmentation algorithm can be easily incorporated with other domain-dependent algorithms by concatenating features from them, such as using trainable word embeddings from neural networks [14].

VI. CONCLUSION

Text Segmentation is a basis for many Natural Language Processing (NLP) tasks, such as Information Retrieval and document summarization. As these tasks are moving toward real world applications, such as personalized learning in online educational courses, this generates a need to improving the fine-granularity of text segmentation with sparse training content. While existing domain-dependent text segmentation methods have better performance in most cases, such methods may not be applicable in educational scenarios. We propose a domain-independent text segmentation method based on topic distribution and pre-trained word embeddings. The proposed algorithm represents each sentence/block with topic modeling features and word semantic features, then segments text documents by evaluating similarity between sentences/blocks. Our method does not require any hand-crafted features, domain-dependent knowledge transfer, nor lengthy training on a large corpus to gain prior knowledge. It significantly outperforms peer domain-independent text segmentation methods by lowering the error rate by 81% on Choi Dataset. We also demonstrate the use of our method on educational materials to automatically

Group	Method	P_k(%)
A	TextTilling [7]	45.25
	C99 [18]	10.50
	LCseg [24]	8.69
	U00 [25]	7.75
	Our Method	**5.08**
B	F04 [26]	4.20
	M09 [22]	2.72
	TopicTilling [12]	0.88

TABLE VI. RESULTS BASED ON THE CHOI DATASET FOR VARIOUS ALGORITHMS IN THE LITERATURE

segment sequences of documents by sub-topical relationships, with an increasing score of coherence. Such segmented documents are able to provide structural information on how course should be organized, and further become the step-stone for personalized education materials.

Moving forward, viable future work may focus on the further usage of word2vec features or other features. For example, even the pre-trained word embedding may provide extra information as they are trained using billions of words [27], training our own embedding on related corpus, such as textbooks and course materials, is still expected to improve segmentation accuracy and quality. Besides, vector representation for n-grams and phrases can also be taken into consideration [28]. In addition, we may further evaluate our method in other datasets to test the ability of generalizing, and improve the method stability when document pool contains mixture of various size. For generalization purposes, we only consider pre-trained word embeddings in this paper, but we believe previously mentioned approaches may contribute to improving the algorithms overall performance.

REFERENCES

[1] D. Beeferman, A. Berger, and J. Lafferty, "Text segmentation using exponential models," *arXiv preprint cmp-lg/9706016*, 1997.

[2] I. Manickam, A. S. Lan, and R. G. Baraniuk, "Contextual multi-armed bandit algorithms for personalized learning action selection," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 6344–6348.

[3] W. Chen, C. G. Brinton, D. Cao, and M. Chiang, "Behavior in social learning networks: early detection for online short-courses," in *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*. IEEE, 2017, pp. 1–9.

[4] A. S. Lan, A. E. Waters, C. Studer, and R. G. Baraniuk, "Sparse factor analysis for learning and content analytics," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1959–2008, 2014.

[5] W. Chen, A. Lan, D. Cao, C. Brinton, and M. Chiang, "Behavioral analysis at scale: Learning course prerequisite structures from learner clickstreams."

[6] O. Manabu and H. Takeo, "Word sense disambiguation and text segmentation based on lexical cohesion," in *Proceedings of the 15th conference on Computational linguistics-Volume 2*. Association for Computational Linguistics, 1994, pp. 755–761.

[7] M. A. Hearst, "Texttiling: Segmenting text into multi-paragraph subtopic passages," *Computational linguistics*, vol. 23, no. 1, pp. 33–64, 1997.

[8] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[9] D. Beeferman, A. Berger, and J. Lafferty, "Statistical models for text segmentation," *Machine learning*, vol. 34, no. 1-3, pp. 177–210, 1999.

[10] Q. Sun, R. Li, D. Luo, and X. Wu, "Text segmentation with lda-based fisher kernel," in *Proceedings of the 46th annual meeting of the association for computational linguistics on human language technologies: short papers*. Association for Computational Linguistics, 2008, pp. 269–272.

[11] T. Brants, F. Chen, and I. Tsochantaridis, "Topic-based document segmentation with probabilistic latent semantic analysis," in *Proceedings of the eleventh international conference on Information and knowledge management*. ACM, 2002, pp. 211–218.

[12] M. Riedl and C. Biemann, "Topictiling: a text segmentation algorithm based on lda," in *Proceedings of ACL 2012 Student Research Workshop*. Association for Computational Linguistics, 2012, pp. 37–42.

[13] J. Chang, J. Boyd-Graber, C. Wang, S. Gerrish, and D. M. Blei, "Reading tea leaves: How humans interpret topic models," in *Neural Information Processing Systems*, 2009. [Online]. Available: docs/nips2009-rtl.pdf

[14] J. Li, A. Sun, and S. Joty, "Segbot: A generic neural text segmentation model with pointer network," *IJCAI. Under Review*, 2018.

[15] M. Sakahara, S. Okada, and K. Nitta, "Domain-independent unsupervised text segmentation for data management," in *Data Mining Workshop (ICDMW), 2014 IEEE International Conference on*. IEEE, 2014, pp. 481–487.

[16] O. Heinonen, "Optimal multi-paragraph text segmentation by dynamic programming," in *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 2*. Association for Computational Linguistics, 1998, pp. 1484–1486.

[17] X. Ji and H. Zha, "Domain-independent text segmentation using anisotropic diffusion and dynamic programming," in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. ACM, 2003, pp. 322–329.

[18] F. Y. Choi, "Advances in domain independent linear text segmentation," in *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*. Association for Computational Linguistics, 2000, pp. 26–33.

[19] C. De Boom, S. Van Canneyt, T. Demeester, and B. Dhoedt, "Representation learning for very short texts using weighted word embedding aggregation," *Pattern Recognition Letters*, vol. 80, pp. 150–156, 2016.

[20] M. Röder, A. Both, and A. Hinneburg, "Exploring the space of topic coherence measures," in *Proceedings of the eighth ACM international conference on Web search and data mining*. ACM, 2015, pp. 399–408.

[21] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *science*, vol. 315, no. 5814, pp. 972–976, 2007.

[22] H. Misra, F. Yvon, J. M. Jose, and O. Cappe, "Text segmentation via topic modeling: an analytical study," in *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 2009, pp. 1553–1556.

[23] E. Kapetanios, D. Tatar, and C. Sacarea, "Natural language processing: Semantic aspects." CRC Press, Taylor and Francis, 2013, pp. 257–258.

[24] M. Galley, K. McKeown, E. Fosler-Lussier, and H. Jing, "Discourse segmentation of multi-party conversation," in *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*. Association for Computational Linguistics, 2003, pp. 562–569.

[25] M. Utiyama and H. Isahara, "A statistical model for domain-independent text segmentation," in *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2001, pp. 499–506.

[26] P. Fragkou, V. Petridis, and A. Kehagias, "A dynamic programming algorithm for linear text segmentation," *Journal of Intelligent Information Systems*, vol. 23, no. 2, pp. 179–197, 2004.

[27] E. Grave, T. Mikolov, A. Joulin, and P. Bojanowski, "Bag of tricks for efficient text classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL*, 2017, pp. 3–7.

[28] S. Vajjala and S. Banerjee, "A study of n-gram and embedding representations for native language identification," in *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, 2017, pp. 240–248.