Taylor & Francis
Taylor & Francis Group

Automatika

Journal for Control, Measurement, Electronics, Computing and Communications

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/taut20

A parallel optimization and transfer learning approach for summarization in electrical power systems

V. Priya, V. Praveena & L. R. Sujithra

To cite this article: V. Priya, V. Praveena & L. R. Sujithra (2023) A parallel optimization and transfer learning approach for summarization in electrical power systems, Automatika, 64:4, 1225-1233, DOI: 10.1080/00051144.2023.2254975

To link to this article: https://doi.org/10.1080/00051144.2023.2254975

Published online: 11 Sep 2023.

AUTOMATIKA
2023, VOL. 64, NO. 4, 1225–1233
https://doi.org/10.1080/00051144.2023.2254975

Taylor & Francis
Taylor & Francis Group

OPEN ACCESS Check for updates

A parallel optimization and transfer learning approach for summarization in electrical power systems

V. Priya, V. Praveena and L. R. Sujithra

Department of Computer Science and Engineering, Dr NGP Institute of Technology, Coimbatore, India

ABSTRACT

Transfer learning approaches in natural language processing have been explored and evolved as a potential solution for solving many problems in recent days. The current research on aspect-based summarization shows unsatisfactory accuracy and low-quality generated summaries. Additionally, the potential advantages of combining language models with parallel processing have not been explored in the existing literature. This paper aims to address the problem of aspect-based extractive text summarization using a transfer learning approach and an optimization method based on map reduce. The proposed approach utilizes transfer learning with language models to extract significant aspects from the text. Subsequently, an optimization process using map reduce is employed. This optimization framework includes an in-node mapper and reducer algorithm to generate summaries for important aspects identified by the language model. This enhances the quality of the summary, leading to improved accuracy, particularly when applied to electrical power system documents. By leveraging the strengths of natural language models and parallel data processing techniques, this model presents an opportunity to achieve better text summary generation. The performance metric used is accuracy, measured with the ROUGE tool, incorporating precision, recall and f-measure. The proposed model demonstrates a 6% improvement in scores compared to state-of-the-art techniques.

ARTICLE HISTORY
Received 15 July 2023
Accepted 29 August 2023

KEYWORDS
Text summarization; transfer learning; map reduce optimization; extractive summarization; aspect summarization; recommender systems

1. Introduction

Text summarization and information understanding are closely interconnected concepts within information science and retrieval, playing a vital role in allowing readers to quickly review numerous papers and extract key information. Automatic summarization, although challenging, has been recognized as a crucial task in natural language processing (NLP).

There are two main approaches to summarizing texts. While extractive methods excel at selecting relevant data, they may lack the fluency and coherency of human-generated summaries. Abstractive text summarization aims to produce concise summaries that capture the main points of the original text, using compressed paraphrasing and potentially introducing new words not present in the source document. Although abstractive summarization has shown the most promise in terms of overcoming challenges with extracting key information from text texts, abstractive generation may produce phrases that were not there in the original input material. The attention-based encoder-decoder paradigm has currently been widely investigated in abstractive summarization, motivated by neural network success in machine translation experiments. The model revisits the input and attends to essential information by dynamically accessing relevant pieces of information based on the hidden states of the decoder during the production of the output sequence.

A recent extractive document summarizing models has not demonstrated their effectiveness significantly. We examine transfer learning for extractive text summarizing in this research to address a significant difficulty in summarization: condensing the original document as much as feasible while retaining the important concepts, collecting opinions from the text and coming to a judgement based on the content.

Sentiment Analysis (SA) methods can help you figure out what a writer is thinking [1]. Summarization software generates simple summaries from a huge number of evaluations in general. The significant characteristics are extracted initially by using a feature-based summarization approach. Opinions are extracted using these features, and textual summaries are prepared using them. This summary can assist all stakeholders in reaching business decisions about a certain product or service in their domain. Depending on how the summary is generated, summarization systems are categorized as extractive or abstractive. For feature-based text summarization systems, there are a variety of machine learning methodologies available in

CONTACT V. Priya priyasarvesh2004@gmail.com Department of Artificial Intelligence and Data Science, Dr NGP Institute of Technology, Coimbatore, India

This article has been corrected with minor changes. These changes do not impact the academic content of the article.

the literature, both supervised and unsupervised techniques [2]. The paper uses a transfer learning approach with a new language model for feature extraction and then uses this for automatic text summarization using a mapreduce algorithm.

The rest of the paper is organized as follows. Section 2 covers the importance of transfer learning and mapreduces used for text summarization. Section 3 highlights thhe proposed research work. Similarly, Section 4 reviews the evaluation parameters and results obtained. Section 5 serves as our conclusion and includes a summary of our on-going efforts.

2. Literature survey

Text mining and sentiment analysis are combined in aspect-based summarization, which makes it a dynamic research topic. This section looks at the literature on text summarizing techniques that use mapreduce and transfer-learning techniques. Many input parameters are commonly seen in these systems. They are user assessments for a particular domain that include evidence of scores for summary output, cumulative ratings for each phrase according to the aspect and important domain-related aspects. Mapreduce-based algorithms are explored first, followed by an analysis of transfer-learning methodologies.

In several systems proposed by Gamon et al. [3], Khairnar & M. Kinikar [4], Nenkova [5], Zhaung et al. [6] and several researchers, aspect-based sentiment summarization has been widely studied in the literature. Before summarizing the content, these studies usually presume that knowledge from the domain is accessible beforehand. MapReduce is now a counterpart for analyzing large amounts of data in all the fields which works under a parallel approach as mentioned by Dean et al [7]. As Zhao et al. [8] showed, numerous researchers have used machine learning methods like k means clustering with a parallel approach to speed up processing when working with large datasets. Some of the ensemble techniques proposed by Priya et al 1[2] utilize MapReduce with bagging is not motivated to enhance the cluster quality. The approach stated by Ferrari et al [9] uses clustering, topic modelling and semantic similarity along with mapreduce to address the summarization challenge. In this system, with multi-document summarization, the author performed better in terms of scalability. Machine learning algorithms such as SVM Classifier (SVM) and two separate MapReduce stages for aspect summarizing are some of the current research works by Shah et al [10], Lin et al [11] and Tadano et al [12] in the field of aspect summarization using MapReduce.

The idea of transfer learning was motivated by people's capacity to use information acquired from one activity to address problems they had never encountered before. The information gained from one domain or task can be applied to another similar domain or task in the field of machine learning. In the field of text summarization, transfer learning has been widely applied to NLP tasks involving pre-trained language models. For extractive summarization, sentence embedding models like BERT (Bidirectional Encoder Representations from Transformers) [13] can be employed. According to several studies [14], pre-training the language models would enhance the functionality of summarization systems.

BERT is popular among them and used in text summarization systems. BERT is adjusted by BERTSUM [15] for both extractive and abstractive summarization systems. At the document level, BERTSUM employs a competent BERT encoder with pre-training to effectively represent phrases semantically. To tackle the problem of massive text input, a randomly initialized Transformer decoder is also used. Without using the copy mechanism or reinforcement learning techniques, an encoder with two-fold objectives is utilized for the optimization of summarization performance at the document level with minimal restrictions. BERT manipulates three types of embedding for each input token: symbol of a token, section and relative position-based embedding. These embeddings are coupled together to customize a distinct input vector, which is then given to the bidirectional Transformer layers in the model. Then the model provides contextual vectors for each token. Researchers have also experimented with auto text summarization for electric power system documents [15,16].

This research proposes an optimization strategy based on node Mapper along with pre-trained BERT encoder to increase the accuracy of text summarization. This method makes use of Hadoop's clean-up mechanism, which makes use of a built-in combiner optimization and a learned language model for the extraction of significant aspects and sentences. The MapReduce Framework is used to efficiently process large collections of reviews. Also, the system aims to test the summarization algorithm for the electric power domain. This research is the first approach to combine the pre-trained language model and mapreduce optimization for aspect-based text summarization to the best of our knowledge.

The following are the significant contributions of this research:

- An in-node mapper-based optimization approach, for optimizing the mapper output inspired by the models employing machine learning, is created.
- A partitioner algorithm to speed up extensive data processing and provide summaries for the significant aspects in distinct records.
- An improved pre-trained language model to extract significant aspects and sentences from the

domain-specific datasets using transfer learning from the existing corpus.

The subsequent section deliberates on the proposed design in an elaborate manner.

3. The proposed aspect summarization technique

The process of synthesizing a text based on the key characteristics or aspects discovered is known as aspect-based summarization. It is crucial in displaying the important evidence about the different identified facets of the script or review. It's most effective when dealing with vast amounts of written documents or reviews.

The proposed architecture of the system is shown in Figure 1.

There are six main stages in the proposed system: Data pre-processing, pre-training the language model, feature extraction, grouping, mapreduce algorithm optimization and the evaluation of summaries. Its various stages are explained in detail in the following section.

3.1. Pre-processing

Stop words are removed and lemmatization is used to eliminate noisy terms from the dataset's reviews. The approach of latent semantic analysis (LSA) was used to extract relevant features. According to Deerwester and colleagues [17], LSA is beneficial because it builds linkages by developing a collection of ideas that relate to documents and phrases. LSA creates assumptions based on terms that have a closer meaning and can be discovered in the related text content. LSA is the most efficient approach for feature extraction from the text, so this approach was chosen. The words that have been determined to be similar are grouped to determine key elements. This concept was given by Blake. Customer testimonials from the hotel, movie and product domains were utilized. Utilizing the Uma Maheswari et al ontology-dependent author-specific aggregation approach [18], the aspects discovered from the three datasets are sorted. This technique uses an ontology graph and user interests to award a rating to each aspect. The LSA technique was used to extract the most significant aspects. This is used for comparing the significant aspects of the word embedding derived from the language model.

3.2. Pre-training the language model

In this research, the most popular BERT model is used for transfer learning. The BERT architecture was chosen because it outperformed other NLP algorithms on sentence embedding. BERT is based on the transformer design; however, it has special pre-training goals. In one step, it randomly masks off 10–15 per cent of the training set's words to forecast the masked words; in the other, it compares an input text sentence with an entrant sentence and determines if the entrant sentence accurately follows the input phrase [16,19–21. Even with a large number of GPUs this operation can take several days to complete. As a result Google made two BERT models available to the general public one having 1.2 billion features and another with 180 billion. Due to its greater performance the bigger BERT model which was trained beforehand was finally used for the summary .

For extractive summarization, the primary BERT implementation employs the Pytorch-pre-trained-BERT package from the "loving face" organization. The package wraps Google's which was before model implementations in a core Pytorch wrapper. In addition to the basic BERT model, the Pytorch-pretrained-BERT library includes the OpenAi GPT-2 prototype, which is a network that builds on the original BERT architecture. The model had been pre-trained in all three domains: film, hotel and electrical power documents. The stages of the transfer learning approach used for

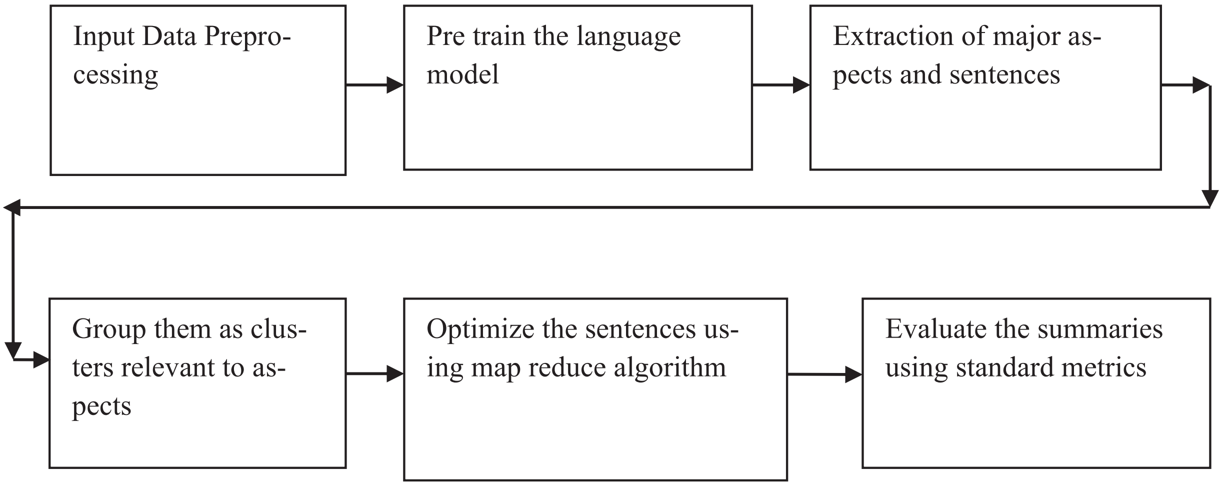
Input Data Preprocessing
Pre train the language model
Extraction of major aspects and sentences
Group them as clusters relevant to aspects
Optimize the sentences using map reduce algorithm
Evaluate the summaries using standard metrics

Figure 1. The proposed architecture of the extractive text summarization system.

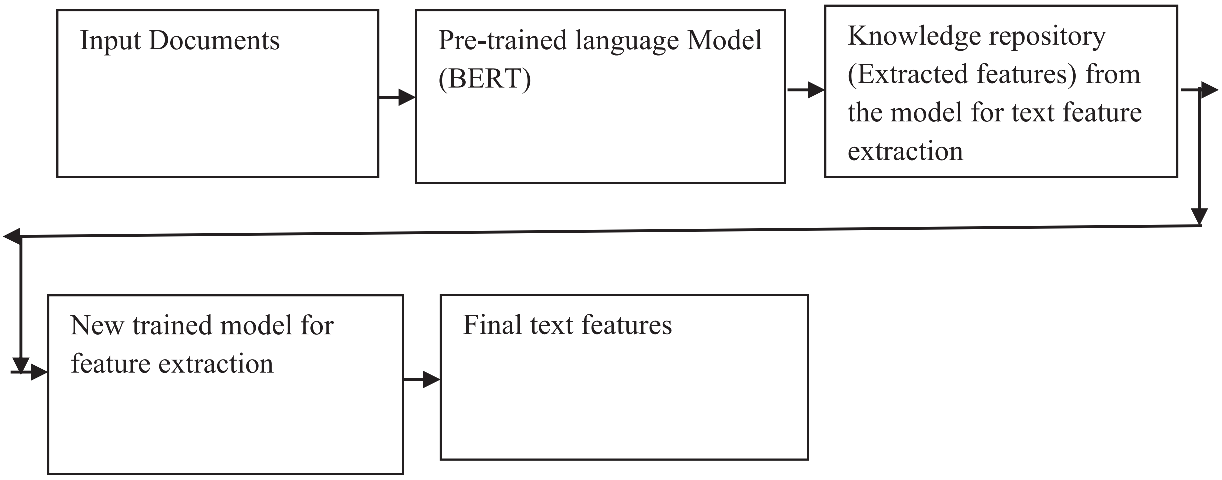
Input Documents
Pre-trained language Model (BERT)
Knowledge repository (Extracted features) from the model for text feature extraction
New trained model for feature extraction
Final text features

Figure 2. Stages of the transfer learning approach for feature extraction for extractive text summarization.

feature extraction that is employed for text summarization are shown in Figure 2.

The transfer learning approach is used mainly for feature extraction compared with the existing language BERT model. For this a convolutional neural network model is used. Features from the BERT model are used for a specific domain and trained with input documents. Then automatic summarization is employed for generating summaries.

3.3. Grouping embeddings

Using the Scikit Learn library's implementation, the K means model was employed for clustering during the experiment. Due to the models' equal performance, K-Means was selected for aggregating inbound word embedding from the BERT model. BERT was modified and pre-trained for the specific domain and used to extract significant aspects particular to the field and its relevant sentences.

Sentences relevant to the aspects are found in a cluster and they are optimized using mapreduce to generate a final summary. Optimization using mapreduce is explained in the next section.

3.4. Map reduce optimization

There are numerous concurrent and scalable algorithms in the literature; however, they do not focus on improving accuracy, according to Golghate [22]. This technique, which makes use of a node combiner, was created especially for text retrieval to improve the calibre of summaries.

Since the combiner is used for every run of the mapper, the main concept is to use combiner optimization to separate and filter out the superfluous sentences. Node combiners are also used in the work by Woo et al. [23] to enhance the performance of the Hadoop system. The node combiner using the map function is meant to focus on text retrieval while eliminating noisy sentences from the summary. Partitioner and reducer modules are designed to provide output files that include summaries of the input aspect words. The next subsections discuss the complete mapreduce framework procedure.

3.4.1. Combiner function for the cluster

Hadoop stores the dataset as a collection of key-value pairs. The value text contains relevant opinions for one facet, and also the tf-idf model's relevancy score and the key is the file's facet. The aspects of the review sentences are retrieved from the BERT model.

The pseudo-code includes a technique for an in-node combiner that includes a partitioner and reducer.

Mapper Method:

Algorithm: Innode Mapper (String arr, as[i], String arrayout, a collection of sentences)

Input:aspectkey [text (aspectkeywords, as[i])], textvalue [reviews for a particular aspect]

Output: < outputkey', outputvalue' > pair, where the aspect phrase is the key, and the value is the value.
– Filtered output sentences are reviewed based on a threshold

Steps in the Mapper Method:

1. Input the reviews for a specific aspect with the aspect keyword retrieved in the previous stage.
2. Create a hashmap with aspects with relevant sentences.
3. Now iterate among the sentences and retrieve the sentences with high relevancy using a threshold. This ensures that sentences with high relevancy are retained for all the aspects used.
4. Now the text key, which is the aspect and the score for each sentence, is obtained as output from the mapper method.

```
Class NodeMapper {
Method nodemap(Text, Text) {
NodeMap < String, String >
AspectnodeMap = newHash Map < char[], char[] > ();
// Use a hash map to match aspects with their relevant sentences
for each aspect as key in aspect keywords do
{
AspectMap.put(key_id,Value);
// method to include relevant sentences
}}
```

The mapper method pseudo-code takes the review sentences along with the aspects as input. Then it maps the aspects with relevant sentences using a hash map based on a sentence score. The hash function generates appropriate sentences for the aspects and every aspect sentence ID along with the value is generated as output. Now these sentences are filtered using a threshold value, which is fixed as 0.5. This is for obtaining highly relevant sentences for all the features.

Clean Up Method:

```
function Clear ()
{
for nodeMap.Entry < text, text >
entry: nodefeatureMap.entryfeatureSet() { //mapset for all the aspects with appropriate sentences
if (similarity (sentenceid, value)) < 0.5 then
emit < outputkey', outputvalue' > pair // keep the sentence in the cluster
else discard values
}}}
```

The partitioner method performs partitions from each node based on the aspect that is given as the key.

Partitioner method

```
public class aspectvaluePartitioner extends MapaspectoutputPartitioner < String, String > {
getmapoutputPartition (public int) (Textpartkey, Text partvalue, int tasksinreducer){
If Text = partkey one then // partkey 1 denotes aspect 1 and so on
return one %tasksinreducer;
elif Text = partkey two then
return two % tasksinreducer
elif Text = partkey three then
return three % tasksinreducer
...
}}
```

The reducer function works in conjunction with the partitioner groups the output sentences selected for the final summary for all the aspects.

Reducer method

```
Static Class aspectvaluePartitionbasedReducer extends nodemapperReducer (string, string, Text, Text)
{
Summaryreduce (public void) (Text outkey', iterableoutvalue'){
For (Text outval:outvalues){
String [ ]outvalTokens = outval.toString().split(",");
output.write (new (Texttype(outkey'), new (Texttype (outvalue'))));
}}
```

The in-node mapper's primary duty is to organize mapper output by identical keys. As with the node combiner, the Combiner object is used in combination with the Map class. A hash map is used to hold most of the review statements and also the key phrases. The lines that are identical to the very first sample sentence are not dumped in the disc during the clean-up process. Only the most appropriate sentences are kept in this case, resulting in memory optimization. To calculate similarity, text-based similarity metrics given by Wael et al [24] were used. This guarantees that superfluous sentences are deleted, which improves the quality of the summary. A custom partitioner is used to separate the summary for all of the aspects related and identified in the result of in combiner with the mapper. Depending on the importance of domain-specific characteristics, the quantity of reducer jobs varies. The reducer runs across each key, writing corresponding sentences to the memory for each aspect key phrase. The method is executed for each of the three domains previously described.

4. Evaluations and discussion

The baseline datasets in text-analytics101 are used in all of investigations (2001). This is about 400 MB in size. This section includes reviews based on the film's domain. The dataset is approximately 190 MB. This section offers user feedback on a variety of hotels in various cities. Datasets are available in the links provided below:

Large Movie Review Dataset. (n.d.). Retrieved from http://ai.stanford.edu/ amaas/data/sentiment/

http://www.text-analytics101.com/ [25,26]. This dataset includes product reviews from a variety of sources. For processing, only reviews based on mobile phones are taken into account.

The third dataset used for testing is the electric power system documents. Experiments were also made based on the electric power dataset that contains more than 2000 electrical papers.

The ROUGE tool has been used to assess the performance. Lin and colleagues (2004). This tool evaluates the textual similarity of the system's automatically generated summaries to human-annotated reference summaries. Three reference summaries are considered. These summaries have three human annotators assigned to them. The ROUGE 2 formula and ROUGE L formula are the measurements that are taken into account. When system-generated summaries are compared against reference summaries from two separate authors, the F measure is calculated. For comparison, some of the standard methods that generate aspect-

Table 1. Embedding derived from the language model for the three domains.

Features/hotel domain	Word embeddings	Movie domain features	Word embeddings	Product domain Features	Word embeddings
Location	Locale Venue	Artist	Actor Hero	Monitor	Screen Display
Provisions	Meals Food	Song	Music Rhythm	Operating speed	Speed Fast
Service	Business Supply	Script	Screenplay Story	Dimensions	Size Weight
Room	Lobbies Hall	Direction	Making	Display	Monitor Resolution
Staff	Waiter Manager	Conversation	Dialogue	Memory	Storage
Price	Cost Budget	Cinematography	Show	Battery	Power Backup
Convenience	Facility	Story	Moral		
Luxury	Comfort Ambience	Sound	Music Background		

or feature-based summaries are employed. They were mapreduce-based algorithms that use parallel processing and PTLM (pre-trained LSTM) for constructing text summaries was also included. The main parameters in the ROUGE tool are Precision, recall and F measure.

Precision

Precision in the context of ROUGE is denoted by Equation (1) and indicates how much of the system summary was, in fact, relevant or needed.

$$\text{Precision} = \text{No of overlapping words/Total words in system summary} \tag{1}$$

Recall

Recall in the context of ROUGE simply means how much of the reference summary, the system summary is recovering or capturing and shown in Equation (2).

$$\text{Recall} = \text{No of overlapping words/Total words in reference summary} \tag{2}$$

F-measure

F-Measure is the harmonic mean of precision and recall as per Equation (3).

$$\text{F} - \text{Measure} = 2 * \text{Precision} * \text{Recall/Precision} + \text{Recall} \tag{3}$$

For generating a reference summary, two annotators were used and they were asked to select the top sentences from the reviews related to the top most features extracted from the system. The metric used in ROUGE for evaluating system-generated summaries is f-measure. These are measured against gold standard reference summaries. An aspect-based summary was generated using the review dataset from the hotel domain. Using MapReduce, the results are compared to other established approaches. Results/two alternate words for the features and aspects in three domains are given. They denote the word embeddings retrieved for specific features for each domain. The BERT model is a well-known language model used for text processing. This model improves summary quality by identifying similar words and overlapping words from the text and also this is utilized for measurement. The word embedding derived from the language model for the three domains is presented in Table 1.

These embeddings have been used for the measurement for overlapping words when summary sentences are retrieved. The system-generated summaries are given for sample features.

> System-Generated Summary:
>
> Location: Its location is great if you want to get a slight feel of what it's like living in China The location was wonderful too only a few minutes of walk to Tiananmen Square Stylish clean reasonable value poor location Excellent value location, not a big problem Location is poor for the typical tourist Nice trendy hotel location not too bad. The location is also quite fantastic.
>
> Staff: Staffs were very friendly and check-in and checkout times were acceptable. We found the reception staff generally very helpful and friendly.
>
> Room: Rooms are not big but adequate with modern bathroom amenities good and strong hot showers. The room was spotless and had a comfortable king-sized bed as far as soft beds go in China. The rooms are very pretty and comfortable with modern western-style bathrooms We booked a three-bedroom room for a night. Each room came with a queen-sized bed. We paid about USD per night for this large bedroom apartment style room. On our return, we also got upgraded to a bigger room.

The metric used for all the domains in ROUGE is f-measure. ROUGE 2 and ROUGE L have been chosen. ROUGE-2 Precision and Recall compare the

Table 2. Hotel domain dataset-Rouge 2 and Rouge L scores.

Hotel domain features		PLTM method	Combiner-based approach	Parallel clustering approach	MR optimization with transfer learning
Locale/venue	R2	0.46	0.6	0.39	0.61
	RL	0.42	0.44	0.35	0.66
Provisions/meals	R2	0.37	0.45	0.46	0.62
	RL	0.4	0.7	0.47	0.65
Service/business	R2	0.28	0.38	0.42	0.45
	RL	0.33	0.32	0.38	0.42
Room/lobbies	R2	0.51	0.48	0.49	0.63
	RL	0.46	0.53	0.50	0.69
Staff/employers	R2	0.17	0.46	0.43	0.5
	RL	0.34	0.51	0.52	0.55
Price/cost	R2	0.32	0.48	0.45	0.59
	RL	0.47	0.53	0.51	0.54
Convenience/facility	R2	0.36	0.55	0.63	**0.69**
	RL	0.39	0.4	0.49	**0.64**
Luxury/comfort	R2	0.32	0.42	0.40	0.44
	RL	0.41	0.45	0.24	0.49

Table 3. Movie domain dataset-Rouge 2 and Rouge L scores.

Movie domain features		PTLM method	Combiner-based approach	Parallel clustering approach	MR optimization with transfer learning
Artist/Actor	R2	0.36	0.42	0.49	0.51
	RL	0.51	0.53	0.55	0.58
Song/Music	R2	0.55	0.56	0.54	0.56
	RL	0.40	0.62	0.56	0.65
Script/Screenplay	R2	0.28	0.38	0.41	0.45
	RL	0.33	0.43	0.45	0.5
Direction	R2	0.51	0.68	0.59	0.58
	RL	0.56	0.67	0.62	0.67
Conversation/Dialogue	R2	0.29	0.46	0.42	0.49
	RL	0.34	0.44	0.44	0.47
Cinematography	R2	0.32	0.48	0.46	0.45
	RL	0.37	0.53	0.43	0.47
Story	R2	0.36	0.54	0.47	**0.59**
	RL	0.41	0.45	0.51	**0.64**
Sound	R2	0.32	0.44	0.38	0.44
	RL	0.35	0.41	0.39	0.49

similarity of bi-grams between reference and candidate summaries. By bi-grams, it means that each token of comparison is 2 consecutive words from the reference and system-generated candidate summaries. ROUGE-L Precision and Recall measures the Longest Common Subsequence (LCS) words between reference and system-generated candidate summaries. By LCS, it refers to word tokens that are in sequence, but not necessarily consecutive. Since these two measures consider consecutive words and also word tokens using subsequences these metrics were used to compute precision and recall, from which f measure is calculated and presented in Tables 2–4 for the three domains considered. The rouge score with the f-measure metric for the aspects discovered in the hotel domain is shown in Table 2. The results suggest that the node mapper algorithm significantly enhances performance. Using f-measure, the approach delivers a result with increased accuracy, which improves by 5% overall.

Using ROUGE 2 and L, Table 2 illustrates the f-measure score values for the essential characteristics in the hotel domain. Bold values represent components of high f-measure values. The three most regularly mentioned features in the evaluations, according to the LSA technique, which represent their word weights are place, food and rooms.

Table 4. Power system documents dataset-Rouge 2 and Rouge L scores.

Power system document features		PTLM Method	Combiner-based approach	Parallel clustering approach	MR optimization with transfer learning
Power	R2	0.46	0.6	0.61	0.68
	RL	0.51	0.4	0.43	0.46
Voltage	R2	0.55	0.45	0.46	0.56
	RL	0.4	0.50	0.52	0.64
Current	R2	0.28	0.38	0.39	0.45
	RL	0.24	0.37	0.36	0.49
Battery	R2	0.34	0.48	0.47	0.49
	RL	0.38	0.53	0.51	0.64

4.1. Movie dataset

Table 3 compares ROUGE 2-based f-measure values for each significant element of the movie domain. The results have improved by 5% over previous findings. Aspects such as direction, song and singer suggest that the system is progressing well. This is because reviews contain more relevant statements for these features. These points are mentioned several times in consumer feedback. The recommended approach performs brilliantly in all domains since it uses term weights and optimization. The three factors most frequently mentioned in reviews are the artist, the song and the direction.

4.2. Power system documents

As previously stated, the dataset related to the product domain was used to construct summaries for each important attribute detected. The top five features of panel, efficiency, screen, size, storage and energy are only summarized since only recommendations for cell phones were accounted for from the business domain dataset. Comparing optimization using the in-node mapper method to other frequently employed strategies in the literature, the results demonstrate a 10% improvement in f-measure values.

Table 4 for the power system documents dataset displays the outcomes of Rouge 2 and the L measure, which assesses the bigram and similarity between the standard and candidate summaries. Because it employs word weights and similarity-based filtering, our approach works admirably across the board.

Finally, we use a Hadoop system to calculate how long it takes to execute our algorithm on single- and multi-point nodes. In a three-node cluster, one name server node and two data processing nodes were employed. It is measured how long it takes for the entire system to operate from input to output. This has already been decreased, and by using numerous nodes, it might be much decreased. These results are summarized in Table 4.

5. Conclusion & future work

The MapReduce Framework along with pre-trained language models was used to examine aspect-based text summarizing systems in this paper. The proposed solution, which employs a language model (BERT) pre-trained for the domain along with an MR-based optimization algorithm, has increased the summary's quality. For large power system documents and review datasets, the system also demonstrates a significant boost in performance in terms of calculation time. The technique specifically for extractive summarization beats other common systems and improves accuracy using ROUGE measurements. Using the MapReduce Framework, the technique generates efficient aspect summaries in big datasets.

5.1. Future work

The researchers can enhance the summary in the future by incorporating semantic interpretations including in-memory computing techniques and new language models, into Hadoop. The BERT model also can be improved to enhance word embedding, which would improve the quality of the summary.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

[1] A Blake Catherine. Comparison of document, sentence and term event spaces. Joint 21st International Conference on Computational Linguistics (COLING) and the 44th Annual Meeting of the Association for Computational Linguistics (ACL), Sydney, Australia; 2006, pp. 601–608. New york: ACM.

[2] Boiy E, Moens MA. Machine learning approach to sentiment analysis in multilingual web texts. J Inf Retriev. 2009;12:526–558. doi:10.1007/s10791-008-9070-z

[3] Gamon M, Aue A, Corston-Oliver S, et al. Pulse: mining customer opinions from free text. 6th International Symposium on Intelligent Data Analysis (IDA), Madrid, Spain, 8 September-10 September 2005, Paper no. LNCS 3646, pp 121–132, Heidelberg: Springer.

[4] Khairnar J, Kinikar M. Latent semantic analysis used for mobile rating and review summarization. Int J Comput Sci Telecommun. 2013;4:61–67.

[5] Ani N, Kathleen M. A survey of text summarization techniques. Chapter in Mining Text Data, pp. 43–76. Springer, Heidelberg, 2012.

[6] Keneshloo Y, Ramakrishnan N, Reddy CK. Deep transfer reinforcement learning for text summarization. Proceedings of the 2019 SIAM International Conference on Data Mining 2019, vol. 6, pp. 675–683. Society for Industrial and Applied Mathematics.

[7] Dean J, Ghemawat S. Map reduce: simplify data processing on large clusters. Commun ACM. 2008;51:107–113. doi:10.1145/1327452.1327492

[8] Large Movie Review Dataset. http://ai.stanford.edu/~amaas/data/sentiment/ (2001, accessed 22 January 2020).

[9] Ferreira R. Assessing sentence scoring techniques for extractive text summarization. Expert Syst Appl. 2013;16:5755–5764.

[10] Gomaa HW, Fahmy AA. A survey of text similarity approaches. Int J Comput Appl. 2013;68(13):13–18. doi:10.5120/11638-7118

[11] Jimmy L, Chris D. Data-intensive text processing with map reduce. University of Maryland, College Park Manuscript, 2010, pp. 28–30.

[12] Lee W-H, Jun H-G, Kim H-J. Hadoop map reduce performance enhancement using in-node combiners. Int J Comput Sci Inf Technol. 2015;7(5):1–17. doi:10.5121/ijcsit.2015.7501

[13] Devlin J, Chang MW, Lee K, et al. Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805. 2018 Oct 11.

[14] Viji D, Revathy S. A hybrid approach of weighted fine-tuned BERT extraction with deep Siamese Bi–LSTM model for semantic text similarity identification. Multimed Tools Appl. 2022;81(5):6131–6157. doi:10.1007/s11042-021-11771-6

[15] Jiang W, Zou Y, Zhao T, et al. A hierarchical bidirectional LSTM sequence model for extractive text summarization in electric power systems. 2020 13th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China; 2020, pp. 290–294, doi:10.1109/ISCID51228.2020.00071

[16] Mbuwir BV, Spiessens F, Deconinck G. Distributed optimization for scheduling energy flows in community microgrids. Electr Power Syst Res . 2020;187:106479. doi:10.1016/j.epsr.2020.106479

[17] Deerwester S, Dumais ST, Furnas GW. Indexing by latent semantic analysis. J Am Soc Inf Sci. 1990;41: 391–407.

[18] Ryosuke T, Shimada K, Endo T. Multi-aspects review summarization based on identification of important opinions and their similarity. 24th pacific Asia Conference on Language, Information and Computation (PACLIC).Sendai, Japan, 4 November–7 November 2010, pp. 685–692. Japan: Institute for Digital Enhancement of Cognitive Development

[19] Lamsiyah S. Unsupervised extractive multi-document summarization method based on transfer learning from BERT multi-task fine-tuning. J Inf Sci 2023;49:164–182. doi:10.1177/0165551521990616

[20] Zhuang F, Qi Z, Duan K, et al. A comprehensive survey on transfer learning. Proc. IEEE, vol. 109; 2021, pp. 43–76.

[21] Liu, Y., Lapata, M., 2019. Text summarization with pretrained encoders. In: EMNLP-IJCNLP 2019 – 2019 Conference Empirical Methods Natural Language Processing 9th International Joint Conference on Natural Language Processing Proceedings Conference, pp. 3730–3740.

[22] Ashish G, Shende Shailendra W. Parallel K-means clustering based on Hadoop and Hama. Int J Comput Technol. 2014;1:33–37. doi:10.2495/ICCST140041

[23] Hotel datasets. http://www.text-analytics101.com/ (2001, accessed 22 January 2020).

[24] Weizhong Z, Huifang M, Qing H. Parallel K-means clustering based on Map reduce. First International Conference on CloudCom 2009, Beijing, China, 1 December–4 December, 2009 Paper no. LNCS 5931: pp 674–679. Heidelberg: Springer.

[25] Fecht P, Blank S, Zorn HP. Sequential transfer learning in NLP for German text summarization. In Swiss Text 2019.

[26] Alomari N, Idris AQ, Sabri I. Alsmadi deep reinforcement and transfer learning for abstractive text summarization: a review. Comp Speech Lang. 2021;10:101276.