applied sciences

MDPI

Article

A Novel Approach for Semantic Extractive Text Summarization

Waseemullah [1], Zainab Fatima [2], Shehnila Zardari [1], Muhammad Fahim [1], Maria Andleeb Siddiqui [2], Ag. Asri Ag. Ibrahim [3,*], Kashif Nisar [3] and Laviza Falak Naz [2]

[1] Computer Science and Information Technology, NED University of Engineering and Technology, Karachi 75270, Pakistan; waseemu@cloud.neduet.edu.pk (W.); shehnilaz@cloud.neduet.edu.pk (S.Z.); fahimku2020@gmail.com (M.F.)
[2] Department of Software Engineering, NED University of Engineering and Technology, Karachi 75270, Pakistan; zainab.ned@cloud.neduet.edu.pk (Z.F.); mandleeb@cloud.neduet.edu.pk (M.A.S.); naz4108845@cloud.neduet.edu.pk (L.F.N.)
[3] Faculty of Computing and Informatics, University Malaysia Sabah, Jalan UMS, Kota Kinabalu 88400, Sabah, Malaysia; kashif@ums.edu.my
* Correspondence: awgasri@ums.edu.my

Abstract: Text summarization is a technique for shortening down or exacting a long text or document. It becomes critical when someone needs a quick and accurate summary of very long content. Manual text summarization can be expensive and time-consuming. While summarizing, some important content, such as information, concepts, and features of the document, can be lost; therefore, the retention ratio, which contains informative sentences, is lost, and if more information is added, then lengthy texts can be produced, increasing the compression ratio. Therefore, there is a tradeoff between two ratios (compression and retention). The model preserves or collects all the informative sentences by taking only the long sentences and removing the short sentences with less of a compression ratio. It tries to balance the retention ratio by avoiding text redundancies and also filters irrelevant information from the text by removing outliers. It generates sentences in chronological order as the sentences are mentioned in the original document. It also uses a heuristic approach for selecting the best cluster or group, which contains more meaningful sentences that are present in the topmost sentences of the summary. Our proposed model extractive summarizer overcomes these deficiencies and tries to balance between compression and retention ratios.

Keywords: text mining; text summarization; text extraction; semantic text extraction

Citation: Waseemullah; Fatima, Z.; Zardari, S.; Fahim, M.; Andleeb Siddiqui, M.; Ibrahim, A.A.A.; Nisar, K.; Naz, L.F. A Novel Approach for Semantic Extractive Text Summarization. *Appl. Sci.* **2022**, *12*, 4479. https://doi.org/10.3390/app12094479

Academic Editors: Federico Divina and Arcangelo Castiglione

Received: 9 December 2021
Accepted: 18 March 2022
Published: 28 April 2022

1. Introduction

Extractive text summarization is a process of extracting the top relevant, informative, and meaningful sentences from a document in a concise form. It is based on statistical and linguistic features by selecting the top relevant features based on frequency, cue phrase, sentence extraction, etc. [1]. Much research has been conducted for producing effective and efficient text extractive summarization by using different techniques; however, the issues of "compression ratio" and "retention ratio" remain the same in all the research such that no one has made efforts towards the above-mentioned issues faced by all extractive summarizers. The following describes the "compression ratio" and "retention ratio": "The text summarizer should generate one-third of whole document text but retain or preserves its meaningful information during compression" [2]. Most researchers have developed extractive text summarizers, but when they shrink the size of a document or text during compression, its retention of text becomes upset, and, mostly, summarizers lose very meaningful information. Some of them lose features, contents, concepts, information, etc.

The proposed extractive summarizer preserves these above-mentioned criteria by effectively removing any "outliers", which produce irrelevant information in the summary and, additionally, maintain facts and figures in the form of date events [3]. First, it takes the input document or file as a source file from the user and fetches all information in the form

of collections of sentences by splitting documents with the help of the sentence tokenized method. The second step includes a pre-processing step in which text normalization is performed by removing punctuation, stopping words like "is", "am", "are", etc., and performing lemmatization by taking the root meaning of each and every word [4]. In the third step, the system performs a scoring of sentences by taking the mode of frequent items and selecting the most frequent items, after which the similarity is calculated for each and every token or word. In order to preserve its domain constraint means, no part of any word containing irrelevant information is included; thus, only filtered domain words or informative words are selected based on the topmost similarity criteria [5]. In the fourth step, the system extracts sentences based on the topmost filtered and frequent words from the document. At last, all the sentences are assembled by preserving their chronological order and producing a summary.

The general steps of automatic text summarization, as shown in Figure 1, are described below:

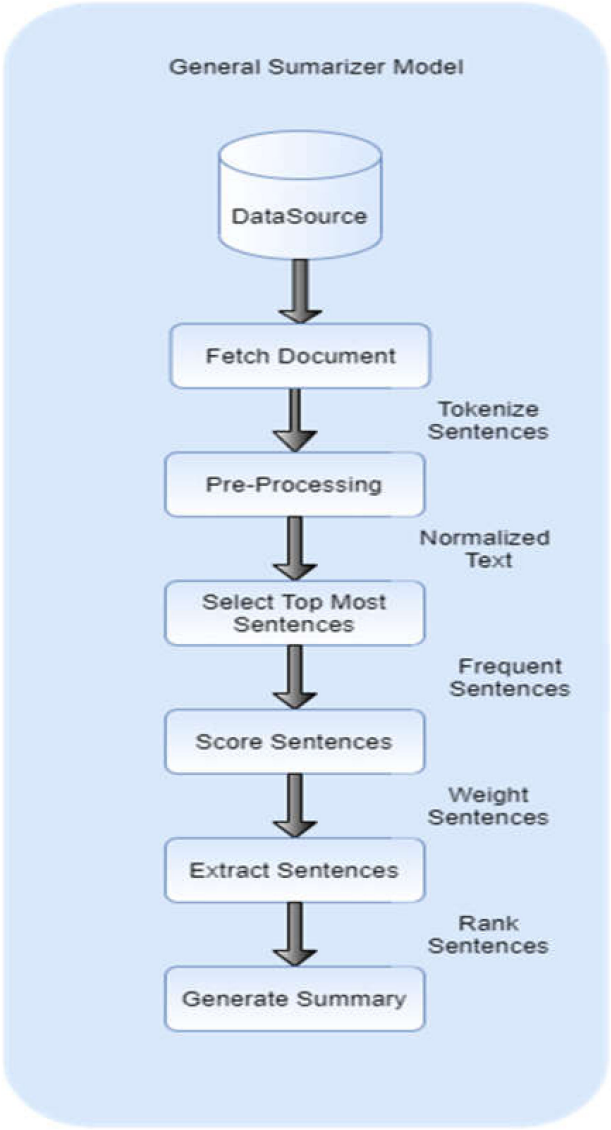


Figure 1. General Summarizer Model.

Extractive text summarizer models were developed based on text rank and frequency and are word-based and featured-based, with many other advancements having been developed during the past few decades [6]. However, all these summarizers work on selection criteria based on the top rank or features, and when they select the topmost

criteria, they leave out some important events, such as important dates, which show very relevant information regarding the title and topic. Secondly, when we shrink the size or compression ratio of any extractive summarizer, it loses the content, features, concepts, and other important information [7]. One of the key limitations of the existing models is that they produce less precision and low accuracy, whereas the proposed model is an attempt to overcome the existing limitation of existing text summarization models.

We analyzed the different shortcomings of existing algorithms, which are explained below:

Luhn's model depends upon the number of frequent words divided by a total number of words. However, it is not suitable for the maximum frequency or minimum frequency of words. In short, it has no semantic techniques; it simply works on a tf–idf frequency. Edmonson's depends on the cue method, sentence position, etc. It considers most of the sentences that contain the title or heading of documents. It also considers the first and last paragraph of sentences with the help of sentence location, but it has no semantic techniques. It cannot judge how important information is given in the body of sentences. LSA works on distributional semantics, calculates the cosine similarity between keywords and the remaining text, and, based on the highest similarity score, gives the highest similar sentences; however, it cannot filter irrelevant text and one of the major drawbacks of this model is it cannot filter repetitive sentences. Thus, it produces redundancy. The important events in the form of dates remain left in the document. LexRank's algorithm is based on Eigenvector centrality, and sentences are placed at the vertex of a graph while the weights are assigned based on a semantic similarity that calculates the cosine similarity among sentences. During sentence overlapping, the redundant sentences are also retrieved, which have to be removed. TextRank's algorithm is based on Eigenvector centrality, and sentences are placed at the vertex of a graph, and weights are assigned based on lexical similarity. One of the major drawbacks of this model is sentence overlapping, which shows redundancy. In KL divergence, redundancy would only lower the efficiency of the algorithm, as adding redundant sentences would affect the unigram distribution of words and increase the divergence from the source document set. Table 1 shows comparisons based on the shortcomings of the algorithms.

Table 1. Comparison of algorithms based on their shortcomings.

Features	LexRank	Edmonson	LSA	TextRank	Kl-Divergence	Proposed
Redundancy	✔	✔	✔	✔	✔	×
Semantics	✔	×	✔	×	×	✔
Events/Dates	×	×	×	×	×	✔
Chronological Order	×	×	×	×	×	✔
CR/RR Balancing	×	×	×	×	×	✔

Here, the symbol ✔ means "allowed", and the symbol × means "not allowed". Column one shows the features of the different text summarizers, including the one proposed in this paper. Columns two to seventh show the comparison among them on the basis of selected features shown in column one. As shown in the table LexRank's algorithms allow redundancy, while the proposed algorithm does not allow it. The same criteria are followed in the remaining features, such as different algorithms allowing some features while some do not allow others. It is clearly shown from the table that one of the major advantages of the proposed model is that it allows all the features and excludes redundancy as compared to the other algorithms. The proposed algorithm's detailed results are shown in the results section.

This paper contains five sections. Section 1 describes the introduction of the research work carried out in this paper. Section 2 describes related work that has been done in relation to text summarization techniques. Section 3 describes the implementation of the methodology and the dataset used for the experiment. Section 4 presents the results of the experiments that were conducted. Section 5 presents the conclusion and future work.

2. Background

The text summarization systems started in the early 1950s by Luhn and Baxendle on the surface-level approach where the word frequency and word position attributes are used in the single document text summarization of technical articles [8,9]. Lehn's algorithm works on a bag-of-words model. It counts the number of words in the document and compresses the text documents based on thematic words. The thematic words of the document show how the word is important in the document. Its importance can be calculated based on the frequency of the words or how many times the word is repeated in the document.

In the 1960s, the other attributes, such as cue words and cue phrases, combined with the previous ones that were used by Edmonson on the surface-level approach for single document text summarization [10]. In 1995, trainable document summarizers were developed using machine learning techniques [11]. In this type of technique, multiple samples of summaries are given as an input to the summarizer system; it finds the relation between the sentences and their title or document and decides whether it includes summary sentences or not. In 1997, Barzilay and Elhadad proposed a lexical chains model that provides the semantic structure of sentences. Lexical chains were constructed using some knowledge base that contains nouns and their various associations [12]. NER and information extraction techniques were applied to news articles, in which the number of words that co-occurred in the questions was extracted [13]. Cut and paste systems were developed by using the statistical technique; they maintain the cohesive structure of the text and create a text that flows logically in structure and meaning from beginning to end [7]. Swesum (Herculus Dalianis) developed the domain-specific statistical summarizer, which summarizes news articles [14]. Conroy J.M. and O' Leary, D.P. developed the single-level lexical summarizer, which summarizes articles and generates lexically related sentences [2]. Graph-based summarizers were also developed [15]. LSA-based summarizers were developed, which extract semantically related sentences [16]. Abdullah Fattah Omar also worked on text summarization using the same approach (LSA) to produce a coherent summary [17].

Researchers developed models for single-document summarization based on joint extraction and syntactic compression. Our model chooses sentences from the document, identifies possible compressions based on constituency parses and scores those compressions with a neural model to produce the final summary [18]. Sergey Gorbachev also developed a neural model with a modification of fuzzy logic in order to produce a coherent summary [19]. Siddhant Upasani developed summarizers based on TextRank, the text rank algorithm, which is the implementation of the page rank algorithm that is used in the Google search engine for ranking web pages [20]. An English news summarizer for the English learning of Indonesian students was developed based on tf–idf features. The topmost sentences are selected based on similarity keywords, for the limitation length of the document, only relevant, top most sentences are extracted. For the summary evaluation, the precision, recall, and f-measure score are used for the summary strength [21]. Michael George developed automatic text summarization based on thickening sentence scores; it fetches sentences with a higher value and density and lower length. The target increases sentence value and reduces unnecessary words and long sentences, which gives the top sentences list more value. Subsequently, that sentence score is equal to the total sentence terms' occurrence divided by the sentence words [22].

A domain-independent, statistical-based method was developed for single-document extractive summarization. The bigrams technique was used, which repeats more than once in the text and are good terms to describe the text's contents, called maximal frequent sentences. We also show that the frequency of repeating bigram terms that occur gives a good result [23]. Rasim proposed a maximum coverage and minimum redundant text summarization model by using an integer linear programming problem technique. One of the advantages of this model is that it can directly discover key sentences in the given document(s) and cover the main content of the original document(s) [24]. This model also guarantees that the summary cannot be multiple sentences that convey the same

information. The proposed model is quite general and can also be used for single- and multi-document summarization implemented on DUC2005 and DUC2007 datasets [25]. The multiple alternative sentence compressions for automatic text summarization were suggested by Nitin Madnani, David Zajic, Bonnie Dorr, Necip Fazil Ayan, and Jimmy Lin, and is a summarizer model in which a parse-and-trim approach consisting of filtering, compression, and candidate selection stages [26], are used. The filtering process contain sentences of high relevance and centrality which are selected for further processing. HMM generates the most probable compressions of a source sentence. Trimmer uses linguistically-motivated trimming rules to remove constituents from a parse tree [27]. Both approaches associate compression-specific feature values with the candidate compressions that can be used in candidate selection. Trimmer generates multiple compressions by treating the output of each Trimmer rule application as a distinct compression [28]. The output of a Trimmer rule is a single parse tree and an associated surface string [29]. A vector space modeling-based evaluation of automatically generated text summaries was suggested by Alaidine Ben Ayed, Ismaïl Biskri, and Jean-Guy Meunier. They presented VSMbM, a new metric for automatically generated text summaries evaluation. VSMbM is based on vector space modeling. It gives insights on to which extent retention and fidelity are met in the generated summaries [30]. Three variants of the proposed metric, namely PCA–VSMbM, ISOMAP–VSMbM, and tSNE–VSMbM are tested and compared to the Recall-Oriented Understudy for Gisting Evaluation (ROUGE), a standard metric used to evaluate automatically generated summaries [31].

3. Methodology and Experimental Setup

The system fetches documents from a Wikipedia data source and performs pre-processing, such as tokenization, stop-word removal, lemmatization, and stemming [32]. After pre-processing, it performs feature engineering by generating a bag-of-words model and calculating how many thematic words are present in a document, which shows important information regarding the title and topic of the document. Then, it also performs some statistical measures, such as mean, mode, variance, and z-score, and, based on the z-score, it eliminates outliers considered irrelevant information [33]. The program primarily focuses on the development of a summarization algorithm that can tackle the limitations of past studies by providing a much more accurate summarization, with a faster processing rate and resulting in a meaningful passage obtained at the end. Many of the existing algorithms fail to provide a meaningful summary and often lead to a clash of requirements as they have been observed to remove major sections of the text without any justification [27]. In this way, the proposed system also extracts unique words by filtering cosine similarities and, based on unique words, extracts sentences from samples by document similarity or fuzzy logic. After removing the punctuation, it generates a summary and evaluates a compression ratio by matching the summary sentences, original document sentences, and retention ratios based on information gained from matching the informative sentences in the summary against the informative sentences in the document with the help of the LDA model [34]. The steps involved in the summarization Algorithm 1 are defined in form of pseudo-code as below:

The architectural diagram of the proposed summarizer is given in Figure 2.

As shown in Figure 2, the proposed model's architecture breaks up into five layers, and each layer performs a separate function. The first layer performs the preprocessing steps for text cleaning and transforms it into a normalized form [35]. The second layer performs a feature selection where all the significant features that are taking part to produce an efficient summary are selected. The third one performs a feature extraction where the topmost relevant feature is extracted and irrelevant features are filtered. The fourth one performs sentences extraction, and the fifth one performs a summary generation [36].

Algorithm 1: Pseudo-Code

Let D → Document
S C D ∴ Sample is a subset of Document D
$S \in \forall Sc_i$ ∴ Sci is list of Sentences that Sample S contains.
$Sc = \{Sc_1, Sc_2, Sc_3,,,,, Sc_n\}$

Remove Stop Words

stop_words = {'is','am','are' . . . .}
$Sc - stop\_words = \{Sc_1, Sc_2, Sc_3,,,,, Sc_n\}$ - {'is','am','are' . . . .}
$Sc = \{Sc_1, Sc_2, Sc_3,,,,, Sc_n\}$ ∴ set of Sentences that don't contain stopwords.

Generate Bag-of-Words by Mode:

$Mode = l + h\ (f_m - f_1/2f_m - f_1 - f_2)$
Keywords = f_{Mode}(Sc) ∴ list of most frequently occurring words contains S.

Compute Mean

$\mu = Sc_1 + Sc_2 + Sc_3 + \ldots\ \ldots . + Sc_n/n$

Compute Standard Deviation

$\delta = \sqrt{\sum_{k=1}^{n}(Xi - \mu)\ 2/(n-1)}$

Compute Z-Score

$Z\text{-Score} = (X_i - \mu)/\delta$
$\forall Sc_i > Z\text{-Score}$
$S_{Pos\_Outlier} \in \forall Sc_i$
And, $\forall Sc_i < Z\text{-Score}$
$S_{Neg\_Outlier} \in \forall Sc_i$.

Eliminate Outliers

$S_{pos} = S - S_{Pos\_Outlier}$ ∴ $\forall S_{pos} \in S_{Pos\_Outlier}$
$S_{neg} = S - S_{neg\_Outlier}$ ∴ $\forall S_{neg} \in S_{neg\_Outlier}$
$Sents = S_{pos} \cup S_{neg}$

Extract Facts from Sample

Ext[i] = $f_{extract\_dates}$ (S) ∴ S is Sample.

Extract Sentences Based on Keywords

Summ[i] = f_{Fuzzy} (Keywords,Sents) ∴ Sents is list of all sentences.

Generate Summary

$Summ\_all = \sum_{k=1}^{n} Ext + \sum_{k=1}^{n} Summ$
(Facts) & (Fuzzy)
Summary = $\mathbf{f}_{Unique}$ (Summ_all)

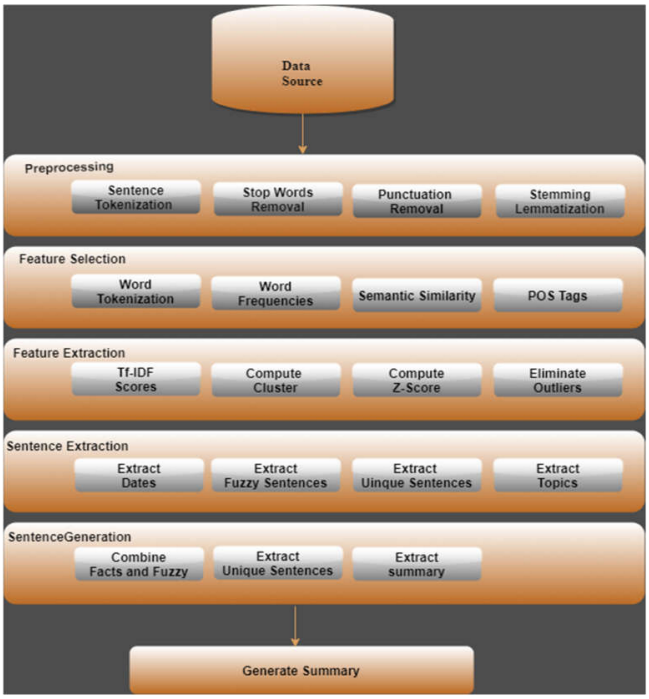


Figure 2. Architectural diagram of proposed summarizer.

3.1. Data Source

The system takes online input documents from Wikipedia, a data source that is an encyclopedia of articles [37]. The main reason for choosing this resource is that it contains structural documents or article formatting where each and every piece of information regarding the search topic is almost always relevant, complete, and in a structured form, e.g., the first paragraph contains each article describing its nature, context, or definition of the topic, and the second one describes its other related information, and so on.

3.2. Preprocessing

Stop Words: The first step of preprocessing is the removal of stop words, such as a, an, the, is, etc.

Tokenization: The second step is to extract tokens of sentences by the word tokenize function in the Python library [38]

POS Tags: The third step extracts part of speech (POS) tags from a document, such as nouns, pronouns, etc.

Facts: The fourth step extracts facts, such as the dates of events in which something happens, e.g., date of birth and death and all other important events, because summarizers usually exclude such important events during shrinking.

Stemming: The fourth step consists of generating the stems of tokens; the stem is the base or root form of words where the second word is generated from its original word, e.g., in the word "become", the stem is "be" (by Port Stemmer in Python [39]).

Lemmatization: The sixth step consists of producing the lemmas of tokens; the lemma is the base or root meaning of words, e.g., the word "living" has a root meaning of the word "live".

Frequent Words: In the seventh step, the most frequent words are identified and selected by using the mode function.

$$\text{Mode} = l + h\,(fm - f_1/2f_m - f_1 - f_2) \quad (1)$$

Semantic Similarity: In this step, the most frequent words are obtained, and some irrelevant words are also selected due to frequent occurrence. In order to avoid false selection, the similarity between words is calculated by taking the cosine of the dot product of two words or vectors, so only relevant words that are semantically similar can be selected [40].

$$\text{Cosine}\,(A.B) = \rightarrow A. \rightarrow B/\surd \rightarrow A. \rightarrow B \quad (2)$$

Unique Tokens: This is the last step of pre-processing where we obtain fetched tokens, but some tokens may be duplicated, so unique functions capture only distinct items, tokens, words, or vectors.

$$U\_tokns = fUnique\ (List\ of\ tokens) \quad (3)$$

Now, after selecting the unique tokens, it is ready for the next stage, which is feature extractions where tokens are converted into numerical vectors [41].

Feature Extraction: Feature extraction is the process of selecting relevant features from a document that are taking part in the summary and excluding the irrelevant ones.

Bow model: The bag-of-words model contains word tokens, and we transform them into the binary number of bits so that the machine is able to learn or maintain its vocabulary. The output of this model is binary feature vectors.

TF–IDF Model: The Tf–Idf model generates the real feature vectors because, in the BOW model, we cannot predict "how much particular term is important in the document and how many times the particular terms appear in the whole document and how much the term is rare?" [42].

Term-Frequency: It refers to "how many times a particular word or token appears in the document into a whole number of documents".

$$Tf = \text{No. of terms appear in document}/\text{Total no. of terms documents} \quad (4)$$

Inverse Document Frequency: It refers to "how many times a rare term or word appears in the document means the more the term rare is, it possesses more importance".

$$\text{IDF(t)} = \log\_e \text{ (Total number of documents/Number of documents with term t in it).} \quad (5)$$

$$\text{Tf - IDF} = \text{Tf} \times \text{IDf} \quad (6)$$

3.3. Extract Sentences

Outliers Treatments: First, calculate the Z-score so that the outliers may be removed, and the system will only generate a relevant cluster or group which belongs to its title or topic [43].

Sentence Boundary Identification: Secondly, the sentences are extracted based on fuzzy logic; it first takes all the input values (crisp values) into fuzzy set values, which is called "fuzzification", and applies if-then-else rules in the inference engine. Finally, it converts the fuzzy-set values into crisp form; this process is called "defuzzification". The fuzzy perform the best boundary identification in text documents as compared to other extraction techniques [44].

Topics Identification: Thirdly, the retention ratio is calculated by the LDA model, which generates relevant topics that are considered important and are participating in summary sentences. It can be calculated as:

$$\text{RR (Coverage)} = \text{No. of Topics in Summary Sentences/Total No. of Topics.} \quad (7)$$

Generate Summary: Finally, the summary is produced by combining all the facts and extracted sentences in chronological order; however, there are some chances of duplicated sentences that may be included in the summary during extraction from the document. Therefore, the unique function captures only non-repeated or non-duplicated sentences so that repetition is eliminated [45].

3.4. Evaluation Technique

The proposed algorithm captures all facts and figures regarding the title or topic of the document by extracting dates or events from the documents and retaining its position (chronological order) and content from start to end in ascending order [46]. It gives better precision and higher accuracy by preserving its content, features, concepts, and information regarding the title or topic of the document. Further, it filters irrelevant information by considering it as an outlier, which is assumed as far away from its topic or title. Then, the system discards it so that a coherent summary can be produced [47]. The next step is to avoid redundancy. It removes redundant information (redundancy) as compared to other extractive summarizers, such as Luhn's [48] and Edmonson's [49] summarizers, etc. It is not domain-specific but is generic for all textual documents. It tries to balance compression and retention ratios with minimal information loss. It removes very short sentences because short sentences usually do not contain meaningful information as compared to long sentences, so it includes only meaningful sentences [50].

$$\text{Precision/Retention (COVERAGE)} = \text{doc\_term } \Omega \text{ ext\_term/doc-term} \quad (8)$$

Clustering is technique to produce or clusters the same or similar groups of items together. As each cluster contains its center where all data points are gathered around its center but some points may not lie inside its domain or boundary and these data points exist far away from its center so it is treated as outliers. The System extracts the sentences from original documents based on fuzzy logic [51]. Basically, fuzzy perform good extraction from the pool of data where the boundaries of data or text is not clearly visible or identified [52]. The fuzzy logic depends upon possibility rather than probability and whenever something is vague and its boundaries are hard to predict, the fuzzy rules perform better extraction as

compare to any probabilistic models by using IF THEN ELSE with combinations of Logical operators [53–55].

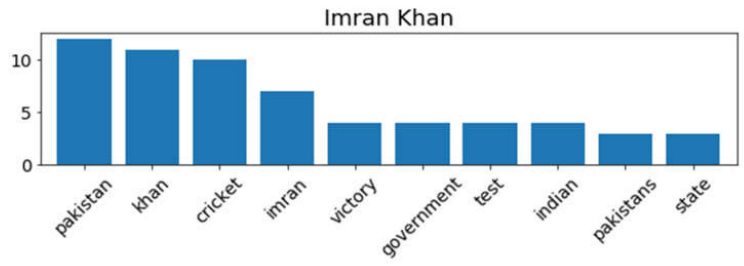
4. Results

The following results, shown in Figure 3a,b, Figure 4a,b and Figure 5a,b were obtained after executing different wiki pages. We also analyzed the compression ratios (CR) of different methods of summarization results (as shown in Table 2) with the proposed algorithm.

Data Sample 1: Imran Khan's Biography From Wikipedia

Original Content:

(**a**)



Summarized Content:

(**b**)

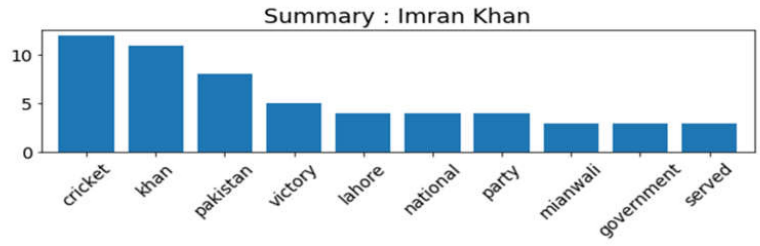
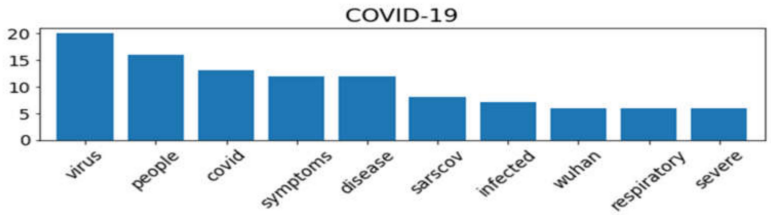


Figure 3. (**a**) Shows the number of semantic words and their frequency in the document. (**b**) Shows the number of semantic words and their frequency in the document.

Data Sample 2: COVID-19 Page From Wikipedia

Original Content:

(**a**)



Summarized Content:

(**b**)

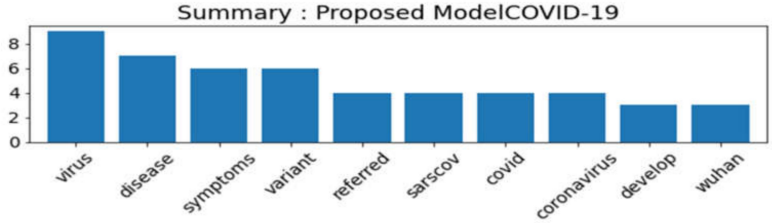
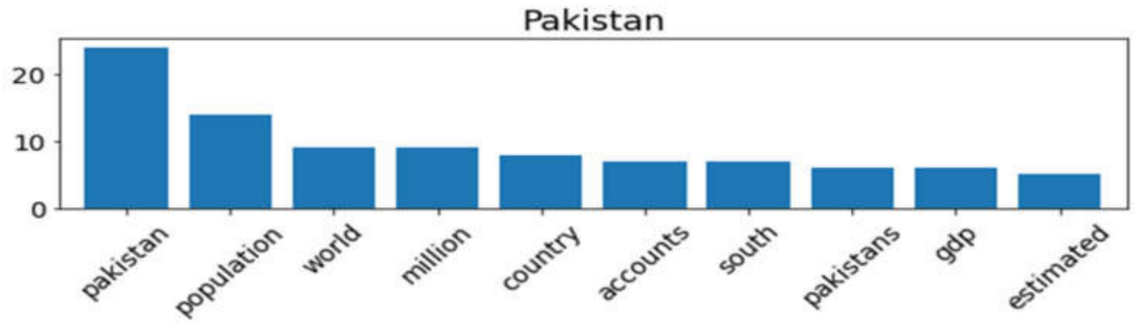


Figure 4. (**a**) Shows the number of semantic words and their frequency in the document. (**b**) Shows the number of semantic words and their frequency in the document.

Data sample 3: Pakistan's page from Wikipedia

Original Content:

(**a**)



Summarized Content:

(**b**)

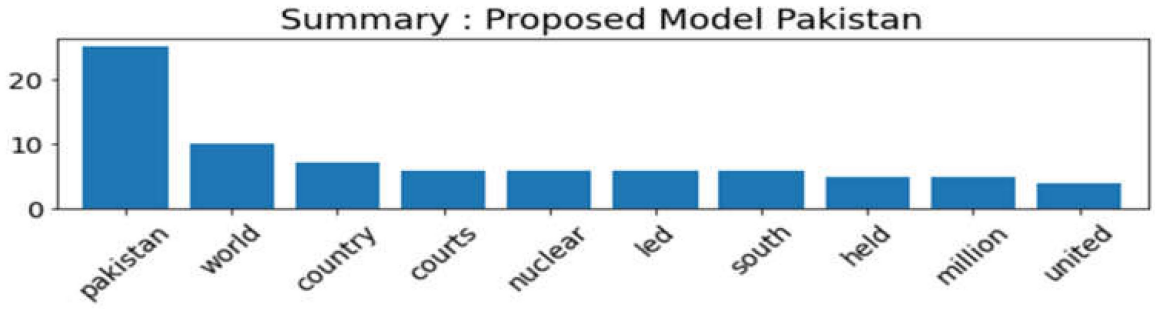


Figure 5. (**a**) Shows the number of semantic words and their frequency in the document. (**b**) Shows the number of semantic words and their frequency in the document.

Table 2. The comparison of different summarizers—Shows the comparison of different summarizer with proposed model.

Topic	Original Words Length	LSA Words Length	CR Ratio %	LEX Words Length	CR Ratio %	KL Words Length	CR Ratio %	Text Words Length	CR Ratio %	Proposed Summarizers Words Length	CR Ratio %
Imran Khan	67,455	1887	27.98	1819	26.98	909	13.48	2909	43.13	7928	11.75
COVID-19	72,424	2217	30.62	1936	26.74	861	11.90	3014	41.62	2121	2.92
Pakistan	116,125	1693	14.58	1585	13.65	7277	6.28	3116	26.84	6605	5.68
Taj Mahal	29,015	16,227	55.92	15,205	52.40	11,384	39.23	18,531	63.86	3700	12.75
Tsunami	34,132	21,745	63.70	17,752	52	6610	19.36	21,250	62.25	2416	7.07

Column one describes the original document's topic, column two shows the original document's length, and columns three to twelve show the different text summarizers' compressed documents' lengths, along with their compression ratios. The result shown in the table suggests that the proposed model has the least compression ratio compared to the LSA and other given models.

The comparison graph of Table 2 is shown below in Figure 6. The graph contains the term coverage ratio captured by the categories of the summarizers. As shown in Figure 6, the proposed model gives the highest percentage of coverage as compared to other summarizers.

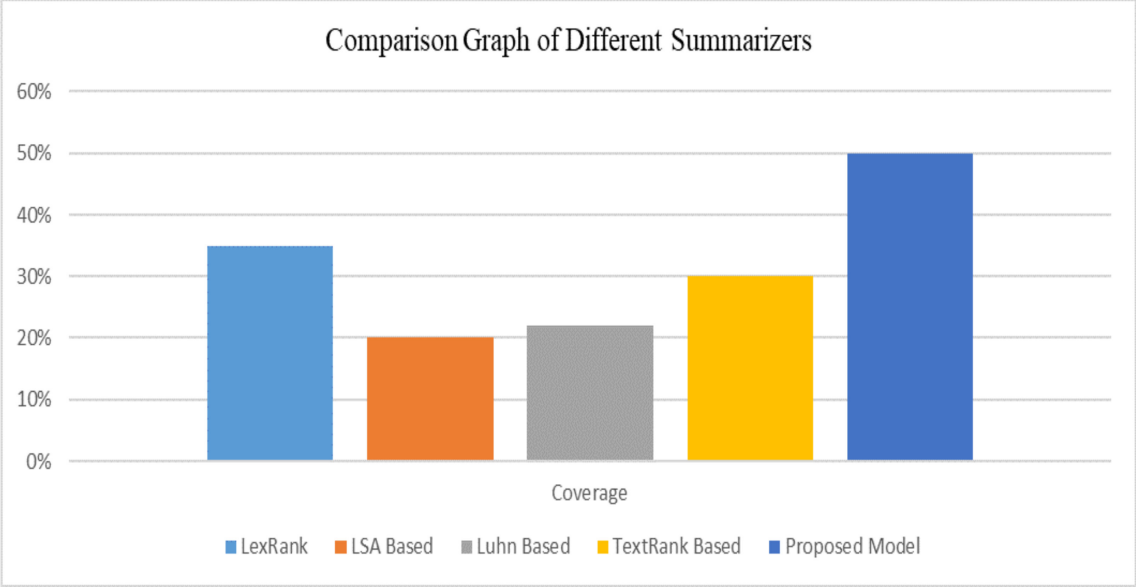


Figure 6. Shows the comparison graph of different summarizers.

The proposed model is also compared with the LDA model, as shown in Figure 7.

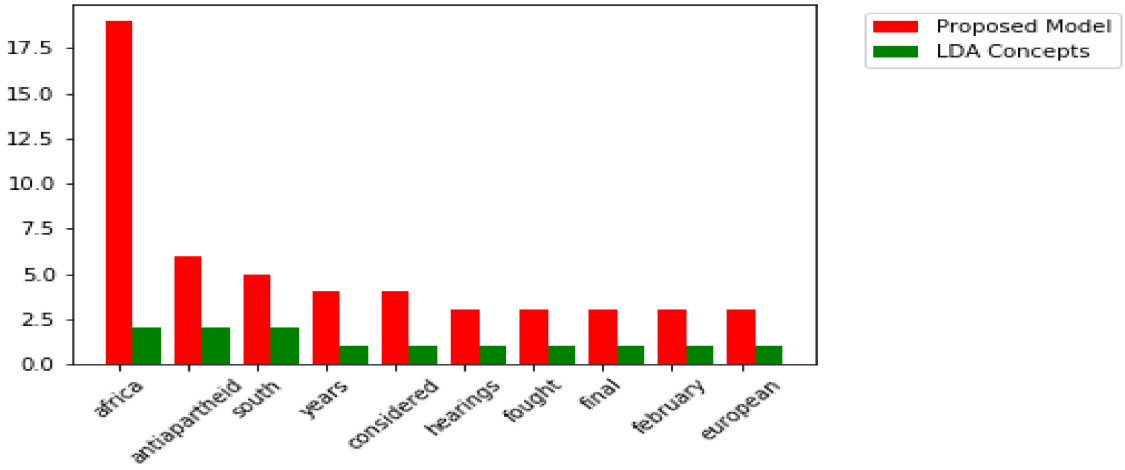


Figure 7. A comparison of the proposed model with the LDA model.

The frequency of the terms is higher than the LDA model's because the output of the LDA sum is very precise; it ignores event dates, which contain very important information [56,57]. In comparison, the proposed model summarizer not only describes concepts but also takes into account event details, which is the important information, in our point of view [58,59]. In both of the above comparisons, the frequency of keywords varies, as shown in Figure 7.

5. Conclusions & Future Recommendations

Table 1 shows the comparison of different summarizers with the proposed model. It shows that during shrinkage from 100–10%, different summarizers, based on rank, features, and concepts, lose their properties. That is why their retention ratio or information gain becomes less, whereas our proposed model tries to preserve maximum information gain with minimum loss by trying to maintain the domain words of the document, which show meaningful information. Figure 6 shows the comparison of the proposed model with the LDA model; the LDA model extracts topics in terms of concepts from the document, and our model also extracted topics from the same document, which shows its topic coverage

is nearly the same as the LDA model's topic coverage. In the proposed summarizer, some information is missing as compared to the original document due to its semantic dissimilarity to either topic or title, which are treated as outliers.

For achieving greater or closer accurate results further, one can experiment with statistical approaches. The ANOVA technique can be used to compare the original one, summarize documents, and check the variance. If its variance is high, then the model can be tuned for better accuracy with minimum variance.

Author Contributions: Conceptualization, W. and Z.F.; methodology, M.A.S. and A.A.A.I.; software, M.F.; validation, S.Z. and M.A.S.; formal analysis, W., Z.F., S.Z. and M.A.S.; resources, Z.F.; data curation, S.Z.; writing—review and editing, K.N. and L.F.N.; visualization, M.F., A.A.A.I. and K.N.; supervision, S.Z and M.A.S.; project administration, M.A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Qaroush, A.; Abu Farha, I.; Ghanem, W.; Washaha, M.; Maali, E. An efficient single document Arabic text summarization using a combination of statistical and semantic features. *J. King Saud Univ. Comput. Inf. Sci.* **2019**, *33*, 677–692. [CrossRef]
2. Mohamed, M.; Oussalah, M. SRL-ESA-TextSum: A text summarization approach based on semantic role labeling and explicit semantic analysis. *Inf. Process. Manag.* **2019**, *56*, 1356–1372. [CrossRef]
3. Khan, A.; Salim, N.; Farman, H.; Khan, M.; Jan, B.; Ahmad, A.; Ahmed, I.; Paul, A. Abstractive Text Summarization based on Improved Semantic Graph Approach. *Int. J. Parallel Program.* **2018**, *46*, 992–1016. [CrossRef]
4. Song, S.; Huang, H.; Ruan, T. Abstractive text summarization using LSTM-CNN based deep learning. *Multimed. Tools Appl.* **2019**, *78*, 857–875. [CrossRef]
5. Sah, S.; Kulhare, S.; Gray, A.; Venugopalan, S.; Prud'Hommeaux, E.; Ptucha, R. Semantic Text Summarization of Long Videos. In *Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 24–31 March 2017*; IEEE: Piscataway, NJ, USA, 2017; pp. 989–997.
6. El-Kassas, W.S.; Salama, C.R.; Rafea, A.A.; Mohamed, H.K. Automatic text summarization: A comprehensive survey. *Expert Syst. Appl.* **2020**, *165*, 113679. [CrossRef]
7. Ma, S.; Sun, X.; Xu, J.; Wang, H.; Li, W.; Su, Q. Improving semantic relevance for sequence-to-sequence learning of chinese social media text summarization. *arXiv* **2017**, arXiv:1706.02459.
8. Sun, X.; Zhuge, H. Summarization of Scientific Paper through Reinforcement Ranking on Semantic Link Network. *IEEE Access* **2018**, *6*, 40611–40625. [CrossRef]
9. Rahman, N.; Borah, B. Improvement of query-based text summarization using word sense disambiguation. *Complex Intell. Syst.* **2020**, *6*, 75–85. [CrossRef]
10. Abu Nada, A.M.; Alajrami, E.; Al-Saqqa, A.A.; Abu-Naser, S.S. Arabic Text Summarization Using AraBERT Model Using Extractive Text Summarization Approach. *Int. J. Acad. Inf. Syst. Res. IJAISR* **2020**, *4*, 6–9.
11. Zhong, M.; Liu, P.; Chen, Y.; Wang, D.; Qiu, X.; Huang, X. Extractive summarization as text matching. *arXiv* **2020**, arXiv:2004.08795.
12. Van Lierde, H.; Chow, T.W. Query-oriented text summarization based on hypergraph transversals. *Inf. Process. Manag.* **2019**, *56*, 1317–1338. [CrossRef]
13. Kanapala, A.; Pal, S.; Pamula, R. Text summarization from legal documents: A survey. *Artif. Intell. Rev.* **2019**, *51*, 371–402. [CrossRef]
14. Muthu, B.; Cb, S.; Kumar, P.M.; Kadry, S.N.; Hsu, C.-H.; Sanjuan, O.; Crespo, R.G. A Framework for Extractive Text Summarization based on Deep Learning Modified Neural Network Classifier. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* **2021**, *20*, 1–20. [CrossRef]
15. Joshi, A.; Fidalgo, E.; Alegre, E.; Fernández-Robles, L. SummCoder: An unsupervised framework for extractive text summarization based on deep auto-encoders. *Expert Syst. Appl.* **2019**, *129*, 200–215. [CrossRef]
16. Gambhir, M.; Gupta, V. Recent automatic text summarization techniques: A survey. *Artif. Intell. Rev.* **2017**, *47*, 1–66. [CrossRef]
17. Fang, C.; Mu, D.; Deng, Z.; Wu, Z. Word-sentence co-ranking for automatic extractive text summarization. *Expert Syst. Appl.* **2017**, *72*, 189–195. [CrossRef]

18. Moratanch, N.; Chitrakala, S. A survey on extractive text summarization. In *Proceedings of the 2017 International Conference on Computer, Communication and Signal Processing (ICCCSP), Chennai, India, 10–11 January 2017*; IEEE: Piscataway, NJ, USA, 2017; pp. 1–6.
19. Al-Sabahi, K.; Zuping, Z.; Nadher, M. A Hierarchical Structured Self-Attentive Model for Extractive Document Summarization (HSSAS). *IEEE Access* **2018**, *6*, 24205–24212. [CrossRef]
20. Rossiello, G.; Basile, P.; Semeraro, G. Centroid-Based Text Summarization through Compositionality of Word Embedding. In Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation across Source Types and Genres, Valencia, Spain, 10 April 2017; pp. 12–21.
21. Wang, Y.; Afzal, N.; Fu, S.; Wang, L.; Shen, F.; Rastegar-Mojarad, M.; Liu, H. MedSTS: A resource for clinical semantic textual similarity. *Comput. Humanit.* **2020**, *54*, 57–72. [CrossRef]
22. Nasar, Z.; Jaffry, S.W.; Malik, M.K. Textual keyword extraction and summarization: State-of-the-art. *Inf. Process. Manag.* **2019**, *56*, 102088. [CrossRef]
23. Miller, D. Leveraging BERT for extractive text summarization on lectures. *arXiv* **2019**, arXiv:1906.04165.
24. Patel, D.; Shah, S.; Chhinkaniwala, H. Fuzzy logic based multi document summarization with improved sentence scoring and redundancy removal technique. *Expert Syst. Appl.* **2019**, *134*, 167–177. [CrossRef]
25. Liu, Y.; Lapata, M. Text summarization with pretrained encoders. *arXiv* **2019**, arXiv:1908.08345.
26. Afsharizadeh, M.; Ebrahimpour-Komleh, H.; Bagheri, A. Query-Oriented Text Summarization Using Sentence Extraction Technique. In *Proceedings of the 2018 4th International Conference on Web Research (ICWR), Tehran, Iran, 25–26 April 2018*; IEEE: Piscataway, NJ, USA; pp. 128–132.
27. Allahyari, M.; Pouriyeh, S.; Assefi, M.; Safaei, S.; Trippe, E.D.; Gutierrez, J.B.; Kochut, K. ext summarization techniques: A brief survey. *arXiv* **2017**, arXiv:1707.02268.
28. Wang, D.; Liu, P.; Zheng, Y.; Qiu, X.; Huang, X.-J. Heterogeneous graph neural networks for extractive document summarization. *arXiv* **2020**, arXiv:2004.12393.
29. Ma, S.; Sun, X.; Lin, J.; Wang, H. Autoencoder as assistant supervisor: Improving text representation for chinese social media text summarization. *arXiv* **2018**, arXiv:1805.04869.
30. Abujar, S.; Hasan, M.; Shahin, M.S.; Hossain, S.A. A Heuristic Approach of Text Summarization for Bengali Documentation. In *Proceedings of the 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Delhi, India, 3–5 July 2017*; IEEE: Piscataway, NJ, USA, 2017; pp. 1–8.
31. Duari, S.; Bhatnagar, V. sCAKE: Semantic Connectivity Aware Keyword Extraction. *Inf. Sci.* **2019**, *477*, 100–117. [CrossRef]
32. Singh, S. Natural language processing for information extraction. *arXiv* **2018**, arXiv:1807.02383.
33. Gupta, S.; Gupta, S.K. Abstractive summarization: An overview of the state of the art. *Expert Syst. Appl.* **2019**, *121*, 49–65. [CrossRef]
34. Al-Abdallah, R.Z.; Al-Taani, A. Arabic single-document text summarization using particle swarm optimization algorithm. *Procedia Comput. Sci.* **2017**, *117*, 30–37. [CrossRef]
35. Gao, Y.; Zhao, W.; Eger, S. SUPERT: Towards new frontiers in unsupervised evaluation metrics for multi-document summarization. *arXiv* **2020**, arXiv:2005.03724.
36. Al-Radaideh, Q.A.; Bataineh, D.Q. A Hybrid Approach for Arabic Text Summarization Using Domain Knowledge and Genetic Algorithms. *Cogn. Comput.* **2018**, *10*, 651–669. [CrossRef]
37. Yousefi-Azar, M.; Hamey, L. Text summarization using unsupervised deep learning. *Expert Syst. Appl.* **2017**, *68*, 93–105. [CrossRef]
38. Saini, N.; Saha, S.; Jangra, A.; Bhattacharyya, P. Extractive single document summarization using multi-objective optimization: Exploring self-organized differential evolution, grey wolf optimizer and water cycle algorithm. *Knowl.-Based Syst.* **2019**, *164*, 45–67. [CrossRef]
39. Lo, K.; Wang, L.L.; Neumann, M.; Kinney, R.; Weld, D.S. S2ORC: The semantic scholar open research corpus. *arXiv* **2019**, arXiv:1911.02782.
40. Maulud, D.H.; Zeebaree, S.R.; Jacksi, K.; Sadeeq, M.A.; Sharif, K.H. State of art for semantic analysis of natural language processing. *Qubahan Acad. J.* **2021**, *1*, 21–28. [CrossRef]
41. Gao, S.; Chen, X.; Li, P.; Ren, Z.; Bing, L.; Zhao, D.; Yan, R. Abstractive text summarization by incorporating reader comments. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 6399–6406.
42. Fu, Z.; Huang, F.; Ren, K.; Weng, J.; Wang, C. Privacy-Preserving Smart Semantic Search Based on Conceptual Graphs Over Encrypted Outsourced Data. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 1874–1884. [CrossRef]
43. Bharti, S.K.; Babu, K.S. Automatic keyword extraction for text summarization: A survey. *arXiv* **2017**, arXiv:1704.03242.
44. Huang, L.; Wu, L.; Wang, L. Knowledge graph-augmented abstractive summarization with semantic-driven cloze reward. *arXiv* **2020**, arXiv:2005.01159.
45. Moradi, M.; Dorffner, G.; Samwald, M. Deep contextualized embeddings for quantifying the informative content in biomedical text summarization. *Comput. Methods Programs Biomed.* **2020**, *184*, 105117. [CrossRef]
46. Cao, M.; Sun, X.; Zhuge, H. The contribution of cause-effect link to representing the core of scientific paper—The role of Semantic Link Network. *PLoS ONE* **2018**, *13*, e0199303. [CrossRef]

47. Sinoara, R.A.; Antunes, J.; Rezende, S. Text mining and semantics: A systematic mapping study. *J. Braz. Comput. Soc.* **2017**, *23*, 9. [CrossRef]
48. Alsaqer, A.F.; Sasi, S. Movie review summarization and sentiment analysis using rapidminer. In *Proceedings of the 2017 International Conference on Networks & Advances in Computational Technologies (NetACT), Thiruvananthapuram, India, 20–22 July 2017*; IEEE: Piscataway, NJ, USA, 2017; pp. 329–335.
49. Sahba, R.; Ebadi, N.; Jamshidi, M.; Rad, P. Automatic text summarization using customizable fuzzy features and attention on the context and vocabulary. In *Proceedings of the 2018 World Automation Congress (WAC), Stevenson, WA, USA, 3–6 June 2018*; IEEE: Piscataway, NJ, USA, 2018.
50. Mallick, C.; Das, A.K.; Dutta, M.; Das, A.K.; Sarkar, A. Graph-based text summarization using modified TextRank. In *Soft Computing in Data Analytics*; Springer: Singapore, 2019; pp. 137–146.
51. Tayal, M.A.; Raghuwanshi, M.M.; Malik, L.G. ATSSC: Development of an approach based on soft computing for text summarization. *Comput. Speech Lang.* **2017**, *41*, 214–235. [CrossRef]
52. Cetto, M.; Niklaus, C.; Freitas, A.; Handschuh, S. Graphene: Semantically-linked propositions in open information extraction. *arXiv* **2018**, arXiv:1807.11276.
53. Lin, H.; Ng, V. Abstractive summarization: A survey of the state of the art. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 9815–9822.
54. Alami, N.; Meknassi, M.; En-Nahnahi, N. Enhancing unsupervised neural networks based text summarization with word embedding and ensemble learning. *Expert Syst. Appl.* **2019**, *123*, 195–211. [CrossRef]
55. Kryściński, W.; McCann, B.; Xiong, C.; Socher, R. Evaluating the factual consistency of abstractive text summarization. *arXiv* **2019**, arXiv:1910.12840.
56. Xu, J.; Gan, Z.; Cheng, Y.; Liu, J. Discourse-aware neural extractive text summarization. *arXiv* **2019**, arXiv:1910.14142.
57. Wei, H.; Ni, B.; Yan, Y.; Yu, H.; Yang, X.; Yao, C. Video Summarization via Semantic Attended Networks. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
58. Hu, J.; Li, S.; Yao, Y.; Yu, L.; Yang, G.; Hu, J. Patent Keyword Extraction Algorithm Based on Distributed Representation for Patent Classification. *Entropy* **2018**, *20*, 104. [CrossRef]
59. Goularte, F.B.; Nassar, S.M.; Fileto, R.; Saggion, H. A text summarization method based on fuzzy rules and applicable to automated assessment. *Expert Syst. Appl.* **2019**, *115*, 264–275. [CrossRef]