Information Processing and Management 57 (2020) 102359

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

Candidate sentence selection for extractive text summarization

Begum Mutlu[a], Ebru A. Sezer[*,b], M. Ali Akcayol[a]

[a] Department of Computer Engineering, Gazi University, Ankara 06570, Turkey
[b] Department of Computer Engineering, Hacettepe University, Ankara 06800, Turkey

ARTICLE INFO

Keywords:
Extractive text summarization
Text summarization features
Summarization dataset
Long short-term memory

ABSTRACT

Text summarization is a process of generating a brief version of documents by preserving the fundamental information of documents as much as possible. Although most of the text summarization research has been focused on supervised learning solutions, there are a few datasets indeed generated for summarization tasks, and most of the existing summarization datasets do not have human-generated goal summaries which are vital for both summary generation and evaluation. Therefore, a new dataset was presented for abstractive and extractive summarization tasks in this study. This dataset contains academic publications, the abstracts written by the authors, and extracts in two sizes, which were generated by human readers in this research. Then, the resulting extracts were evaluated to ensure the validity of the human extract production process. Moreover, the extractive summarization problem was reinvestigated on the proposed summarization dataset. Here the main point taken into account was to analyze the feature vector to generate more informative summaries. To that end, a comprehensive syntactic feature space was generated for the proposed dataset, and the impact of these features on the informativeness of the resulting summary was investigated. Besides, the summarization capability of semantic features was experienced by using GloVe and word2vec embeddings. Finally, the use of ensembled feature space, which corresponds to the joint use of syntactic and semantic features, was proposed on a long short-term memory-based neural network model. ROUGE metrics evaluated the model summaries, and the results of these evaluations showed that the use of the proposed ensemble feature space remarkably improved the single-use of syntactic or semantic features. Additionally, the resulting summaries of the proposed approach on ensembled features prominently outperformed or provided comparable performance than summaries obtained by state-of-the-art models for extractive summarization.

1. Introduction

Today, tremendous data is available on the Internet. With its proliferation, it becomes difficult to efficiently gather the main information from this massive amount of data. Regarding the text documents, it is a complex and exhaustive process to gather and perceive the primary information from huge amount of resources in sufficient time for human-beings. Fortunately, these processes have been automatically performed by information retrieval methods for decades. However, the rise in the quantity of information causes some performance issues such as insufficient solutions and unwieldy applications of information retrieval tasks. The use of high technology machines may reduce the loss caused by these issues. However, it may cost more. As a more suitable alternative, dimension reduction can be employed in raw data to handle these issues and accelerate the implementations of these tasks. Regarding

* Corresponding author.
E-mail address: ebruakcapinarsezer@gmail.com (E.A. Sezer).

https://doi.org/10.1016/j.ipm.2020.102359
Received 4 April 2020; Received in revised form 27 June 2020; Accepted 11 July 2020

the domain of text processing, automatic text summarization is a good and highly interpretable experience for dimension reduction.

Text summarization is a process of generating a brief version of a single document or a set of documents. Automatic summarization of text documents is a challenging problem because it is highly vital to make the resulting summaries cover basic information of the source document(s) as much as possible. In literature, this problem has been studied according to two principle strategies: abstraction and extraction. In abstractive text summarization, the summarization process of humans has been imitated. People summarize documents by gathering salient information and reorganizing this information in idiosyncratic sentences. Imitating such a process lets more natural and artifact-like summaries to be generated. In abstractive text summarization, the critical concepts in the source document(s) have been determined first, and these concepts have been paraphrased with regards to the grammatical rules and constraints of corresponding natural language by natural language processing tools. This process is indeed highly challenging because of its language-dependency and semantic restrictions while paraphrasing, but still, it has been successfully applied on relatively short documents for simple tasks such as title/headline/keyword generation (Lopyrev, 2015; Nallapati, Zhou, dos Santos, Gulcehre, & Xiang, 2016; Nasar, Jaffry, & Malik, 2019), sentence compression (Knight & Marcu, 2002; Miao, Cao, Li, & Guan, 2020; Zajic, Dorr, Lin, & Schwartz, 2007) and sentence fusion (Krahmer, Marsi, & van Pelt, 2008) etc. The larger documents (or document sets), on the other hand, have been mostly summarized by extractive strategy. Here, the salient text units have been determined, and the most salient text units have been included in the summaries according to a ratio of compression. As a typical application strategy, these text units have corresponded to the sentences in the document(s). Reflecting the salient sentences to the summary has been made this kind of summarization more readable and plausible since the sentences in system summaries are grammatically correct and semantically proper human-written texts.

Several different approaches have handled the extractive text summarization process. Frequency-based term weighting approaches have been one of the preliminary studies on this area (Balabantaray, Sahoo, Sahoo, & Swain, 2012; García-Hernández & Ledeneva, 2009; Ledeneva, Gelbukh, & García-Hernández, 2008). Subsequently, the latent semantic analysis (Gong & Liu, 2001; Hachey, Murray, Reitter et al., 2005; Steinberger & Ježek, 2009), hidden markov models (Brdiczka & Chu, 2011; Conroy & O'leary, 2001), and graph-based unsupervised approaches (Aliguliyev, 2006; Fang, Mu, Deng, & Wu, 2017; Mihalcea & Tarau, 2004; Wan, 2010) have been gathered attention. More recently, it has been considered an optimization problem, and the best summary sentences that maximize the evaluation metrics have been selected for model summaries. Here, the general approach for sentence selection is selecting the most related and less redundant sentences while avoiding to convey similar information as much as possible (Alguliev, Aliguliyev, & Hajirahimova, 2012; Alguliev, Aliguliyev, Hajirahimova, & Mehdiyev, 2011a; Alguliev, Aliguliyev, & Isazade, 2013; Alguliev, Aliguliyev, & Mehdiyev, 2011b; Alguliyev, Aliguliyev, & Isazade, 2015; Alguliyev, Aliguliyev, Isazade, Abdi, & Idris, 2019).

As the state-of-the-art, text summarization is related to machine learning's classification problem. This kind of extractive text summarization method aims to determine the salience degree to sentences in the document and select the most salient sentences. Since it does not include sentence construction and paraphrasing processes, it has the advantage of language independence. To that end, extractive summarization has been gathering more attention in the literature (Allahyari et al., 2017; Nenkova & McKeown, 2012).

Simplifying the extractive text summarization, it consists of sentence scoring and sentence selection steps. In other words, a salience degree (or *summary-worthiness*) is determined for each sentence, the sentences are ranked according to their salience degree, and the most salient k sentences are selected as *summary-worthy*. These steps intrinsically associate the problem with classification problems in the machine learning field, since the determination of salience is a machine-learning problem which can be ideally handled by supervised learning.

In literature, sentence scoring for extractive text summarization has been handled by syntactic or semantic approaches. In syntactic approach, predetermined hand-crafted features for each sentence have been acquired, and considered during sentence scoring (Fattah & Ren, 2009; Ferreira et al., 2013; Goularte, Nassar, Fileto, & Saggion, 2019; Meena & Gopalani, 2014; Mutlu, Sezer, & Akcayol, 2019; 2020; Oliveira et al., 2016; Suanmali, Salim, & Binwahlan, 2009; Wan, 2010; Wang, Li, Wang, & Zheng, 2017). In the semantic approach, on the other hand, the meaning of words/phrases, and the semantic relations of them have been taken into account (Chen, Liu, Chen, & Wang, 2017; Cheng & Lapata, 2016; Denil, Demiraj, & De Freitas, 2015; Mohamed & Oussalah, 2019; Narayan, Cohen, & Lapata, 2018; Ren et al., 2018; Yin & Pei, 2015; Zhang, Lapata, Wei, & Zhou, 2018). However, humans take into account both the meaning and semantic relations of sentences and some structural properties of text during summarization. However, as much as our literature knowledge, the use of an enhanced feature space that comprehensively handles these two feature types has not been gathered remarkable attention yet.

Being one of a few studies that consider syntactic and semantic features in one respect, the authors analyzed document-dependent and document-independent features for summarization in Cao et al. (2015). While the document-independent features corresponded to sentence embeddings which carry the meaning of sentences, the dependent features substituted the syntactic features which are sentence position of the sentence, the averaged term frequency values of words in the sentence, and the averaged cluster frequency values of words in the sentence. The basing sentence scoring method was the convolutional neural network, and the experiments on a well-known benchmark dataset (DUC2002) showed that 0.366 ROUGE-1 (Recall-Oriented Understudy for Gisting Evaluation) recall could be reached.

In Nallapati, Zhai, and Zhou (2017), a Recurrent Neural Network based Sequence Model (SummaRuNNer) was proposed for extractive summarization. The SummaRuNNer is a two-layer recurrent neural network (RNN) based sequence classifier where the first layer operates at word level within each sentence, and the following layer runs over sentences for classification purposes. Bidirectional Gated Recurrent Unit (Bi-GRU) based RNN was used as the basic building block of sequence classifier. While semantically representing the words and sentences by word2vec word embeddings, two positional features were also played a contributing role in the model's decision making procedure, The ROUGE-1 recall value obtained from SummaRuNNer was 0.46 $\pm$ 0.8

on DUC2002 dataset.

Very recently, Joshi et al. represented a model for extractive text summarization based on deep auto-encoders in Joshi, Fidalgo, Alegre, and Fernández-Robles (2019). This model has been considered the sentence content relevance and sentence novelty relevance scores obtained from word and sentence embeddings and the syntactic sentence position relevance score to improve the salience of the first few sentences in the document. The ROUGE-1 score obtained from the resulting summaries on DUC2002 corpus was 0.517. Although these studies presented promising capacity in selecting the salient sentences from the text, their syntactic feature spaces were minimal and mostly relied only on the position of the sentences. However, several syntactic information can be extracted from the text, and it is still unknown, which of them affect the summarization performance in what ratio. Therefore, these existing approaches can not be considered as mature solutions yet. Here, the more important point is not the sentence scoring method itself, but what information was passed to the sentence scoring method. To that end, the input features which correspond to the *importance of the sentence* should be clearly determined and optimized to be used in sentence scoring.

In this study, the syntactic and semantic features used in extractive text summarization were deeply investigated, and their individual and joint contribution to summarization problem was analyzed by several experiments. Enhancing these two types of features, a comprehensive feature space with both syntactic and semantic summarization features were proposed, and it was shown that, when compared to their individual use, the combined use of these two types of feature spaces can contribute more according to both selecting the most informative sentences and in preserving the main information in the source document.

In automatic text summarization literature, there are few datasets created exactly for summarization task. In the optimum case, a summarization dataset has to provide proper goal summaries which are generated by a human. It is a crucial requirement for both extraction and abstraction. Although there exist plenty of text datasets in literature, a tiny portion of these datasets contain the goal summaries which are manually generated by a human. With this distinctive feature, the Document Understanding Conference (DUC) (DUC, 2007) has been the most convenient benchmark dataset for this task and employed in most of the studies on extractive summarization.

To be an alternative benchmark dataset to DUC, a summarization corpus was created and proposed in this study. The corpus was obtained from the proceedings of 41st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR, 2018). The proceedings are publicly available for academic purposes in SIGIR (2018). In the proposed corpus, the introduction section of each proceeding was considered as a source document and acquired from SIGIR (2018). Additionally, the abstracts, concepts, and keywords that were written by the original authors of publication were obtained to be possible baseline for abstractive summarization or text classification tasks, respectively. Besides these already determined text blocks, the sentences in introduction sections were manually labeled as *summary-worthy* or *summary-unworthy* by three human readers by asking them to select a subset of sentences from the source documents to reduce the original text to its 33% by preserving the basic information and the entire coherency as much as possible. As a result of this labeling process, a candidate sentence list was created for each reader, and three candidate extractive sentence sets were obtained for each document. Blending these candidate sentence sets, two extracts ($Ext_{\cup}$ and $Ext_{\cap}$) were obtained by the union and intersection of candidate sentence sets. The $Ext_{\cup}$ extracts contain the sentences which were selected as *summary-worthy* by at least one reader. The $Ext_{\cap}$ extracts, on the other hand, were obtained from the sentences which were selected as *summary-worthy* by at least two human readers. Naturally, the $Ext_{\cup}$ extracts are larger documents than $Ext_{\cap}$ extracts. To have an overall understanding of proposed corpus, statistical information was first given. Then, a ROUGE-based evaluation was performed several perspectives to validate and verify the consistency of human extracts ($Ext_{\cup}$ and $Ext_{\cap}$).

Using the new SIGIR 2018 corpus, enhanced feature space containing syntactic and semantic features were extracted, and an extensive feature space was proposed to be used in determining the salience of sentences. Additionally, a Long Short-Term Memory (LSTM)-based Neural Network (LSTM-NN) was proposed to classify the sentences as *summary-worthy* or *summary-unworthy*. This model processes the semantic and syntactic features in separate LSTMs, and combines the output vectors in a deeper layer. Then a two-layer fully connected neural network is applied for the classification of sentences. The sentences which were labeled as *summary-worthy* by this model were considered as model summaries, and these summaries were evaluated by ROUGE metrics obtained from 5-fold cross-validation. The evaluation was first based on measuring the contribution of individual and joint use of syntactic and semantic feature spaces. Here, it was observed that using the enhanced feature space significantly improves the ROUGE-values. Secondly, SummaRuNNer (Nallapati et al., 2017), and BanditSum (Dong, Shen, Crawford, van Hoof, & Cheung, 2018) were implemented to compare the informativeness of resulting summaries with the summaries of state-of-the-art deep learning methods. The obtained results showed that the LSTM-NN model fed by enhanced feature space provided more informative summaries by also including fewer sentences to the resulting summaries than SummaRuNNer, and it achieves comparable results with BanditSum, and it also outperform this baseline method basing the phrase-based assessments.

2. Objectives and contribution

In this study, the extractive text summarization was handled by three objectives based on the summarization dataset, the feature space and the methods used for determining an importance degree to sentences.

There are plenty of text documents available for numerous information retrieval or text mining tasks. However, a dataset for summarization purposes needs to fulfill some key requirements. First of all, it should include human-written documents on a specific topic. However, most of the existing datasets were obtained from short text blocks such as reviews (Hu, Chen, & Chou, 2017). Performing a summarization task on this kind of text may be considered as linguistic summarization, or opinion mining but not text summarization. Besides, a summarization dataset needs to contain the goal summaries obtained from the full text of the documents. It

is a critical need because it is not only crucial for supervised learning; there are indeed unsupervised summarization methods; however, it is also vital for the evaluation of model summaries.

Furthermore, these goal summaries should be generated by considering the type of summarization; the goal summaries should differ for abstraction and extraction. And these goal summaries should be generated by humans since humans have this expertise for this task, and models try to imitate their behaviors. However, it is a very costly process to gather all this information. Therefore, a few summarization datasets have been used in literature, and this has inevitably limited the comparative analysis of summarization methods.

To that end, a new English dataset containing the proceedings of SIGIR 2018 was proposed in this study. Since original authors have summarized the proceedings, the abstractive summaries have been naturally obtained. Additionally, the documents were read by human participants and manually labeled as *summary-worthy* or *summary-unworthy*. Blending the candidate summaries obtained from each reader, two human extracts were generated by taking the union and intersection of candidate sentence sets. Then, this manual extraction was verified by several measurements and experiments to ensure the manual labeling process of dataset production was neither random or algorithmic.

In literature, the automatic summarization methods for extraction has been considered either the syntactic features or the semantic features. Syntactic features are the hand-crafted properties of the text to be summarized or the characteristics of the corresponding sentence whose summary worthiness is being investigated. These features are predefined, and how to calculate the feature value of sentences has been predetermined by several metrics. Regarding the semantic features, the sentences are represented by embeddings that have been provided from the fusion of belonging word vectors. Regarding either the syntactic or the semantic representation of sentences, the model decides the summary worthiness of sentences, and selects the most important ones to include in summary.

Humans, on the other hand, have an intention for summarization. This intention may vary, such as generating the most informative summary by selecting the sentences with high coverage of text, generating a conclusive summary by selecting the sentences which reflect the obtained results from the reading, or generating a general summary that only contains the main topics of the text. Undoubtedly, the definition of *importance* (or the summary-worthiness) of the sentence varies according to the readers' intention. Additionally, it varies based on the characteristics of the text, such as type and writing styles. E.g., the writing style of an academic article is different from the news articles'; the e-mails, on the other hand, has a totally different structure. Hypothetically, for sentence position may be notable to define the importance of academic writing since sentences in some specific positions usually give the core information about the corresponding paragraph (Davis & Liss, 2006). On the contrary, this judgment is highly likely to be failed for an unstructured text such as blogs or e-mails where the semantic of a sentence may be the most leading property of importance. While the definition of importance varies, the human's summarization strategy varies as well according to how the reader finds the important sentence by using which properties of it. In this regard, it is necessary to take into account the importance of the sentence and all possible features of the text in determining sentence significance, which makes it insufficient to perform automatic text summarization through only syntactic features or only semantic features. In the single use of syntactic or semantic attributes, the risk arises that a group of features likely to be decisive in the summarization has been excluded. Under the circumstances, for the proposed solution to adequately address the text summary intentions and document diversity, the system must ensemble the learning channels; in other words, the utilized features. Hence, representing the sentences according to their syntactic features besides the semantics should be comprehensively investigated for the automatic text summarization process. Nevertheless, the majority of existing studies have relied either on the syntactic or the semantic features, and they have focused on improving their summarization performance basing one of these features. As much as our literature knowledge, an enhanced feature space corresponds to the semantic and syntactic information of document sentences has not been comprehensively studied yet.

In this study, a summarization procedure based on ensembled feature space was proposed. Here, the importance of the sentences was determined based on both semantic and syntactic features. To that end, LSTM-NN was proposed in this study to provide a summarization method that handles the joint of semantic and syntactic features. This model was a hierarchical structure where the first layer contains two LSTMs that simultaneously process the semantic and syntactic feature spaces, respectively. Then, the outputs of LSTMs were concatenated in a deeper layer to obtain the enhanced feature space. Subsequently, a classification task is performed to determine a degree of importance to the sentences and subsequently select the *summary-worthy* ones.

3. A new benchmark dataset for text summarization

In automatic text summarization literature, there are few datasets created exactly for summarization task. As already mentioned in the introductory section, having manually generated extracts that contain the important sentences in the source document is a crucial need for extractive text summarization. However, there exist a few dataset fulfilling this necessity. In this section, the most commonly used datasets are revisited with their advantages and weaknesses, and then the proposed dataset is introduced.

3.1. Existing summarization datasets

DUC (DUC, 2007) is a summarization dataset on news articles that are gathered from different sources for each topic. There are seven versions of this dataset from the year 2001 to 2007 in which up to 40 topics have been considered each contains a varying number of documents. The corpus provides human-written goal summaries, and in some versions (DUC2001 and DUC2002) the manually generated extracts have been presented, which correspond the documents with important sentences on the current topic. Having several documents for each news topic and containing the goal summaries compiled by considering each document on that

topic, multi-document summarization can also be employed on DUC. With these aspects, it has been as a base dataset, and most of the studies have been given their performance results considering DUC data in both abstractive and extractive summarization literature (Cao et al., 2015; Cheng & Lapata, 2016; Kumar, Salim, Abuobieda, & Albaham, 2014; Lin & Och, 2004; Nallapati et al., 2017; Ren et al., 2018; Sinha, Yadav, & Gahlot, 2018; Suanmali et al., 2009; Yin & Pei, 2015).

CL-SciSumm (Jaidka, Chandrasekaran, Rustagi, & Kan, 2016) is a summarization dataset contains academic papers on computational linguistic (CL) domain. There are three versions of this dataset entitled with their publication year as SciSumm2016-SciSumm2017-SciSumm2018 so that the data set acquired in the next version includes data from the previous version. Each version of this corpus contains several ACL Computational Linguistics research papers (as reference citation), their citing papers (approximate number of which is 10 for each reference paper) and 3 types of summaries each: abstract (the traditional self-summary of the paper written by the authors), the community summary (the collection of citation sentences called *citances*) and a human-written abstract by a trained annotator. There are 20 reference papers for SciSumm2016, 40 paper for SciSumm2017, and SciSumm2018 is the latest and largest dataset with 60 papers. Recently, the dataset was extended by having 1000 scientific papers with having 15 (in average) citation sentences as community summary in Yasunaga et al. (2019). Although these three summaries address the salient information of the papers, it may not be efficient to be used in extractive summarization. Because these three summaries contain rewritten sentences, and it is more convenient to use it for abstractive summarization. In order to apply these datasets into a machine learning approach-based extractive summarization, the extracts may be obtained from the manual abstracts basing some optimization approaches such as the extractive training in Nallapati et al. (2017) or oracle construction in Xu and Durrett (2019).

It should be also noted that within the CL-SciSumm corpus, each citance is also mapped to its referenced text in the reference paper, which provides to mark the important sentence in the reference paper. This advantage may make the extractive summarization applicable. However, these sentences may not be considered as salient for the reference paper, but the citing article. Therefore, the community summary can not be referenced for an extractive summarization as well.

It should also be noted that there surely exist some other datasets for summarization purposes such as CNN/DailyMail (Hermann et al., 2015; See, Liu, & Manning, 2017), MultiLing (Multiling community site, 2020) and WikiHow (Koupaee & Wang, 2018) data collections. Indeed, they proposed goal summaries as well. However, they have some lack of extractive summarization. More specifically, CNN/DailyMail corpus presents the highlights as goal summaries, which is more applicable for abstractive summarization. The MultiLing corpus suggests using the overview section in the document as goal summaries. This dataset may be applicable for abstractive summarization. However, there exists a more serious concern that the information in the overview section may not be obtained from the entire document. Differently from these summarization corpora, WikiHow provides the goal extracts besides the source documents containing questions and answers, and these extracts were provided from the source documents by labeling the sentences as *summary-worthy* or *summary-unworthy*. However, this labeling was not manual but an algorithmic approach that the first sentences of each answer were considered as *summary-worthy*. Because of these restrictions, the most applicable option for text summarization has been the DUC corpus, for decades.

3.2. Proposed dataset: SIGIR 2018

In this study, a new benchmark summarization dataset was presented, which contains 125 research papers in 41th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2018). The conference proceedings have been publicly available for academic purposes in SIGIR (2018). The full papers in all of the sessions and short papers in *short paper proceedings-1* have been downloaded from SIGIR (2018). The acquired data from each paper to build the entire dataset is as follows:

Table 1
A document representation sample from SIGIR 2018.

```
<INTRODUCTION
DOCREF= "p495-wu"
SESSION= "4D"
CONFERENCE= "SIGIR2018"
TITLE = <sdq>'Learning Contextual Bandits in a Non-stationary Environment'</sdq>
READER=<sdq>'Reader1-Reader2-Reader3"></sdq>
<s num="1'</sdq> paragraphID="1" sentID="1" belongs-summary="3"> Multi-armed bandit algorithms provide a principled solution to the explore exploit dilemma, which exists in many important real-world applications such as display advertisement recommender systems, and online learning to rank.</s>
<s num="2" paragraphID="1" sentID="2" belongs-summary="0"> Intuitively, bandit algorithms adaptively designate a small amount of traffic to collect user feedback in each round while improving their model estimation quality on the fly. </s>
⋮
<s num="27" paragraphID="5" sentID="5" belongs-summary="0"> However, as the change points are unknown to the algorithm ahead of time, any early or late detection of the changes can only result in an increased regret. </s>
<s num="28" paragraphID="5" sentID="6" belongs-summary="1"> More importantly, we prove that if an algorithm fails to model the changes a linear regret is inevitable.</s>
<s num="29" paragraphID="5" sentID="7" belongs-summary="2"> Extensive empirical evaluations on both a synthetic dataset and three real-world datasets for content recommendation confirmed the improved utility of the proposed algorithm, compared with both state-of-the-art stationary and non-stationary bandit algorithms. </s>
</INTRODUCTION>
```

Table 2
A sample abstract from SIGIR 2018.

```
<ABSTRACT
DOCREF="p495-wu"
SESSION="4D"
CONFERENCE="SIGIR2018"
TITLE = <sdq>'Learning Contextual Bandits in a Non-stationary Environment'</sdq>
SUMMARIZER=<sdq>'Author'</sdq>>
<s num="1" paragraphID="1" sentID="1" BelongsSummary="1"> Multi-armed bandit algorithms have become a reference solution for handling the explore/
    exploit dilemma in recommender systems, and many other important real-world problems, such as display advertisement. </s>
<s num="2" paragraphID="1" sentID="2" BelongsSummary="1"> However, such algorithms usually assume a stationary reward distribution, which hardly
    holds in practice as users preferences are dynamic. </s>
⋮
<s num="7" paragraphID="1" sentID="7" BelongsSummary="1"> Extensive empirical evaluations on both synthetic and real-world datasets for
    recommendation confirm its practical utility in a changing environment. </s>
</ABSTRACT>
```

- Source: Introduction section of papers (Table 1)
- Extract: The set of important sentences (determined by human participants) (Table 1)
- Abstract: Abstract section (written by authors) (Table 2)
- Keywords (written by authors) (Table 3)
- Concept information (selected by authors from conference submission system) (Table 4)

Besides these already determined text blocks, the sentences in introduction sections were manually labeled as *summary-worthy* or *summary-unworthy* by three human readers who have at least a bachelor degree in the computer engineering department. In this manual labeling process, it was asked from the readers to select a subset of sentences from the source documents so as to reduce the original text to its 33% by preserving the fundamental information and the entire coherency as much as possible. Here, the sentences of each document are predetermined, and each sentence has a *'belongs-summary'* attribute initially set to '0'. The readers labeled this attribute '1' from '0' if they think the sentence should be included in their summary. Inevitably, each reader is blind to the other readers' extracts.

As a result of the labeling process, a candidate sentence list was created for each reader, and three candidate extractive summaries were obtained for each document. Note that having multiple candidate extractive summaries for each paper is a considerable advantage of the proposed corpus. Because there is not a strict point of view for summarization, it is objective, and each reader generates the extracts of the papers in her/his way. To obtain the final human extracts from these three candidate sentence lists, they were blended by two strategies: union and intersection, and it resulted in two human extracts to be generated per paper as follows:

- Ext $_{\cup}$ extract: Human extracts obtained from the sentences which are selected by at least one reader
- Ext $_{\cap}$ extract: Human extracts obtained from the sentences which are selected by at least two readers

Table 1 shows a part of a source document by example. Here, the value of *belongs-summary* attribute of each sentence varied as 0 for low, 1 for medium-low, 2 for medium-high or 3 for high *summary-worthiness*. This value is the number of readers who labeled the corresponding sentence as *summary-worthy*. The two human extracts (Ext $_{\cup}$ and Ext $_{\cap}$) were built according to the value of this attribute.

Selecting the *summary-worthy* sentences from source document and generating the human extracts made the proposed SIGIR 2018 dataset applicable for extractive text summarization, which this study was focused on. However, extractive summarization is not the only possible application of SIGIR 2018. As mentioned before, the abstract sections of documents (Table 2), the keywords (Table 3) and the concept information (Table 4) were also acquired to obtain this dataset. To that end, SIGIR 2018 can be used as a benchmark dataset to apply several text mining tasks such as:

Table 3
A sample keywords document from SIGIR 2018.

```
<KEYWORDS
DOCREF="p495-wu"
SESSION="4D"
CONFERENCE="SIGIR2018"
TITLE = <sdq>'Learning Contextual Bandits in a Non-stationary Environment'</sdq>
SUMMARIZER=<sdq>'Author'</sdq>>
<s num="0" paragraphID="0" sentID="0" belongs-summary="1"> Non-stationary Bandit </s>
<s num="1" paragraphID="1" sentID="1" belongs-summary="1"> Recommender Systems </s>
<s num="2" paragraphID="2" sentID="2" belongs-summary="1"> Regret Analysis </s>
</KEYWORDS>
```

Table 4
A sample concept document from SIGIR 2018. Concepts had been determined by the authors of the proceeding via conference submission system.

```
Name of the session: 4D Recommender Systems Methods
Concept 1: Information systems → Recommender systems
Concept 2: Theory of computation → Online learning algorithms
Concept 3: Theory of computation → Regret bounds
<CONCEPTS
DOCREF="p495-wu"
SESSION="4D"
CONFERENCE="SIGIR2018"
TITLE = <sdq>'Learning Contextual Bandits in a Non-stationary Environment'</sdq>
SUMMARIZER=<sdq>'Author'</sdq>>
<s num="0" paragraphID="0" sentID="0" belongs-summary="1"> Recommender Systems Methods </s>
<s num="1" paragraphID="1" sentID="1" belongs-summary="1"> Information systems </s>
<s num="2" paragraphID="1" sentID="2" belongs-summary="1"> Recommender systems </s>
<s num="3" paragraphID="2" sentID="1" belongs-summary="1"> Theory of computation </s>
<s num="4" paragraphID="2" sentID="2" belongs-summary="1"> Online learning algorithms </s>
<s num="5" paragraphID="2" sentID="3" belongs-summary="1"> Regret bounds </s>
</CONCEPTS>
```

1. Extractive text summarization
2. Abstractive text summarization
   Automatic generation of abstracts via original abstracts of the corresponding proceeding, which is written by authors' theirselves.
   Title/headline/keyword generation by using the title, keywords, and concept information of the proceedings as the baseline.
3. Text document classification/clustering via concept document as predetermined class labels.

To obtain some statistics and observe the challenges of the proposed corpus on summarization tasks, several analyzes have been performed on its acquired and indirectly obtained data. Please remind that the corpus has 125 document sets each contain the introduction section with labeled sentences, abstracts written by the authors, keywords, and concepts determined by authors. 77 papers of document set were obtained from full papers, and the remaining documents were obtained from short papers. The statistics on vocabulary size of documents have been presented in Table 5 according to the average number of sentences and the number of distinct words (μ), and their standard deviation (σ). In total, 55,585 words appeared in the dataset (does not include stop-words). In Table 5, the source documents (the introduction section of papers), the Ext $_{\cup}$ extracts, the Ext $_{\cap}$ extracts and human-written abstracts were considered individually. In the entire dataset, each source document contains around 31 sentences and 443 distinct words, and the sentences contain 21 words (13 distinct words without stop-words) on average. The Ext $_{\cup}$ extracts have around 15 sentences (that means almost 50% of sentences have been labeled as *summary-worthy* by at least one reader) while Ext $_{\cap}$ extracts contain around 7 sentences (that corresponds 22% of number of sentences).

3.3. Dataset evaluation

In Table 6, the statistics of frequently used text summarization datasets have been presented. The corresponding datasets are CNN/DailyMail (Hermann et al., 2015; See et al., 2017), NYT (Sandhaus, 2008), DUC(DUC, 2007), TeMario (Pardo & Rino, 2003) and ScisummNet(Yasunaga et al., 2019).

As seen in Table 6, the majority of popular text summarization datasets rely on news articles or columns written in journalists writing style. Among these datasets, the CNN/DailyMail has been commonly used for text summarization because of its being one of the most comprehensive dataset for this task. It has been used especially for abstraction since the goal summaries are highlight sentences that are more suitable for abstraction. The machine learning-based extractive summarization methods, on the other hand, utilized this dataset by automatically labeling the sentences before the training procedure. To that end, they usually used a greedy

Table 5
SIGIR2018 dataset specification.

		Source documents		Ext $_{\cup}$ extracts		Ext $_{\cap}$ extracts		Abstracts	
		μ	σ	μ	σ	μ	σ	μ	σ
Number of sentences	Entire dataset	31.31	11.84	15.18	5.96	7.46	3.53	7.73	2.42
	Long papers	37.25	9.82	17.96	5.50	9.19	3.20	8.53	2.38
	Short papers	21.79	7.96	10.71	3.42	4.67	1.85	6.46	1.89
Number of distinct words	Entire dataset	443.13	160.33	250.69	93.30	132.82	61.05	115.86	32.80
	Long papers	528.48	126.67	300.60	76.00	166.03	50.48	128.95	30.70
	Short papers	306.21	103.89	170.63	54.74	79.56	31.76	94.88	24.17

Table 6
Text summarization datasets.

Dataset name	Document type	Dataset size	Document length	Summary type	Number of summary sentences	Summary length
CNN	News article	92,579	540	Manual abstract	3	37
DailyMail	News article	219,506	593	Manual abstract	3	61
NYT	News article	167,223	727	Manual abstract	5	88
DUC2002	News article	567	612	Manual extract	11 22	200 400
TeMario	Column (Portuguese)	100	613	Manual abstract Automatic extract	NA NA	193 232
ScisummNet151	Scientific paper	1000	NANA	Manual abstract Community abstract	NA 15	110 151
SIGIR 2018	Conference proceeding	125	444	Abstract Manual extract Manual extract	7 7 16	115 133 166

approach by selecting the sentences which maximize the ROUGE metrics, and then they trained their models basing these labeled sentences (Nallapati et al., 2017; Xu & Durrett, 2019).

Among the datasets on the news, DUC dataset satisfies the need for labeled sentences. It is relatively small than CNN/Dailymail. However, it has the advantage of having manually generated extracts. Please note that, in this study, it was aimed to construct an alternative dataset to DUC on another document structure. Differently from DUC, the proposed dataset focused on academic writings.

Similar to the proposed dataset, SciSummNet 2019 has been recently constructed basing the scientific papers. Here, the abstracts of the papers are considered as goal summaries. Besides, community summaries have also been gathered to obtain a second approach for goal summaries. These community summaries contain citation sentences of the source document. It is a highly comprehensive dataset on scientific papers, but it does not contain labeled sentences for extraction. To that end, SIGIR 2018 dataset is distinguished according to the labeled data provision and the structure of corresponding documents.

3.4. Summary evaluation

ROUGE is the main evaluation method for automatic summaries (Ermakova, Cossu, & Mothe, 2019; Patil Pallavi & Mane, 2014). It is based on the similarity of n-grams. N-gram corresponds a subsequence of *n* words from a text. ROUGE evaluation can be implemented for changing *n* values such as uni-grams (ROUGE-1), bi-grams (ROUGE-2), the longest common sequences (ROUGE-L) etc. In Eq. 1, the calculation of ROUGE-n score of a candidate summary has been presented where RSS refers referenced summary set; $count(\text{gram}_n)$ is the number of n-grams in the referenced summary; and $count_{\text{match}}(\text{gram}_n)$ is the maximum number of n-grams co-occurrence in a candidate summary(s) and the referenced summary.

$$\text{ROUGE-n} = \frac{\sum_{s \in RSS} \sum_{\text{gram}_n \in s} count_{\text{match}}(\text{gram}_n)}{\sum_{s \in RSS} \sum_{\text{gram}_n \in s} count(\text{gram}_n)} \quad (1)$$

As mentioned before, ROUGE-based evaluation has been performed to compare the reference summary with the automatically generated model summary. It makes it possible to observe the frequent n-grams of reference and model summaries and provides a numerical output, which is the measure of how close is the information provided by the model summaries to the information provided by the reference summaries. In this study, the automatic summaries were evaluated by ROUGE metrics by basing the human abstracts/extracts. Besides, the ROUGE-based summary evaluation was manipulated a little to measure the preserved information of source documents after summarization. In other words, the ROUGE metrics were also measured between summary-document pairs in addition to summary-summary pairs. Considering the recall values of this approach, the overlap between the source document and the corresponding summary, the magnitude of information preservation during summarization was obtained.

3.5. Validation of manually labeled extracts

To validate and verify the manually labeled extracts in the proposed SIGIR 2018 corpus, ROUGE-based evaluation was respectively adapted on the abstract, the Ext $_{\cap}$ extracts, and Ext $_{\cup}$ extracts. This evaluation was not performed between summaries of each other but the compiled summaries with the source documents. To that end, it was aimed to measure how much information was preserved after reducing the source documents to the abstract, the Ext $_{\cap}$ extracts, and Ext $_{\cup}$ extracts.

As already mentioned, the main advantage of proposed corpus is its production of 2 extract summaries (Ext $_{\cup}$ extracts and Ext $_{\cap}$ extracts) which contain human-labeled sentences as *summary-worthy* or *summary-unworthy*. Surely, this is a difficult task for

Table 7
ROUGE values of the original abstracts, manually-labeled extracts, random summaries and the resulting summaries from TextRank.

	ROUGE-1			ROUGE-2			ROUGE-L		
Summary	F	P	R	F	P	R	F	P	R
Abstract	0.343	0.660	0.237	0.116	0.261	0.076	0.161	0.413	0.149
Ext $_{\cap}$ extracts	**0.548**	1.000	*0.384*	**0.449**	0.966	*0.298*	**0.418**	1.000	*0.384*
$Random_{small}$	0.421	1.000	0.269	0.312	0.946	0.188	0.186	0.665	0.176
$TextRank_{small}$	0.519	1.000	0.353	0.437	0.980	0.283	0.381	1.000	0.353
Ext $_{\cup}$ extracts	**0.785**	1.000	*0.649*	**0.711**	0.971	*0.565*	**0.721**	1.000	*0.649*
$Random_{large}$	0.743	1.000	0.592	0.632	0.938	0.478	0.337	0.512	0.302
$TextRank_{large}$	0.749	1.000	0.600	0.700	0.984	0.545	0.670	1.000	0.600

a human to perform properly, and different people can select different sentences as *summary-worthy*. Therefore, it is highly essential to validate the labeling process of dataset collection and to ensure the manual sentence labeling was neither random nor algorithmic (selecting the sentences periodically, selecting the first sentences in each paragraph, selecting the longest sentences etc.). To that end, the obtained manual extracts were compared with randomly generated extracts and the resulting summaries of a graph-based unsupervised summarization method (TextRank).

Randomly generated extracts were obtained 5 times, and the resulting extracts were ROUGE-based evaluated by basing the entire source document. These random extracts were in different lengths, and referred as $Random_{small}$, and $Random_{large}$ where the number of sentences in $Random_{small}$ extracts was equal to the average number of sentences in Ext $_{\cap}$ extracts, and the number of sentences in $Random_{large}$ was equal to the average number of sentences in Ext $_{\cup}$ extracts. Here, the $Random_{small}$ was a subset of the $Random_{large}$. In other words, the set of sentences in the larger random extract includes the set of sentences in the smaller one.

TextRank is a graph-based ranking model for text processing (Mihalcea & Tarau, 2004). Its custom implementation about sentence ranking in text summarization is an unsupervised approach for the automated summarization of texts that can also be used to obtain the most salient text units in a document (Barrios, Lopez, Argerich, & Wachenchauzer, 2016). The algorithm applies a variation of PageRank algorithm over a graph. It produces a ranking of the elements in the graph: the most important nodes in the graph are the ones that better describe the source document. It is a broadly accepted summarization method because it allows TextRank to be applied in summarization task without the need of a training corpus or labeling (Alzuhair & Al-Dhelaan, 2019; Barrios et al., 2016; Li, Du, & Shen, 2012; Sun & Zhuge, 2018). Besides the random extracts, the extracts obtained from TextRank algorithm were used to validate the human extracts as well. Similar to differently length random extracts, the extracts of TextRank was also in small and large form.

Table 7 presents these results of experiments aim to validate the manually labeled extracts in SIGIR 2018. Regarding only the manually labeled extracts and source document abstracts, it is clear to obtain from this table that the maximum n-gram overlap to the source document has been obtained by Ext $_{\cup}$ extracts for every value of *n*. However, this success was because the Ext $_{\cup}$ extracts contain almost half of the sentences of the source document (average number of selected sentences is 15.18). The abstracts, on the other hand, could reflect the information of source document at the minimum level because there are several distinct terms and phrases that the source document does not contain while the authors' abstract does. Ext $_{\cap}$ extracts, on the other hand, serve reasonable ROUGE values considering its limitations on summary length (average number of selected sentences is 7.46).

In Table 7, the extracts were grouped (and separated by a horizontal line) by the number of sentences they have. The document abstract, Ext $_{\cap}$ extracts, $Random_{small}$ and $TextRank_{small}$ are in the same group with having around 7 sentences. On the other hand, Ext $_{\cup}$ extracts, $Random_{large}$ and $TextRank_{large}$ are in another group with around 15 sentences. Regarding these groups, it is obvious that randomly generated extracts can not provide good ROUGE values. Note that though the $Random_{large}$ has a great advantage on summary length since it labels one of two sentences as important, it can not provide close ROUGE values to its counterpart Ext $_{\cup}$ extracts. Regarding the extracts obtained from TextRank, they could provide better ROUGE values according to the corresponding random extracts. However, the human extracts still seem to be the best extracts, which means that although each reader has labeled a sentence in its way and taking into account its unique features, it can be concluded that all of them have a common denominator in the selection of important sentences.

4. Extractive text summarization on SIGIR 2018

In this study, the proposed SIGIR 2018 corpus was handled by extractive text summarization task. The problem was handled as sentence ranking and classification by basing the semantic features, syntactic features, and ensembled features.

4.1. Sentence selection for summarization based on semantic features

The semantic representation of document sentences for summarization task relies on the word embeddings (Cheng & Lapata, 2016; Denil et al., 2015; Nallapati et al., 2017; Narayan et al., 2018; Ren et al., 2018; Yin & Pei, 2015; Zhang et al., 2018). A word embedding is a vector with a specific embedding size that represents the corresponding word. In this study, pre-trained word2vec (Le & Mikolov, 2014) and GloVe (Pennington, Socher, & Manning, 2014) word embeddings were utilized in different embedding sizes

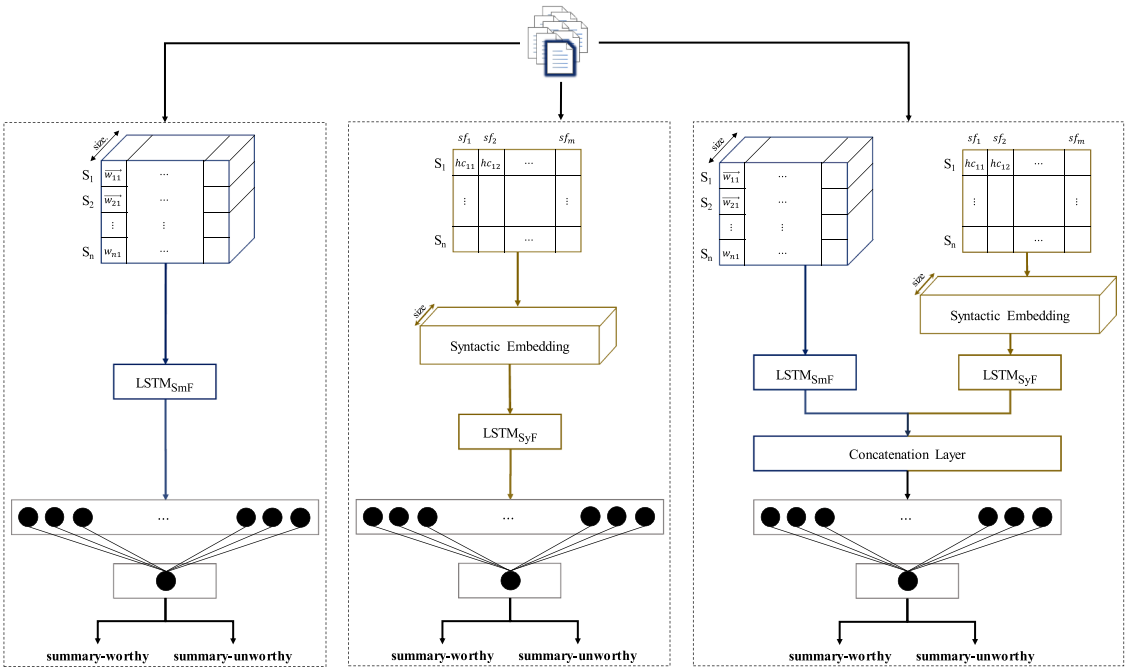


Fig. 1. LSTM-NN models for sentence selection in summarization based on (a) semantic features, (b) syntactic features, and (c) ensembled features.

(50, 100, 200, and 300). The sentences were located sequentially throughout the document, and the word embeddings of the words in the sentences were included in the input space as an extra dimension to the sentence representation (similar to the existing approaches in literature). In Fig. 1(a) the generation of this embedding-based input feature space and the classification procedure for extractive summarization task is illustrated. Here, the resulting 3-dimensional feature space was directly given to the LSTM as input, and the summary worthiness of sentences was obtained by the next layers containing a 2-layer fully connected neural network.

The sentences in documents are in different lengths. In other words, the number of words in the sentences varies, and the LSTM in this study was modeled as it accepts the feature space in a fixed length. Therefore, padding was required to fix the sequence length and locating the word vectors. The sequence length was empirically studied as (*i*) the maximum number of sentences in a sentence, (*ii*) the average number of words in a sentence, (*iii*) the minimum number of words in a sentence, and (*iv*) the sum of average and standard deviation of words in a sentence. Moreover, the padding strategy was also varied as (*a*) locating the word vectors respectively, and filling the rest of the sequence with zero vectors, (*b*) filling the empty part of the sequence first, and then locating the actual word vectors, and (*c*) locating the word vectors in the middle of the sequence, and filling the rest of the sequence with zero vectors. Among all these implementations, the use of the maximum number of words as sequence length (*i*), and as padding strategy, locating the word vectors (*a*) firstly provided the best summaries. Therefore, the rest of the experiments were performed by this approach.

Table 8 presents the 5-fold cross-validation results obtained from the LSTM-NN, which only uses the semantic feature space rely on GloVe and word2vec embeddings in different sizes (50, 100, 200, and 300) as seen in Fig. 1. Here, the value of Δ addresses the subtraction of the average number of sentences in the model summary and the average number of sentences in reference summary ($Ext_{\cap}$). It should be noted that the problem was handled by regular classification problems for every experiment, and 0.5 was used for the classification threshold to label sentences as *summary-worthy* and *summary-unworthy* automatically. Since selecting the top *k* sentences was not the main strategy for this labeling, the number of sentences in reference summary, and the model summary was different between each other. That is why the value of Δ is important. The closer this value is to zero, the better the success of the auto-generated summary. However, evaluation of the summary cannot, of course, be made only on this value. The results of ROUGE analysis were also presented in Table 8 by basing the full text of source document and the human extracts ($Ext_{\cap}$) respectively. Using these two texts as reference, it was aimed to measure (*i*) how much information of source document was preserved after summarization, and (*ii*) how close the resulting summary to the human extract. The closer the ROUGE values are to 1, the better the success of the auto-generated summary. Regarding the GloVe embeddings, while the highest ROUGE-1 values are around 0.41 according to the full documents, it reaches 0.60 when $Ext_{\cap}$ extracts were referenced. For word2vec, these values are 0.41 and 0.58, respectively.

4.2. Sentence selection for summarization based on syntactic features

One of the most popular feature vector for sentence extraction is hand-crafted features correspond to syntactic information of the

Table 8
ROUGE values obtained from the testing process of 5-fold cross-validation of LSTM-NN fed with pure GloVe and word2vec (W2V) embeddings with changing embedding sizes.

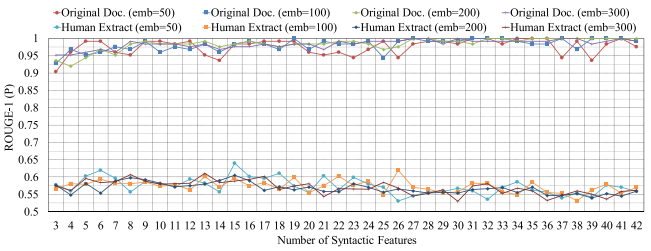
				ROUGE-1			ROUGE-2			ROUGE-L		
Reference Text	Embedding	Size	Δ	F	P	R	F	P	R	F	P	R
Source Document	GloVe	50	-	0.539	1.000	**0.385**	0.445	0.961	0.303	0.420	0.992	0.385
		100	-	0.573	1.000	**0.419**	0.481	0.968	0.336	0.459	1.000	0.419
		200	-	0.553	1.000	**0.401**	0.460	0.962	0.319	0.438	0.992	0.401
		300	-	0.572	1.000	**0.416**	0.482	0.960	0.336	0.455	0.992	0.416
	W2V	50	-	0.550	1.000	**0.396**	0.460	0.970	0.317	0.433	1.000	0.396
		100	-	0.542	1.000	**0.384**	0.449	0.969	0.304	0.419	1.000	0.384
		200	-	0.571	1.000	**0.414**	0.480	0.970	0.332	0.453	1.000	0.414
		300	-	0.546	1.000	**0.390**	0.455	0.969	0.311	0.426	1.000	0.390
$Ext_{\cap}$ extrats	GloVe	50	2.04	0.521	0.548	**0.541**	0.398	0.419	0.420	0.446	0.494	0.493
		100	2.44	2.44	0.538	**0.583**	0.416	0.412	0.466	0.463	0.490	0.536
		200	2.84	2.84	0.559	**0.574**	0.423	0.438	0.458	0.468	0.513	0.528
		300	3.48	3.48	0.558	**0.605**	0.442	0.444	0.502	0.482	0.518	0.568
	W2V	50	0.88	0.527	0.547	**0.553**	0.409	0.424	0.441	0.458	0.503	0.511
		100	1.16	1.16	0.579	**0.574**	0.442	0.462	0.468	0.489	0.536	0.534
		200	1.64	1.64	0.544	**0.583**	0.425	0.421	0.470	0.474	0.498	0.538
		300	2.92	2.92	0.581	**0.573**	0.439	0.467	0.466	0.482	0.540	0.533

sentences. These features have been widely used both by human in extracting the primary information throughout the document and by the automatic summarizers during the sentence scoring (Fattah & Ren, 2009; Ferreira et al., 2013; Goularte et al., 2019; Meena & Gopalani, 2014; Mutlu et al., 2019; 2020; Oliveira et al., 2016; Suanmali et al., 2009; Wan, 2010; Wang et al., 2017). However, which features should be determined, and how to measure the feature scores have not been standardized yet. This issue was discussed in Mutlu, Sezer, and Akcayol (2020), and it was proposed to calculate the score of each feature by several metrics. Being the most comprehensive feature space with 40 hand-crafted features, term frequency (TF), term frequency-inverse sentence frequency (TF-ISF), length of text unit, similarity/overlap with title, similarity/overlap with document keywords, relative paragraph identifier, relative sentence identifier, the position of text unit (based on source document and corresponding paragraph), sentence-sentence cohesion, bushy-path, aggregate similarity, phrasal information (the intensity of noun phrases (NP), verbal phrases (VP), prepositional phrases (PP), and adjective phrases (AP)) and sentence inclusion of name entity were taken into account for the generation of this feature space (Mutlu et al., 2020).

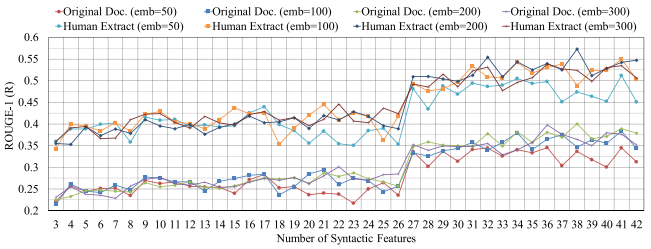
In this study, the feature space with 40 hand-crafted features was utilized, and they were extracted from the proposed SIGIR 2018 corpus by using the same metrics in Mutlu et al. (2020). How to calculate the feature score for each sentence can be found in Mutlu et al. (2020) for further reading. Here, there is a small extension to position features. in Mutlu et al. (2020), the position feature was obtained only based on the belonging paragraph. Here, on the other hand, 2 extra position features were considered by basing the sentence position in the entire document. According to this extension, the scores of 42 hand-crafted syntactic features were calculated for each sentence as follows:

- 2 feature scores for TF
- 4 feature scores for TF-ISF
- 3 feature scores for length
- 3 feature scores for title
- 1 feature score for keywords
- 1 feature score for relative paragraph identifier
- 1 feature score for relative sentence identifier
- 3 feature scores for position in entire document
- 2 feature scores for position in belonging paragraph
- 5 feature scores for sentence-sentence cohesion
- 1 feature score for bushy-path
- 1 feature score for aggregate similarity
- 12 feature scores for noun, verbal, adjective phrases and prepositions
- 3 feature scores for sentence inclusion of name entity

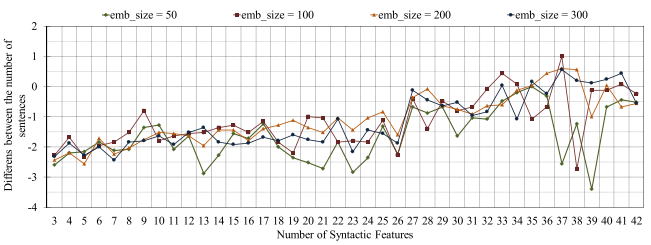
As already mentioned, the sentences in the documents were labeled as their summary worthiness during the generation of the proposed corpus. Using this labeled sentence via their syntactic features, it becomes possible to model a binary classifier. Here the LSTM-NN was employed for modeling the sentences since the sentences in paragraphs and the paragraphs in the documents can be considered as sequences. In Fig. 1(b) the classification procedure was illustrated. First, the raw form of the feature vector, containing



(a) Precision (P)



(b) Recall (R)



(c) Difference between number of sentences in model and reference summaries (Δ)

Fig. 2. ROUGE values obtained by LSTM-NN fed with hand-crafted features. The ROUGE values were obtained by basing the human Ext $_{\cap}$ extract and the source document respectively.

the feature score (sf_m) for each sentence (S_n), was transformed into a 3-dimensional embedding space, then this embedding space was given to the LSTM sequentially. Here, the reason for using an embedding layer was mostly because of making a fair comparison between the semantic-based LSTM and this syntactic-based LSTM. Because it was first aimed to compare the effect of using semantic and syntactic feature spaces, and the rest of the summarization model was aimed to be identical as much as possible. To that end, the embedding size of this layer was also varied (as 50, 100, 200, and 300), just like the variations in the embedding size of word vectors (in Section 4.1). After the embedding layer, the output of LSTM was presented to a fully connected 2-layer neural network for classification.

The experiments on the syntactic feature space were performed by changing the number of features to observe the contribution of each feature to the summarization task. The best k features were incrementally selected by a univariate statistical test based on chi-square where $3 \leq k \leq 42$.

Similar to the summary evaluation procedure in Section 4.1, the difference between the average number of sentences in automatic summaries and the average number of sentences in reference Ext $_{\cap}$ extracts were measured beside the ROUGE analysis. ROUGE analysis was performed by basing the full text of source documents, and basing the human-generated Ext $_{\cap}$ extracts because of the same motivation detailed in Section 4.1. In Fig. 2a, and b the ROUGE-1 precision and recall values obtained from the testing stage of 5-fold cross-validation were presented respectively by increasing the number of features in syntactic feature space. Here, the increase in the number of syntactic features improved the ROUGE-1 recall value which achieves the highest scores (0.57 basing the Ext $_{\cap}$ extracts, and 0.4 basing the full text of documents) in the experiments using the 200 sized embedding vectors and the 38-feature input space. In the experiments with 39 and more features, the recall values first decreased and then started to increase from the experiments with 40 and more features.

To measure the difference in the length of reference and model summaries, the subtraction of the number of sentences in predicted extract and the number of reference extract (Ext $_{\cap}$) was also presented in Fig. 2c for changing number of syntactic features. Here, it can be seen that the average number of sentences in automatically generated summaries can at most one sentence bigger than the reference summaries, and in the experiments, with 38 and more features the number of sentences in model summaries are around the number of sentences in reference summaries that is the desirable situation. The experiments on the syntactic feature space were performed by changing the number of features to observe the contribution of each feature to the summarization task. The best k

features were incrementally selected by a univariate statistical test based on chi-square where $3 \leq k \leq 42$.

4.3. Sentence selection for summarization based on ensembled features

In Section 4.1 and in Section 4.2, the limits of using semantic features and syntactic features for summarization was analyzed, and the upper limits of their usage on summarization performance on the proposed dataset was presented. The effect of the individual use of these two feature spaces on text summarization was observed, and it was seen that both vectors contributed highly to the summarization problem. Regarding the use of syntactic feature space, which contains 42 hand-crafted features gathered from the document sentences, the ROUGE-1 value of resulting summaries was 0.57 on average. The utilized syntactic feature space is the most comprehensive in literature, and the obtained ROUGE value is highly convincing. On the other hand, the use of semantic features based on GloVe embeddings presented more improved summaries with 0.60 ROUGE recall value.

As already mentioned in the previous section, this study handled the text summarization problem by selecting the most salient sentences to include the model extracts. Here, the main theoretical idea is that this selection process did not only rely on the importance of the belonging words or phrases but the structural organization of text and syntactical information of sentences. Hence, this study proposes to use the combination of semantic and syntactic features for summarization purposes to improve summarization performance. To that end, the semantic word embeddings were sequentially processed by an LSTM. Meanwhile, the syntactic features that correspond to the sentences were handled by a second LSTM. The outputs of these LSTMs compose an ensembled feature space.

In this study, an LSTM-NN model was proposed so as to it was fed with an ensembled feature space containing semantic and syntactic features to select the *summary-worthy* sentences in the document. The use of ensembled features in proposed LSTM-NN has been presented in Fig. 1(c). Here, LSTM_{SyF} processes the syntactic feature space, while LSTM_{SmF} processes the semantic feature space. The output vectors of LSTMs (for semantic and syntactic features respectively) were concatenated horizontally, and the resulting vector was given to the 2-layer neural network as input to perform classification. Using this approach, the semantic and syntactic features were equally considered in classification layers. Here, all of the LSTM implementation details were the same as the single-use of corresponding feature space.

In LSTM_{SmF}, the sequence length was determined as the maximum number of words in a sentence, and accordingly, the padding was performed by first locating the word vectors, and filling the rest of the sequence with zero vectors.

Rectified Linear Unit (RELU) was chosen for activation function in LSTMs and the first layer of neural network (with 64 neurons) except for the sigmoid-based neuron in the last layer. Adam optimizer was used to maximize the area under the receiver operating characteristic curve (ROC-AUC) and training accuracy. Binary cross-entropy was utilized as the loss function since the problem is structures as a binary classification problem. The classification threshold was 0.5. The batch size was 32. The training was early stopped when the training loss was less than $\delta = 1.00E - 07$ for 10 epochs.

First experimental organization was set up on DUC dataset to prove the improvement of enhanced feature space on the pure usage of syntactic and semantic features. The pretrained GloVe and word2vec word embeddings have been utilized for conducting the pure semantic features. The results of the experiments are presented in Table 9. In these experiments, the LSTM-NN determines a degree of summary-worthiness to each sentence, and the top k summary-worthy sentences were selected to be included in the final summary. Here, k is the number of sentences that the reference summary of the corresponding document has. In Table 9, it is clearly seen that the use of enhanced feature space prominently improves the ROUGE values according to the individual use of syntactic and semantic features.

In the experiments with syntactic feature space, it was concluded that the feature spaces containing 38 and more features were the most contributing ones since the best ROUGE values to be obtained. To that end, further studies and experiments focused only on these feature spaces. Basing the full texts of source documents and the Ext $_{\cap}$ extracts, the use of proposed ensembled feature space was ROUGE-based evaluated for these feature spaces, respectively, and the results were presented in Tables 11 and 10. Regarding the semantic feature side of these experiments, GloVe and word2vec were utilized in different embedding sizes. For each experiment, it is clear that the use of ensembled feature space outperformed the single-use of any feature space. To present a more clear representation, the experiments with 42-feature syntactic feature space and GloVe and word2vec semantic feature spaces were given in Fig. 3. Here the change in ROUGE-1-2-L values in 5-fold testing was illustrated, and the contribution of ensembled feature space to the use of individual semantic or syntactic feature space was clearly represented. The ROUGE-1 recall can be reached around 0.65, with almost 5% of an increase in the individual use of feature spaces. Fig. 3 also includes the results obtained from the standard

Table 9
The ROUGE values of LSTM-NN, obtained from the experiments on DUC, with pure semantic features, pure syntactic features and ensembled features.

	ROUGE-1			ROUGE-2			ROUGE-L		
Features	F	P	R	F	P	R	F	P	R
SyF	0.443	0.436	0.450	0.243	0.245	0.241	0.413	0.406	0.420
W2V	0.295	0.392	0.236	0.097	0.134	0.076	0.272	0.365	0.217
W2V + SyF	0.475	0.441	0.514	0.289	0.266	0.316	0.447	0.415	0.485
GloVe	0.438	0.413	0.466	0.239	0.228	0.251	0.407	0.384	0.433
GloVe + SyF	0.456	0.435	0.480	0.267	0.253	0.282	0.428	0.408	0.451

Table 10
Referencing the human extracts (Ext $_{\cap}$) of documents, the ROUGE-1 values obtained from LSTM-NN fed by ensembled feature space.

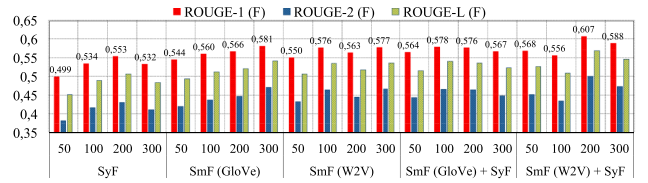
			38 Features			39 Features			40 Features			41 Features			42 Features		
	Embedding	Size	Δ	P	R	Δ	P	R	Δ	P	R	Δ	P	R	Δ	P	R
Train	GloVe + SyF	50	0.00	0.995	0.995	0.06	0.999	0.995	0.23	0.995	0.996	0.06	0.996	0.995	−0.05	0.999	0.998
		100	0.17	0.997	0.998	0.00	0.999	0.999	−0.05	0.999	0.997	−0.27	0.999	0.991	0.00	0.999	0.998
		200	−0.01	1.000	0.999	0.01	1.000	0.999	0.00	0.998	0.999	0.00	1.000	0.999	0.00	1.000	1.000
		300	0.00	1.000	1.000	0.00	1.000	0.998	0.00	0.999	0.998	0.00	1.000	0.999	0.00	1.000	0.999
	W2V + SyF	50	0.03	0.988	0.998	0.21	0.994	0.994	0.02	0.992	0.995	0.08	0.991	0.997	0.24	0.994	0.997
		100	0.00	0.998	0.998	−0.06	0.999	0.993	0.00	0.999	0.998	0.00	0.999	0.998	0.00	1.000	0.999
		200	0.00	1.000	0.998	0.00	0.993	0.984	0.00	0.999	0.997	0.00	1.000	0.999	0.00	0.999	0.999
		300	0.00	0.997	0.998	−0.02	1.000	1.000	−0.17	0.949	0.888	0.00	0.999	0.999	0.00	0.985	0.983
Test	GloVe + SyF	50	3.88	0.530	0.587	3.40	0.519	0.611	4.52	0.553	0.606	3.64	0.570	0.590	4.32	0.540	0.591
		100	2.44	0.525	0.604	2.48	0.537	0.589	4.08	0.555	**0.627**	0.44	0.551	0.539	2.08	0.543	0.618
		200	4.08	0.559	**0.630**	3.72	0.534	0.604	3.36	0.554	0.592	4.24	0.548	0.592	3.92	0.570	0.582
		300	4.16	0.565	0.603	4.16	0.541	0.614	3.32	0.568	0.617	4.92	0.539	**0.631**	3.88	0.546	0.589
	W2V + SyF	50	1.44	0.557	0.541	3.96	0.562	0.580	1.48	0.570	0.528	2.88	0.551	0.612	2.64	0.555	0.581
		100	4.16	0.590	0.601	1.92	0.576	0.523	3.24	0.558	0.554	2.36	0.566	0.573	1.40	0.574	0.539
		200	2.24	0.594	0.607	3.84	0.582	0.570	4.28	0.565	0.590	3.28	0.560	0.604	5.32	0.568	**0.652**
		300	2.52	0.583	0.556	4.08	0.574	**0.638**	2.00	0.615	0.569	4.36	0.565	**0.642**	5.04	0.551	**0.631**

classification performance evaluation methods by basing the classification threshold as 0.5 (see Fig. 3d). Since the Adam optimizer was based on the ROC-AUC and the classification accuracy, there is not a remarkable difference in the value of these metrics (ROC-AUC = 0.65 ± 0.02, accuracy = 0.70 ± 0.01), and they were not included in Fig. 3d. Here, the recall values are highly essential (Mutlu et al., 2019), and it is clear that the use of ensembled feature space improves the models' recall values in most of the implemented cases.

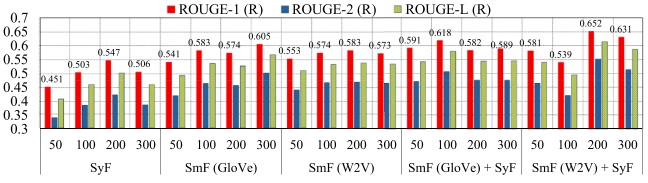
The proposed LSTM-NN based on ensembled feature space, was also compared with 2 types of baseline methods (Lead (Xu & Durrett, 2019) and TextRank (Mihalcea & Tarau, 2004)) and 2 types of state-of-the-art deep learning models (SummaRuNNer (Nallapati et al., 2017) and BanditSum (Dong et al., 2018)) to relatively determine its contribution to the literature. The results were provided in Table 12. The Lead was implemented in two strategies. First, the original implementation was performed similarly to the implementation in Xu and Durrett (2019). Accordingly, first k sentences were selected to be included in the summary. This strategy was powerful when the source documents are written by an inverted pyramid style such as news (Pottker, 2003). However, SIGIR 2018 dataset differs from the existing summarization datasets, which are mostly based on news articles, and it relies on academic writing styles. To that end, as a second Lead approach, the first around m sentences were selected from each paragraph homogeneously by round-robin so as the length of resulting summary be k. This lead implementation is referred as Lead (ours) in Table 12. It can be seen in Table 12 that this modification improves the ROUGE values. Being a robust text summarization method, TextRank

Table 11
Referencing the entire documents. the ROUGE-1 values obtained from LSTM-NN fed by ensembled feature space.

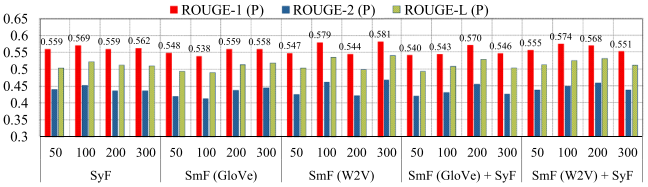
			38 features			39 features			40 features			41 features			42 features		
	Embedding	Size	Δ	P	R	Δ	P	R	Δ	P	R	Δ	P	R	Δ	P	R
Train	GloVe + SyF	50	0.00	1.000	0.385	0.06	1.000	0.384	0.23	1.000	0.385	0.06	1.000	0.384	−0.05	1.000	0.384
		100	0.17	1.000	0.385	0.00	1.000	0.384	−0.05	1.000	0.384	−0.27	1.000	0.382	0.00	1.000	0.384
		200	−0.01	1.000	0.384	0.01	1.000	0.384	0.00	1.000	0.385	0.00	1.000	0.384	0.00	1.000	0.384
		300	0.00	1.000	0.384	0.00	1.000	0.384	0.00	1.000	0.384	0.00	1.000	0.384	0.00	1.000	0.384
	W2V + SyF	50	0.03	1.000	0.389	0.21	1.000	0.385	0.02	1.000	0.386	0.08	1.000	0.387	0.24	1.000	0.386
		100	0.00	1.000	0.385	−0.06	1.000	0.382	0.00	1.000	0.384	0.00	1.000	0.385	0.00	1.000	0.384
		200	0.00	1.000	0.384	0.00	1.000	0.381	0.00	1.000	0.384	0.00	1.000	0.384	0.00	1.000	0.385
		300	0.00	1.000	0.385	−0.02	1.000	0.384	−0.17	0.994	0.355	0.00	1.000	0.385	0.00	0.998	0.383
Test	GloVe + SyF	50	3.88	0.992	0.430	3.40	0.984	0.460	4.52	1.000	0.427	3.64	1.000	0.416	4.32	1.000	0.436
		100	2.44	1.000	0.447	2.48	1.000	0.434	4.08	1.000	0.449	0.44	1.000	0.383	2.08	1.000	0.451
		200	4.08	1.000	0.439	3.72	0.976	0.437	3.36	1.000	0.417	4.24	1.000	0.426	3.92	1.000	0.406
		300	4.16	1.000	0.415	4.16	1.000	0.449	3.32	0.992	0.424	4.92	0.984	0.451	3.88	0.992	0.418
	W2V + SyF	50	1.44	1.000	0.382	3.96	0.992	0.411	1.48	0.992	0.363	2.88	1.000	0.435	2.64	1.000	0.410
		100	4.16	0.992	0.405	1.92	1.000	0.359	3.24	0.984	0.380	2.36	1.000	0.389	1.40	1.000	0.370
		200	2.24	0.992	0.399	3.84	1.000	0.395	4.28	0.992	0.402	3.28	0.992	0.413	5.32	1.000	0.454
		300	2.52	1.000	0.375	4.08	1.000	0.434	2.00	1.000	0.368	4.36	1.000	0.452	5.04	1.000	0.460



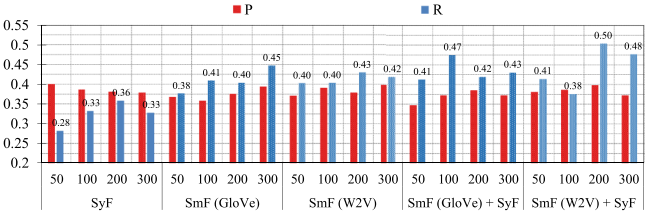
(a) ROUGE (F-Measure)



(b) ROUGE (Recall)



(c) ROUGE (Precision)



(d) Evaluation of standard classification metrics

Fig. 3. Referencing the human extracts (Ext $_{\cap}$) of documents, the results obtained from ROUGE analysis and standard classification evaluation metrics obtained by LSTM-NN on pure syntactic features (SyF), pure semantic features (SmF) and ensembled features.

outperformed these two Lead strategies; however, its ROUGE values are still behind the state-of-the-art.

SummaRuNNer is one of the implemented deep learning approach which is performed by using three different structures as (*ii*) convolutional neural network (CNN) for word-level and Bi-GRU for sentence-level (*ii*) hierarchical two-layered attention-based Bi-GRU, and (*iii*) Bi-GRU for word-level and Bi-GRU for sentence level, The obtained ROUGE values were presented in Table 12. Here, it is obviously seen that among these three structures, the SummaRuNNer with hierarchical Bi-GRU performs best both for 5-fold training and testing. The ROUGE-1 recall value reached 0.626 for testing. Since these base-line models use word2vec embeddings, the results of experiments on this embedding were also presented in the corresponding table. As seen here, the proposed LSTM-NN, outperformed all of the variations of SummaRuNNer by producing 0.652 ROUGE-1 recall value. The reason behind the improvement in the summaries produced by SummaRuNNer models is not entirely related to the utilized summarization method since the LSTM-NN provides a similar approach with SummaRuNNer, and it uses the same word embeddings for word representation. The exact reason for this improvement is the utilized syntactic features. SummaRuNNer uses position features very effectively; however, it does not consider other syntactic properties of text. The proposed LSTM-NN, on the other hand, uses a large syntactic feature space with several calculation methods besides the semantic features. It allows producing more similar extracts to human-generated extracts.

In Table 12, the ROUGE values of BanditSum performed on SIGIR 2018 dataset are presented. Here, the implementation was based on the shared source code recommended in Dong et al. (2018) by using exactly the same RNN-based reinforcement learning method on GloVe embeddings and using the recommended parameter configurations. Here, only two modifications were performed. First, the number of training iteration was increased to ensure the model convergences precisely. Additionally, the oracle sentence length was set to the number of sentences in the goal summary. Please note that these two modifications were in favor of the summarization performance of BanditSum, which also makes the comparisons fairer.

According to the ROUGE values obtained from 5-fold cross-validation, it can be concluded from Table 12 that BanditSum provides

Table 12
Comparison of LSTM-NN with different deep learning models models which also utilizes semantic and syntactic information for selecting the salient sentences.

		ROUGE-1			ROUGE-2			ROUGE-L		
	Sentence Selection Method	F	P	R	F	P	R	F	P	R
Train	Hierarchical attention-based Bi-GRU	0.877	0.832	0.929	0.850	0.804	0.905	0.877	0.832	0.929
	CNN & Bi-GRU	0.872	0.829	0.921	0.850	0.805	0.903	0.872	0.829	0.921
	Bi-GRU & Bi-GRU (SummaRuNNer)	0.879	0.836	0.929	0.852	0.806	0.906	0.879	0.836	0.929
	BanditSum	0.629	0.602	0.668	0.475	0.455	0.504	0.595	0.570	0.628
	LSTM-NN	**0.999**	**0.999**	**0.999**	**0.998**	**0.998**	**0.998**	**0.999**	**0.999**	**0.999**
Test	Lead (Xu & Durrett, 2019)	0.437	0.478	0.429	0.298	0.339	0.290	0.422	0.462	0.414
	Lead (ours)	0.462	0.516	0.439	0.307	0.360	0.286	0.442	0.495	0.420
	TextRank	0.486	0.493	0.479	0.326	0.326	0.326	0.464	0.471	0.457
	Hierarchical attention-based Bi-GRU	0.557	0.535	0.607	0.437	0.416	0.489	0.513	0.491	0.561
	CNN & Bi-GRU	0.545	0.526	0.592	0.418	0.402	0.463	0.495	0.476	0.541
	Bi-GRU & Bi-GRU (SummaRuNNer)	0.564	0.534	0.626	0.449	0.425	0.510	0.524	0.495	0.583
	BanditSum	0.616	0.592	0.652	0.458	0.440	0.485	0.581	0.558	0.611
	LSTM-NN (SmF(GloVe) + SyF)	**0.578**	**0.543**	**0.618**	**0.464**	**0.430**	**0.507**	**0.540**	**0.507**	**0.579**
	LSTM-NN (SmF(W2V) + SyF)	**0.607**	**0.568**	**0.652**	**0.501**	**0.458**	**0.551**	**0.569**	**0.531**	**0.614**

promising results on SIGIR 2018 dataset, and it outperformed the Lead, TextRank, and SummaRuNNer. On the one hand, the obtained ROUGE-1 values of BanditSum was on par with our LTSM-NN on enhanced feature space with W2V embeddings. For LSTM-NN with GloVe embeddings, ROUGE-1 values are behind the ROUGE-1 values of BanditSum. On the other hand, bigram-based ROUGE-2 values are higher than BanditSum's based on both of embedding types. It means that BanditSum's model summaries can contain more unigrams that the summary should have. However, LSTM-NN selects the phrases more precisely. Specifically, the ROUGE-1 only measures the unigram overlaps, which means that the corresponding unigrams may be in other sentences. On the other hand, phrase-based ROUGE scores are more related to the accuracy of sentence selection. Here in Table 12, the ROUGE-2, and ROUGE-L of the BanditSum and LSTM-NN are comparable, and LSTM-NN slightly increased the phrase-based ROUGE values.

5. Conclusion

In this study, the extractive text summarization was revisited according to three research areas: the utilized dataset, the feature space, and the method which is applied for sentence selection to obtain its summary worthiness.

Dataset: A new English dataset containing the proceedings of 41th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2018) was proposed. The introduction section of each proceeding was considered as a source document since, in an academic publication, this section should give the essential information of the entire study. The sentences in introduction sections were manually labeled as *summary-worthy* or *summary-unworthy* by three readers. In this way, a candidate sentence list was created for each reader, and three candidate extractive summaries were obtained for each document. Blending these candidate summaries, two human extracts (Ext $_{\cup}$ and Ext $_{\cap}$ were generated by taking the union and intersection of candidate sentence sets. Then, this manual extraction was verified by several measurements and experiments to ensure the manual labeling process of dataset production was neither random or algorithmic.

Summarization Features: In literature, the extractive text summarization has been handled either the syntactic feature space (which contains hand-crafted features extracted from the entire text) or the semantic feature space (as a combination of word-embeddings for each sentence). In this study, the effect of each feature space on selecting the salient sentences was individually investigated, and by combining these feature spaces, a new ensembled feature space was proposed. The experiments performed by using the syntatic features and semantic features respectively, and summarization performance was evaluated by ROUGE-n metrics. The results of the experiments on proposed SIGIR 2018 dataset showed that the more syntactic features were used, the more informative summaries were produced, and the ROUGE-1 score was 0.57. Regarding the single-use of semantic features 0.60 ROUGE-1 score could be reached. Comparing the single-use of syntactic and semantic feature spaces, the use of ensembled feature space remarkably improved the summarization performance based on both standard classification performance evaluation measures, and the ROUGE-based summary evaluation analysis. The ROUGE-1 score reached 0.65 by using the proposed ensembled feature space.

Summarization Method: A Long Short-Term Memory based neural network model (LSTM-NN) was proposed in this study. This model was a hierarchical structure where the first layer contains two LSTMs that simultaneously process the semantic and syntactic feature spaces. The outputs of LSTMs were concatenated in a deeper layer, and finally, the resulting enhanced feature space was given to a fully connected 2-layer neural network for classification. LSTM-NN model was empirically compared with, two types of Lead method, TextRank, and two state-of-the-art deep learning models (SummaRuNNer and BanditSum), and the resulting summaries were evaluated by ROUGE. The results showed that LSTM-NN outperformed the baseline methods of Lead and TextRank, and all of the variations of SummaRuNNer by improving the ROUGE-1 score by around 3%, and produced more similar extracts to what humans do. Regarding BanditSum, the ROGUE-1 values were comparable, but still, the ROUGE-2 values were higher, which means that LSTM-NN selects the phrases more precisely. These results support the main idea of this study as handling the text summarization

by enhanced feature space.

In this study, the pre-trained word embeddings were utilized not to have a bias on a specific domain and to generate more general summaries. However, it should be also be investigated that how does it affect the summarization performance to train the word embeddings for the proposed corpus to obtain more domain-based summaries. To that end, the following studies will be focusing on domain-specific text summarization.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.ipm.2020.102359

References

Aliguliyev, R. M. (2006). *A novel partitioning-based clustering method and generic document summarization. 2006 IEEE/WIC/ACM international conference on web intelligence and intelligent agent technology workshops*. IEEE626–629.

Annual international ACM SIGIR conference on research and development in information retrieval. Sigir2018 proceedings.

Alguliev, R. M., Aliguliyev, R. M., & Hajirahimova, M. S. (2012). Gendocsum + mclr: Generic document summarization based on maximum coverage and less redundancy. *Expert Systems with Applications, 39*(16), 12460–12473.

Alguliev, R. M., Aliguliyev, R. M., Hajirahimova, M. S., & Mehdiyev, C. A. (Aliguliyev, Hajirahimova, Mehdiyev, 2011a). Mcmr: Maximum coverage and minimum redundant text summarization model. *Expert Systems with Applications, 38*(12), 14514–14522.

Alguliev, R. M., Aliguliyev, R. M., & Isazade, N. R. (2013). Multiple documents summarization based on evolutionary optimization algorithm. *Expert Systems with Applications, 40*(5), 1675–1689.

Aliguliyev, R. M., Aliguliyev, R. M., & Isazade, N. R. (2015). An unsupervised approach to generating generic summaries of documents. *Applied Soft Computing, 34*, 236–250.

Alguliyev, R. M., Aliguliyev, R. M., Isazade, N. R., Abdi, A., & Idris, N. (2019). Cosum: Text summarization based on clustering and optimization. *Expert Systems, 36*(1), e12340.

Alguliev, R. M., Aliguliyev, R. M., & Mehdiyev, C. A. (Aliguliyev, Mehdiyev, 2011b). Sentence selection for generic document summarization using an adaptive differential evolution algorithm. *Swarm and Evolutionary Computation, 1*(4), 213–222.

Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., & Kochut, K. (2017). Text summarization techniques: A brief survey. arXiv:1707.02268.

Alzuhair, A., & Al-Dhelaan, M. (2019). An approach for combining multiple weighting schemes and ranking methods in graph-based multi-document summarization. *IEEE Access, 7*, 120375–120386.

Balabantaray, R. C., Sahoo, D., Sahoo, B., & Swain, M. (2012). Text summarization using term weights. *International Journal of Computer Applications, 38*(1), 10–14.

Barrios, F., Lopez, F., Argerich, L., & Wachenchauzer, R. (2016). *Variations of the similarity function of textrank for automated summarization. Argentine symposium on artificial intelligence*.

Brdiczka, O., & Chu, M. K. (2011). Measuring document similarity by inferring evolution of documents through reuse of passage sequences. US Patent 8,086,548.

Cao, Z., Wei, F., Li, S., Li, W., Zhou, M., & Wang, H. (2015). *Learning summary prior representation for extractive summarization. Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 2: Short papers)*829–833.

Chen, K.-Y., Liu, S.-H., Chen, B., & Wang, H.-M. (2017). An information distillation framework for extractive summarization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing, 26*(1), 161–170.

Cheng, J., & Lapata, M. (2016). *Neural summarization by extracting sentences and words. Annual meeting of the association for computational linguistics*484–494.

Conroy, J. M., & O'leary, D. P. (2001). *Text summarization via hidden Markov models. Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval*406–407.

Document understanding conference (duc) dataset, (2001–2007). https://www-nlpir.nist.gov/projects/duc/data.html.

Davis, J., & Liss, R. (2006). *Effective academic writing 3*. New York: Oxford University Press.

Denil, M., Demiraj, A., & De Freitas, N. (2015). Extraction of salient sentences from labelled documents. *Computing Research Repository*, 1–9.

Dong, Y., Shen, Y., Crawford, E., van Hoof, H., & Cheung, J. C. K. (2018). *Banditsum: Extractive summarization as a contextual bandit. Proceedings of the 2018 conference on empirical methods in natural language processing*3739–3748.

Ermakova, L., Cossu, J. V., & Mothe, J. (2019). A survey on evaluation of summarization methods. *Information Processing & Management, 56*(5), 1794–1814. https://doi.org/10.1016/j.ipm.2019.04.001.

Fang, C., Mu, D., Deng, Z., & Wu, Z. (2017). Word-sentence co-ranking for automatic extractive text summarization. *Expert Systems with Applications, 72*, 189–195.

Fattah, M. A., & Ren, F. (2009). GA, MR, FFNN, PNN And GMM based models for automatic text summarization. *Computer Speech and Language, 23*(1), 126–144. https://doi.org/10.1016/j.csl.2008.04.002.

Ferreira, R., Lins, R. D., Freitas, F., Cavalcanti, G. D. C., Lima, R., Simske, S. J., ... Others (2013). Assessing sentence scoring techniques for extractive text summarization. *Expert Systems with Applications, 40*(14), 5755–5764.

García-Hernández, R. A., & Ledeneva, Y. (2009). *Word sequence models for single text summarization. 2009 second international conferences on advances in computer-human interactions*. IEEE44–48.

Gong, Y., & Liu, X. (2001). *Generic text summarization using relevance measure and latent semantic analysis. Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval*19–25.

Goularte, F. B., Nassar, S. M., Fileto, R., & Saggion, H. (2019). A text summarization method based on fuzzy rules and applicable to automated assessment. *Expert Systems with Applications, 115*, 264–275. https://doi.org/10.1016/j.eswa.2018.07.047.

Hachey, B., Murray, G., Reitter, D., et al. (2005). *The embra system at duc 2005: Query-oriented multi-document summarization with a very large latent semantic space. Proceedings of the document understanding conference (DUC) 2005, vancouver, bc, canada*.

Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., & Blunsom, P. (2015). *Teaching machines to read and comprehend. Advances in neural information processing systems*1693–1701.

Hu, Y.-H., Chen, Y.-L., & Chou, H.-L. (2017). Opinion mining from online hotel reviews a text summarization approach. *Information Processing & Management, 53*(2), 436–449. https://doi.org/10.1016/j.ipm.2016.12.002.

Jaidka, K., Chandrasekaran, M. K., Rustagi, S., & Kan, M.-Y. (2016). *Overview of the cl-scisumm 2016 shared task. In proceedings of joint workshop on bibliometric-enhanced information retrieval and NLP for digital libraries (BIRNDL 2016)*.

Joshi, A., Fidalgo, E., Alegre, E., & Fernández-Robles, L. (2019). Summcoder: An unsupervised framework for extractive text summarization based on deep auto-encoders. *Expert Systems with Applications, 129*, 200–215.

Knight, K., & Marcu, D. (2002). Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artificial Intelligence, 139*(1), 91–107. https://doi.org/10.1016/S0004-3702(02)00222-9.

Koupaee, M., & Wang, W. Y. (2018). Wikihow: A large scale text summarization dataset. *Computing Research Repository*.

Krahmer, E., Marsi, E., & van Pelt, P. (2008). *Query-based sentence fusion is better defined and leads to more preferred results than generic sentence fusion. Proceedings of ACL-08: Hlt, short papers*193–196.

Kumar, Y. J., Salim, N., Abuobieda, A., & Albaham, A. T. (2014). Multi document summarization based on news components using fuzzy cross-document relations. *Applied Soft Computing, 21*, 265–279.

Le, Q., & Mikolov, T. (2014). *Distributed representations of sentences and documents. International conference on machine learning*1188–1196.

Ledeneva, Y., Gelbukh, A., & García-Hernández, R. A. (2008). *Terms derived from frequent sequences for extractive text summarization. International conference on intelligent text processing and computational linguistics*. Springer593–604.

Li, X., Du, L., & Shen, Y.-D. (2012). Update summarization via graph-based sentence ranking. *IEEE transactions on knowledge and data engineering, 25*(5), 1162–1174.

Lin, C.-Y., & Och, F. (2004). *Looking for a few good metrics: Rouge and its evaluation. NTCIR workshop*.

Lopyrev, K. (2015). Generating news headlines with recurrent neural networks. *Journal of Controlled Release, 230*, 73–78. https://doi.org/10.1023/A.
Multiling community site, http://multiling.iit.demokritos.gr/. Accessed: 2020-01-16.
Meena, Y. K., & Gopalani, D. (2014). *Analysis of sentence scoring methods for extractive automatic text summarization. Proceedings of the 2014 international conference on information and communication technology for competitive strategies*. ACM53.
Miao, L., Cao, D., Li, J., & Guan, W. (2020). Multi-modal product title compression. *Information Processing & Management, 57*(1), 102123. https://doi.org/10.1016/j.ipm.2019.102123.
Mihalcea, R., & Tarau, P. (2004). *Textrank: Bringing order into text. Proceedings of the 2004 conference on empirical methods in natural language processing*.
Mohamed, M., & Oussalah, M. (2019). Srl-esa-textsum: A text summarization approach based on semantic role labeling and explicit semantic analysis. *Information Processing & Management, 56*(4), 1356–1372. https://doi.org/10.1016/j.ipm.2019.04.003.
Mutlu, B., Sezer, E. A., & Akcayol, M. A. (2019). Multi-document extractive text summarization: A comparative assessment on features. *Knowledge-Based Systems, 183*, 104848.
Mutlu, B., Sezer, E. A., & Akcayol, M. A. (2020). *Investigation of text summarization features by fuzzy systems and convolutional neural networks. Submitted: Semantics conference 2020*.
Nallapati, R., Zhai, F., & Zhou, B. (2017). *Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. Thirty-first aaai conference on artificial intelligence*.
Nallapati, R., Zhou, B., dos Santos, C. N., Gulcehre, C., & Xiang, B. (2016). Abstractive text summarization using sequence-to-sequence rnns and beyond. *ICLR,* 4–7. https://doi.org/10.18653/v1/K16-1028.
Narayan, S., Cohen, S. B., & Lapata, M. (2018). *Ranking sentences for extractive summarization with reinforcement learning. Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: Human language technologies1*.
Nasar, Z., Jaffry, S. W., & Malik, M. K. (2019). Textual keyword extraction and summarization: State-of-the-art. *Information Processing & Management, 56*(6), 102088. https://doi.org/10.1016/j.ipm.2019.102088.
Nenkova, A., & McKeown, K. (2012). *A survey of text summarization techniques. Mining text data*. Springer43–76.
Oliveira, H., Ferreira, R., Lima, R., Lins, R. D., Freitas, F., Riss, M., & Simske, S. J. (2016). Assessing shallow sentence scoring techniques and combinations for single and multi-document summarization. *Expert Systems with Applications, 65*, 68–86. https://doi.org/10.1016/j.eswa.2016.08.030.
Pardo, T. A. S., & Rino, L. H. M. (2003). *TeMario: a corpus for automatic text summarizationTechnical Report*. NILC Tech. Report NILC-TR-03-09.
Patil Pallavi, D., & Mane, P. (2014). A comprehensive review on fuzzy logic & latent semantic analysis techniques for improving the performance of text summarization. *International Journal of Advance Research in Computer Science and Management Studies, 2*(11), 476–485.
Pennington, J., Socher, R., & Manning, C. D. (2014). *Glove: Global vectors for word representation. Empirical methods in natural language processing (EMNLP)*1532–1543.
Pottker, H. (2003). News and its communicative quality: The inverted pyramid when and why did it appear? *Journalism Studies, 4*(4), 501–511.
Ren, P., Chen, Z., Ren, Z., Wei, F., Nie, L., Ma, J., & De Rijke, M. (2018). Sentence relations for extractive summarization with deep neural networks. *ACM Transactions on Information Systems (TOIS), 36*(4), 1–32.
Sandhaus, E. (2008). New york times corpus: Corpus overview. LDC catalogue entry LDC2008T19.
See, A., Liu, P. J., & Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. *Computing Research Repository* abs/1704.04368
Sinha, A., Yadav, A., & Gahlot, A. (2018). Extractive text summarization using neural networks. *Computing Research Repository*.
Steinberger, J., & Ježek, K. (2009). *Update summarization based on latent semantic analysis. International conference on text, speech and dialogue*. Springer77–84.
Suanmali, L., Salim, N., & Binwahlan, M. S. (2009). Fuzzy logic based method for improving text summarization. *International Journal of Computer Science and Information Security (IJCSIS), 2*(1), 6.
Sun, X., & Zhuge, H. (2018). Summarization of scientific paper through reinforcement ranking on semantic link network. *IEEE Access, 6*, 40611–40625.
Wan, X. (2010). *Towards a unified approach to simultaneous single-document and multi-document summarizations. Proceedings of the 23rd international conference on computational linguistics*. Association for Computational Linguistics1137–1145.
Wang, W., Li, Z., Wang, J., & Zheng, Z. (2017). How far we can go with extractive text summarization? heuristic methods to obtain near upper bounds. *Expert systems with applications, 90*, 439–463.
Xu, J., & Durrett, G. (2019). *Neural extractive text summarization with syntactic compression. Proceedings of the 2019 conference on empirical methods in natural language processing*. Hong Kong, China: Association for Computational Linguistics.
Yasunaga, M., Kasai, J., Zhang, R., Fabbri, A., Li, I., Friedman, D., & Radev, D. (2019). *ScisummNet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks. Proceedings of AAAI 2019*.
Yin, W., & Pei, Y. (2015). *Optimizing sentence modeling and selection for document summarization. Twenty-fourth international joint conference on artificial intelligence*1383–1389.
Zajic, D., Dorr, B. J., Lin, J., & Schwartz, R. (2007). Multi-candidate reduction: Sentence compression as a tool for document summarization tasks. *Information Processing and Management, 43*(6), 1549–1570. https://doi.org/10.1016/j.ipm.2007.01.016.
Zhang, X., Lapata, M., Wei, F., & Zhou, M. (2018). *Neural latent extractive document summarization. 2018 conference on empirical methods in natural language processing*.