Faceted Text Segmentation via Multitask Learning

Bei Wu, Bifan Wei, Jun Liu, Kewei Wu, and Meng Wang

***Abstract*— Text segmentation is a fundamental step in natural language processing (NLP) and information retrieval (IR) tasks. Most existing approaches do not explicitly take into account the facet information of documents for segmentation. Text segmentation and facet annotation are often addressed as separate problems, but they operate in a common input space. This article proposes FTS, which is a novel model for faceted text segmentation via multitask learning (MTL). FTS models faceted text segmentation as an MTL problem with text segmentation and facet annotation. This model employs the bidirectional long short-term memory (Bi-LSTM) network to learn the feature representation of sentences within a document. The feature representation is shared and adjusted with common parameters by MTL, which can help an optimization model to learn a better-shared and robust feature representation from text segmentation to facet annotation. Moreover, the text segmentation is modeled as a sequence tagging task using LSTM with a conditional random fields (CRFs) classification layer. Extensive experiments are conducted on five data sets from five domains: data structure, data mining, computer network, solid mechanics, and crystallography. The results indicate that the FTS model outperforms several highly cited and state-of-the-art approaches related to text segmentation and facet annotation.**

***Index Terms*— Facet annotation, long short-term memory, multitask learning (MTL), text segmentation.**

I. Introduction

A TEXT segment, regardless of whether it is explicitly marked, is defined as a sequence of clauses or sentences that display local coherence [1]. Automatically, segmenting a document into many text segments has been extensively investigated within the tasks of natural language processing (NLP) and information retrieval (IR). Most existing approaches focus on segmenting a document based on its topic: establishing boundaries in the locations, where the transition from one topic to another topic occurs [2], [3]. For example, in [4]–[6], latent Dirichlet allocation (LDA) [7] is used to exploit text semantic similarities that help to segment documents into topical segments. In this article, a topic is defined as a domain-specific term that describes a specific concept [8], which is represented as a phrase, such as binary tree, stack data structure, and linked list.

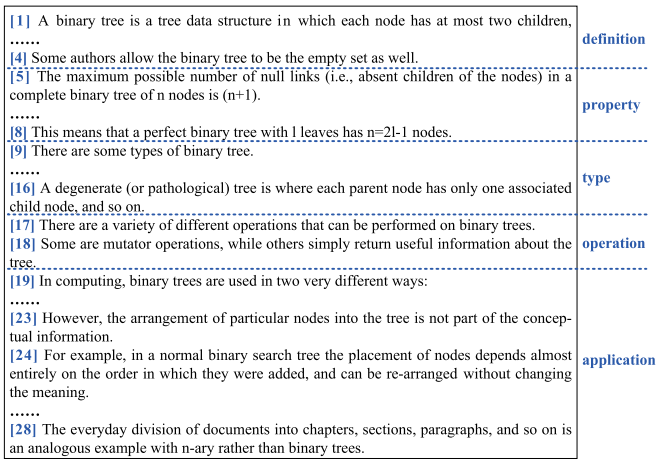

Fig. 1. Sample of document from Quora with 28 sentences, describes only one topic, binary tree in domain data structure. [·] signifies the index of a sentence.

However, a topic typically contains more fine-grained aspects or facets [9], such as definition, property, and application. For instance, as shown in Fig. 1, a document from Quora (refer to https://www.quora.com/What-is-a-binary-tree) describes only one topic: binary tree. More specifically, it describes several facets of the topic binary tree, including definition, property, type, operation, and application.

Obviously, segmenting a document using the facet granularity offers an opportunity to navigate, refine, and group the documents more effectively. If we mark each faceted segment with a facet label, users can retrieve the desired content more quickly and accurately.

In this article, we focus on faceted text segmentation which jointly carries out text segmentation for each document and facet annotation for each text segment within a document. To achieve the goal of faceted text segmentation, we confront two critical challenges.

Manuscript received April 16, 2019; revised January 22, 2020 and May 24, 2020; accepted August 8, 2020. Date of publication September 7, 2020; date of current version September 1, 2021. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB1401300 and Grant 2017YFB1401302; in part by the National Natural Science Foundation of China under Grant 61672419, Grant 61672418, Grant 61877050, and Grant 61937001; in part by MoE-CMCC "Artifical Intelligence" Project under Grant MCM20190701; in part by the Innovative Research Group of the National Natural Science Foundation of China under Grant 61721002; in part by the Project of China Knowledge Centre for Engineering Science and Technology; and in part by the Consulting Research Project of Chinese Academy of Engineering "The Online and Offline Mixed Educational Service System for 'The Belt and Road' Training in MOOC China." *(Corresponding author: Jun Liu.)*

Bei Wu and Kewei Wu are with MOEKLINNS, Department of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: wubei784872134@stu.xjtu.edu.cn; kewei_307@stu.xjtu.edu.cn).

Bifan Wei is with the National Engineering Laboratory of Big Data Analytics, School of Continuing Education, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: weibifan@xjtu.edu.cn).

Jun Liu is with the National Engineering Laboratory of Big Data Analytics, Department of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: liukeen@mail.xjtu.edu.cn).

Meng Wang is with the School of Computer Science and Engineering, Southeast University, Nanjing 211189, China (e-mail: wangmengsd@outlook.com).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2020.3015996

1) The key distinction between faceted text segmentation and most text segmentation tasks is that the former requires a facet label for each segment. Facet labels of segments are usually generated by facet annotation. These labels typically require time-consuming hand annotation, which causes poor scalability of the faceted text segmentation model.
2) Facet annotation and text segmentation are usually addressed as order as separate NLP tasks with separate training sets and disregard the dependence between these two subtasks. On the one hand, existing pipeline methods cause the error accumulation from the former subtask to the latter subtask. On the other hand, information about the latter subtask cannot be fed back to the former subtask for error correction.

Given these challenges, we propose a novel model in this article, which is named the faceted text segmentation via multitask learning (FTS) model. The main contributions are summarized as follows.

1) We are the first researchers to formalize the problem of faceted text segmentation, which represents a text as the sequences of semantically coherent segments and assigns each segment as a facet label.
2) We propose a novel model for faceted text segmentation via multitask learning (MTL)—named FTS—which simultaneously executes text segmentation as a sequence tagging task and facet annotation in an optimization model with robust text representation.
3) To validate the proposed model, exhaustive experiments are conducted on the data sets from five domains: data structure, data mining, computer network, solid mechanics, and crystallography. The experimental results show that FTS outperforms the state-of-the-art approaches and demonstrate that facet information is effective in boosting the performance of text segmentation.

The remainder of this article is organized as follows. In Section II, we review the related work of text segmentation and MTL. In Section III, we formalize the problem and describe the proposed FTS model. Section IV discusses the experimental setup and results. We give the conclusion and future work in Section V.

II. Related Work

In this article, we focus on faceted text segmentation. This work is mainly related to text segmentation, facet annotation, and MTL.

A. Text Segmentation

Automatic text segmentation has received considerable attention in NLP and IR tasks due to its effectiveness for text summarization and text indexing [10]. The approaches of text segmentation can be divided into three categories: discourse structure-based, topic-based, and graph-based.

Discourse structure-based approaches aim to utilize dependencies among discourse units of documents to divide the document into different discourse segments. Each segment reflects different intentions. The discourse structure is divided into three levels: linguistic structure, intentional structure, and attentional state [11]. Mann and Thompson [12] developed a hierarchical tree among sentences using rhetorical relations that belong to the discourse structure and recognized the leaf node to segment the documents. Galley *et al.* [13] combined content-based information with acoustic cues to detect discourse shifts for written text segmentation. Some discourse structure-based approaches also exploit lexical chain information to segment documents [1], [14], [15]. A lexical chain is a list of words, which captures a portion of the cohesive structure of text[1], in which related or similar words tend to be repeated in the coherent segments and segment boundaries often correspond to a change.

Topic-based approaches develop a heuristic and probabilistic topic model for identifying whether two sentences discuss the same topic [16], [17]. Brants *et al.* [18] employed probabilistic latent semantic analysis (pLSA) [19] to derive the latent representations of segments and segmented the documents based on the similarities of segments' latent vectors. Sun *et al.* [6] introduced an LDA-based Fisher kernel to compute words' semantic similarities and employed dynamic programming to achieve global best text segmentation. Misra *et al.* [4] proposed an LDA-based method for the text segmentation task, which jointly computes topic distributions with segmentation and enables the collection of information about the thematic content of each segment. Riedl and Biemann [5] also employed the LDA to identify topical changes within documents and segmented a document into topically similar units.

Graph-based approaches cast text segmentation in a graph-theoretic framework. Malioutov and Barzilay [20] modeled text segmentation as a graph-partitioning task aiming to simultaneously optimize the total pairwise sentence similarities within each segment and dissimilarities across various segments. Ferret [21] proposed an unsupervised method to segment documents using a similarity graph that is based on the topical similarities among text segments. Glavaš *et al.* [10] developed an unsupervised algorithm for linear text segmentation that exploited word embeddings and extended a measure of semantic relatedness to construct a semantic relatedness graph of a document. The text segmentation results are then determined by the maximal cliques of the semantic relatedness graph.

The probabilistic topic approaches and graph-based approaches can identify the probabilistic topic of each segment. The probabilistic topics, however, are clusters of similar words based on the statistics of the words in each document, which means that these words regularly appear in the documents. Facets are important aspects of a specific topic in a knowledge domain. They are rarely appear in documents [22]. Therefore, these approaches may fail to achieve the satisfactory results.

B. Facet Annotation

In facet annotation, one or more facet labels of a specific topic are assigned to a given text. Facets in IR are often referred to as subtopics, which can be represented by several keywords or clicked items [23]. Subtopic mining aims to

[1]https://en.wikipedia.org/wiki/Lexical_chain

identify possible subtopics for a given text to represent potential intents, which can be categorized into two groups: graph-based and feature-based.

Graph-based approaches focus on the dependence structure of text. Beeferman and Berger [24] treated the query log as a click-through bipartite graph, where clusters are interpreted as subtopics that address diverse queries. Yu and Ren [25] proposed to mine query subtopics by clustering on a modifier graph, which refers to an undirected weighted graph that is derived from coappearing terms with a given query. Yi *et al.* [23] finely integrated the nonnegative sparse LSA model with a subtractive clustering algorithm to mine subtopics from a search behavior tripartite graph.

Feature-based approaches mine subtopics based on the features extracted from texts. Wang *et al.* [26] proposed a subtopic-based diversification algorithm that considers the richness, importance, and novelty of concurrently mined subtopics. Ullah *et al.* [27] introduced multiple query-dependent and query-independent features for computing the balancing relevancy and novelty to mine subtopics. Song *et al.* [28] exploited the semantic composition features for the distributed representations of query subtopic candidates in query subtopic mining.

Some researchers proposed subtopic mining approaches that combine graph-based and feature-based approaches. For example, Ullah and Aono [29] proposed a bipartite graph-based ranking method to mine and rank the subtopics of a query with multiple semantic and content-aware features.

A data set does not contain obvious facet labels. Most subtopic mining approaches may become less effective in the facet annotation task and, hence, cause the degradation of the quality of facet label assignment for segments.

C. Multitask Learning

MTL is an approach to inductive transfer that improves generalization by sharing the domain information between complementary tasks. MTL uses a shared representation to learn multiple tasks—what is learned from one task can help to learn the other tasks [30]. MTL has recently been employed in many tasks, such as image classification [31], visual tracking [32], multiview action recognition [33], and text recommendation [34]. MTL can be divided into two categories: hard parameter sharing and soft parameter sharing.

Hard parameter sharing approaches comprise the most common way of MTL in neural networks, where multiple tasks are simultaneously learned based on their common parameters. Heskes *et al.* [35] proposed an MTL model to learn the feature space transformation via a maximum likelihood estimation of the weight parameters of a large network. Argyriou *et al.* [36] presented an algorithm to learn common sparse function representations across a pool of related tasks. Ranjan *et al.* [37] presented an MTL method named HyperFace based on a CNN to learn common features for simultaneously detecting faces, localizing landmarks, estimating head pose, and identifying gender.

Soft parameter sharing approaches regularize the distance between the parameters of multiple tasks to ensure the parameter similarities, where each task has separate parameters. Duong *et al.* [38] proposed a dependence parsing framework with soft parameter sharing by L2 regularization between the source and the target language. Misra *et al.* [39] proposed an end-to-end approach that is based on two separate model architectures to learn shared representations in ConvNets using a soft parameter-sharing MTL. Yang and Hospedales [40] regularized the parameters from multiple models by a tensor trace norm to simultaneously train multiple neural networks.

Text segmentation and facet annotation are often addressed as separate problems, but they are the parts of text processing that operates in a common input space. Intuitively, more accurate facet annotation will have better text segmentation, and more accurate text segmentation will have better facet annotation for each segment. When relationships between the subtasks to learn exist, simultaneously learning all subtasks instead of following the conventional pipeline approach, in which each subtask is independently learned, can be advantageous. MTL with common parameters can help an optimization model with automatic error discovery to learn a better-shared and robust feature representation shared from text segmentation to facet annotation. We strategically design the network architecture based on MTL to simultaneously learn text segmentation and facet annotation.

III. Methodology

We begin with a formal definition and a brief overview of the FTS model and then describe each module.

A. Problem Definition

This work aims to establish boundaries, where the transition from one facet to another facet occurs, namely, represents a document as the sequences of semantically coherent segments and assigns each segment a facet label.

Given the set of documents $D = \{d_1, d_2, \ldots, d_i, \ldots, d_n\}$, $d_i = (s_1^i, s_2^i, \ldots, s_j^i, \ldots, s_m^i)$ denotes the ith document, where s_j^i is the jth sentence within document d_i. $F = \{f_1, f_2, \ldots, f_u, \ldots, f_p\}$ denotes the candidate facet label set. Faceted text segmentation can be modeled as a multitask problem, in which two tasks are related.

1) *Text Segmentation:* The aim of text segmentation is to split document $d_i \in D$ into q $(q \leq p)$ segments, usually the blocks of sentences.
   $d_i \rightarrow d_i' = (\text{sg}_1^i, \text{sg}_2^i, \ldots, \text{sg}_k^i, \ldots, \text{sg}_q^i)$, and $\text{sg}_k^i = (s_{b_{k-1}+1}^i, s_{b_{k-1}+2}^i, \ldots, s_{b_k}^i)$, where b_k indexes the kth boundary sentence.
2) Facet Annotation: Based on the results of the text segmentation d_i', we must assign exactly one facet label z_k^i from the facet label set F to each segment sg_k^i within d_i'. We generate $(z_1^i, z_2^i, \ldots, z_k^i, \ldots, z_q^i)$ for each $d_i' = (\text{sg}_1^i, \text{sg}_2^i, \ldots, \text{sg}_k^i, \ldots, \text{sg}_q^i)$. A pair (sg_k^i, z_k^i) can consist of segment sg_k^i within document d_i' and target facet label $z_k^i \in F$.

B. Framework

In this article, we construct a model–the FTS model–which consists of three modules, as shown in Fig. 2.

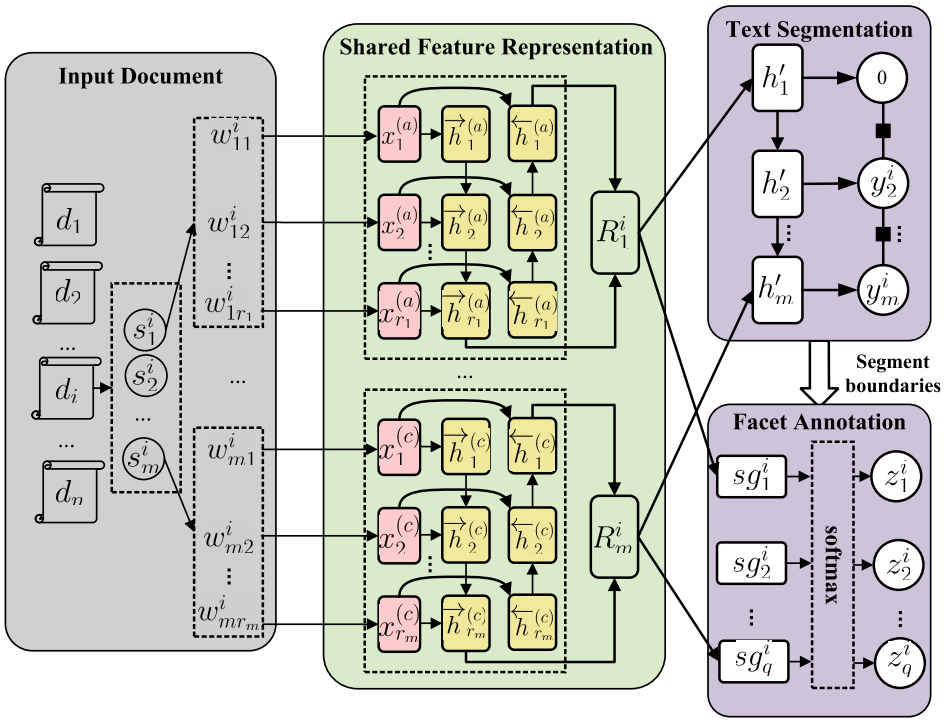

Fig. 2. Schematic framework of the FTS model. We utilize the facet information generated by facet annotation to improve text segmentation via adjusting the shared feature representation.

1) *Shared Feature Representation:* The segmentation model considers a list of tokenized sentences $d_i = (s_1^i, s_2^i, \ldots, s_j^i, \ldots, s_m^i)$ as the input. Each sentence s_j^i within document d_i is embedded into the feature vector R_j^i via a bidirectional long short-term memory (Bi-LSTM). The Bi-LSTM learns robust feature representation shared from text segmentation to facet annotation.
2) *Text Segmentation:* We treat the text segmentation task as a sequence tagging task due to the sequential characteristics of sentences. The feature vector R_j^i that corresponds to sentence s_j^i within document d_i is fed into an LSTM to extract the sequence information of sentence s_j^i. Considering the global level of a sentence within a document instead of individual positions of sentences, we adopt the conditional random fields (CRFs) to utilize the neighbor sentence tag information and predict the current sentence conditional tag.
3) *Facet Annotation:* We concatenate the feature vectors $\{R_j^i \mid b_{k-1}+1 \le j \le b_k\}$ of sentences $\{s_j^i \mid b_{k-1}+1 \le j \le b_k\}$ which belong to the same segment sg_k^i based on text segmentation and feed into a softmax layer to obtain the final facet label $z_k^i \in F$ for the segment sg_k^i. Therefore, the facet labels of segments can be automatically generated by facet annotation.

We simultaneously learn text segmentation and facet annotation in an optimization model with automatic error discovery. In the training process, we adjust the shared feature representation from text segmentation to facet annotation.

C. Shared Feature Representation

The Bi-LSTM [41] has been applied and obtained great achievements in a variety of NLP tasks, such as sequence tagging [42], [43], named entity recognition [44], [45], and relation classification [46], [47]. A Bi-LSTM maintains a memory based on history and future information, which can predict the current output conditioned on long distance features.

In this article, we employ the Bi-LSTM proposed by Graves *et al.* [48] as the representation learner to scan both left to right and right to left, which captures the left and right context from sentence s_j^i of document d_i as follows:

$$\overrightarrow{h}_t = \mathcal{H}\left(W_{x\overrightarrow{h}} x_t + W_{\overrightarrow{h}\overrightarrow{h}} \overrightarrow{h}_{t-1} + b_{\overrightarrow{h}}\right) \quad (1)$$

$$\overleftarrow{h}_t = \mathcal{H}\left(W_{x\overleftarrow{h}} x_t + W_{\overleftarrow{h}\overleftarrow{h}} \overleftarrow{h}_{t+1} + b_{\overleftarrow{h}}\right). \quad (2)$$

1) x_t is the input vector of the tth word within sentence s_j^i, where $t \in (1, r_j)$, and r_j is the number of words within sentence s_j^i.
2) $\overrightarrow{h}_t$ denotes the forward hidden sequence of the t-th position, and $\overleftarrow{h}_t$ denotes the backward hidden sequence of tth position.
3) W_{xh} denotes the input-hidden weight matrix, W_{hh} denotes the hidden-hidden weight matrix, and b_h is the hidden bias vector.
4) $\mathcal{H}$ is the hidden layer function, which is usually an element-wise application of a sigmoid function.

As a result, we concatenate the final hidden vectors of the forward and backward LSTM network as the final output

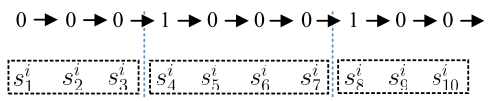

Fig. 3. Example of text segmentation.

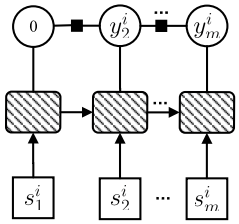

Fig. 4. LSTM-CRF for text segmentation.

feature vector R_j^i of sentence s_j^i

$$R_j^i = \overrightarrow{h}_{r_j} \oplus \overleftarrow{h}_1. \tag{3}$$

$\oplus$ refers to the concatenation operation of two vectors. R_j^i is fed into the text segmentation and concatenated with other sentence feature vectors within the same segment to be the input of the facet annotation.

D. Text Segmentation

Text segmentation represents a document as the sequences of semantically coherent segments, which can be treated as a sequence tagging task. An example is shown in Fig. 3. In document d_i, $y_j^i = 0$ means that s_j^i and s_{j-1}^i belong to the same segment and $y_j^i = 1$ otherwise.

Considering that the Boolean tag of the current sentence is related to its former sentence, we employ a forward LSTM that can efficiently utilize the past temporal features. We focus on the document level instead of an individual position of a sentence, thus leading to a CRF model. Therefore, to model the tag dependence, we combine an LSTM network and a CRF network to form an LSTM-CRF model [49] that follows the sequence tagging framework outlined by Huang *et al.* [42], which is shown in Fig. 4.

We introduce the transition score matrix $\mathbf{A}$ to model the probability A_{uv} of jumping from the uth state to the vth state for a pair of consecutive time steps. For the input document feature matrix $\mathbf{R}_i = (R_1^i, R_2^i, \ldots, R_j^i, \ldots, R_m^i)$ of the input document $d_i = (s_1^i, s_2^i, \ldots, s_j^i, \ldots, s_m^i)$ with the Boolean tag sequence $\mathbf{Y}_i = (y_1^i, y_2^i, \ldots, y_j^i, \ldots, y_m^i)$, a document level score is given by the sum of the transition scores and network tagging scores

$$s(\mathbf{R}_i, \mathbf{Y}_i) = \sum_{j=1}^{m} \left(A_{y_j^i, y_{j+1}^i} + P_{ij, y_j^i} \right). \tag{4}$$

Different from the conventional CRF, P_{ij,y_j^i} is utilized to assist the transfer matrix $\mathbf{A}$ which may be underfitting or overfitting to the data. P_{ij,y_j^i} is the output score by the forward LSTM network before the CRF with the parameter θ_1 for the Boolean tag y_j^i at j-sentence within document d_i. y_j^i is 1 if the current sentence and its former sentence belong to the same segment and 0 otherwise.

According to the Boolean tag sequence, we can recognize the boundaries of segments within a document.

E. Facet Annotation

Following the results of the text segmentation, the shared feature representation $V_k^i = R_{b_{k-1}+1}^i \oplus R_{b_{k-1}+2}^i \oplus \cdots \oplus R_{b_k}^i$ of segment $\text{sg}_k^i = (s_{b_{k-1}+1}^i, s_{b_{k-1}+2}^i, \ldots, s_{b_k}^i)$ is formed based on the sentence feature representation. We acquire the segment-level facet information generated by the facet annotation.

We generate the segment feature representation V_k^i, which summarizes all information of sentences in segment sg_k^i that can be employed as features for facet annotation. We utilize a softmax layer to obtain the facet label of a segment

$$\overrightarrow{p}_k^i = \text{softmax}\left(W_s V_k^i + b_s\right) \tag{5}$$

$$z_k^i = \arg\max_{f_u \in F} \left[\overrightarrow{p}_k^i, u\right] \tag{6}$$

where W_s and b_s are the parameters of the softmax layer. $\overrightarrow{p}_k^i$ is the probabilistic vector of the candidate facet labels for segment sg_k^i. u is the index of the maximum value within vector $\overrightarrow{p}_k^i$. z_k^i is the output as the facet label from facet set F.

F. Multitask Learning

In this work, we aim to leverage the facet information to improve text segmentation. MTL simultaneously considers several tasks since the related tasks would help induce more robust feature representations. Therefore, we employ MTL to learn parameters and use the shared representations to the desired outputs.

Given the set of training documents $D = (d_1, d_2, \ldots, d_n)$, the training loss of text segmentation is

$$\mathcal{L}_1 = \frac{1}{n} \sum_{i=1}^{n} \max\left(0, |s\left(\mathbf{R}_i, \hat{\mathbf{Y}}_i, \theta_1\right)| + \Delta\left(\hat{\mathbf{Y}}_i, \mathbf{Y}_i\right) - |s(\mathbf{R}_i, \mathbf{Y}_i, \theta_1)|\right) \tag{7}$$

$$\Delta\left(\hat{\mathbf{Y}}_i, \mathbf{Y}_i\right) = \sum_{j=1}^{m} \eta \mathbf{I}\left(\hat{y}_j^i \neq y_j^i\right). \tag{8}$$

$\hat{\mathbf{Y}}_i = (\hat{y}_1^i, \hat{y}_2^i, \ldots, \hat{y}_j^i, \ldots, \hat{y}_m^i)$ is the predicted tag sequence for document $d_i \in D$, and η is the discount parameter.

$\mathbf{I}(\cdot)$ is an indicator function, which is computed as follows:

$$\mathbf{I}\left(\hat{y}_j^i \neq y_j^i\right) = \begin{cases} 1 & \hat{y}_j^i \neq y_j^i \\ 0 & \hat{y}_j^i = y_j^i. \end{cases} \tag{9}$$

We use the negative log likelihood of the correct facet labels as the training loss of the facet annotation

$$\mathcal{L}_2 = -\sum_{d_i \in D} \sum_{k=1}^{q} \sum_{j=1}^{m_k} \log p_{\hat{z}_k^i} \tag{10}$$

where $p_{\hat{z}_k^i}$ is the corresponding probabilistic that indicates that the predicted facet label $\hat{z}_k^i$ $(k = 1, 2, \ldots, q)$ is the facet label for segment $\text{sg}_k^i = (s_{b_{k-1}+1}^i, s_{b_{k-1}+2}^i, \ldots, s_{b_k}^i)$ within

document d_i $(i = 1, 2, \ldots, n)$, namely, $\hat{z}_k^i$ is the predicted facet label for each sentence within segment sg_k^i, and m_k is the sentence number of segment sg_k^i.

The parameter set of the multitask model is $\theta_2 = \{W_{x\overrightarrow{h}}, W_{\overrightarrow{h}\overrightarrow{h}}, W_{x\overleftarrow{h}}, W_{\overleftarrow{h}\overleftarrow{h}}, W_s, \theta_1, b_{\overrightarrow{h}}, b_{\overleftarrow{h}}, b_s\}$. The regularized loss function with a L2 norm term of the model is

$$\mathcal{L} = \lambda_1 \mathcal{L}_1 + \lambda_2 \mathcal{L}_2 + \frac{\lambda_3}{2}\|\theta_2\|_2^2. \tag{11}$$

The training goal is to minimize $\mathcal{L}$. λ_1, λ_2, and λ_3 are the weight parameters for the loss of the text segmentation, facet annotation, and L2 norm term, respectively. Since we focus on the text segmentation optimization in this article, λ_1 is set to 1, namely

$$\mathcal{L} = \mathcal{L}_1 + \lambda_2 \mathcal{L}_2 + \frac{\lambda_3}{2}\|\theta_2\|_2^2. \tag{12}$$

The training procedure of MTL for faceted text segmentation is shown as follows.

Training Procedure MTL for Faceted Text Segmentation

while the model does not converge
  for each epoch **do**
    for each batch **do**
      1) execute Bi-LSTM for shared feature representation: Equations (1) (2) (3)
      2) execute LSTM-CRF for text segmentation: Equation (4)
      3) execute softmax layer for facet annotation: Equations (5) (6)
      4) update parameters to minimize loss: Equation (12)
        – update parameters of softmax layer
        – update parameters of LSTM-CRF for text segmentation
        – update parameters of Bi-LSTM for shared feature representation
    end for
  end for
end while

G. Time Complexity Analysis

We apply the computational complexity as the time complexity of the model. The model considers a document that consists of a list of tokenized sentences as the input. The words of the sentence are embedded in a Bi-LSTM to learn robust sentence representations. The sentence representations within a document are fed into an LSTM with a CRF model to obtain the boundaries of the segments. The segment representation as the output of the text segmentation module is mapped to a probability distribution of the facet labels using a one-layer multilayer perceptron (MLP) followed by a softmax function. As the complexity of the learning LSTM model per weight and time step is $\mathcal{O}(1)$, the complexity of an LSTM model per time is estimated as $\mathcal{O}(W)$, where $\mathcal{W}$ is the number of weights of the LSTM model [50]. Assuming that $\overline{m}$ sentences exist within a document, $\overline{r}$ words within a sentence and $\overline{q}$ segments within a document, on average, the time complexity is $\mathcal{O}(\overline{r}(|W_1|) + \overline{m}(|W_2| + |\theta_1|) + \overline{q}|W_s|)$. $|W_1| = |W_{x\overrightarrow{h}}| + |W_{\overrightarrow{h}\overrightarrow{h}}| + |W_{x\overleftarrow{h}}| + |W_{\overleftarrow{h}\overleftarrow{h}}|$ is the number of Bi-LSTM weights in the shared feature representation module. $|W_2|$ is the number of LSTM weights, and $|\theta_1|$ is the number of CRF weights in the text segmentation module. $|W_s|$ is the number of softmax layer weights in the facet annotation module. The time complexity of this model is nearly linear to the parameter size of the model.

IV. Experiment and Results

In this section, we introduce the data sets that we use in subsection *A*. The experimental setup and results are discussed in Section IV-B and Section IV-C, respectively.

A. Data Sets and Evaluation Metrics

1) Data Sets: As the existing public data sets lack the facet annotation information, we introduce five data sets from five domains—data structure, data mining, computer network, solid mechanics, and crystallography—to evaluate the model. These five data sets with different domains and diverse sizes are automatically acquired from the online encyclopedia Wikipedia as follows.

Step 1 (Facet label extraction): We extract the headwords of clauses from contents' sections of Wikipedia article pages as shown in Fig. 5 and lemmatize the headwords as the initial facet labels. For example, the initial facet labels of topic binary tree contain definition, type, property, and so on.

Step 2 (Facet Label Propagation): We propagate initial facet labels among similar topics within a domain by using a facet label mining approach, TF-Miner, proposed in the previous work [51], to replenish the facet labels of each topic. For example, binary tree and quadtree are two similar topics in domain data structure, which are both tree data structures. We replenish facet labels operation and method for topic quadtree via facet label propagation from topic binary tree.

Step 3 (Original Facet Label Set Generation): We make a facet label union of all topics within a specific domain to generate an original facet label set of the domain. Taking domain data structure as an example, its original facet label set contains facet labels definition, type, property, method, operation, and so on.

Step 4 (Facet Label Merging): We merge some facet labels from the original facet label set by fuzzy matching. For each facet label, we give some synonyms that are taken from the WordNet [52] and merge them into a facet label. For example, operation, process, and procedure are merged into facet label operation.

Step 5 (Candidate Facet Label Set Generation): We conduct a frequency statistical analysis for the facet labels in five domains, whose results are shown in Fig. 6. We select the top 10 facet labels to generate the candidate facet label set $F =$ {*definition, application, example, property, implementation, history, operation, method, type, algorithm*}, which encompasses approximately about 83.86% facet labels of topics in five domains on average (83.16% in data structure, 86.23% in data mining, 78.41% in computer network, 86.04% in solid mechanics, and 85.48% in crystallography). The ratio $\mathcal{C}$ in a domain is computed as follows:

$$\mathcal{C} = \frac{1}{n_t}\sum_{i=1}^{n_t}\frac{|F_i \cap F|}{|F_i|} \tag{13}$$

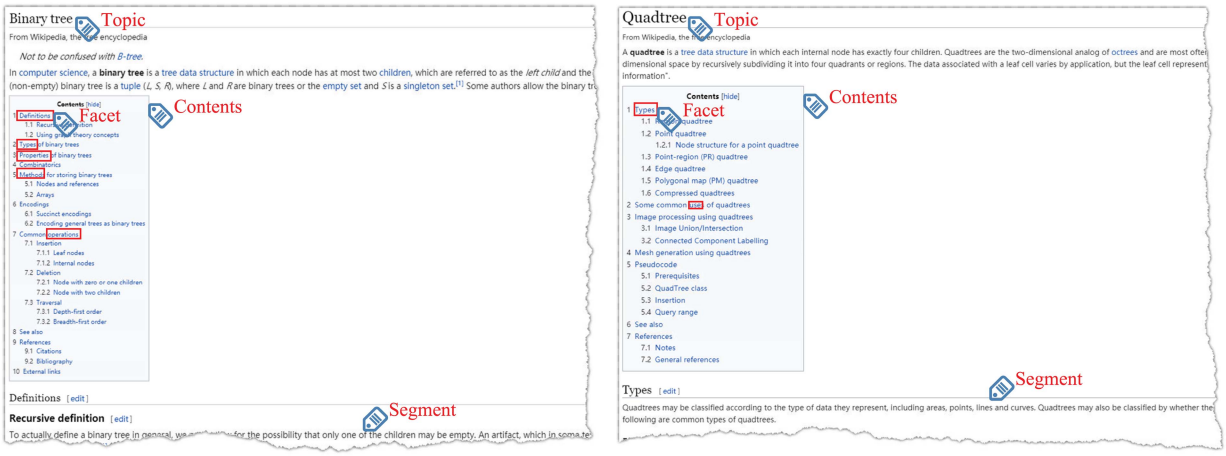

Fig. 5. Examples of the Wikipedia article pages corresponding to topics binary tree and quadtree.

TABLE I
STATISTICS OF FIVE DATA SETS

Dataset	# document	*Aver(sentences)*	*k*	# topic	*Aver(facets)*
Data Structure	48,636	9	5	193	11.25
Data Mining	23,688	15	8	94	8.27
Computer Network	21,168	12	6	84	14.63
Solid Mechanics	13,860	12	6	55	8.36
Crystallography	11,088	10	5	44	8.91

* $Aver(sentences)$ is the average sentence number of segments.
* $k = \lceil \frac{Aver(sentences)}{2} \rceil$ decides the window size in evaluation metrics.
* $Aver(facets)$ is the average facet number of topics.

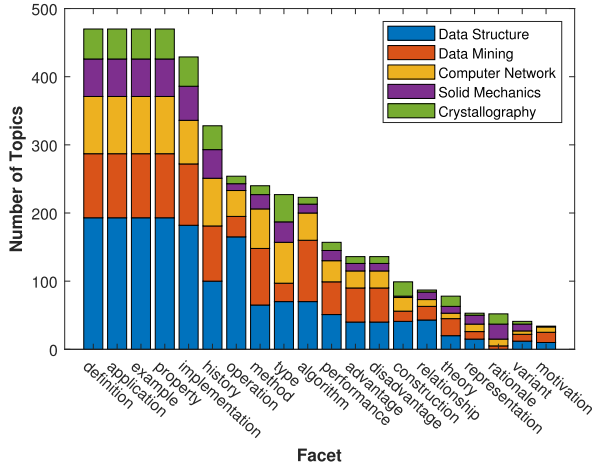

Fig. 6. Results of facet label frequency statistics.

where n_t is the number of topics, F_i is the facet label set of topic t_i, F is the candidate facet label set, and $|\cdot|$ is the number of facet labels.

Step 6 (Document Generation): We develop a parser to parse Wikipedia article pages and extract the corresponding texts linked to the facet labels as segments. We reorganize and concatenate the segments using random numbers marked with facet labels as documents.

According to the previously mentioned automatic construction method (construction code see https://github.com/xjtu-BeiWu/Text_segmentation_datagen), users can easily construct data sets without tedious facet annotation.

The statistics of the five data sets are shown in Table I. In these experiments, 70% of the data sets are used as the training set, 10% of the data sets are used as the validation set, and the remaining 20% of the data sets are used as the test set.

2) Evaluation Metrics: We evaluate the performance of the text segmentation using two evaluation metrics P_k [53] and WindowDiff (WD) [54], which are computed as follows:

$$P_k = P(\text{seg})P(\text{miss}) + (1 - P(\text{seg}))P(\text{false alarm}) \tag{14}$$

$$\text{WD}(r, h) = \frac{1}{m-k}\sum_{i=1}^{m-k}(|b(r_i, r_{i+k}) - b(h_i, h_{i+k})| > 0). \tag{15}$$

1) $P(\text{seg})$ is the prior probability that a pair of sentences with a distance of k sentences belongs to different segments.
2) $P(\text{miss})$ is the probability of missing a segment.
3) $P(\text{false alarm})$ is the probability that one more segment exists.

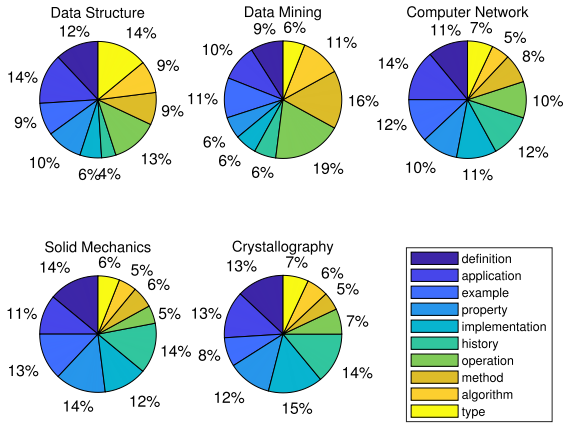

Fig. 7. Facet label distribution corresponding to the segments in five data sets.

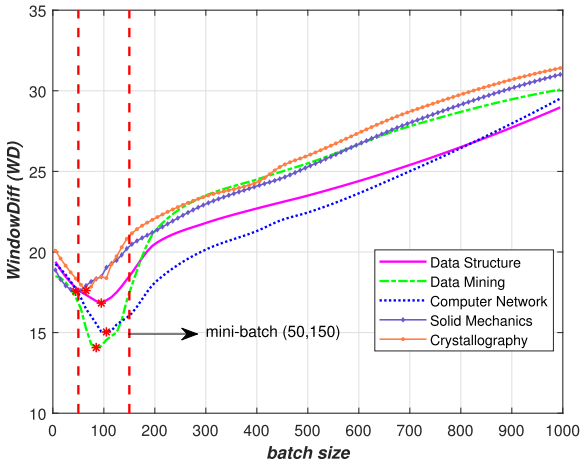

Fig. 8. Results of WD with different batch sizes.

4) k is often half of the average length of the segments [5]; the values are shown in Table I.
5) P_k is the probability that two randomly drawn sentences that are mutually k sentences apart are classified incorrectly.
6) WD is a stricter version of P_k.
7) $b(\cdot)$ represents the number of boundaries between the two parameter index position. r represents the parameter index position of the reference segmentation (ground truth) boundary, and h represents the parameter index position of the hypothetical segmentation (result predicted by the FTS model in this article) boundary.
8) m is the number of sentences within a document.

The lower values of P_k and WD indicate a better performance.

Considering the balance characteristics of the data set, as shown in Fig. 7, the Marco_$F1$ score (abbreviated as $F1$) is selected as the main metric in the evaluation of the facet annotation task, which is extensively employed in the multi-class classification task. Given a binary label representation for the facet annotation of segments, Marco_$F1$ is constructed via separately averaging with respect to examples for each label. The higher values of $F1$ indicate a better performance.

A t-test is used to determine whether the performance difference is statistically significant.

B. Experimental Setup

The 50-D vectors for word embedding are pre-trained on a 12.2G corpus from the English Wikipedia 2015 by the word2vec utility [56]. We use a learning rate of 0.1 to train the model. The training requires less than 80 epochs to converge and a few hours in general.

In each epoch, we divide the whole training data into batches and process one batch at a time. Each batch contains a list of documents which is determined by the parameter batch size. The results of WD with different batch sizes are shown in Fig. 8, which indicate that the model performs better with a mini-batch. An excessively small batch size makes the result difficult to converge in a certain number of epochs, or it needs a large epoch. An excessively large batch size makes that the target function tends to converge to sharp minima (similar to local minima), which causes the degradation of the generalization performance [57], as shown in Fig. 8. Furthermore, a larger memory capacity is needed. Considering the memory capacity, we set a batch size to 100.

The FTS model is learned with the mini-batch Adam [58] and the base learning rate $\lambda = 0.001$. The loss function $\mathcal{L}$ has two hyperparameters: λ_2 and λ_3. We set the dropout rate to 0.5. We conduct a simple grid search to estimate the values of λ_2 and λ_3 and select $\lambda_2 = 0.3$ and $\lambda_3 = 0.1$.

C. Experimental Results

1) Text Segmentation: To evaluate the effectiveness of the FTS model in the text segmentation task, we develop the following four baseline experiments.

1) *Choi's:* Choi [55] proposed a domain-independent approach for segmenting documents using a ranking scheme and cosine similarity measure.
2) *pLSA:* Brants *et al.* [18] employed pLSA to derive the latent representations of segments and segmented the documents based on the similarities of segments' latent vectors.
3) *TopicTiling:* Riedl and Biemann [5] presented an LDA-based text segmentation model, named TopicTiling. This model uses the topic model LDA to identify topical changes within documents. Topics are used to calculate the cosine similarity between two adjacent blocks of sentences.
4) *GRAPHSEG:* Glavaš *et al.* [10] constructed a semantic relatedness graph of a document and derived coherent segments from maximal cliques of the relatedness graph.

Table II shows the performance of the proposed FTS model and other baseline approaches in the text segmentation task. The FTS model outperforms all baseline approaches. Choi's [55] yields the worst performance. One reason for this result is that the direct cosine similarity measure cannot effectively calculate the relatedness between two sentences. The pLSA-based approach [18] can only recognize the semantics of the training samples instead of the semantics of the test samples, whereas the training samples and test samples

TABLE II
COMPARISONS OF THE FTS MODEL WITH BASELINE APPROACHES IN THE TEXT SEGMENTATION TASK

Approach	Data Structure		Data Mining		Computer Network		Solid Mechanics		Crystallography	
	P_k	WD	P_k	WD	P_k	WD	P_k	WD	P_k	WD
Choi's [55]	31.84	33.36	26.93	30.27	28.51	31.82	31.99	34.07	32.14	34.31
pLSA [18]	23.62	26.58	25.37	27.31	24.10	26.85	23.95	27.04	25.64	27.86
TopicTiling [5]	16.65	20.23	16.01	21.30	16.58	20.09	19.39	23.41	18.13	22.63
GRAPHSEG [10]	17.12	21.07	18.78	22.05	17.34	21.13	19.84	23.63	19.64	22.98
FTS	**13.05**	**16.93**	**12.10**	**14.29**	**12.37**	**15.18**	**16.04**	**18.56**	**15.78**	**18.20**

* FTS vs TopicTiling significantly different ($p < 0.05$)
* The numbers in bold are lowest values, which indicate the best performance.

TABLE III
COMPARISONS OF THE FTS MODEL WITH THREE APPROACHES IN THE FACET ANNOTATION TASK

Approach	Data Structure	Data Mining	Computer Network	Solid Mechanics	Crystallography
SVM [59]	79.33	79.69	79.62	78.90	78.06
CNN-static [60]	81.69	80.95	80.90	80.21	79.23
C-LSTM [61]	81.70	81.05	81.42	80.93	79.54
FTS	**82.99**	**81.18**	**82.10**	**81.27**	**80.13**

* FTS vs C-LSTM significantly different ($p < 0.05$)
* The numbers in bold are highest values of $F1$, which indicate the best performance.

in the experiment are separate. This limitation yields the unsatisfactory results in the data sets. The topic-based model TopicTiling [5] outperforms other baseline approaches. Different to the pLSA-based approach, TopicTiling can recognize the semantics of all samples, and the boost performance originates from segments' infusion.

2) Facet Annotation: To further evaluate the effectiveness of the model in the facet annotation task, we compared the FTS model with the following three classification approaches that are related to facet annotation as follows.

1) *SVM:* Zhang and Lee [59] adopted an SVM with a bag-of-words for short text classification, which is highly cited. The SVM utilizes the syntactic structure of sentences via a tree kernel that can be computed by dynamic programming. This method is employed as the baseline as it is popular and effective in sentence classification. We improve the method with efforts in the facet annotation task, which corresponds to the adaptation for multiclass classification with a one-vs-the-rest multiclass strategy. The large data set limits and hinders the ability to perform a full grid search to adjust the parameters of the SVM. We utilize LibSVM and the radial basis function kernel with the parameters $\mathcal{C} = 1.0$ and $\gamma = 0.05$ by a grid search with a random 10% of the data set.
2) *CNN-Static:* This approach, which is proposed by Kim [60], applies a simple CNN with one layer of convolution built on word vectors. The approach is a highly cited approach. Kim trained a simple CNN with one layer of convolution on the top of word vectors obtained from an unsupervised neural language model to classify sentences. We choose a CNN-static with 50-D pretrained vectors from word2vec utility [56].
3) *C-LSTM:* Zhou *et al.* [61] proposed an embedding-based approach to learn phrase-level features via a convolutional layer and learn long-term dependencies by an LSTM, which is highly cited. We choose 50-D pretrained vectors from word2vec utility [56] and set the hidden layer size to 50 for the LSTM.

The segments obtained from the text segmentation module are fed to the facet annotation module in the FTS model. To ensure the comparability of the FTS model with the three approaches, we proposed the following experimental setup.

1) *FTS:* During the training and testing process, we retain all intermediate data generated by the model. During the metric calculation process, we use a sentence as a unit.
2) *SVM/CNN-static/C-LSTM:* During the training and testing process, we use sentences as an input. During the metric calculation, we employ a sentence as a unit.

Table III shows the results, which indicate that the model outperforms the three approaches. SVM yields a worse performance than the FTS model partly as the context information of a tested sentence is not taken into account in the training and testing process of using the SVM with the bag-of-words. With a more robust feature representation adjusted by text segmentation, we surpass the embedding models—CNN-static presented in [60] and C-LSTM presented in [61]. However, the FTS model slightly increases compared with other approaches. For example, an average increase in the $F1$ score of 2.08% (computed by $82.99 - (79.33 + 81.69 + 81.70)/3$) is observed in the data set data structure. During the training and testing process, we employ the ground truth as

TABLE IV
ABLATION EXPERIMENT RESULTS

Approach		Data Structure	Data Mining	Computer Network	Solid Mechanics	Crystallography
P_k	FTS-TS	17.32	17.10	18.01	19.76	19.22
	FTS	**13.05**	**12.10**	**12.37**	**16.04**	**15.78**
WD	FTS-TS	21.27	20.09	21.15	22.76	22.32
	FTS	**16.93**	**14.29**	**15.18**	**18.56**	**18.20**
$F1$	FTS-FA	81.36	80.54	81.28	80.05	79.36
	FTS	**82.99**	**81.18**	**82.10**	**81.27**	**80.13**

* The number in bold are the lowest values of P_k and WD, or the highest value of $F1$, which indicate the best performance.
* FTS-TS is the single-task model only containing the text segmentation module.
* FTS-FA is the single-task model only containing the facet annotation module.

the input of three baseline approaches, whereas the results of the text segmentation module of FTS are applied as the input of the facet annotation module of FTS. The incorrect segments generated by the text segmentation module cause only a slight improvement in the performance using FTS compared with the three baseline approaches.

3) Ablation Experiment: To further evaluate the effectiveness of MTL, an ablation experiment is conducted between FTS and the single-task model, such as the FTS-TS model, which only contains a text segmentation module, and the FTS-FA model, which only contains a facet annotation module.

The results are shown in Table IV. The P_k scores and WD scores of the FTS model are lower than those of the FTS-TS model in five data sets, which indicates that the facet information obtained from the facet annotation module contributes to the performance improvement of the text segmentation. The $F1$ scores of the FTS model are higher than those of the FTS-FA model in five data sets, which indicates that the text segmentation module improves the performance of facet annotation.

4) Case Study: Five sample documents[2] from five domains—data structure, data mining, computer network, solid mechanics, and crystallography—are adopted for the case study. The results are shown in Fig. 9. The ref results refer to the ground truths, which are annotated manually, and the hyp results refer to the predicted results obtained by the FTS model.

This figure shows that the results of the reference and the hypothetical results are not exactly consistent. The FTS model has some misses and redundancies.

1) Segmenting the text is difficult when sentences within this text have very strong dependencies. For example, the boundary between the fourth sentence and the fifth sentence, which corresponds to the case in data structure (sample document shown in Fig. 1), and the boundary between the second sentence and the third sentence, which corresponds to the case in crystallography, are missing. A reason for this outcome is that sentences for definition and property are not easily distinguished for a topic. Another example is that a redundant boundary, which corresponds to computer network, exists between the 16th sentence and the 17th sentence. A reason for

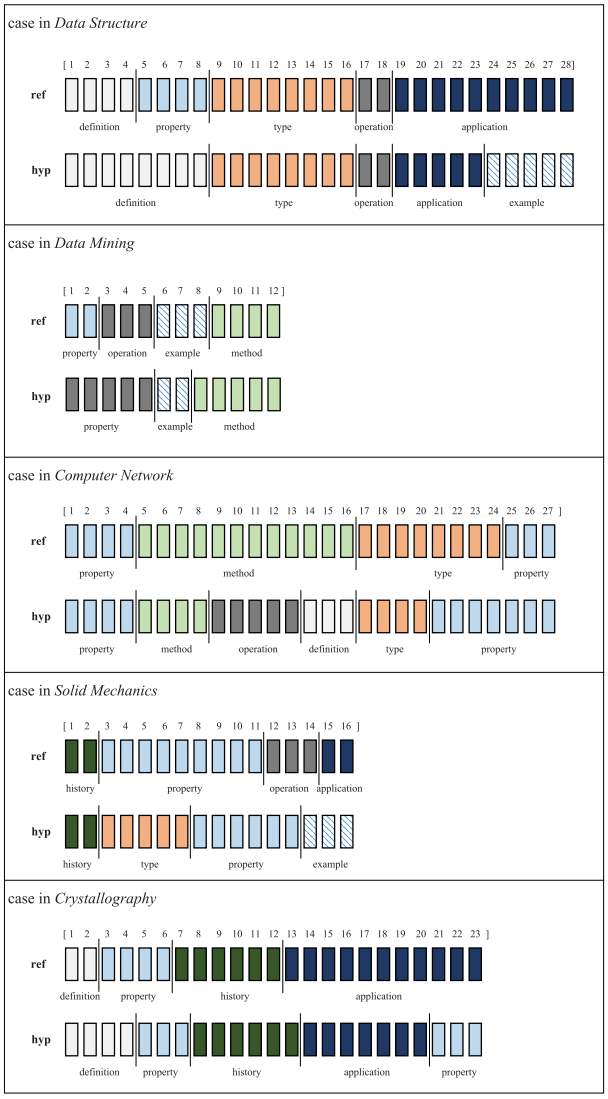

Fig. 9. Illustration of segmentation result. The boxes indicate the sentences of subdivision. Solid lines indicate that the boundaries are assigned. The ref results refer to the ground truths annotated manually, and the hyp results refer to the predicated results obtained by the FTS model.

[2]https://github.com/xjtu-BeiWu/Text_segmentation_datagen/tree/master/cases

this outcome may be that some sentences describe the property of something, which is just a type.

2) Some iconic phrases may affect the text segmentation results. For example, according to the case in data structure, the boundary between the 23rd sentence and the 24th sentence is redundant. The 24th sentence begins with the words "For example," which may produce a redundant boundary and the presence of facet label example. The same situation applies to the case of *Solid Mechanics*. In the case of data mining, a wrong boundary between the seventh sentence and the eighth sentence exists. "Another very interesting recent approach" appears at the beginning of the eighth sentence.

Although the analysis might not cover all poor cases, it sheds light on the future directions.

V. Conclusion

In this work, we propose FTS, which is a novel model that leverages facet information to improve text segmentation via MTL. The FTS model employs the Bi-LSTM learner to learn a robust feature representation that is shared from text segmentation to facet annotation. This model treats text segmentation as a sequence tagging task using an LSTM-CRF model. This model utilizes MTL to simultaneously train the text segmentation and facet annotation to implement faceted text segmentation and improve the performance of these two subtasks. The results of comprehensive experiments indicate that the FTS model outperforms the highly cited and popular approaches related to text segmentation, and the facet information is effective in boosting the performance of text segmentation.

In the future, incorporating the hierarchical facet structure and the heterogeneous facet information of different topics should be incorporated to enhance the differentiation among the different facets and minimize the semantic similarities among the different facets. In addition, we would like to introduce a hierarchical attention mechanism to assign different weights to different words and sentences to extract more critical information. This approach will enable more accurate judgments without increased overhead in the calculation and storage of the model.

References

[1] H. Kozima, "Text segmentation based on similarity between words," in *Proc. 31st Annu. Meeting Assoc. for Comput. Linguistics*, 1993, pp. 286–288.

[2] M. A. Hearst, "Text Tiling: Segmenting text into multi-paragraph subtopic passages," *Comput. Linguistics*, vol. 23, no. 1, pp. 33–64, 1997.

[3] T. Jo, "Long text segmentation by string vector based KNN," in *Proc. 19th Int. Conf. Adv. Commun. Technol. (ICACT)*, 2017, pp. 805–810.

[4] H. Misra, F. Yvon, J. M. Jose, and O. Cappe, "Text segmentation via topic modeling: An analytical study," in *Proc. 18th ACM Conf. Inf. Knowl. Manage.*, 2009, pp. 1553–1556.

[5] M. Riedl and C. Biemann, "TopicTiling: A text segmentation algorithm based on LDA," in *Proc. ACL Student Res. Workshop*, 2012, pp. 37–42.

[6] Q. Sun, R. Li, D. Luo, and X. Wu, "Text segmentation with LDA-based Fisher kernel," in *Proc. 46th Annu. Meeting Assoc. Comput. Linguistics Hum. Lang. Technol. Short Papers (HLT)*, 2008, pp. 269–272.

[7] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Jan. 2003.

[8] B. Wei, J. Liu, Q. Zheng, W. Zhang, C. Wang, and B. Wu, "DF-miner: Domain-specific facet mining by leveraging the hyperlink structure of wikipedia," *Knowl.-Based Syst.*, vol. 77, pp. 80–91, Mar. 2015.

[9] B. Zheng, W. Zhang, and X. F. B. Feng, "A survey of faceted search," *J. Web Eng.*, vol. 12, nos. 1–2, pp. 041–064, 2013.

[10] G. Glavaš, F. Nanni, and S. P. Ponzetto, *Unsupervised Text Segmentation Using Semantic Relatedness Graphs*. Stroudsburg, PA, USA; Association for Computational Linguistics, 2016.

[11] B. J. Grosz and C. L. Sidner, "Attention, intentions, and the structure of discourse," *Comput. Linguistics*, vol. 12, no. 3, pp. 175–204, 1986.

[12] W. C. Mann and S. A. Thompson, "Rhetorical structure theory: Toward a functional theory of text organization," *Text-Interdiscipl. J. Study Discourse*, vol. 8, no. 3, pp. 243–281, 1988.

[13] M. Galley, K. McKeown, E. Fosler-Lussier, and H. Jing, "Discourse segmentation of multi-party conversation," in *Proc. 41st Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2003, pp. 1–8.

[14] M. A. Hearst, "Multi-paragraph segmentation of expository text," in *Proc. 32nd Annu. Meeting Assoc. Comput. Linguistics*, 1994, pp. 9–16.

[15] M. Utiyama and H. Isahara, "A statistical model for domain-independent text segmentation," in *Proc. 39th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2001, pp. 499–506.

[16] H. Chen, S. R. K. Branavan, R. Barzilay, and D. R. Karger, "Global models of document structure using latent permutations," in *Proc. Hum. Lang. Technol., Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics (NAACL)*, 2009, pp. 371–379.

[17] O. Koshorek, A. Cohen, N. Mor, M. Rotman, and J. Berant, "Text segmentation as a supervised learning task," 2018, *arXiv:1803.09337*. [Online]. Available: http://arxiv.org/abs/1803.09337

[18] T. Brants, F. Chen, and I. Tsochantaridis, "Topic-based document segmentation with probabilistic latent semantic analysis," in *Proc. 11th Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2002, pp. 211–218.

[19] T. Hofmann, "Probabilistic latent semantic analysis," in *Proc. 15th Conf. Uncertainty Artif. Intell.* San Mateo, CA, USA: Morgan Kaufmann, 1999, pp. 289–296.

[20] I. Malioutov and R. Barzilay, "Minimum cut model for spoken lecture segmentation," in *Proc. 21st Int. Conf. Comput. Linguistics 44th Annu. Meeting (ACL)*, 2006, pp. 25–32.

[21] O. Ferret, "Finding document topics for improving topic segmentation," in *Proc. 45th Annu. Meeting Assoc. Comput. Linguistics*, 2007, pp. 480–487.

[22] B. Wu, B. Wei, J. Liu, Z. Guo, Y. Zheng, and Y. Chen, "Facet annotation by extending CNN with a matching strategy," *Neural Comput.*, vol. 30, no. 6, pp. 1647–1672, Jun. 2018.

[23] D. Yi, Y. Zhang, and B. Wei, "Query subtopic mining via subtractive initialization of non-negative sparse latent semantic analysis," *J. Inf. Sci. Eng.*, vol. 32, no. 5, pp. 1161–1181, 2016.

[24] D. Beeferman and A. Berger, "Agglomerative clustering of a search engine query log," in *Proc. 6th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2000, pp. 407–416.

[25] H.-T. Yu and F. Ren, "Subtopic mining via modifier graph clustering," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*. Cham, Switzerland: Springer, 2014, pp. 337–347.

[26] C.-J. Wang, Y.-W. Lin, M.-F. Tsai, and H.-H. Chen, "Mining subtopics from different aspects for diversifying search results," *Inf. Retr.*, vol. 16, no. 4, pp. 452–483, Aug. 2013.

[27] M. Z. Ullah, M. Shajalal, A. N. Chy, and M. Aono, "Query subtopic mining exploiting word embedding for search result diversification," in *Proc. Asia Inf. Retr. Symp.* Cham, Switzerland: Springer, 2016, pp. 308–314.

[28] W. Song, Y. Liu, L.-Z. Liu, and H.-S. Wang, "Semantic composition of distributed representations for query subtopic mining," *Frontiers Inf. Technol. Electron. Eng.*, vol. 19, no. 11, pp. 1409–1419, Nov. 2018.

[29] M. Z. Ullah and M. Aono, "A bipartite graph-based ranking approach to query subtopics diversification focused on word embedding features," *IEICE Trans. Inf. Syst.*, vol. 99, no. 12, pp. 3090–3100, 2016.

[30] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, 1997.

[31] X.-T. Yuan, X. Liu, and S. Yan, "Visual classification with multitask joint sparse representation," *IEEE Trans. Image Process.*, vol. 21, no. 10, pp. 4349–4360, Oct. 2012.

[32] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via structured multi-task sparse learning," *Int. J. Comput. Vis.*, vol. 101, no. 2, pp. 367–383, Jan. 2013.

[33] A.-A. Liu, Y.-T. Su, P.-P. Jia, Z. Gao, T. Hao, and Z.-X. Yang, "Multiple/single-view human action recognition via part-induced multitask structural learning," *IEEE Trans. Cybern.*, vol. 45, no. 6, pp. 1194–1208, Jun. 2015.

[34] T. Bansal, D. Belanger, and A. McCallum, "Ask the GRU: Multi-task learning for deep text recommendations," in *Proc. 10th ACM Conf. Recommender Syst.*, 2016, pp. 107–114.
[35] T. Heskes *et al.*, "Solving a huge number of simular tasks: A combination of multi-task learning and a hierarchical Bayesian approach," in *Proc. ICML*, vol. 15, 1998, pp. 233–241.
[36] A. Argyriou, T. Evgeniou, and M. Pontil, "Multi-task feature learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 41–48.
[37] R. Ranjan, V. M. Patel, and R. Chellappa, "HyperFace: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 1, pp. 121–135, Jan. 2019.
[38] L. Duong, T. Cohn, S. Bird, and P. Cook, "Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process.*, vol. 2, 2015, pp. 845–850.
[39] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert, "Cross-stitch networks for multi-task learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3994–4003.
[40] Y. Yang and T. M. Hospedales, "Trace norm regularised deep multi-task learning," 2016, *arXiv:1606.04038*. [Online]. Available: http://arxiv.org/abs/1606.04038
[41] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Netw.*, vol. 18, nos. 5–6, pp. 602–610, Jul. 2005.
[42] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF models for sequence tagging," 2015, *arXiv:1508.01991*. [Online]. Available: http://arxiv.org/abs/1508.01991
[43] X. Ma and E. Hovy, "End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF," 2016, *arXiv:1603.01354*. [Online]. Available: http://arxiv.org/abs/1603.01354
[44] J. P. C. Chiu and E. Nichols, "Named entity recognition with bidirectional LSTM-CNNs," 2015, *arXiv:1511.08308*. [Online]. Available: http://arxiv.org/abs/1511.08308
[45] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," 2016, *arXiv:1603.01360*. [Online]. Available: http://arxiv.org/abs/1603.01360
[46] Y. Xu, L. Mou, G. Li, Y. Chen, H. Peng, and Z. Jin, "Classifying relations via long short term memory networks along shortest dependency paths," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1785–1794.
[47] S. Zhang, D. Zheng, X. Hu, and M. Yang, "Bidirectional long short-term memory networks for relation classification," in *Proc. 29th Pacific Asia Conf. Lang., Inf. Comput.*, 2015, pp. 73–78.
[48] A. Graves, N. Jaitly, and A.-R. Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in *Proc. IEEE Workshop Autom. Speech Recognit. Understand.*, Dec. 2013, pp. 273–278.
[49] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12 pp. 2493–2537, Aug. 2011.
[50] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," 2014, *arXiv:1402.1128*. [Online]. Available: http://arxiv.org/abs/1402.1128
[51] Z. Guo, B. Wei, J. Liu, and B. Wu, "TF-Miner: Topic-specific facet mining by label propagation," in *Proc. Int. Conf. Database Syst. Adv. Appl.* Cham, Switzerland: Springer, 2019, pp. 457–460.
[52] G. A. Miller, "WordNet: A lexical database for English," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.
[53] D. Beeferman, A. Berger, and J. Lafferty, "Statistical models for text segmentation," *Mach. Learn.*, vol. 34, nos. 1–3, pp. 177–210, 1999.
[54] L. Pevzner and M. A. Hearst, "A critique and improvement of an evaluation metric for text segmentation," *Comput. Linguistics*, vol. 28, no. 1, pp. 19–36, Mar. 2002.
[55] F. Y. Choi, "Advances in domain independent linear text segmentation," in *Proc. 1st North Amer. Chapter Assoc. Comput. Linguistics Conf.*, 2000, pp. 26–33.
[56] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.
[57] N. Shirish Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. Tak Peter Tang, "On large-batch training for deep learning: Generalization gap and sharp minima," 2016, *arXiv:1609.04836*. [Online]. Available: http://arxiv.org/abs/1609.04836
[58] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980
[59] D. Zhang and W. S. Lee, "Question classification using support vector machines," in *Proc. 26th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2003, pp. 26–32.
[60] Y. Kim, "Convolutional neural networks for sentence classification," 2014, *arXiv:1408.5882*. [Online]. Available: http://arxiv.org/abs/1408.5882
[61] C. Zhou, C. Sun, Z. Liu, and F. C. M. Lau, "A C-LSTM neural network for text classification," 2015, *arXiv:1511.08630*. [Online]. Available: http://arxiv.org/abs/1511.08630

Bei Wu received the B.S. degree in computer science and technology from Xi'an Jiaotong University, Xi'an, China, in 2014, where she is currently pursuing the Ph.D. degree with the Institute of Computer Science.

Her research interests include natural language processing and data mining.

Bifan Wei received the Ph.D. degree in computer science and technology from Xi'an Jiaotong University, Xi'an, China, in 2014.

He is currently a Senior Engineer with Xi'an Jiaotong University. His research interests include web data mining, taxonomy learning, and distance education.

Jun Liu received the B.S. and Ph.D. degrees in computer science and technology from Xi'an Jiaotong University, Xi'an, China, in 1995 and 2004, respectively.

He is currently a Professor with the Department of Computer Science and Technology, Xi'an Jiaotong University. He has authored more than 70 research papers in various journals and conference proceedings. His current research focuses on data mining and text mining.

Dr. Liu has served as a Guest Editor for many technical journals, such as *Information Fusion*, the *IEEE Systems Journal*, and *Future Generation Computer Systems*. He also acted as a conference/workshop/track chair at numerous conferences.

Kewei Wu received the B.S. and M.S. degrees in computer science and technology from Xi'an Jiaotong University, Xi'an, China, in 2017 and 2020, respectively.

His research interests include visualization and virtual reality.

Meng Wang received the Ph.D. degree from Xi'an Jiaotong University, Xi'an, China, in 2018, and the B.S. degree in computing science from Sichuan University, Chengdu, China, in 2012.

He is currently a Lecturer with the School of Computer Science and Engineering, Southeast University, Nanjing, China. His research interests include knowledge graph, network embedding, SPARQL query language, and data mining.