

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/349470963>

Unsupervised extractive multi-document summarization method based on transfer learning from BERT multi-task fine-tuning

Article in *Journal of Information Science* · February 2021

DOI: 10.1177/0165551521990616

CITATIONS

0

READS

83

4 authors, including:



Salima Lamsiyah

Sidi Mohamed Ben Abdellah University

5 PUBLICATIONS 2 CITATIONS

[SEE PROFILE](#)



Abdelkader El Mahdaouy

Mohammed VI Polytechnic University

17 PUBLICATIONS 71 CITATIONS

[SEE PROFILE](#)



Bernard Espinasse

Aix-Marseille Université

99 PUBLICATIONS 654 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Relation Classification [View project](#)



Symbolic-Based Information Extraction [View project](#)

Unsupervised extractive multi-document summarization method based on transfer learning from BERT multi-task fine-tuning

Journal of Information Science

1–19

© The Author(s) 2021

Article reuse guidelines:

sagepub.com/journals-permissions

DOI: 10.1177/0165551521990616

journals.sagepub.com/home/jis**Salima Lamsiyah** 

Laboratory of Informatics, Signals, Automatic, and Cognitivism, FSDM, Sidi Mohamed Ben Abdellah University, Morocco; Laboratory of Engineering Sciences, National School of Applied Sciences, Ibn Tofail University, Morocco

Abdelkader El Mahdaouy 

School of Computer Science (UM6P-CS), Mohammed VI Polytechnic University (UM6P), Morocco

Saïd El Alaoui Ouatik

Laboratory of Informatics, Signals, Automatic, and Cognitivism, FSDM, Sidi Mohamed Ben Abdellah University, Morocco; Laboratory of Engineering Sciences, National School of Applied Sciences, Ibn Tofail University, Morocco

Bernard Espinasse

LIS UMR CNRS 7020, Aix-Marseille Université/Université de Toulon, France

Abstract

Text representation is a fundamental cornerstone that impacts the effectiveness of several text summarization methods. Transfer learning using pre-trained word embedding models has shown promising results. However, most of these representations do not consider the order and the semantic relationships between words in a sentence, and thus they do not carry the meaning of a full sentence. To overcome this issue, the current study proposes an unsupervised method for extractive multi-document summarization based on transfer learning from BERT sentence embedding model. Moreover, to improve sentence representation learning, we fine-tune BERT model on supervised intermediate tasks from GLUE benchmark datasets using single-task and multi-task fine-tuning methods. Experiments are performed on the standard DUC'2002–2004 datasets. The obtained results show that our method has significantly outperformed several baseline methods and achieves a comparable and sometimes better performance than the recent state-of-the-art deep learning-based methods. Furthermore, the results show that fine-tuning BERT using multi-task learning has considerably improved the performance.

Keywords

BERT fine-tuning; multi-document summarization; multi-task learning; sentence representation learning; transfer learning

1. Introduction

In today's society, we are facing an inevitable and challenging problem of information overload that highlighted the need of developing effective and specific tools to deal with this problem. Automatic text summarization (ATS), by condensing and creating shortened versions of texts, can efficiently help users save their time and effort while finding concise and suitable information for their needs. Thus, the main idea of ATS is to find a subset of data that capture the core

Corresponding author:

Salima Lamsiyah, Laboratory of Informatics, Signals, Automatic, and Cognitivism, FSDM, Sidi Mohamed Ben Abdellah University B.P. 1796, Fez-Atlas, 30003, Morocco.

Email: salima.lamsiyah@usmba.ac.ma

information of the entire set. Different types of automatic text summarization have been proposed. For instance, regarding the number of input documents, single and multi-document summarization have been introduced. Similarly, based on the purpose of the summarization, it can be either generic or query-focused. Generic summaries represent all relevant facts of a source document without considering the users' information needs, whereas in query-focused summaries, the content of the summary is driven by the user's information need or simply the user's query. In this article, we focus on generic multi-document summarization.

Since the pioneering work of Luhn [1], several methods have been proposed for automatic text summarization. They are mainly classified into *extractive* and *abstractive* approaches. The former generates summaries by extracting the most salient sentences as they exactly appear in the source documents, while abstractive summarization techniques produce summaries by concisely paraphrasing the document's content. Abstractive methods are considered complex and sometimes they are not completely automatic; they require resources previously built that demand a high computational effort. Thus, as part of our research, we will use the extractive approach for generic multi-document summarization.

Document representation is considered as a fundamental process that affects the effectiveness of many extractive text summarization methods. For instance, Bag-of-Words (BOW) representation has been widely used in automatic text summarization [2]. However, the latter representation suffers from two major drawbacks: it does not take into consideration the ordering of words and ignores the semantic relationships between them. Although Bag-of-N-grams representation considers the order of the words in a short context, it suffers from data sparsity and curse of dimensionality problems [3]. Recently, word embeddings such as Word2vec [4] and GloVe [5] have been emerged as an effective representational basis for text summarization methods [6–10]. They aim to represent words as dense vectors in low-dimensional vector space using various pre-trained models, inspired from neural networks language modelling. Even though word embeddings have shown good performance in several natural language processing tasks [11,12], representing relationships among multiple words and phrases in a single dense vector is still an emerging problem. For example, considering these two sentences 'you are going there to play not to study' and 'you are going there to study not to play', these two sentences will have identical representation based on word embedding and BOW representations while their meanings are completely different.

In order to deal with the abovementioned problems, several sentence embedding models have been developed to represent variable-length sentences by dense vectors in a low dimensional vector space, which capture the semantic relationships among their constituents. We distinguish between two popular approaches for learning sentence embeddings, namely *multi-task learning* and *language model pre-training*. On the one hand, Multi-Task Learning (MTL) is an inductive mechanism that aims to improve generalisation performance by getting the benefit of the domain information contained in training signals of related tasks [13]. For two fundamental reasons, applying MTL to sentence representation learning using deep neural networks has shown promising results [14–18]. First, the limited amount of task-specific training data that is typically required to train any supervised deep neural network. Second, MTL benefits from regularisation effect that avoids over-fitting to a specific task, and hence, providing universal representations across tasks. On the other hand, the idea behind the language model pre-training approach is to first train deep neural network models on a large amount of unlabeled data using unsupervised objectives. Then, apply these pre-trained models to other supervised natural language understanding tasks using task-specific data. This allows building general-purpose models that have the ability to transfer the learned knowledge to other similar tasks as well as to help in saving time and computational power. For instance, pre-trained sentence embedding models such as the Bidirectional Encoder Representations from Transformers (BERT) [19], Skip-Thoughts [20] and ELMo [21] have shown to be effective for learning universal sentences representations, which are useful for many natural language processing tasks including automatic text summarization methods [22].

In this article, we propose an unsupervised extractive method for multi-document summarization based on transfer learning from the fine-tuned BERT models. We combine the strengths of both multi-task learning and the pre-trained language model BERT [19] to encode the input documents and obtain their sentences representation. We argue that these two technologies are complementary to each other and can be combined for improving sentence representation learning to boost the performance of text summarization task.

To summarise, the main contributions of this work are as follows:

1. We propose an unsupervised method for extractive multi-document summarization based on transfer learning from BERT fine-tuning on Natural Language Understanding tasks for sentence representation learning.
2. We fine-tune BERT model on supervised intermediate tasks from GLUE benchmark [23] using single-task and multi-task fine-tuning.
3. We investigate the impact of the two BERT fine-tuning methods on extractive multi-document summarization, and we showcase that BERT multi-task fine-tuning achieves substantial performance improvement.

To demonstrate the effectiveness of the proposed method, we empirically evaluate its performance on the standard DUC'2002–2004 multi-document summarization datasets, using the state-of-the-art ROUGE method [24]. Specifically, ROUGE-N (ROUGE-1, ROUGE-2, ROUGE-4) and ROUGE-L metrics aim to measure the content similarity between the generated summaries and their corresponding reference summaries (gold summaries). The obtained results show that the proposed method significantly outperforms several baseline methods and is on a par with recent state-of-the-art methods for extractive multi-document summarization. Moreover, the proposed method is unsupervised, fast and easy to implement.

The remainder of this article is organised as follows. We review the related work in section 2. We describe the proposed method in section 3. The experimental results on benchmark datasets are presented in section 4 and discussed in section 5. Finally, section 6 concludes the paper and draws lines for future works.

2. Related work

In this section, we first review some previous methods for extractive text summarization. Then, we provide a brief overview of the transfer learning methods applied in text representation learning. In addition finally, we cover BERT model architecture.

2.1. Extractive text summarization

Extractive text summarization methods identify the most salient sentences of a document and then subsequently concatenate them as they appear in the original document to create the final summary. Traditional text summarization methods are rule-based; they rely on hand-crafted features and expert knowledge [25–29]. With the recent advancements of neural network architectures, text summarization methods based on deep neural networks have received much attention and have achieved promising results. In the rest of this section, a brief description of extractive text summarization methods based on deep learning models is given. For readers who are interested in a detailed overview of automatic text summarization approaches and methods, they may refer to the recent surveys on the field [30,31].

In Zhong et al. [32], the authors have proposed an unsupervised method for multi-document summarization based on a deep learning model with three layers: concepts extraction, summary generation and reconstruction validation. Cao et al. [33] have used a recursive neural network to automatically rank sentences for multi-document summarization. While Yasunaga et al. [34] have introduced a neural multi-document summarization system (GBN-MDS) based on both graph convolutional networks and recurrent neural networks. Besides, Lebanoff et al. [35] have proposed to adapt encoder–decoder models trained on single-document datasets to the multi-document summarization task by introducing an external maximal marginal relevance module to select relevant sentences from multi-document input.

Moreover, deep neural networks have been also exploited to learn word/sentence representations for extractive text summarization. Word embedding representations have been proven to be effective for improving the performances of several text summarization methods [6–10]. The representation power of neural networks is related to their ability to learn high-level features across multiple layers and create accurate decision boundaries for the input instances. Following this success, other works have explored the potential of deep learning models for sentence representation learning. In particular, convolutional neural networks models have been widely used to learn sentence representations for extractive text summarization. For instance, Denil et al. [36] have developed a hierarchical convolutional model to introspect the structure of the document where a convolutional network is applied to learn sentences representations based on their words embeddings. In a similar work, Yin and Pei [37] have introduced a convolutional neural network–based method, where each sentence is projected to a continuous vector space, and then an optimization process is run to select relevant sentences taking into consideration their diversity and prestige cost. In the same context, Cao et al. [38] have developed a system based on enhanced convolutional neural networks that aims to automatically learn summary's prior features for extractive summarization task.

In recent years, sentence embedding models have been widely used for extractive text summarization task [6,39,40]. Sentence embedding models aim to map sentences into dense vectors that encode their semantics. Zhang et al. [41] have proposed a sentence vector encoding framework based on a deep LSTM model; it embeds sentences into continuous vectors for single-line text summarization task. Joshi et al. [22] have proposed an unsupervised method for extractive text summarization based on the Skip-Thoughts pre-trained sentence embedding model [20]. In the same context, Bouscarrat et al. [42] have introduced STRASS, another extractive text summarization method based on sentence embedding representations. Where, the summary is created by selecting the sentences that are more similar to the entire embedding of the original document. Recently, Liu and Lapata [43] have developed a general framework for both extractive and abstractive text summarization based on the Bidirectional Encoder Representations from Transformers (BERT) [19], where authors

have introduced a novel-based BERT document-level encoder able to capture the semantics of a document and thus generate representations of its sentences.

In contrast to the existing methods, we introduce a simple, effective and unsupervised method for extractive generic multi-document summarization. We explore transfer learning from BERT sentence embedding model [19] to improve sentence representation learning, which helps boost the performance of the proposed method. Inspired by human activities where people often apply the knowledge learned from previous tasks to help learn a new task, we fine-tune BERT on intermediate natural language understanding tasks from GLUE datasets before applying it to the text summarization task. Our idea is justified by the fact that transfer learning allows benefitting from knowledge learned from other natural language understanding tasks. Moreover, rather than fine-tuning BERT on a single task, we propose to fine-tune BERT on multiple related tasks simultaneously, which helps learn more universal sentence representation. Therefore, in this work, we showcase how the combination of transfer and multi-task learning approaches can be helpful in learning suitable sentence representation for extractive multi-document summarization. To the best of our knowledge, this is the first work that exploits the potential of BERT model fine-tuning on natural language understanding tasks (GLUE datasets) for unsupervised multi-document summarization task.

2.2. Transfer learning

Transfer learning is considered as a promising strategy that allows leveraging knowledge learned from one or more related tasks to boost the performance of a target task. It plays a key role in many natural language processing tasks, more specifically in text representations learning. Most of the transfer learning models are based on deep neural networks where there are two ways to apply neural network-based transfer learning methods for text representation learning: *parameter initialization* and *multi-task learning*. In some cases, parameter initialization and multi-task learning methods are used together to construct precise target models.

In the case of *parameter initialization approach*, a deep neural network is first trained on a source task, and then, the learned parameters are used to initialise the target task neural network model. In fact, two methods exist for applying the initialization parameter approach, including parameters freezing and fine-tuning methods. The freezing method uses the model trained on the source task data to extract features in order to use them as input to the target task without any modification; it is mainly helpful when there is a limited labelled data for the target task. While the fine-tuning method trains the neural network model in a source task data, the pre-trained model is applied to the target task where parameters of some layers are fixed and the others are learned on the target data. Therefore, with the success of distributed representations, pre-trained language models such as ELMo [21], BERT [19], Skip-Thoughts [20] and InferSent [44] have demonstrated promising benefits in learning universal sentence representations.

Nevertheless, most of the existing text representation models have been trained using single tasks such as predicting the next word or sentence [4,20] or text entailment [44]. Hence, the use of a small amount of task-specific labelled data can affect the performance of these models because supervised deep neural networks require a large amount of labelled data, which is not always available. To overcome this issue, *multi-task learning approach* proposes to learn text representations using supervised data from multiple related tasks. The use of this approach allows benefitting from a regularisation effect that helps in reducing the risk of over-fitting, as well as making the learned representations universal across tasks. Several works have been proposed for learning text representation using multi-task learning [14,15,16,18]. For instance, Liu et al. [14] have proposed a novel method for learning text representation across multiple tasks. The proposed method aims to leverage large amounts of task-specific data, as well as it benefits from a regularisation effect that leads to learn universal and useful representations for other new tasks. Luong et al. [15] have proposed to combine multi-task learning and sequence to sequence learning to learn suitable text representations for machine translation task. Moreover, Hashimoto et al. [45] have presented a joint many-task model with growing depth in a single end-to-end model, which makes use of linguistic hierarchies to solve increasingly complex natural language processing tasks. Jernite et al. [46] have developed a novel unsupervised method for learning sentence representation inspired by the notion of discourse coherence. The proposed method combines three auxiliary tasks to train the neural network model, namely the binary ordering of sentences task, the newt sentence prediction task and the conjunction prediction task. Furthermore, Guo et al. [16] have developed an abstractive method for document summarization based on a novel multi-task learning architecture that combines two auxiliary tasks including question generation and entailment generation tasks. Recently, Ruder et al. [18] have introduced a novel method for learning text representation based on latent multi-task learning.

In addition to the initialization parameter and the multi-task learning approaches, other works have been proposed to combine the strengths of both of them. For instance, the Multi-Task Deep Neural Networks (MT-DNN) model [47] has recently emerged as a new technology for learning contextual text representations among multiple natural language

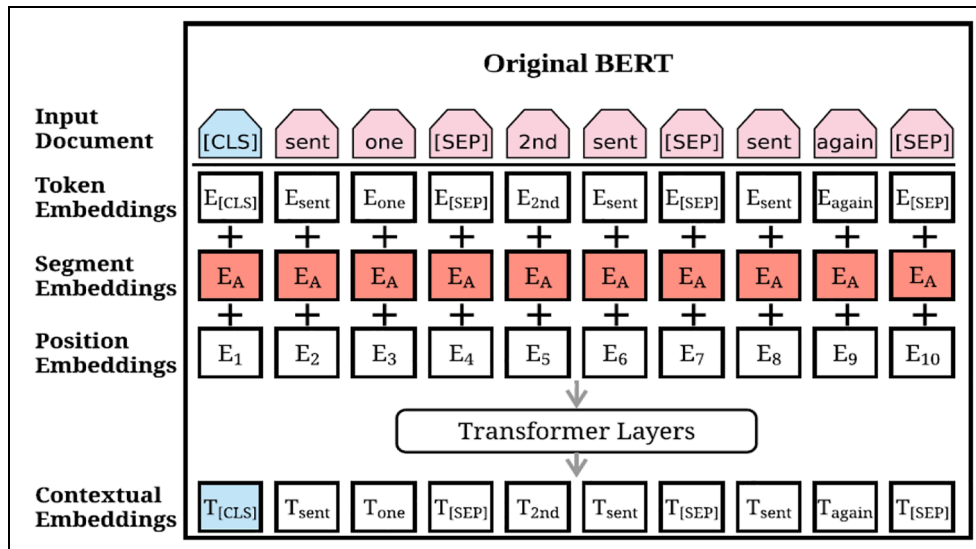


Figure 1. Architecture of the original BERT model [43]. The sequence on the top represents the input document, namely a sentence S or a pair of sentences packed together. Followed by the summation of token, segment and positional embeddings for each token. The latter embedding vectors are used as input to the Transformer encoder that generates contextual embeddings for each token.

understanding related tasks. The MT-DNN differs from the traditional pre-trained language models in that, instead of using only pre-training to learn text representations, MT-DNN model adds multi-task objectives in order to learn more general and pertinent representations. Moreover, Liu et al. [48] have introduced the distilled Multi-Task Deep Neural Networks model (MT-DNN_{KD}) using the same architecture as that of the MT-DNN model; however, the former is trained using the knowledge distillation method [49] in the multi-task learning settings.

2.3. BERT model architecture

The Bidirectional Encoder Representations from Transformers (BERT) model [19] is based on a multi-layer bidirectional transformer [50] with attention mechanisms. BERT is trained on large amount of source data (about 3300M words) from English Wikipedia and BookCorpus [51], using two unsupervised tasks including the masked language modelling and the next sentence prediction. Then, the pre-trained model can be applied to a new natural language processing task by adding a few layers to the source model, such as text classification [52], question/answering systems [19] and automatic text summarization [43]. The original architecture of BERT model is illustrated in Figure 1. Given the input S , which is a sequence of words (either a sentence or a pair of sentences packed together), the lexicon encoder maps S into a sequence of embeddings vectors, one for each word, built by summing the corresponding word, segment and positional embeddings. Then, the transformer encoder captures the contextual information for each word via self-attention and generates a sequence of contextual embeddings vectors for the input S . Two versions of BERT model have been trained, which are $BERT_{BASE}$ and $BERT_{LARGE}$, described as follows:

- $BERT_{BASE}$: $L = 12$, $H = 768$, $A = 12$, Number of parameters = 110M
- $BERT_{LARGE}$: $L = 24$, $H = 1024$, $A = 16$, Number of parameters = 340M

Where L , H and A denote, respectively, the number of layers, the hidden size and the number of self-attention heads. In all cases, the feed-forward/filter size is set to $4H$ (i.e. 3072 for the $H = 768$ and 4096 for $H = 1024$).

3. Proposed method

In this section, we present our method for generic extractive multi-document summarization. It consists of two main steps: BERT fine-tuning and multi-document summarization. In the first step, we fine-tune BERT model using both single-task and multi-task learning from the GLUE benchmark datasets [23]. In the second step, the fine-tuned BERT

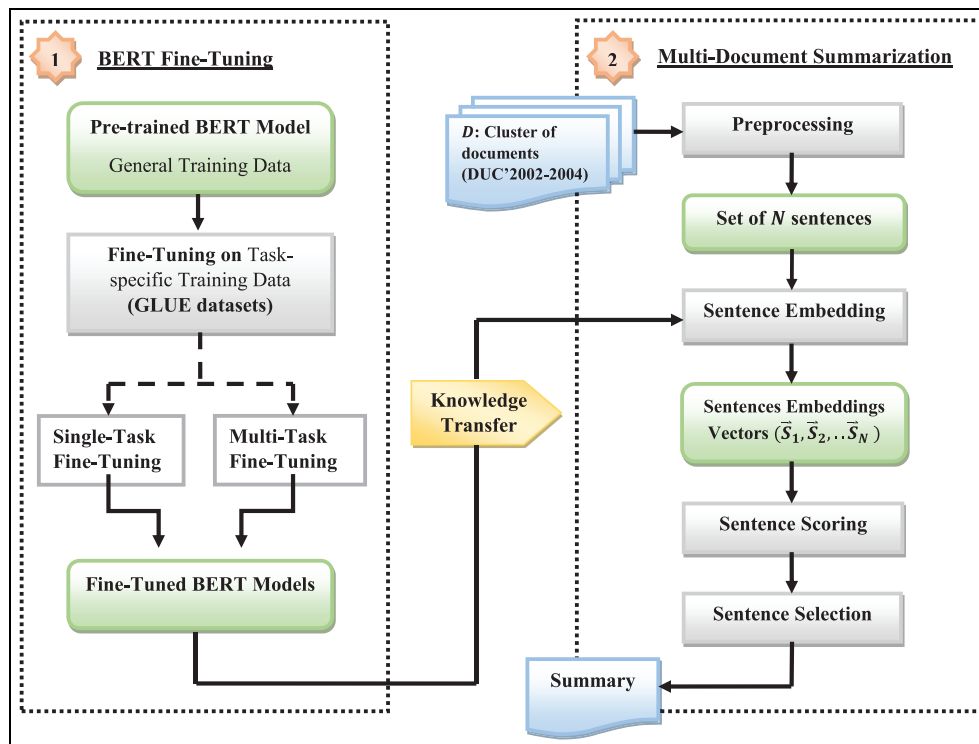


Figure 2. Architecture of the proposed method for extractive multi-document summarization.

models are used to extract sentence representations for text summarization. Based on the latter sentences representation, a score is assigned to each sentence in the cluster by linearly combining three sentence scoring measures (Sentence Content Relevance, Sentence Novelty and Sentence Position Scores). The flowchart of the proposed method is shown in Figure 2. In what follows, we present the two steps of the proposed method (cf. sections 3.1 and 3.2).

3.1. BERT fine-tuning

Since we introduce an unsupervised method for extractive multi-document summarization, we benefit from transfer learning abilities. In order to generate an embedding vector for each sentence of the input documents, we fine-tune the pre-trained BERT model on intermediate tasks, namely GLUE benchmark tasks, before applying it in text summarization task. We fine-tune BERT model using two methods, including BERT single-task fine-tuning and BERT multi-task fine-tuning. In the rest of this section, we present in detail the two BERT fine-tuning methods.

3.1.1. BERT Single-Task Fine-Tuning. In this step, the pre-trained model BERT is fine-tuned for each GLUE task using task-specific data. We use a model-based transfer learning approach, where the pre-trained model BERT is applied to a specific task by adding additional few layers to the source model with parameters of some layers stay the same while parameters of other layers are learned using the new task-specific data. In the following, we describe briefly the tasks used to fine-tune the pre-trained BERT model.

Single-Sentence Classification Task aims to predict a class label c for an input sentence S . We take as an example the SST-2 task [53], the probability that S is labelled as class c (i.e. the sentiment) is obtained by applying a logistic regression with softmax, described as follows

$$Probability(c|S) = softmax(W_{SST-2}^T \cdot x) \quad (1)$$

where x is the contextual semantic representation of the sentence S obtained using the pre-trained encoder BERT. In addition, W_{SST-2} is the task-specific parameter matrix.

Pairwise Text Classification Task: taking as an example the Natural Language Inference (NLI) task, given a pair of sentences (S_1, S_2) ; this task aims to find a logical relationship R between S_1 and S_2 , which means to predict whether the sentence S_2 is an entailment, contradiction or neutral according to the sentence S_1 . The probability distribution of the relation R is obtained by applying the Stochastic Answer Network model (SAN) [54], the state-of-the-art neural NLI, which uses multi-step reasoning rather than directly predicts the entailment given the input.

Text Similarity Task: given a pair of sentences (S_1, S_2) , the aim of this task is to measure the semantic similarity between these two sentences. Taking as an example the STS-B task [55], the similarity between S_1 and S_2 is computed as follows

$$\text{Similarity}(S_1, S_2) = W_{STS-B}^T \cdot x \quad (2)$$

where $\text{Similarity}(S_1, S_2) \in \mathbb{R}$, x represents the contextual semantic representation of the input pair of sentences (S_1, S_2) obtained using the pre-trained model BERT and W_{STS-B}^T is the task-specific parameter vector.

Relevance Ranking Task: We take as an example the Stanford Question Answering (QNLI) task [56]. Given a pair of a question and its candidate answer (Q, A) , the model ranks all the candidates answers in order to their relevance to the query. Where, the relevance ranking score is computed as follows:

$$\text{Relevance}(Q, A) = g(W_{QNLI}^T \cdot x) \quad (3)$$

where x is the contextual semantic representation of the pair (Q, A) obtained using the pre-trained encoder BERT and W_{QNLI} is the task-specific parameter matrix. For a given question Q , the candidates' answers are ranked according to their relevance scores that are obtained using equation (3).

3.1.2. BERT Multi-Task Fine-Tuning. In this step, we use the MT-DNN method [46] for BERT multi-task learning. We fine-tune the pre-trained BERT model simultaneously on four Natural Language Understanding tasks, including the single-sentence classification, pair-wise text classification, text similarity and relevance ranking, where each task has its own task-specific output layers (described in the previous section 3.1.1). The pre-trained BERT model is first used to initialise the parameters of the shared layers. Then, a mini-batch-based stochastic gradient descent is used to learn the model parameters, namely the parameters of shared layers and task-specific output layers. In detail, as shown in Algorithm 1 [48], the training samples from multiple tasks are packed into mini-batches denoted by b_t , where each b_t contains only the samples from task t . Then, in each epoch, a mini-batch b_t is selected and the model is updated according to the task-specific objective for the task t , denoted by L_t . This can lead to optimising the sum of all multi-task objectives.

Algorithm 1 BERT Multi-Task Fine-Tuning [47]

```

Initialise model parameters randomly
Initialise the shared layers using the pre-trained model BERT
Define the tasks:  $T$ 
Define the number max of epoch:  $epoch_{max}$ 
Begin
  for  $epoch \leftarrow 1$  to  $epoch_{max}$  do
    1. Merge all the datasets:
       $D = D_1 \cup D_2 \dots \cup D_T$  //  $D_t$  is the dataset of the task  $t$ 
    2. Shuffle  $D$ 
    for  $b_t$  in  $D$  do
      //  $b_t$  corresponds to a mini-batch of a task  $t$ 
      3. Compute task-specific loss  $L_t$ 
      4. Compute gradient
      5. Update model
    end for
  end for
End

```

3.2. Multi-document summarization

Extractive multi-document summarization task is defined as the process of selecting the most relevant sentences that represent the content of the entire cluster of documents. It consists of three main steps: sentence embedding, sentence scoring and sentence selection.

3.2.1. Sentence Embedding. Let denote D a cluster containing n documents $D = [d_1, d_2, \dots, d_n]$. First, we split each document d_i from the cluster D into sentences using the open-source software library for Advanced Natural Language Processing spaCy.¹ Then, we convert the words of the obtained sentences to lower case, as well as we remove special characters and redundant whitespace using the Natural Language Toolkit (NLTK²) and regular expressions. We obtain a cluster D of N sentences, formally denoted as $D = [S_1, S_2, \dots, S_N]$. Finally, we apply the fine-tuned BERT models (obtained in the first step) on the cluster of documents to be summarised in order to map each sentence S_i in the cluster D into an embedding vector S_i^D .

3.2.2. Sentence Scoring. In this step, we assign a score for each sentence S_i in the cluster D to measure its relevance. We combine linearly three scores, including the sentence content relevance, the sentence novelty and the sentence position. The latter scores are described as follows:

Sentence Content Relevance Score: formally, given a cluster of documents D that contains N sentences $D = [S_1, S_2, \dots, S_N]$, where each sentence S_i is represented by an embedding vector S_i^D . In order to obtain the sentence content relevance score for each sentence S_i in the cluster D . First, we build the centroid vector $\overrightarrow{C_D}$ for the cluster D by computing the mean vector of this cluster 'sentences embeddings vectors. We obtain real-valued vector in k -dimensional Euclidian space R^k , as shown in equation 4. Where $\overrightarrow{C_D}$ is the centroid embedding of the cluster D , N is the number of sentences in D and S_i^D refers to the embedding vector of the sentence S_i

$$\overrightarrow{C_D} = \frac{1}{N} \sum_{i=1}^N \overrightarrow{S_i^D} \quad (4)$$

Second, we compute the cosine similarity between each sentence embedding vector $\overrightarrow{S_i^D}$ and the centroid embedding vector $\overrightarrow{C_D}$ of the cluster D , as described in equation (5). The idea behind using the centroid to get the content relevance sentence score relies on the fact that the centroid aims to condense the meaningful information of the entire cluster in one vector. In addition hence, it is plausible that relevant sentences are those more similar to the centroid

$$score^{content\ Relevance}(S_i, D) = cosineSimilarity(\overrightarrow{S_i^D}, \overrightarrow{C_D}) = \frac{\overrightarrow{S_i^D} \cdot \overrightarrow{C_D}}{\|\overrightarrow{S_i^D}\| \cdot \|\overrightarrow{C_D}\|} \quad (5)$$

where $score^{content\ Relevance}$ represents the content relevance score of the sentence S_i in the cluster D , $\overrightarrow{C_D}$ is the centroid embedding vector of the cluster D and S_i^D is the embedding vector of the sentence S_i . The $score^{content\ Relevance}$ is bounded in $[0,1]$, where sentences with higher scores are considered more relevant.

Sentence Novelty Score: since we address in this work the multi-document summarization task, redundancy represents a critical problem. The risk to select sentences that convey the same information is more prominent in contrast to single-document summarization. Thus, in order to deal with redundancy and produce summaries with good information diversity, we use the sentence novelty metric [22]. It assigns a low score to the sentence when it is redundant and a high score when it is novel. Hence, to get the novelty score of a sentence S_i in the cluster D , we measure its similarity with all the sentences in the cluster D using the cosine similarity between their corresponding embedding vectors, as illustrated in equation 6. Where, S_i^D and S_k^D are the embedding vectors of the sentence S_i and the sentence S_k respectively, and N is the number of sentences in the cluster D

$$sim(S_i, S_k) = cosineSimilarity(\overrightarrow{S_i^D}, \overrightarrow{S_k^D}) = \frac{\overrightarrow{S_i^D} \cdot \overrightarrow{S_k^D}}{\|\overrightarrow{S_i^D}\| \cdot \|\overrightarrow{S_k^D}\|}, 1 \leq k \leq N, i \neq k \quad (6)$$

Then, if the maximum of the obtained similarities $sim(S_i, S_k)$, $1 \leq k \leq N$, $i \neq k$ is below a given threshold τ , then the sentence S_i is considered novel. However, when two sentences have almost the same similarity beyond a given

threshold, then the sentence with the higher value of the content relevance score gets a higher score of novelty. The sentence novelty score is calculated as follows

$$score^{novelty}(S_i, D) = \begin{cases} 1, & \text{if } \max(sim(S_i, S_k)) < \tau, 1 \leq k \leq N, i \neq k \\ 1, & \text{if } \max(sim(S_i, S_k)) > \tau, score^{contentRelevance}(S_i, D) > score^{contentRelevance}(S_l, D), \\ & l = \operatorname{argmax}(sim(S_i, S_k)), 1 \leq k \leq N, i \neq k \\ 1 - \max(sim(S_i, S_k)), & \text{otherwise,} \end{cases} \quad (7)$$

where $sim(S_i, S_k)$ represents the similarity between the sentence S_i and the other sentences in the cluster D , as described in equation (5). l is the argmax of the $sim(S_i, S_k)$ which means that l represents the index of the sentence that is the most similar to the sentence S_i . The $score^{novelty}$ and the $sim(S_i, S_k)$ are bounded in $[0,1]$. τ represents the threshold, in order to determine the best value of τ , we have tested several values of τ , namely the values comprised between $[0.5, 0.95]$ with constant steps of 0.05.

Sentence Position Score has been frequently applied in automatic text summarization [27,22], and it is considered as one of the effective methods for selecting relevant sentences, especially in newswire documents. The idea behind the sentence position hypothesis is that the first sentences of a document represent the most pertinent ones, and the importance decreases while we get further away from the beginning of the document. The sentence position in our case is used as a complementary handcrafted metric for sentence scoring. Formally, given D a cluster of n documents, and each document d consists of M sentences. The sentence position score is computed as described in equation (8)

$$score^{position}(S_i^d) = \max\left(0.5, \exp\left(\frac{-p(S_i^d)}{\sqrt[3]{M}}\right)\right) \quad (8)$$

where $score^{position}(S_i^d)$ represents the position score of a sentence S_i in a document d , $-p(S_i^d)$ is the i th position of S in d with $p(S_i^d)$ starting by 1, and M is the number of sentences in the document d . The obtained score is bounded in $[0.5, 1]$, assuming that the first sentences in a document are the most relevant. The sentence importance decreases when it occurs far from the leading sentences of the document, noticing that the score remains constant at the value of 0.5 after a number of sentences.

3.2.3. Sentence Selection. Finally, in order to get the final score of a sentence S_i in the cluster D , we combine linearly the three scores, namely the sentence content relevance (equation (5)), the sentence novelty (equation (7)) and the sentence position (equation (8)). Then, the top-ranked sentences are selected to form the summary with respect to a compression rate (pre-given summary length), assuming that relevant sentences are those that maximise the weighted sum of the three scores. Thus, the final score of a sentence S_i in a cluster D , denoted as $score^{final}(S_i, D)$, is formally defined in equation (9)

$$score^{final}(S_i, D) = \alpha * score^{contentRelevance}(S_i, D) + \beta * score^{novelty}(S_i, D) + \lambda * score^{position}(S_i^d) \quad (9)$$

where, $\alpha + \beta + \lambda = 1$ with $\alpha, \beta, \lambda \in [0, 1]$ with constant steps of 0.1.

4. Experimental results

In this section, we present a comparative analysis of the results obtained using the proposed method. Several experiments have been conducted to address the following issues: (1) Investigating the use of BERT fine-tuning using single-task as well as multi-task learning for sentence representation learning. (2) Determining the natural language understanding tasks that allow transferring useful knowledge to extractive text summarization task. In addition, (3) assessing the performance of the proposed method in contrast to the baseline and state-of-the-art methods. However, before presenting the experimental results, we provide a brief description of the datasets used, the evaluation measures, the experimental setup and the approaches used for comparative analysis.

4.1. Datasets

We evaluate the proposed method on three standard datasets for generic multi-document summarization, namely DUC (Document Understanding Conference) (2002-2004) datasets, created by the National Institute of Standards and Technology (NIST). DUC'2002 and DUC'2003 consist, respectively, of 59 and 30 clusters where each cluster contains

Table 1. A description of DUC'2002, DUC'2003 and DUC'2004 datasets [55].

Dataset	Domain	Clusters	Documents	Sentences	Tasks
DUC'2002	News	59	576	14,370	Generic single- and multi-document
DUC'2003	News	30	309	7691	Generic single- and multi-document
DUC'2004	News	50	500	13,135	Generic multi-document

DUC: document understanding conference.

Table 2. Descriptions and statistics of GLUE tasks [23].

Corpus	Task	#Train	#Dev	#Test	#Label	Domain
Single sentence classification						
CoLA	Acceptability	8.5k	1k	1k	2	Misc.
SST-2	Sentiment	67k	872	1.8k	2	Movie reviews
Pair-wise text classification						
RTE	NLI	2.5k	276	3k	2	News, Wikipedia
MNLI	NLI	393k	20k	20k	3	Misc.
QQP	Paraphrase	364k	40k	391k	2	Social QA questions
MRPC	Paraphrase	3.7k	408	1.7k	2	News
Text similarity						
STS-B	Similarity	7k	1.5k	1.4k	1	Misc.
Relevance ranking						
QNLI	QA/NLI	108k	5.7k	5.7k	2	Wikipedia

GLUE: general language understanding evaluation; CoLA: corpus of linguistic acceptability; SST: semantic textual similarity; RTE: recognising textual entailment; NLI: Natural Language Inference; MNLI: Multi-Genre Natural Language Inference; QQP: Quora Question Pairs; MRPC: Microsoft Research Paraphrase Corpus; STS-B: Semantic Textual Similarity Benchmark; QNLI: Question Answering Language Inference.

approximately 10 English newswire articles, distributed by TREC. Besides, Task 2 of DUC'2004 dataset comprises 50 clusters, where each cluster includes 10 documents, coming from the Associated Press and New York Times newswires. Table 1 summarises some basic statistics of these datasets.

For BERT fine-tuning, we have used the General Language Understanding Evaluation (GLUE) benchmark [23]; it is considered well-designed for evaluating the performance and generalisation of Natural Language Understanding (NLU) models that share general linguistic knowledge across a diverse set of related tasks. The GLUE benchmark represents a collection of multiple NLU tasks, including single-sentence classification, pairwise text classification, text similarity and relevance ranking, as summarised in Table 2. As well, it includes diverse datasets that are described in the following:

- Corpus of Linguistic Acceptability (*CoLA*) [57] and Stanford Sentiment Treebank (*SST-2*) [53] are both single binary sentence classification tasks. The main aim of *CoLA* is to predict the linguistic plausibility of an English sentence. While *SST-2* aims to classify the sentiment of a set of sentences extracted from movies reviews to positive and negative ones.
- The Semantic Textual Similarity Benchmark (*STS-B*) [55] represents a regression task that predicts the semantic similarity score between a pair of sentences.
- Recognising Textual Entailment (*RTE*) [58] and Multi-Genre Natural Language Inference (MNLI) [59] are both language inference tasks. Given a pair of sentences, the objective is to predict whether the second sentence is an entailment, contradiction or neutral according to the first one.
- Quora Question Pairs³ (*QQP*) dataset and Microsoft Research Paraphrase Corpus (*MRPC*) [60] are pairwise text classification tasks; the goal is to predict if two sentences are semantically equivalent.
- Question Answering Language Inference (*QNLI*) is a version of the Stanford Question Answering Dataset (SQuAD) [56] that is defined as a binary classification task in GLUE. Given a query, the goal is to predict the relevant answer to this query.

4.2. Evaluation measures

For evaluation, we have adopted the widely used method in evaluating automatic text summarization systems, namely ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [24], particularly ROUGE- N (ROUGE-1 (R-1), ROUGE-2 (R-2) and ROUGE-4 (R-4)). ROUGE, as it is defined in equation (10), measures the similarity between system summaries and a collection of summaries models (human summaries) based on the n -gram comparison and overlap. We have calculated ROUGES scores with the ROUGE toolkit (version 1.5.5), adopting the same ROUGE settings⁴ that are used on DUC datasets for generic extractive multi-document summarization

$$ROUGE - N = \frac{\sum_{S \in (\text{ReferenceSummary})} \sum_{N\text{-gram} \in (S)} Count_{match}(N - gram)}{\sum_{S \in (\text{ReferenceSummary})} \sum_{N\text{-gram} \in (S)} Count(N - gram)} \quad (10)$$

where N is the length of $N - gram$ and $Count_{match}(N - gram)$ is the maximum number of $N - grams$ that occur in both gold summary and candidate summary. ROUGE-1 and ROUGE-2 are considered as the most used ROUGE measures, and they calculate, respectively, the number of overlapping unigrams and bigrams. While ROUGE-L (R-L) evaluates the fluency of the summary, it is based on the Longest Common Subsequence (LCS) that takes into account the sentence-level structure similarity. Let us denote X a candidate summary and Y a gold summary that contains n words. ROUGE-L is calculated as follows

$$ROUGE - L = \frac{LCS(X, Y)}{n} \quad (11)$$

where $LCS(X, Y)$ is the length of the longest subsequence of X and Y .

4.3. Experimental setup

The proposed method has been developed using Python and based on the PyTorch implementation of BERT model,⁵ Multi-Task Deep Neural Networks (MT-DNN) model,⁶ and Transformers.⁷

For a complete comparison, we explore the potential of BERT model using three approaches: (1) feature-based approach, (2) BERT single-task fine-tuning and (3) BERT multi-task fine-tuning. In the feature-based approach, fixed features are extracted using the pre-trained encoder BERT and directly used as input features for the task at hand. In the second approach, a simple output layer is added to the pre-trained model, where all the parameters are jointly fine-tuned using task-specific data. In the third approach, BERT is fine-tuned simultaneously on four natural language processing tasks. All the experiments were performed based on tasks from GLUE benchmark datasets using BERT_{BASE} and BERT_{LARGE}. BERT_{BASE} model is designed to embed a sentence into 768-dimensional vectors, while BERT_{LARGE} provides sentence embeddings vectors of 1024 dimensions.

For BERT single-task fine-tuning, the only new parameters are those of the new additional output layer related to the downstream task. Most hyper-parameters are the same as in the pre-training step [19], except for the batch size, number of training epochs and the learning rate. Generally, the optimal values of these hyper-parameters depend on task-specific data; however, we found a possible range of values that work well across all tasks. For BERT_{BASE}, we use a batch size of 32, 3 epochs and a learning rate of 5e-5 for all the GLUE tasks. For BERT_{LARGE}, we use a batch size of 16, 5 epochs and a learning rate of 5e-5 for all the GLUE task. For BERT Multi-Task fine-tuning, we followed the same setup of the MT-DNN model [46]. We used Adamax optimizer [61] with a learning rate of 5e-5 and a batch size of 32 and 16 for BERT_{BASE} and BERT_{LARGE}, respectively. The number of epochs was set to 3 for BERT_{BASE} and 5 for BERT_{LARGE}.

All our experiments as well as BERT fine-tuning have been performed using an Intel(R) Xeon(TM) CPU 3.00 GHz server equipped with Nvidia Tesla K40c GPU having 12 Go of RAM. The checkpoints of the fine-tuned BERT models have been stored and used in an off-line mode to generate the embeddings vectors for all the input clusters sentences contained in DUC'2002–2004 datasets. It is worth mentioning that BERT models Fine-Tuning time depends on the task at hand and its size (BASE or LARGE). For instance, the time of BERT_{BASE} and BERT_{LARGE} Fine-Tuning on RTE task is approximately 3 and 11 min, respectively. In addition, the execution time of summarising each dataset, using the fine-tuned BERT_{BASE} (CoLA_{BASE}, RTE_{BASE}, ...) and BERT_{LARGE} (CoLA_{LARGE}, RTE_{LARGE}, ...) models, is depicted in Table 3.

As we have mentioned in section 3, the final score of a sentence S_i in the cluster D is calculated by summing three weighted scores, including the sentence content relevance, the sentence novelty and the sentence position. We have studied the sensitivity performance of the proposed method according to four parameters, τ , α , β and λ . Thus, in order to optimise these four parameters, we have performed the K-fold cross-validation method using the three DUC (2002-2004)

Table 3. Execution time in minute for summarising each dataset DUC'2002–2004 using the fine-tuned BERT_{BASE} and BERT_{LARGE} models.

	DUC'2002	DUC'2003	DUC'2004
BERT _{BASE}	04:46.8	01:57.7	03:30.6
BERT _{LARGE}	04:57.2	02:05.4	03:50.1

DUC: document understanding conference; BERT: bidirectional encoder representations from transformers.

datasets. The used values of the threshold τ are comprised between $[0.5, 0.95]$ with constant steps of 0.05. The parameters α , β and λ are comprised between $[0, 1]$ with constant steps of 0.1 and $\alpha + \beta + \lambda = 1$. Generally, the best values of the threshold $\tau \in [0.8, 0.95]$, the best values of $\alpha \in [0.7, 0.9]$, and the best values of β and $\lambda \in [0.1, 0.3]$. Furthermore, we performed the paired Student's *t*-test for statistical significance testing and attached a superscript to the performance number in the tables when the *p* value is < 0.05 .

4.4. Results

First, the experiments are conducted to investigate the impact of using transfer learning from BERT model to improve the task of extractive multi-document summarization. The main goal of these experiments is to answer the following research question: *Given the recent successes of BERT fine-tuning on intermediate-tasks, which tasks allow to transfer useful knowledge to extractive multi-document summarization?*

To answer the latter question, we have applied BERT on the three summarization datasets DUC (2002–2004) by adopting two approaches, described as follows:

- Approach 1: we have used the pre-trained BERT_{BASE} and BERT_{LARGE} models with no fine-tuning (feature-based approach);
- Approach 2: we have fine-tuned BERT_{BASE} and BERT_{LARGE} models for each GLUE task on task-specific data (CoLA, SST-2, RTE, MNLI, QQP, MRPC, STS-B and QNLI);

Table 4 illustrates the overall obtained results on the three multi-document summarization datasets DUC'2002, DUC'2003 and DUC'2004 in terms of ROUGE-1, ROUGE-2 and ROUGE-4 recall scores. The results show clearly that applying BERT_{BASE} and BERT_{LARGE} models on extractive multi-document summarization without fine-tuning leads to poor performance on the three DUC (2002–2004) datasets. These results can be due to the fact that BERT model is trained on a masked language model where the output vectors are grounded to tokens instead of sentences, while most of extractive multi-document summarization methods manipulate sentence-level representations. Moreover, the overall obtained results demonstrate that fine-tuning BERT model yields significant improvements over its pre-trained model (without fine-tuning). For instance, the R-1 score on DUC'2004 has been improved from 28.02% to 39.04% after fine-tuning BERT_{BASE} using the RTE dataset of the GLUE benchmark.

For DUC'2002 dataset, and based on R-1 measure, RTE_{BASE} model has achieved the best performance that leads to significant improvements over most other models, including its variant RTE_{LARGE}. Furthermore, SST-2_{BASE} model has also obtained a good result that significantly outperforms all the other models, except RTE_{LARGE} and QNLI_{LARGE} models. Regarding the R-2 score, the best performance is obtained by RTE_{BASE}, SST-2_{BASE} and MNLI_{LARGE}, where the differences between them and the other models are significant, except RTE_{LARGE} and QNLI_{LARGE}. For the R-4 measure, RTE_{BASE} has also achieved the best performance with significant improvements over most other models, except its variant RTE_{LARGE}, MNLI_{LARGE} and QNLI_{LARGE} models.

For DUC'2003 corpus, the results show that MNLI_{LARGE}, CoLA_{LARGE}, RTE_{BASE}, RTE_{LARGE}, SST-2_{BASE} and QNLI_{LARGE} have achieved the best performance and significantly outperformed most other models regarding R-1 and R-2 scores. Furthermore, MNLI_{BASE} and SST-2_{LARGE} showed better performances than QQP_{BASE}, QQP_{LARGE}, MRPC_{BASE}, MRPC_{LARGE}, STS-B_{BASE} and STS-B_{LARGE} models. For the R-4 score, MNLI_{LARGE}, CoLA_{LARGE}, RTE_{BASE} and RTE_{LARGE} models obtain the best performance and significantly outperform most other models, except CoLA_{BASE}, MNLI_{BASE} and QNLI_{LARGE} models. Moreover, QQP_{LARGE}, MRPC_{LARGE} and QNLI_{BASE} have achieved good performance in comparison to QQP_{BASE}, MRPC_{BASE}, STS-B_{BASE} and STS-B_{LARGE} models.

For DUC'2004 dataset, and based on the three scores (R-1, R-2 and R-4), the obtained results show that fine-tuning BERT-BASE and BERT_{LARGE} on Pairwise Text Classification Task using RTE dataset has achieved the best

Table 4. Comparison results of the different tasks used for BERT Fine-Tuning on the three extractive multi-document summarization datasets.

Models	DUC'2004				DUC'2003				DUC'2002			
	R-1	R-2	R-4	R-4	R-1	R-2	R-4	R-4	R-1	R-2	R-4	R-4
BERT with No Fine-Tuning												
BERT _{BASE} ¹	28.02 ^{2,15}	4.12	0.38	0.47	28.18 ^{2,15}	4.53 ²	0.47	0.47	28.85 ^{2,15}	4.03 ²	0.28	0.28
BERT _{LARGE} ²	23.52 ¹⁵	3.46	0.3	0.24	22.21	2.99	0.24	0.24	22.94 ¹⁵	3.24	0.27	0.27
BERT Single Sentence Classification Task Fine-Tuning												
CoLA ₃	37.16 ^{-2,11-17}	8.09 ^{-2,11-17}	1.11 ^{-2,13-16}	1.35 ^{-2,113-15}	35.83 ^{-2,11-16}	8.26 ^{-2,113-15}	1.35 ^{-2,113-15}	1.35 ^{-2,113-15}	35.34 ^{-2,4,11-17}	7.29 ^{-2,11-16}	1.08 ^{-2,13-16}	1.08 ^{-2,13-16}
CoLA ₄	37.74 ^{-2,11-17}	8.58 ^{-2,11-17}	1.28 ^{-2,113-16}	1.61 ^{-2,11-16}	36.74 ^{-2,11-17}	8.48 ^{-2,11-17}	1.61 ^{-2,11-16}	1.61 ^{-2,11-16}	33.18 ^{-2,113-16}	6.65 ^{-2,113-16}	0.96 ^{-2,14-15}	0.96 ^{-2,14-15}
SST-2 _{BASE}	37.07 ^{-2,11-17}	8.08 ^{-2,11-17}	1.12 ^{-2,11-17}	1.39 ^{-2,13,15-17}	36.47 ^{-2,11-17}	8.52 ^{-2,11-17}	1.39 ^{-2,13,15-17}	1.39 ^{-2,13,15-17}	37.01 ^{-4,6,9-17}	7.81 ^{-2,4,6,9,11-17}	0.95 ^{-2,13-16}	0.95 ^{-2,13-16}
SST-2 _{LARGE}	36.52 ^{-2,11-17}	8.29 ^{-2,11-17}	1.23 ^{-2,11-17}	1.4 ^{-2,13,15}	34.01 ^{-2,113-16}	7.54 ^{-2,13,15}	1.4 ^{-2,13,15}	1.4 ^{-2,13,15-17}	34.17 ^{-2,113-16}	6.54 ^{-2,113-16}	0.81 ^{-2,13-16}	0.81 ^{-2,13-16}
BERT Pairwise Text Classification Task Fine-Tuning												
RTE _{BASE}	39.04 ^{-6,9-19}	9.31 ^{-3,5-6,9-18}	1.47 ^{-3,5,11-17}	1.63 ^{-2,11-17}	36.51 ^{-2,11-17}	8.57 ^{-2,11-16}	1.63 ^{-2,11-16}	1.63 ^{-2,11-17}	38 ^{-6,8-18}	8.21 ^{-4,6,9,11-18}	1.25 ^{-4,6,9,11-16}	1.25 ^{-4,6,9,11-16}
RTE _{LARGE}	38.77 ^{-3,5-6,9-18}	8.92 ^{-3,5,9,11-17}	1.37 ^{-2,5,11-17}	1.56 ^{-2,11-16}	36.54 ^{-2,11-17}	8.45 ^{-2,11-17}	1.56 ^{-2,11-16}	1.56 ^{-2,11-17}	35.86 ^{-2,4,11-17}	7.63 ^{-2,4,6,9,11-17}	1.2 ^{-2,4,9,11-16}	1.2 ^{-2,4,9,11-16}
MNLI _{BASE}	36.23 ^{-2,113-16}	7.71 ^{-2,113-17}	1.24 ^{-2,13-16}	1.51 ^{-2,13,15-16}	35.23 ^{-2,113-16}	7.82 ^{-2,13,15}	1.51 ^{-2,13,15-16}	1.51 ^{-2,113,15-16}	34.21 ^{-2,113-16}	6.61 ^{-2,113-16}	0.69 ^{-2,15}	0.69 ^{-2,15}
MNLI _{LARGE}	37.34 ^{-2,11-17}	8.45 ^{-2,9,11-17}	1.28 ^{-2,113-16}	1.79 ^{-2,11-17}	37.45 ^{-2,6,11-17}	9.17 ^{-2,9,11-16}	1.79 ^{-2,11-16}	1.79 ^{-2,11-17}	35.03 ^{-2,4,6,9,11-17}	7.72 ^{-4,6,9,11-18}	1.17 ^{-2,9,11-16}	1.17 ^{-2,9,11-16}
QQP _{BASE}	32.29 ^{-2,13-16}	6.48 ^{-2,13,15-16}	0.98 ^{-2,13,15-16}	0.99 ^{-2,13,15}	31.63 ^{-2,13,15}	6.46 ^{-2,13,15}	0.99 ^{-2,13,15}	0.99 ^{-2,15}	31.01 ^{-2,13-16}	5.58 ^{-2,15}	0.62 ⁻²	0.62 ⁻²
QQP _{LARGE}	34.77 ^{-2,113-16}	7.16 ^{-2,13-16}	1 ^{-2,13,15-16}	1.17 ^{-2,13,15}	32.81 ^{-2,13,15}	6.99 ^{-2,13,15}	1.17 ^{-2,13,15}	1.17 ^{-2,15}	32.91 ^{-2,13-16}	6.37 ^{-2,13-16}	0.87 ^{-2,15}	0.87 ^{-2,15}
MRPC _{BASE}	26.65 ^{2,15}	5.08 ^{-2,15}	0.54 ²	0.87 ⁻²	25.56 ^{2,15}	5.12 ²	0.87 ⁻²	0.87 ⁻²	27.28 ^{2,15}	5.37 ^{-2,15}	0.64 ⁻²	0.64 ⁻²
MRPC _{LARGE}	29.68 ^{2,13,15}	6.27 ^{-2,13,15-16}	0.81 ^{-2,13,15}	1.22 ^{-2,13,15}	29.32 ^{2,13,15}	6.4 ^{-2,13,15}	1.22 ^{-2,13,15}	1.22 ^{-2,13,15}	27.53 ^{2,15}	5.14 ^{-2,15}	0.54	0.54
BERT Text Similarity Task												
Fine-Tuning												
STS-B _{BASE} ¹⁵	19.29	3.53	0.34	0.55 ²	19.78	3.98 ²	0.34	0.55 ²	20.21	3.66	0.38	0.38
STS-B _{LARGE} ¹⁶	28.12 ²	5.45 ⁻²	0.73 ^{2,15}	0.97 ^{-2,15}	29.42 ^{-2,13,15}	6.12 ^{-2,13,15}	0.73 ^{2,15}	0.97 ^{-2,15}	27.41 ^{2,15}	5.18 ^{-2,15}	0.62 ⁻²	0.62 ⁻²
BERT Relevance Ranking Task Fine-Tuning												
QNLI _{BASE} ¹⁷	35.08 ^{-2,113-16}	7.02 ^{-2,13,15-16}	1.04 ^{-2,13,15-16}	1.18 ^{-2,13,15}	33.62 ^{-2,13,15-16}	7.49 ^{-2,13,15}	1.04 ^{-2,13,15-16}	1.18 ^{-2,13,15}	32.32 ^{-2,113,15-16}	6.45 ^{-2,113,15-16}	1.03 ^{-2,13-16}	1.03 ^{-2,13-16}
QNLI _{LARGE} ¹⁸	36.95 ^{-2,11-17}	8.45 ^{-2,11-17}	1.35 ^{-3,11-17}	1.39 ^{-2,11-16}	35.98 ^{-2,11-16}	8.51 ^{-2,11-16}	1.35 ^{-3,11-17}	1.39 ^{-2,13,15-17}	35.95 ^{-2,4,6,9-17}	7.47 ^{-2,11-16}	1.13 ^{-2,9,11,13-16}	1.13 ^{-2,9,11,13-16}
BERT Multi-Task Fine-Tuning												
MT _{BASE} ¹⁹	37.61 ^{-2,11-17}	8.78 ^{-2,9,11-17}	1.29 ^{-2,13-16}	1.73 ^{-2,113-17}	36.79 ^{-2,11-17}	9.07 ^{-2,9,11-16}	1.29 ^{-2,13-16}	1.73 ^{-2,113-17}	37.82 ^{-2,4,6,9-17}	8.32 ^{-2,4,6,9,11-16}	1.32 ^{-4,6,9,11-16}	1.32 ^{-4,6,9,11-16}
MT _{LARGE} ²⁰	39.08 ^{-6,9-19}	9.68 ^{-6,9-18}	1.58 ^{-3,9,11-17}	1.76 ^{-2,113-17}	37.75 ^{-2,6,11-17}	9.09 ^{-2,9,11-16}	1.58 ^{-3,9,11-17}	1.76 ^{-2,113-17}	38.62 ^{-4,6,8-19}	8.64 ^{-2,4,6,9,11-16}	1.21 ^{-4,6,9,11-16}	1.21 ^{-4,6,9,11-16}

DUC: document understanding conference; BERT: bidirectional encoder representations from transformers; SST: semantic textual entailment; MNLI: Multi-genre natural language inference; QQP: quora question pairs; MRPC: microsoft research paraphrase corpus; STS-B: semantic textual similarity benchmark; QNLI: question answering language inference. For denoting statistical significance results, the superscript numbers indicate significant improvement (p value < 0.05) over the task that has the same superscript number attached. The interval $i-j$ indicates a significant improvement over models that have a superscript number attached ranging from i to j .

performance and leads to significant improvements over most other GLUE tasks, except R-2 and R-4 scores of CoLA_{LARGE} and R-4 score of QNLI_{LARGE} model. Moreover, fine-tuning BERT_{BASE} and BERT_{LARGE} on CoLA, SST-2, MNLI and QNLI datasets have also scored good results and provides significant improvements over QQP, MRPC and STS-B datasets. Furthermore, for the most used GLUE tasks, BERT_{LARGE} outperforms BERT_{BASE}, but the differences between them are not statistically significant.

To summarise, the overall obtained results on the three DUC (2002–2004) datasets demonstrated that fine-tuning the pre-trained BERT model on coLA, SST-2, RTE, MNLI and QNLI datasets allows transferring useful and suitable knowledge for extractive multi-document summarization. Moreover, the comparison results show that the performance provided by the tasks mentioned above (CoLA, SST-2, RTE, MNLI and QNLI) does not depend on the used text summarization dataset, which means the best performing models are always the same for the three datasets (DUC'2002, DUC'2003 and DUC'2004).

Second, the experiments are conducted to address the following research question: (2) *Does BERT multi-task fine-tuning improve the performance of extractive multi-document summarization task?*

The last block of Table 4 reports the obtained results by the fine-tuned BERT models using multi-task learning: MT_{BASE} and MT_{LARGE} models. Where, MT_{BASE} and MT_{LARGE} models stand for the fine-tuned BERT_{BASE} and BERT_{LARGE} models, respectively. For DUC'2002, the MT_{BASE} model has outperformed most of the other models for all evaluation measures (R-1, R-2 and R-4) significantly, except for the R-1 score of RTE_{BASE} model. Moreover, for DUC'2003 corpus, and based on the three scores R-1, R-2 and R-4, MT_{BASE} has achieved better results than all the other models, except the MNLI_{LARGE} model, where the differences between them are not statistically significant. Finally, for DUC'2004 dataset, MT_{BASE} model has achieved better results than most of the other models for most used evaluation measures; however, it was not able to surpass RTE_{BASE} and RTE_{LARGE} models where the differences between them are statistically significant. Nevertheless, the results show that MT_{LARGE} outperforms all the other models on the three DUC (2002–2004) datasets for all the used evaluation measures, leading to significant improvements over most of the used fine-tuned BERT models. These results reveal that the learned representations, using BERT multi-task fine-tuning (MT_{BASE} and MT_{LARGE} models), are more effective for extractive multi-document summarization.

4.5. Comparison evaluation with state-of-the-art methods

To validate the robustness of the proposed method, we compare its performance with existing state-of-the-art methods for generic multi-document summarization task [62], using the evaluation measures R-1, R-2, R-4 and R-L. For this purpose, we used the repository developed by [63], which contains a set of summaries generated by several systems for DUC'2004 dataset. The generated summaries are published in GitHub⁸ while the source codes of most of these systems are available in Sumy repository.⁹ The obtained results are shown in Table 5. The first block of the table summarises the results of the systems used for the comparison, while the second block shows the best results obtained by our method.

On the one hand, the first set of analysis aims to highlight the importance of documents representations for extractive text summarization task. For this purpose, we compare our method with two state-of-the-art methods, including the Centroid_BOW method that uses bag-of-words representations and the Centroid_WordEmbedding method that exploits the capabilities of word embedding models. The obtained results show that our method using the fine-tuned BERT model, considered as a sentence embedding model, has significantly outperformed the centroid method based on BOW representations for most evaluation measures. Furthermore, the overall comparison results demonstrate that our method achieved far better performances than the Centroid_WordEmbedding method in terms of R-1 measure. For instance, with the MT_{LARGE} and RTE_{BASE} models, we obtained an increment of 1.17% and 1.14%, respectively. Besides, in terms of R-2 and R-4 evaluation measures, our method using MT_{LARGE} model outperformed both the Centroid_BOW and Centroid_WordEmbedding methods. Therefore, the obtained results prove the effectiveness of both BERT fine-tuning methods for text summarization.

On the other hand, the second set of evaluations is conducted to compare our method with several baselines and state-of-the-art methods for extractive generic multi-document summarization. For the R-1 evaluation measure, our method using RTE_{BASE}, RTE_{LARGE} and MT_{LARGE} models has outperformed all the baseline methods that are used for comparison, except DPP system and show significant improvements over LexRank, KLSum, CLASSY04 and Centroid_BOW methods. Moreover, it has obtained far better performance than GBN-MDS and PG-MMR deep learning-based methods. Furthermore, our method using CoLA_{LARGE} and MT_{BASE} has also achieved good results that outperformed PG-MMR method and leads to significant improvements over LexRank and Centroid_BOW methods. In addition, in terms of R-2 and R-4 evaluation measures, our method using MT_{LARGE} model has also outperformed all baseline and state-of-the-art methods used for comparison, except ISCISum and RegSum systems and obtain significant improvements over LexRank and Centroid_BOW methods. Even though ISCISum and RegSum systems have achieved better performance than our method, the difference is

Table 5. Systems performance on DUC'2004, using state-of-the-art methods and the proposed method. The highest performance for each of the group of methods is printed in boldface.

Methods	R-1	R-2	R-4	R-L
Baseline and state-of-the-art methods				
LexRank ¹ [64]	35.54	7.47	0,82	31,1
KLSum ² [65]	37.68	8.54	1,27	32,93
CLASSY04 ³ [66]	37.32	8.96	1,52	32,26
ISCIsum ⁴ [67]	38.12	9.77	1,73	33,62
OCCAMS_V ⁵ [68]	38.05	9.69	1,32	34,29
RegSum ⁶ [69]	38.27	9.73	1,61	34,13
Submodular ⁷ [70]	38.83	9.29	1,38	33,77
DPP ⁸ [71]	39.46	9.58	1,56	34,93
Centroid_BOW ⁹ [2]	36.03	7.9	1,19	31,21
Centroid_WordEmbedding [‡] [8]	37.91	9.53	1,56	–
GBN-MDS [‡] [34]	38.23	9.48	–	–
PG-MMR [‡] [35]	36.42	9.36	–	–
Proposed Method				
CoLA _{LARGE}	37.74 ^{1,9}	8.58 ¹	1.28 ¹	33.03 ^{1,9}
MT _{BASE}	37.61 ^{1,9}	8.78 ¹	1.29 ¹	33.05 ^{1,9}
RTE _{LARGE}	38.77 ^{1,3,9}	8.92 ^{1,9}	1.37 ¹	33.9 ^{1,3,9}
RTE _{BASE}	39.04 ^{1,2,3,9}	9.3 ^{1,9}	1.47 ¹	34.04 ^{1,2,3,9}
MT _{LARGE}	39.08 ^{1,2,3,9}	9.68 ^{1,9}	1.58 ¹	34.2 ^{1,2,3,9}

For denoting statistical significance results, the superscript numbers indicate significant improvement (p value < 0.05) over the task used for BERT Fine-Tuning that has the same superscript number. The interval $i - j$ indicates significant improvement over models that have a superscript number ranging from i to j .

[‡]The results of models are taken from their original articles.

not statistically significant. Besides, our method using CoLA_{LARGE}, MT_{LARGE}, RTE_{BASE} and RTE_{LARGE} models has also shown better results than most other baselines and state-of-the-art methods. For the R-L evaluation measure, the overall results show that our method significantly outperformed LexRank, KLSum, CLASSY04 and Centroid_BOW methods. Moreover, it achieved better performances than ISCIsum, RegSum, and Submklmodular systems.

5. Discussion

The pre-trained BERT model has shown outstanding performances in several natural language processing applications. However, the obtained results show that its application in extractive text summarization without fine-tuning does not achieve good results and lags significantly behind the performance of the most fine-tuned BERT models. The latter can be explained by the fact that BERT model is trained on a masked language model where the output vectors are grounded to tokens instead of sentences, while sentence representation is the cornerstone of our unsupervised extractive multi-document summarization methods. Therefore, to improve the performance of sentence representation learning, we fine-tuned BERT model on intermediate tasks before applying it in our method using both single-task and multi-task fine-tuning. Hence, we used GLUE benchmark tasks, including the natural language inference (NLI) tasks (RTE, MNLI and QNLI), the similarity and paraphrasing tasks (STS-B, MRPC and QQP), and the single sentence classification tasks (CoLA and SST-2).

The overall obtained results show that fine-tuning BERT model on NLI tasks (RTE, MNLI and QNLI) has achieved the best performances, where fine-tuning BERT on the RTE dataset outperforms all others. Moreover, fine-tuning BERT on MNLI or QNLI has obtained comparable performance. These results might be due to the fact that the latter are among high-quality labelled corpus, designed for textual entailment tasks, which might constitute a class of problems relevant to text summarization task [72,73]. Moreover, Fine-tuning BERT on single sentence classification tasks (CoLA and SST-2) has shown promising results. Even though CoLA task is a challenging task with much smaller in-domain data than other tasks, it was able to transfer useful knowledge to the text summarization task. The latter can be explained by the fact that CoLA aims to predict the linguistic acceptability of a sentence, which may require a model that captures syntactic and semantic information of the sentences. Although fine-tuning BERT on sentence similarity task (STS-B) and paraphrasing tasks (QQP and MRPC) has outperformed the pre-trained model, these tasks have led to poor performance.

Besides, it is important to highlight that our method has achieved better performance when BERT is fine-tuned on tasks that only have small amounts of training datasets than those of large training datasets. Even if the tasks belong to the same type, for example, the two NLI tasks RTE vs MNLI, and the single sentence classification tasks CoLA vs SST-2. In addition, we also note that the performance of BERT_{LARGE} model become unstable when we fine-tune it on small training datasets such as RTE, CoLA, and SST-2, which confirms the findings of [74].

Furthermore, the results show that BERT multi-task fine-tuning has outperformed models that are fine-tuned on single-tasks for text summarization. Hence, multi-task learning is an important paradigm to learn universal sentence representations. This stems from the fact that the latter improve generalisation performance by taking advantage of the training signals of related tasks [13,47,48].

Finally, the overall comparison results show that our method has achieved significant improvements over several baselines and state-of-the-art methods. This finding proves the effectiveness of fine-tuning BERT model on intermediate tasks that improves sentence representation learning.

6. Conclusion

In this article, we introduced an unsupervised method for extractive multi-document summarization that consists of two main tasks: BERT fine-tuning and multi-document summarization. First, we fine-tune the pre-trained BERT model using two different approaches, namely BERT single-task fine-tuning and BERT multi-task fine-tuning. Second, we use the obtained fine-tuned BERT models to represent the sentences of the input documents. Then, based on the latter representations, we assign a score to each sentence in the cluster of documents by combining three scores including sentence content relevance, sentence novelty and sentence position where the top-ranked sentences are iteratively selected to form the final summary.

To assess the effectiveness of the proposed multi-document summarization method, we have performed an extensive experimental analysis. Indeed, several experiments have been conducted on the standard DUC benchmark datasets (2002–2004) to investigate the GLUE tasks that allow transferring useful knowledge to multi-document summarization task. The overall obtained results showed that fine-tuning BERT model on CoLA, SST-2, RTE, MNLI and QNLI tasks had achieved the best performance and improved the results significantly. In addition, according to ROUGE evaluation measures, the results demonstrate that fine-tuning BERT model using multi-task learning has improved the performance of the extractive multi-document summarization task. Moreover, the overall comparison results show that our method obtained promising results; it has significantly outperformed several baseline methods and achieved far better performances than two state-of-the-art deep learning-based methods (GBN-MDS, PG-MMR) for most evaluation measures.

The use of fine-tune BERT models in extractive multi-document summarization has shown to be effective. Hence, in future work, we plan to investigate their efficiency on other summarization tasks such as query-focused summarization. In addition, we plan to explore the potential of BERT model for language generation.


Declaration of conflicting interests


The author(s) declared no potential conflicts of interest with respect to the research, authorship and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iDs

Salima Lamsiyah  <https://orcid.org/0000-0001-8789-5713>

Abdelkader El Mahdaouy  <https://orcid.org/0000-0003-4281-2472>

Notes

1. <https://spacy.io/>
2. <https://www.nltk.org/>
3. data.quora.com/First-Quora-Dataset-Release-Question-Pairs
4. ROUGE-1.5.5 with parameters '-n 4 -m -l 100 -c 95 -r 1000 -f A -p 0.5 -t 0', 'l' is used for length limit
5. <https://github.com/huggingface/pytorch-pretrained-BERT>
6. <https://github.com/namisan/mt-dnn>
7. <https://github.com/huggingface/transformers>

8. <https://github.com/stuartmackie/duc-2004-rouge>
9. <https://github.com/miso-belica/sumy>

References

- [1] Luhn HP. The automatic creation of literature abstracts. *IBM J Res Develop* 1958; 2(2): 159–165.
- [2] Radev DR, Jing H, Sty's M et al. Centroid-based summarization of multiple documents. *Inform Proces Manag* 2004; 40(6): 919–938.
- [3] Le QV and Mikolov T. Distributed representations of sentences and documents. In: *Proceedings of the 31th international conference on machine learning, ICML*, Beijing, China, 21–26 June 2014, pp. 1188–1196. New York: ACM.
- [4] Mikolov T, Sutskever I, Chen K et al. Distributed representations of words and phrases and their compositionality. In: *Advances in neural information processing systems*, Lake Tahoe, Nevada, United States, 5–8 December 2013, pp. 3111–3119. New York: ACM.
- [5] Pennington J, Socher R and Manning C. Glove: Global vectors for word representation. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, Doha, 25–29 October 2014, pp. 1532–1543. Stroudsburg, PA: ACL.
- [6] Kobayashi H, Noguchi M and Yatsuka T. Summarization based on embedding distributions. In: *Proceedings of the 2015 conference on empirical methods in natural language processing (EMNLP)*, Lisbon, 17–21 September 2015, pp. 1984–1989. Stroudsburg, PA: ACL.
- [7] Jain A, Bhatia D and Thakur MK. Extractive text summarization using word vector embedding. In: *2017 international conference on machine learning and data science (MLDS)*, Noida, India, 14–15 December 2017.
- [8] Rossiello G, Basile P and Semeraro G. Centroid-based text summarization through compositionality of word embeddings. In: *Proceedings of the MultiLing 2017 workshop on summarization and summary evaluation across source types and genres*, Valencia, 3 April 2017.
- [9] Ghalandari DG. Revisiting the centroid-based method: a strong baseline for multi-document summarization. In: *Proceedings of the Workshop on New Frontiers in Summarization, Nfis@emnlp*, Copenhagen, 7 September 2017, pp. 85–90. Stroudsburg, PA: ACL.
- [10] Mohd M, Jan R and Shah M. Text document summarization using word embedding. *Expert Syst Appl* 2020; 143: 112958.
- [11] Qian Y, Du Y, Deng X et al. Detecting new chinese words from massive domain texts with word embedding. *J Inform Sci* 2019; 45(2): 196–211.
- [12] El Mahdaouy A El Alaoui SO | Gaussier E. Word-embedding-based pseudo-relevance feedback for Arabic information retrieval. *J Inform Sci* 2019; 45(4): 429–442.
- [13] Caruana R. Multitask learning. *Machine Learn* 1997; 28(1): 41–75.
- [14] Liu X, Gao J, He X et al. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In: *Proceedings of the 2015 conference of the north american chapter of the association for computational linguistics: human language technologies*, Denver, CO, 31 May–5 June 2015, pp. 912–921. Stroudsburg, PA: ACL.
- [15] Luong M, Le QV, Sutskever I et al. Multi-task sequence to sequence learning. In: *Proceedings of the 4th international conference on learning representations*, San Juan, Puerto Rico, 2–4 May 2016. ICLR.
- [16] Guo H, Pasunuru R and Bansal M. Soft layer-specific multi-task summarization with entailment and question generation. In: *Proceedings of the 56th annual meeting of the association for computational linguistics*, Melbourne, VIC, Australia, 15–20 July 2018, pp. 687–697. Stroudsburg, PA: ACL.
- [17] Subramanian S, Trischler A, Bengio Y et al. Learning general purpose distributed sentence representations via large scale multi-task learning. In: *Proceedings of the 6th International Conference on Learning Representations, ICLR*, Vancouver, BC, Canada, 30 April–3 May 2018.
- [18] Ruder S, Bingel J, Augenstein I et al. Latent multi-task architecture learning. *Proc AAAI Conf Artif Intel* 2019; 33: 4822–4829.
- [19] Devlin J, Chang M, Lee K et al. BERT: pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of the 2019 Conference of the North American chapter of the association for computational linguistics: human language technologies, NAACL-HLT*, Minneapolis, MN, 2–7 June 2019, pp. 4171–4186. Stroudsburg, PA: ACL.
- [20] Kiros R, Zhu Y, Salakhutdinov RR et al. Skip-thought vectors. In: *Advances in neural information processing systems*, pp. 3294–3302, <https://arxiv.org/abs/1506.06726>
- [21] Peters ME, Neumann M, Iyyer M et al. Deep contextualized word representations. In: *Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: human language technologies, NAACL-HLT*, New Orleans, LA, 1–6 June 2018, pp. 2227–2237. Stroudsburg, PA: ACL.
- [22] Joshi A, Fidalgo E, Alegre E et al. Summcode: an unsupervised framework for extractive text summarization based on deep auto-encoders. *Expert Syst Appl* 2019; 129: 200–215.
- [23] Wang A, Singh A, Michael J et al. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Brussels, 14 November, pp. 353–355. Stroudsburg, PA: ACL.

- [24] Lin CY. Rouge: A package for automatic evaluation of summaries. In: *Text Summarization Branches Out. Association for Computational Linguistics*, Barcelona, 3–7 July 2004, pp. 74–81. New York: ACL.
- [25] Ozsoy MG, Alpaslan FN and Cicekli I. Text summarization using latent semantic analysis. *J Inform Sci* 2011; 37(4): 405–417.
- [26] Ferreira R, de Souza Cabral L, Lins RD et al. Assessing sentence scoring techniques for extractive text summarization. *Expert Syst Appl* 2013; 40(14): 5755–5764.
- [27] Oliveira H, Ferreira R, Lima R et al. Assessing shallow sentence scoring techniques and combinations for single and multi-document summarization. *Expert Syst Appl* 2016; 65: 68–86.
- [28] Shafiee F and Shamsfard M. Similarity versus relatedness: a novel approach in extractive Persian document summarisation. *J Inform Sci* 2018; 44(3): 314–330.
- [29] Priya V and Umamaheswari K. Aspect-based summarisation using distributed clustering and single-objective optimisation. *J Inform Sci* 2020; 46(2): 120–148.
- [30] Yao Jg, Wan X and Xiao J. Recent advances in document summarization. *Knowl Inform Syst* 2017; 53(2): 297–336.
- [31] Aries A, Zegour DE and Hidouci W. Automatic text summarization: What has been done and what has to be done, 2019, <https://arxiv.org/abs/1904.00688v1>
- [32] Zhong SH, Liu Y, Li B et al. Query-oriented unsupervised multi-document summarization via deep learning model. *Expert Syst Appl* 2015; 42(21): 8146–8155.
- [33] Cao Z, Wei F, Dong L et al. Ranking with recursive neural networks and its application to multi-document summarization. In: *Twenty-ninth AAAI conference on artificial intelligence*, <https://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9414>
- [34] Yasunaga M, Zhang R, Meelu K et al. Graph-based neural multi-document summarization. In: *Proceedings of the 21st conference on computational natural language learning (Conll)*, Vancouver, BC, Canada, 3–4 August 2017, pp. 452–462. Stroudsburg, PA: ACL.
- [35] Lebanoff L, Song K and Liu F. Adapting the neural encoder-decoder framework from single to multi-document summarization. In: *Proceedings of the 2018 conference on empirical methods in natural language processing*, Brussels, 31 October–4 November 2018, pp. 4131–4141. Stroudsburg, PA: ACL.
- [36] Denil M, Demiraj A and de Freitas N. Extraction of salient sentences from labelled documents, 2014, <https://arxiv.org/abs/1412.6815>
- [37] Yin W and Pei Y. Optimizing sentence modeling and selection for document summarization. In: *Proceedings of the twenty-fourth international joint conference on artificial intelligence, IJCAI*, Buenos Aires, Argentina, 25–31 July 2015, pp. 1383–1389. Reston, VA: AIAA.
- [38] Cao Z, Wei F, Li S et al. Learning summary prior representation for extractive summarization. In: *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing*, Beijing, China, 26–31 July 2015, pp. 829–833. Association for Computational Linguistic.
- [39] Kågebäck M, Mogren O, Tahmasebi N et al. Extractive summarization using continuous vector space models. In: *Proceedings of the 2nd workshop on continuous vector space models and their compositionality (CVSC)*, Gothenburg, Sweden, 26–30 April 2014, pp. 31–39. Association for Computational Linguistics.
- [40] Yogatama D, Liu F and Smith NA. Extractive summarization by maximizing semantic volume. In: *Proceedings of the 2015 conference on empirical methods in natural language processing*, Lisbon, 17–21 September 2015, pp. 1961–1966. Lisbon, 17–21 September 2015.
- [41] Zhang C, Sah S, Nguyen T et al. Semantic sentence embeddings for paraphrasing and text summarization. In: *2017 IEEE global conference on signal and information processing (GlobalSIP)*, Montreal, QC, Canada, 14–16 November 2017, pp. 705–709. New York: IEEE.
- [42] Bouscarrat L, Bonnefoy A, Peel T et al. STRASS: A light and effective method for extractive summarization based on sentence embeddings. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, Florence, 28 July 2019.
- [43] Liu Y and Lapata M. Text summarization with pretrained encoders. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China, 3–7 November 2019, pp. 3730–3740. Stroudsburg, PA: ACL.
- [44] Conneau A, Kiela D, Schwenk H et al. Supervised learning of universal sentence representations from natural language inference data. In: *Proceedings of the 2017 conference on empirical methods in natural language processing, EMNLP*, Copenhagen, 7–11 September 2017, pp. 670–680. Stroudsburg, PA: ACL.
- [45] Hashimoto K, Xiong C, Tsuruoka Y et al. A joint many-task model: Growing a neural network for multiple NLP tasks. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, 7–11 September 2017, pp. 1923–1933. Stroudsburg, PA: ACL.
- [46] Jernite Y, Bowman SR and Sontag DA. Discourse-based objectives for fast unsupervised sentence representation learning, 2017, <https://arxiv.org/abs/1705.00557>
- [47] Liu X, He P, Chen W et al. Multi-task deep neural networks for natural language understanding. In: *Proceedings of the 57th conference of the association for computational linguistics, ACL*, Florence, 11 July 2019, pp. 4487–4496. Stroudsburg, PA: ACL.

- [48] Liu X, He P, Chen W et al. Improving multi-task deep neural networks via knowledge distillation for natural language understanding, 2019, <https://arxiv.org/abs/1904.09482v1>
- [49] Hinton GE, Vinyals O and Dean J. Distilling the knowledge in a neural network, 2015, <https://arxiv.org/abs/1503.02531>
- [50] Vaswani A, Shazeer N, Parmar N et al. Attention is all you need. In: *Advances in neural information processing systems*, pp. 5998–6008, <https://arxiv.org/abs/1706.03762>
- [51] Zhu Y, Kiros R, Zemel R et al. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In: *Proceedings of the IEEE international conference on computer vision*, Santiago, Chile, 7–13 December 2015, pp. 19–27. New York: IEEE.
- [52] Sun C, Qiu X, Xu Y et al. How to fine-tune Bert for text classification? In: *china national conference on Chinese computational linguistics*, Kunming, China, 18–20 October 2019, pp. 194–206. New York: Springer.
- [53] Socher R, Perelygin A, Wu J et al. Recursive deep models for semantic compositionality over a sentiment treebank. In: *Proceedings of the 2013 conference on empirical methods in natural language processing*, Seattle, WA, 18–21 October 2013, pp. 1631–1642. Stroudsburg, PA: ACL.
- [54] Liu X, Shen Y, Duh K et al. Stochastic answer networks for machine reading comprehension. In: *Proceedings of the 56th annual meeting of the association for computational linguistics*, Melbourne, VIC, Australia, 15–20 July 2018.
- [55] Cer D, Diab M, Agirre E et al. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In: *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*, Vancouver, Canada, 3–4 August 2017, pp. 1–14. Association for Computational Linguistics.
- [56] Rajpurkar P, Zhang J, Lopyrev K et al. SQuAD: 100,000+ questions for machine comprehension of text. In: *Proceedings of the 2016 conference on empirical methods in natural language processing*, Austin, TX, 1–5 November 2016, pp. 2383–2392. Stroudsburg, PA: ACL
- [57] Warstadt A, Singh A and Bowman SR. Neural network acceptability judgments. *Trans Assoc Comput Linguist* 2019; 7: 625–641.
- [58] Bentivogli L, Dagan IK, Hoa D et al. The fifth Pascal recognizing textual entailment challenge. In: *TAC 2009 workshop*, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.232.1231&rep=rep1&type=pdf>
- [59] Williams A, Nangia N and Bowman S. A broad-coverage challenge corpus for sentence understanding through inference. In: *Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: human language technologies*, New Orleans, LA, 1–6 June 2018, pp. 1112–1122. Stroudsburg, PA: ACL.
- [60] Dolan WB and Brockett C. Automatically constructing a corpus of sentential paraphrases. In: *Proceedings of the third international workshop on paraphrasing (IWP2005)*, <https://www.aclweb.org/anthology/I05-5002/>
- [61] Kingma DP and Ba J. Adam: A method for stochastic optimization. In: *Proceedings of the 3rd International Conference on Learning Representations, ICLR, 2015*, <https://arxiv.org/abs/1412.6980>
- [62] Hong K, Conroy JM, Favre B et al. A repository of state of the art and competitive baseline summaries for generic news summarization. In: *The international conference on language resources and evaluation LREC'14*, Reykjavik, 26–31 May 2014.
- [63] Démoncourt F, Ghassemi M and Chang W. A repository of corpora for summarization. In: *Proceedings of the eleventh international conference on language resources and evaluation (LREC-2018)*, Miyazaki, Japan, 7–12 May 2018.
- [64] Erkan G and Radev DR. Lexrank: graph-based lexical centrality as salience in text summarization. *J Artif Intel Res* 2004; 22: 457–479.
- [65] Haghighi A and Vanderwende L. Exploring content models for multi-document summarization. In: *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics*, Boulder, CO, 31 May–5 June 2009, pp. 362–370. Association for Computational Linguistics.
- [66] Conroy JM, Goldstein J, Schlesinger JD et al. Left-brain/right-brain multi-document summarization. In *Proceedings of the Document Understanding Conference DUC', 2004*, <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.148.1457>
- [67] Gillick D, Favre B and Hakkani-Tür D. The ICSI summarization system at TAC 2008. In: *Proceedings of the First Text Analysis Conference, TAC*, Gaithersburg, Maryland, USA, 17–19 November 2008. National Institute of Standards and Technology.
- [68] Davis ST, Conroy JM and Schlesinger JD. Occams—an optimal combinatorial covering algorithm for multi-document summarization. In: *IEEE 12th International Conference on Data Mining Workshops, ICDM*, Barcelona, 12–15 December 2016, pp. 454–463. New York: IEEE.
- [69] Hong K and Nenkova A. Improving the estimation of word importance for news multi-document summarization. In: *Proceedings of the 14th conference of the European chapter of the association for computational linguistics*, Gothenburg, 14 April 2014, pp. 712–721. Stroudsburg, PA: ACL.
- [70] Lin H and Bilmes JA. Learning mixtures of submodular shells with application to document summarization. In: *Proceedings of the Twenty-Eighth conference on uncertainty in artificial intelligence*, Arlington, Virginia, USA, 15–17 August 2012, pp. 479–490. Association for Computing Machinery.
- [71] Kulesza A and Taskar B. Determinantal point processes for machine learning. *Found Trend Mach Learn* 2012; 5(2–3): 123–286.
- [72] Gupta A, Kaur M, Mirkin S et al. Text summarization through entailment-based minimum vertex cover. In: *Proceedings of the third joint conference on lexical and computational semantics*, Dublin, 23–24 August 2014.
- [73] Naserasadi A, Khosravi H and Sadeghi F. Extractive multi-document summarization based on textual entailment and sentence compression via knapsack problem. *Natural Lang Eng* 2019; 25(1): 121–146.
- [74] Phang J, Févry T and Bowman SR. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks, 2018, <https://arxiv.org/abs/1811.01088>