Multi-document Text Summarization using Semantic Word and Sentence Similarity: A Combined Approach

Rajendra Kumar Roul
Thapar Institute of Engineering and Technology
Patiala, Punjab, India
raj.roul@thapar.edu

Abstract

The exponential growth in the number of text documents produced daily on the web poses several difficulties to people who are responsible for collecting, organizing, and searching different textual content related to a particular topic. Automatic Text Summarization is effective in this direction, as it can evaluate a large number of documents and extract essential information. However, the limits of automatic text summarization must be overcome by devising practical solutions. Even though current research efforts focus on this direction for future advances, they still face numerous obstacles. This work suggests a combined semantic-based word and sentence similarity technique to summarise a corpus of text documents. KL-divergence approach is used to organize the sentences in the final summary. Experimental work is conducted using DUC datasets, and the obtained results are promising.

1 Introduction

With the widespread adaptation of technology, a large number of documents are getting digitized, resulting in a rapid influx of textual data. This data often contains crucial information; however, absorbing all this information can be difficult and time-consuming. Automatic Text Summarization (ATS) is the process of condensing data into useful and comprehensible information. By distilling out meaningful details, ATS makes referring documents much more efficient. ATS can be done in two ways: *Extractive* and *Abstractive*. Extractive summarization selects sentences of importance directly from the source text, which can either be within a single document (called single document text summarization) or a group of documents (called multi-document text summarization)(Gupta and Lehal, 2010)(Roul and Arora, 2019). On the other hand, abstractive text summarization is an understanding of the main concept of its expression in clear natural language. When abstraction is used for text summarization in deep learning issues, it can overcome the extractive method's grammatical inconsistencies.

1.1 Motivation

There has already been a vast amount of research on text summarization such as 'graph-based summarization (Elbarougy et al., 2020)', 'clustering-based summarization(Wang et al., 2011)(Roul et al., 2016)', 'machine learning based summarization(Roul et al., 2017)(Abdi et al., 2018)', 'summarization based on Fuzzy logic' (Suanmali et al., 2009), topic-modeling based summarization(Roul et al., 2019)(Alami et al., 2021)(Roul, 2021) etc. As listed below, all of these existing text summarising approaches have some common limitations:

i. Two different sentences made up of completely different words can share a similar meaning, and it should be taken care of when the summary is generated.

ii. Stop-words like 'a,' 'an,' 'the,' 'of,' and so on are often excluded from surface matching algorithms since they are relatively prevalent throughout all articles in the collection. However, these words play a significant part in calculating sentence similarity since they provide structural information that is used to infer the content of the phrase, and hence they should not be ignored.

iii. The significance of the words in the scope of the sentence is ignored.

Proceedings of the 18th International Conference on Natural Language Processing, pages 423–430
Silchar, India. December 16 - 19, 2021. ©2021 NLP Association of India (NLPAI)

iv. When computing the similarity of sentences, giving equal weight to each word is still lacking.

This study considers the extractive approach towards achieving summarization and presents a snapshot of the content of a group of related documents. Semantic-based word and sentence similarities are combined to generate a coherent summary at the end.

1.2 Contribution

The following is a summary of the paper's contributions:

i. The problem of organizing and logically displaying the gathered data has not yet received attention. The suggested method computes each sentence's cohesiveness score to eliminate redundancy and picks the top 'm' percent of sentences based on the cohesion score to generate the coherent summary.

ii. Every word of the generated coherent summary gets equal algebraic treatment by considering modified harmonic mean.

iii. The suggested method, which includes all the stop-words, treats each word in a sentence separately according to its semantic structure.

iv. The proposed approach computes the semantic similarity between the sentences to get a more information-rich coherent summary.

v. In the generated summary, all the sentences are arranged as per their importance using the Kullback-Leibler divergence technique.

Empirical results show that the suggested approach is more efficient than the existing extractive text summarization approaches.

2 Proposed Approach

Assume a corpus P consists of D number of documents. At first, all D documents are merged into a single huge collection. All the sentences of this huge collection are extracted to form a set of sentences $S = \{s_1, s_2, s_3, \cdots, s_n\}$. Below steps discuss how the coherent summary is generated from the corpus P.

1. Word similarity calculation:
   Semantic similarity between two words (a and b) is calculated using WordNet (Miller, 1995) having 206942 words and 117660 synsets. Figure 1 shows the hierarchy of these synsets (or concepts) of the semantic database of the WordNet. The symbol '$\cdots$' is used to represent more synonym words of a synset. To extract the information from the semantic database, WordNet.Net[1], a public framework is used here. The semantic sim-

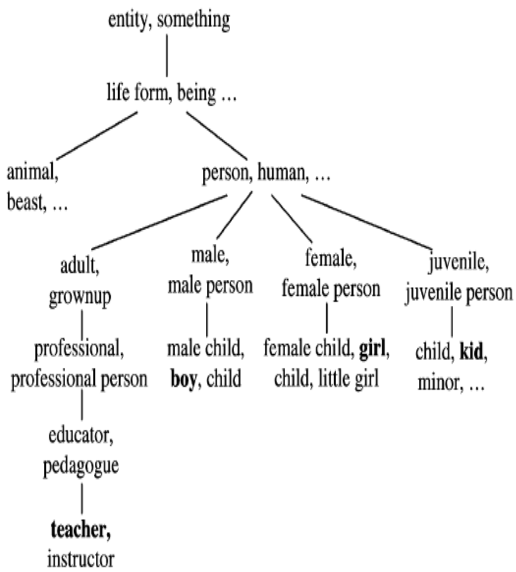

Figure 1: Hierarchical Semantic Net

ilarity $sim(a, b)$ is calculated using two functions:

- minimum path length (min_path)
- depth of the subsumer ($depth\text{-}sub$)

$$sim(a, b) = function(f_1(min\_path), f_2(depth\text{-}sub)) \quad (1)$$

i. Computing minimum path length:
   There are 3 posibilities as mentioned below while computing the $sim(a, b)$:
   - a and b are belong to the same synset: since they have the same meaning, a semantic path length of zero is allocated between them.
   - a and b have not belonged to the same synset: here the shortest path between the two synsets is calculated by 'max-similarity' algorithm (Pedersen et al., 2005) using Pywsd[2].
   - a and b do not belong to the same synset, but their corresponding

[1] http://en.wikipedia.org/wiki/Brown_Corpus
[2] https://github.com/alvations/pywsd

synset consists of one or more common words: in this circumstance, a semantic path length of one is allocated since both synsets share some of the same terms.

In light of the three situations presented above, the $f_1(min\_path)$ of equation 1 is fix to be a steadily reducing function as shown in equation 2.

$$f_1(min\_path) = e^{-\alpha(min\_path)} \tag{2}$$

here $\alpha \in [0,1]$ is constant.

ii. Depth of the subsumer computation:
The depth of the subsumer is determined by counting the levels from the subsumer to the hierarchical net's top. Words in the top layers of the hierarchy have a broader meaning and fewer semantic concepts than words in the lower layers. When computing the similarity, this behavior must be taken into account. Thus, it is necessary to scale up the $sim(a,b)$ for subsuming words at bottom layers, and for subsuming words at higher layers, one needs to scale down the $sim(a,b)$. This shows $f_2(depth\text{-}sub)$ of equation 1 should be monotonically increasing function as shown illustrated in equation 3.

$$f_2(depth\text{-}sub) = \frac{e^{\beta.depth\text{-}sub} - e^{-\beta.depth\text{-}sub}}{e^{-\beta.depth\text{-}sub} + e^{\beta.depth\text{-}sub}} \tag{3}$$

where $\beta \in [0,1]$ is a smoothing factor, and it determines the contribution of depth of subsumer. With respect to α of equation 2,The percentage contribution of subsumer depth reduces as $beta$ rises. The word's depth in the hierarchy is not considered when $\beta > \infty$ (Shepard, 1987). The optimum values of β and α are set to 0.46 and 0.2 respectively (Erkan and Radev, 2004).

iii. Finally, semantic similarity between a and b is measured using equation 4.
$sim(a,b) =$

$$e^{-\alpha.min\_path} * \frac{e^{\beta.depth\text{-}sub} - e^{-\beta.depth\text{-}sub}}{e^{\beta.depth\text{-}sub} + e^{-\beta.depth\_sub}} \tag{4}$$

The value of $sim(a,b) \in [0,1]$.

2. *Word score calculation based on modified harmonic mean*:
The harmonic mean can't be determined without taking into account all of the words in the corpus. It gives each word equal weight and is excellent for qualitative data. The modified Harmonic Mean (HM) formula is used to produce a ranking score for each word in relation to the total corpus, as shown in equation 5.

$$HM_q = \frac{n-1}{\sum\limits_{p,p\neq q} \frac{1}{sim(a,b)+k}} \tag{5}$$

n indicates the number of words in P, $sim(a,b)$ represents the similarity score between the two words a and b as shown in equation 4. Except for the reflexive pair, all of the pairs of words are summed up. k is a factor that must be included in every similarity score in the algorithm to ensure that the score, when divided by 1, does not provide an exception.

3. *Selection of representative words*:
Upon calculating the modified harmonic of every word in P, the top $l\%$ words[3] are saved in a list L_{rep} as representative words of the corpus P. Now, as stated in equation 6, the cosine-similarity ($cos\text{-}sim$) between s and the list L_{rep} is calculated.

$$cos\text{-}sim(s, L_{rep}) = \frac{s.L_{rep}}{||s|| * ||L_{rep}||} \tag{6}$$

Sentences having cosine similarity more than 0.75[4] are considered.

4. *Calculation of sentence similarity*:
The following steps are used to measure similarity between two sentences :

   i. A combined word set construction:
   To compare the similarity of two sentences s_1 and s_2, create a combined set of words $J_s = \{w_1, w_2, \cdots, w_n\}$, where each w_i is an unique word from s_1 and s_2. This means there are no common terms in J_s between s_1 and s_2. Because they carry syntactic information, J_s also contains function words. The word form is maintained in the same way as it appears in the sentence.

[3] chosen by the experiment
[4] decided by experiment

ii. Similarity between sentences:
The structural semantic vector ($lsv_i, i \in [1, 2]$ of s_1 and s_2 is computed first. Each lsv_i entry corresponds to a word in J_s. For every $w \in J_s$, the following steps are utilised to calculate the $lsv_i, i = 1$ for s_1 (denoted as lsv_1). Prior to the commencement of the procedure, a semantic vector sv is considered, with every entries set to zero.

case-a'. $w \in s_1$: the sv entry corresponding to s_1 is set to 1. This value is indicated in equation 7.

$$lsv_1 = I(w)^2 * sv \tag{7}$$

case-b'. $w \notin s_1$: an identical word (designated as $\overline{w}$) is found in $s1$ by analyzing the semantic relatedness of w to each word in s_1 (semantic relatedness is determined using equation 4). The related entry in the (sv) is fix to the estimated similarity, if it exceeds a per-defined threshold value[5], otherwise it is set to zero. Equation 8 shows the detail.

$$lsv_1 = I(w) * I(\overline{w}) * sv \tag{8}$$

In the same manner for s_2, the lexical-semantic vector lsv_2 is generated by converting all entries of sv to zero, and then executing case-a' and b' as mentioned above.

The cosine coefficient between lsv_1 and lsv_2, as stated in equation 9, is the final value of the semantic sentence similarity.

$$sim(lsv_1, lsv_2) = \frac{lsv_1.lsv_2}{||lsv_1|| * ||lsv_2||} \tag{9}$$

The value of $sim(lsv_1, lsv_2) \in [0, 1]$.

iii. Corpus Statistics:
One can measure the value of distinct words of a sentence using corpus statistics. This is critical because, as indicated in equation 10, one must incorporate stop-words with lower priority relating to other words in a sentence.

$$I(w) = 1 - \frac{Log(x + 1)}{Log(S + 1)} \tag{10}$$

The frequency of the word w in P is represented by x, while the total number of words in P is represented by S. To prevent zero, x and S are both increased by one. $I(w) \in [0, 1]$.

5. *Cohesion score calculation*:
The cohesiveness score of each sentence in relation to the related document is calculated by determining the Equlidean distance between s_j and the document's centroid dc, as illustrated in equation 11.

$$coh(s_j) = ||(dc - s_j)|| \tag{11}$$

dc calculated using the equation 12.

$$dc = \frac{\sum_{i=1}^{n'} s_i}{n'} \tag{12}$$

Here, $n' \in d_i$ is the number of sentences.

6. *Generating final summary list*
The top m percent sentences based on the *cohesion score* are picked and saved in a new list *NL*, which constitutes the final summary (given in equation 11).

7. *Organising sentences*
An entropy-based mechanism is presented to organise all of the sentences in *NL* according to their relevance (i.e., weight), which is explained below:

- Each word's probability for a sentence is calculated using equation 13.

$$P(w|s) = \frac{term\text{-}frequency(w, s)}{|s|} \tag{13}$$

- Each word's probability for a document d is calculated using equation 14.

$$P(w|d) = \frac{term\text{-}frequency(w, d)}{|d|} \tag{14}$$

The weight of s (referred as $Weight_s$) is determined by its comparison to the document d and is evaluated using equation 16. As illustrated in the equation 15, KL-divergence (*KLD*) (Kumar

[5]determined by experiment

et al., 2009) is used to make the comparison between s and d.

$$KLD(s,d) = \sum_{w} P(w|s) Log(\frac{P(w|s)}{P(w|d)}) \quad (15)$$

$$Weight_s = \frac{1}{KLD(s,d)} \quad (16)$$

Sentences are ordered in the final summary *NL* according to their weights, and constitute the *system-generated summary*.

2.1 *Extractive Gold Summary (EGS) generation*

Sentences containing important information should be categorised as "Important," else they should be labeled as "Not-Important." The sentences identified as' 'Important" are considered for inclusion in the document's summary. The procedures outlined below show how EGS is created from the DUC dataset (P_{duc}) [6].

i) Every document $d \in P_{duc}$ is processed one sentence at a time. For this, the Natural Language Toolkit [7] is employed.

ii) A list L contains all terms of 4 human-written summaries. The number of related words r' between L and s is calculated for each sentence $s \in d$, where r' fluctuates from one sentence to another.

iii) The score for s is measured by the value of r'. Finally, the sentences are ranked and placed in a new list L' depending on these scores.

iv) The extractive gold summary of d is generated by selecting the top m words from L'. The value of m is used to conduct the experiment. In this approach, each P_{duc} document received a 5-sentence extractive gold summary.

3 Analysis of Experimental Results

The description of DUC datasets[8] used for experimental purposes are shown in Table 1. Two most popular techniques, such as ROUGE-N and summary readability, are used to compare the proposed approach with the state-of-the-art approaches, and those are discussed in the following sections.

[6] http://www.duc.nist.gov
[7] http://www.nltk.org/
[8] http://www.duc.nist.gov

3.1 Comparing the performances using ROUGE-N score

i. ROUGE-2 and ROUGE-1 scores (shown in Figures 3 and 2) of the propose model using DUC-2002 dataset are compared with 5 conventional text summarization models (TGRAPH(Parveen et al., 2015), ILP(Woodsend and Lapata, 2010), URANK(Wan, 2010), TextRank(Mihalcea and Tarau, 2004), NN_SE(Cheng and Lapata, 2016)).

ii. Similar way, ROUGE-1, ROUGE-2, and ROUGE-SU4 scores (shown in Figures 4 - 6) of the propose approach using DUC-2006 dataset are compared with 6 conventional text summarization approaches (OnModer(Ye et al., 2007), CTMSUM (Yang et al., 2015), TopicalN(Wang et al., 2007), IIITH-Sum(Jagarlamudi et al., 2006), RMSUM (Zhai and Lafferty, 2017), SFU_v36(Melli)).

iii. Results on the DUC-2002 dataset show that both ROUGE-1 and ROUGE-2 scores of the proposed approach are better than conventional approaches.

iv. Results on DUC-2006 dataset shows that the ROUGE-1 and ROUGE-SU4 scores of the proposed approach are better, but for ROUGE-2 score, CTMSUM and IIITH-Sum are better compared to the proposed approach.

v. Overall from the results of both DUC datasets, it can be concluded that the proposed model is either better or comparable with the existing text summarization techniques.

3.2 Comparing the performance using readability of the summary

Readability of summary means how system-generated summary can read and understand by others in a better manner and is affected by many parameters like sentence weight, sentence length, sentence density etc.(Zamanian and Heydari, 2012). For computing the readability of the summary, statistical methods are generally used (Kondru, 2007), and some of the methods are used by the proposed approach (Table 2). When the

Table 1: DUC Datasets

Dataset	Number of sets	Number of documents	Avg. number of sentence per document	Summary Length	Source
DUC-2006	48	1230	32.22	240	AQUAINT
DUC-2002	54	532	34.55	140	TREC-9

Table 2: Methods of summary readability

Method	Formula
Coleman Liau (CL)	5.89 * (characters/words) - 0.3 * (sentences/words) -15.8
Flesch Kincaid Grade Level (FKGL)	0.39 * (words/sentences) + 11.8 * (syllables/words) - 15.59
Automated Readability Indexing (ARI)	3.70 * (characters/words) + 0.4 * (words/sentences) - 20.42

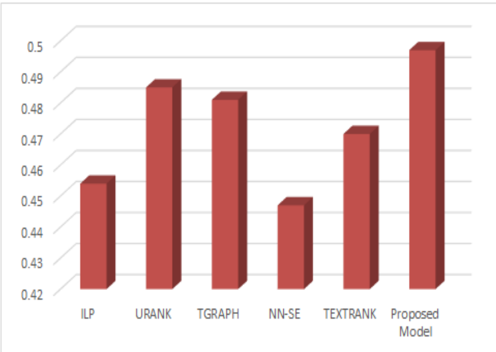

Figure 2: DUC-2002 (ROUGE-1)

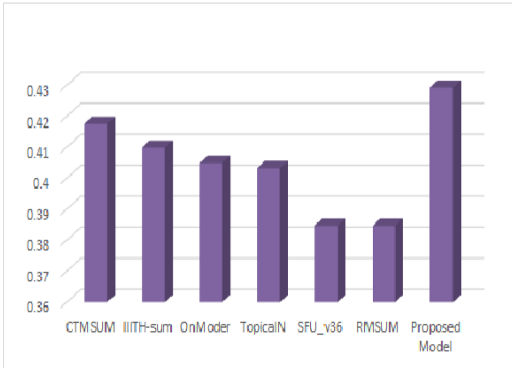

Figure 4: DUC-2006 (ROUGE-1)

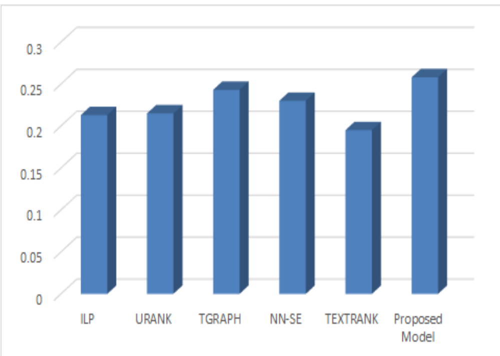

Figure 3: DUC-2002 (ROUGE-2)

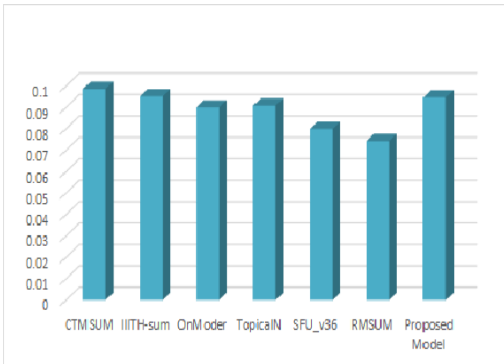

Figure 5: DUC-2006 (ROUGE-2)

summary readability score is very high, it indicates that the system-generated summary is highly user-friendly in terms of understanding and reading. Figures 7 and 8 show the results. Experimentally, it can be concluded that the obtained results of the proposed model are more promising.

4 Conclusion

By combining semantic-based word and sentence similarity, the proposed method suggested a novel

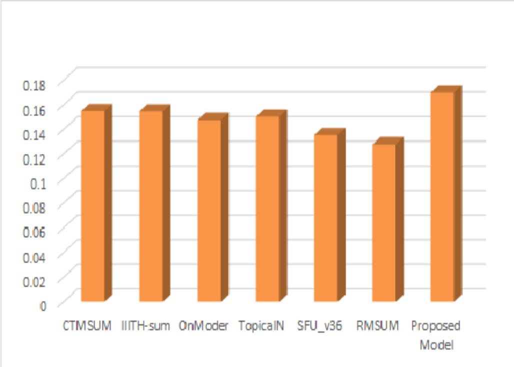

Figure 6: ROUGE-SU4 (DUC-2006)

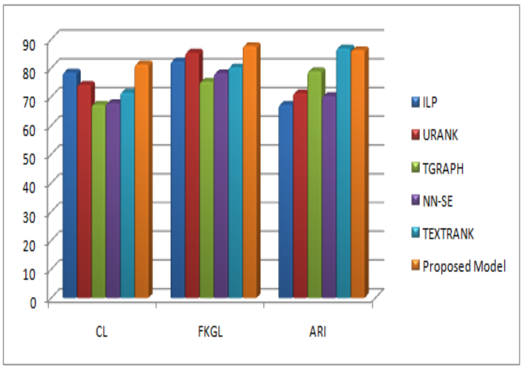

Figure 7: Readbility of summary (DUC-2002)

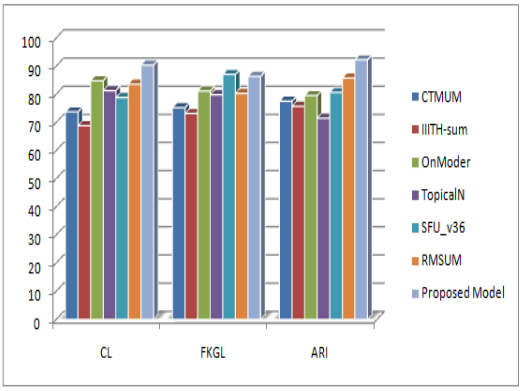

Figure 8: Readbility of summary (DUC-2006)

extractive text summarisation technique. Modified harmonic mean is used to select the important words of each sentence, and then the sentence similarity is computed. Based on the cohesion score, top sentences are selected that constitute the final summary. The sentences in the final summary are organized using KL-divergence approach. The proposed method's experimental work is carried out on two DUC datasets. The proposed approach outperforms the standard approaches on DUC-2006 and DUC-2002 datasets, according to empirical results. This work can be improved even more by using the abstractive text summarization technique to produce a more grammatical-based summary. In the medical domain, many summarization models are proposed, but still, the medical documents have vague terms that make it difficult to extract useful information. The proposed model can use the medical data as the input documents to generate a useful summary that can help the medical system.

References

Asad Abdi, Siti Mariyam Shamsuddin, Shafaatunnur Hasan, and Jalil Piran. 2018. Machine learning-based multi-documents sentiment-oriented summarization using linguistic treatment. *Expert Systems with Applications*, 109:66–85.

Nabil Alami, Mohammed Meknassi, Noureddine En-nahnahi, Yassine El Adlouni, and Ouafae Ammor. 2021. Unsupervised neural networks for automatic arabic text summarization using document clustering and topic modeling. *Expert Systems with Applications*, 172:114652.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, page 484–494.

Reda Elbarougy, Gamal Behery, and AKRAM EL KHATIB. 2020. Graph-based extractive arabic text summarization using multiple morphological analyzers. *Journal of Information Science & Engineering*, 36(2).

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Vishal Gupta and Gurpreet Singh Lehal. 2010. A survey of text summarization extractive techniques. *Journal of emerging technologies in web intelligence*, 2(3):258–268.

Jagadeesh Jagarlamudi, Prasad Pingali, and Vasudeva Varma. 2006. Query independent sentence scoring approach to duc 2006. In *In Proceeding of document understanding conference (DUC-2006)*.

Jagadeesh Kondru. 2007. Using part of speech structure of text in the prediction of its readability. *Computer Science & Engineering*.

Chandan Kumar, Prasad Pingali, and Vasudeva Varma. 2009. A light-weight summarizer based on language model with relative entropy. In *Proceedings of the 2009 ACM symposium on Applied Computing*, pages 1752–1753.

Gabor Melli. Description of squash, the sfu question answering summary handler for the duc-2006 summarization task. *safety*, 1:14345754.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Daraksha Parveen, Hans-Martin Ramsl, and Michael Strube. 2015. Topical coherence for graph-based extractive summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1949–1954.

Ted Pedersen, Satanjeev Banerjee, and Siddharth Patwardhan. 2005. Maximizing semantic relatedness to perform word sense disambiguation. *Research Report UMSI 2005/25, University of Minnesota Supercomputing Institute*, 25:2005.

Rajendra Kumar Roul. 2021. Topic modeling combined with classification technique for extractive multi-document text summarization. *Soft Computing*, 25(2):1113–1127.

Rajendra Kumar Roul and Kushagr Arora. 2019. A nifty review to text summarization-based recommendation system for electronic products. *Soft Computing*, 23(24):13183–13204.

Rajendra Kumar Roul, Aditya Bhalla, and Abhishek Srivastava. 2016. Commonality-rarity score computation: a novel feature selection technique using extended feature space of elm for text classification. In *Proceedings of the 8th annual meeting of the Forum on Information Retrieval Evaluation*, pages 37–41.

Rajendra Kumar Roul, Samarth Mehrotra, Yash Pungaliya, and Jajati Keshari Sahoo. 2019. A new automatic multi-document text summarization using topic modeling. In *International conference on distributed computing and internet technology*, pages 212–221. Springer.

Rajendra Kumar Roul, Jajati Keshari Sahoo, and Rohan Goel. 2017. Deep learning in the domain of multi-document text summarization. In *International Conference on Pattern Recognition and Machine Intelligence*, pages 575–581. Springer.

Roger N Shepard. 1987. Toward a universal law of generalization for psychological science. *Science*, 237(4820):1317–1323.

Ladda Suanmali, Naomie Salim, and Mohammed Salem Binwahlan. 2009. Feature-based sentence extraction using fuzzy inference rules. In *2009 International Conference on Signal Processing Systems*, pages 511–515. IEEE.

Xiaojun Wan. 2010. Towards a unified approach to simultaneous single-document and multi-document summarizations. In *Proceedings of the 23rd international conference on computational linguistics*, pages 1137–1145. Association for Computational Linguistics.

Dingding Wang, Shenghuo Zhu, Tao Li, Yun Chi, and Yihong Gong. 2011. Integrating document clustering and multidocument summarization. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(3):1–26.

Xuerui Wang, Andrew McCallum, and Xing Wei. 2007. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pages 697–702. IEEE.

Kristian Woodsend and Mirella Lapata. 2010. Automatic generation of story highlights. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 565–574. Association for Computational Linguistics.

Guangbing Yang, Dunwei Wen, Nian-Shing Chen, Erkki Sutinen, et al. 2015. A novel contextual topic model for multi-document summarization. *Expert Systems with Applications*, 42(3):1340–1352.

Shiren Ye, Tat-Seng Chua, Min-Yen Kan, and Long Qiu. 2007. Document concept lattice for text understanding and summarization. *Information Processing & Management*, 43(6):1643–1662.

Mostafa Zamanian and Pooneh Heydari. 2012. Readability of texts: State of the art. *Theory & Practice in Language Studies*, 2(1).

Chengxiang Zhai and John Lafferty. 2017. A study of smoothing methods for language models applied to ad hoc information retrieval. In *ACM SIGIR Forum*, volume 51, pages 268–276. ACM.