Intelligent Systems with Applications 16 (2022) 200129

Contents lists available at ScienceDirect

Intelligent Systems with Applications

journal homepage: www.journals.elsevier.com/intelligent-systems-with-applications

# CORPURES: Benchmark corpus for urdu extractive summaries and experiments using supervised learning

Muhammad Humayoun [*,a], Naheed Akhtar [b]

[a] Computer Information Science Division, Higher Colleges of Technology, Abu Dhabi, United Arab Emirates
[b] Department of Computer Science, University of Education, Lahore, Pakistan

ARTICLE INFO

*Keywords:*
Natural language processing
Automatic text summarization
Single document summarization
Extraction based summarization
Extracts
Urdu summary corpus
Supervised learning
Urdu language
Resource poor language

ABSTRACT

Text summarization is the process of shortening the text so that it conveys the key points. Several text summarization methods and benchmark corpora are available for languages like English. A significant hurdle in developing and evaluating existing or new text summarization methods is the unavailability of standardized benchmark corpora, especially for South Asian languages. Among other things, a reference corpus enables researchers to compare existing state-of-the-art methods. Our study addresses this gap by developing a benchmark corpus for one of the widely spoken yet under-resourced language Urdu. The reported corpus contains 161 documents with manually written extractive summaries from the newswire domain. We also perform several experiments on the corpus to show how it can be used to develop, evaluate, and compare text summarization systems using a supervised learning approach for the Urdu language. Our results show that the state of the art classifiers are good candidates for Urdu text summarization when supervised learning techniques are employed. Also, a radical word segmentation technique such as fixed-length segmentation outperforms all other settings (Senetnce Match $F_1$ = 57%, ROUGE-2 $F_1$ = 64.4%). On the basic preprocessing of Urdu texts, we observe that tokenization of words on space is a reliable approach until the proper word segmentation tools for Urdu are mature enough. On word similarity features needed for supervised learning, it is observed that a radical stemming such as Ultra stemming with length (1 and 2) works better than the existing stemming and lemmatization tools for Urdu. Finally, the artificially generated datasets do not significantly improve results compared to the original data.

## 1. Introduction

With the rapid growth of the internet, an overwhelming amount of reading sources are available today for user consumption causing information overload. Automatic text summarization has the potential to play an essential role in our times. Automatic text summarization is the process of shortening a text as a summary to convey the text's key points. The text might be gathered from a single document or multiple documents written on the same topic. Generally, the produced summary is significantly less than the source text but never more than half of the source text (Radev, Hovy, & McKeown, 2002). Distinguishing the more informative parts of a text from the less informative parts is the main challenge in automatic text summarization (Das & Martins, 2007; Steinberger & Ježek, 2012).

Two main approaches used in automatic text summarization are *abstraction based* and *extraction based*. Abstraction based approach gathers important information from sentences and constructs a coherent summary from the original content by eliminating insignificant details. Such a summary may also consist of new sentences that are not present in the actual document (Ye, Chua, Kan, & Qiu, 2007). This requires solving challenging problems such as semantic representation, inference, natural language generation, among others (Radev et al., 2002). Abstractive summarization is still an open research problem that "remains a researcher's dream" (Radev et al., 2002).

In contrast, extraction based techniques rely on identifying and ranking the most "informative" sentences from the source text. These selected sentences are kept intact as a unit even when some of the parts do not contain the most important information. Extractive methods have been used over the last few decades and often provide robust summaries, if not always better. The literature proposes several supervised and unsupervised methods to produce extractive summaries (also known as *extracts*). In supervised machine learning, instances are labeled. It means

* Corresponding author.
*E-mail addresses:* mhumayoun@hct.ac.ae (M. Humayoun), naheedswl@ue.edu.pk (N. Akhtar).

https://doi.org/10.1016/j.iswa.2022.200129
Received 25 August 2021; Received in revised form 18 July 2022; Accepted 20 September 2022
Available online 28 September 2022

that the corresponding correct output (i.e., the right answer) is given along with each instance. In contrast, the labels (i.e., the right answers) are not provided in unsupervised methods.

Some of the unsupervised methods are based on statistics. For instance, the use of word frequency (Luhn, 1958) and sentence position (Baxendale, 1958; Edmundson, 1969) are classic examples. Some popular unsupervised learning approaches are Latent Semantic analysis (LSA) (Murray, Renals, & Carletta, 2005; Steinberger & Jezek, 2004; Steinberger & Ježek, 2009), Maximum Marginal Relevance (MMR), and Integer Programming (Gillick & Favre, 2009). Graph-based unsupervised approaches have also gathered attention (Erkan & Radev, 2004; Fang, Mu, Deng, & Wu, 2016; Mihalcea & Tarau, 2004). Related work in supervised Learning in the context of text summarization is given in Section 2.1.

The lack of benchmark corpora is the main bottleneck in developing and evaluating computational methods for automatic text summarization. The majority of benchmark corpora for automatic text summarization are mainly concerned with the English language. It is evident from the shared tasks offered by, first, Document Understanding Conference (DUC),[1] and later, by Text Analysis Conference (TAC)[2] – both focused on English mainly. In terms of language resource development (in general), and automatic text summarization (in particular), South Asian Languages have mostly been neglected (Becker & Riaz, 2002). Given that South Asia has one of the highest language diversity[3] spoken by more than 1.8 billion people,[4] we think there is a desperate need to develop language resources for South Asian languages to foster research for these languages. This view is supported by (Baker & McEnery, 1999; McEnery, Baker, & Burnard, 2000). However, in recent times, the NLP community seems to show more interest in South Asian languages (most notebly Urdu, Hindi, Bengali and Punjabi), for which language resources are the prerequisite. For Urdu, such related studies are discussed in Section 2.1.

More than 100 million people worldwide, including Pakistan and India, speak Urdu (Grimes, 2021). It is an Indo-Aryan language and having a modified Perso-Arabic alphabet (Humayoun, Hammarström, & Ranta, 2007b). Urdu is written in Nastalique writing style – a highly complex writing system that is cursive and context-sensitive. Urdu draws its advanced vocabulary from Persian and Arabic and day-to-day usage vocabulary from the native languages of South Asia (Virk, Humayoun, & Ranta, 2010). Urdu lacks capitalization. It makes identifying proper nouns, titles, acronyms, and abbreviations a difficult task. Similar to Arabic and Persian, diacritic marks (vowels) are optional and hardly present in the written text. Thus, words are often guessed with the help of context. Urdu is a free word order (Subject Object Verb) language Humayoun, Hammarström, & Ranta (2007a).

### 1.1. Contribution

We present three key contributions to the advancement of the Urdu text summarization research. Firstly, we report the development of a benchmark **CORP**us of **UR**du **E**xtractive **S**ummaries (CORPURES). CORPURES contains 161 documents with manually written extractive summaries from the newswire domain (news articles from BBC Urdu). The Urdu language experts wrote the reference summaries. We used a simpler version of crowdsourcing to develop reference summaries of good quality in a short time. There are four summaries per document, three of which are produced by the contributors, whereas the fourth summary is produced using majority voting. We have publicly released[5] CORPURES that happens to be the first Urdu extractive summary corpus. Also, the development guidelines can be followed to produce a much bigger corpus in the future.

Secondly, we produce a general framework to generate a textual summary (extract) of the Urdu language text. An extract can be generated by many supervised learning methods (classifiers).

Thirdly, we perform several benchmark experiments on CORPURES such as:

1. The Urdu extractive summarization task is considered a binary classification problem, and six classifiers are employed to see how they perform for the task.
2. The common features reported in the literature are used, and the importance of each feature is measured using three attribute selection algorithms for various preprocessing settings given below.
3. Space is not a reliable indicator of a word in Urdu. Thus, the effect of space segmented and properly segmented documents in CORPURES is evaluated for the task with various preprocessing settings given below.
4. The effect of the following preprocessing settings is evaluated in detail for the extractive summarization task:
   (a) Stopword removal
   (b) Adding lexical information such as various lemmatization & stemming techniques.
   (c) Adding semantic information using Levenshtein distance and word embeddings as a similarity measure (instead of string equality).
5. A radical technique such as fixed-length token segmentation as words is employed on CORPURES, and its effect is evaluated.
6. Two synthetic datasets are produced from CORPURES, and the effectiveness of synthetic data for the task is evaluated.

It is worth mentioning that before this research work, the said approach was never empirically evaluated for the Urdu News text summarization task for extractive summaries.

## 2. Related work

### 2.1. Text summarization and supervised learning

Over the last two decades, supervised learning has shown excellent results for natural language processing in general and text summarization in particular. Extractive summarization has been the focus of many studies where important sentences are marked from the original text. Then, these important sentences are put together as summary by keeping the sentences unchanged. Extractive summaries may have problems such as lack of coherence, but usually, they are significantly robust and more straightforward to implement. That is why most existing summarization systems are extraction-based. A method inherited from Edmundson (1969) is reported in (Kupiec, Pedersen, & Chen, 1995) to produce a text summary using Naive Bayes classifier trained on a dataset of 188 documents taken from 21 publications in the scientific/technical domain. A similar study is reported by Aone, Okurowski, & Gorlinsky (1998). The study describes a text summarization system called DimSum using a Naive Bayes classifier but with more features. Another study (Neto, Freitas, & Kaestner, 2002) proposed a trainable summarizer that uses statistical and linguistic features. Two classifiers, the C4.5 decision tree and Naive Bayes, are utilized on a dataset of 200 documents. The results show that the Naive Bayes classifier performed well for all selected compression ratios.

The work (Chuang & Yang, 2000) proposed an approach to automatic text summarization based on extracting sentence segments.

[1] http://duc.nist.gov/

[2] http://www.nist.gov/tac/2011/Summarization/

[3] For instance, Ethnologue listed 447+74=521 living languages in India and Pakistan respectively.

[4] WorldBank statistics: https://data.worldbank.org/indicator/SP.POP.TOTL?locations=8S ; last visited: No. 2021.

[5] CORPURES download link: https://github.com/humsha/CORPURES

Sentences of a document are segmented on special markers using rhetorical structure theory. In addition to the standard features, special features from the rhetorical structure theory are added, such as antithesis, contrast, reason, and cause. Three classifiers (C4.5, Naive Bayes, and Interpattern distance based constructive Neural Network – DistAI) were used to perform experiments on a U.S patent dataset. Moreover, the reference summaries in the dataset were generated manually with the consensus of three people.

The work (Kaikhah, 2004) is reported for the summarization of news articles by using Neural Networks. Every sentence of the source document is represented as a vector of seven features. A feature fusion phase is introduced to eliminate the uncommon features, generalizing the effects of features.

The work (Kianmehr et al., 2009) compares the performance of two classifiers for text summarization: Support Vector Machines (SVM) and Neural Networks. Experiments show that the accuracy of 65% is obtained when three features are used. This accuracy increases to 77% when features are increased from three to six. It was also found that Neural Network is slower than SVM for larger datasets. Our work is similar to the studies mentioned above, as we also employ traditional machine learning classifiers.

Recent studies are focussing on summarization task through Deep Learning. The two studies (Nallapati, Zhai, & Zhou, 2016a; Nallapati, Zhou, & Ma, 2016b) propose novel Recurrent Neural Network (RNN) based architectures for extractive summarization of documents. It is demonstrated that the study (Nallapati et al., 2016b) outperforms state-of-the-art supervised models on two different corpora (CNN/Daily Mail corpus (Hermann et al., 2015)[6] and DUC 2002 single-document summarization dataset). The study (Narayan, Papasarantopoulos, Lapata, & Cohen, 2017) proposes to exploit *side information* in the context of single-document extractive summarization. The *side information* can be the title and image captions which are often available for newswire articles. The summarization model is evaluated on CNN/Daily Mail corpus (Hermann et al., 2015). Moreover, it is shown that it consistently outperforms in terms of both informativeness and fluency. Collins, Augenstein, & Riedel (2017) released a dataset of 10k documents for summarization of computer science publications, and claimed to be used for both abstractive and extractive summarization. It exploits the fact that scholarly publications on the ScienceDirect website contain highlighted statements (i.e., author-provided summaries). The study benchmark several Neural as well as traditional summarization methods. It is demonstrated that their best-performing model outperforms several well-established baseline methods on extractive summarization task. A bottleneck in using deep learning techniques is the availability of a large enough corpus. Unfortunately, the lack of such a corpus for Urdu limits us from using deep learning techniques presently.

*Some studies for languages other than English using supervised learning are as follows* Kutlu, Cigir, & Cicekli (2010) proposed a Turkish text summarizer. It uses surface-level features such as term frequency, key phrase (KP), centrality, title similarity and sentence position. A dataset of 120 news articles is prepared, along with their summaries generated by the human experts. A second dataset consisting of 100 Turkish journal articles and their summaries is also used in this study. Najibullah (2015) presents an Indonesian text summarization system. A computationally inexpensive single document summarizer is reported for Arabic text that is based on keyphrase-base extractive summarization (El-Shishtawy & El-Ghannam, 2012) and a recent work on Arabic text summarization is (Qaroush, Abu Farha, Ghanem, Washha, & Maali, 2019) that utilizes a combination of statistical and semantic features. A prototype FarsiSum based on SweSum is a summarizer for Persian (Hassel & Mazdak, 2004).

*Finally, the following studies are more related to our work* Verma, Pal, & Om (2019) reports a comparative analysis on Hindi and English text summarization. The study relies on existing datasets for Hindi and English. The analysis is performed using thirteen different summarization techniques on Hindi and English newswire datasets. Evaluation is performed using metrics such as precision, recall, $F_1$, cohesion, non-redundancy, readability, and significance. Though Urdu and Hindi share the everyday language, both deviate from each other significantly in script[7], vocabulary[8] and formal writing. So the analysis performed for Hindi may not be considered for Urdu.

Humayoun & Yu (2016) reports a comparative analysis of Urdu text summarization. The reported benchmark experiments analyze the effect of preprocessing such as stopword removal, lemmatization, and stemming using state-of-the-art graph based algorithms for extractive summarization. The algorithms are LexRank, and TextRank. The study relies on an existing benchmark dataset of Urdu *abstracts* (Humayoun, Nawab, Uzair, Aslam, & Farzand, 2016). Urdu faces word segmentation problem as space is not always a reliable marker of a word ending. The corpus (Humayoun et al., 2016) provides two versions of the same document collection: properly segmented and space segmented. Humayoun & Yu (2016) analyzes the effect of preprocessing settings on both versions. However, it is reported that the proper segmentation of words does not add any significant performance for the extractive text summarization due to the low coverage and/or high error rate of existing tools for stemming, morphological analysis and POS tagging. The results are reported as an average of four evaluation measures ROUGE-N (with $n = 1, 2, 3$) and ROUGE-L, and $F_1$ scores are reported. The best $F_1$ scores are reported when stemming is performed and stopwords are removed together. The highest score for Properly segmented dataset is 0.497 (Lead based baseline: 0.492) and 0.4925 (Lead based baseline: 0.487) for space segmented dataset. It can be seen that the results are reported using a strict measure, so even smaller changes in the scores matter a lot. In our work, we try to incorporate the findings learned from this study. However, this study evaluates the effect of preprocessing settings on graph-based algorithms, whereas we apply supervised learning classifiers.

The work of Burney, Sami, Mahmood, Abbas, & Rizwan (2012) is one of the pioneering efforts. It proposes a summarizer using an unsupervised method, namely sentence weight algorithm. The outcome is an add-on "Auto Summarizer for Urdu language" for Microsoft Word. This work lacks a proper evaluation of the quality of generated summaries. A manual evaluation with five evaluators is performed on 20 documents (news and articles). However, the score for only one summary is reported. The similarity score of a sentence match between generated summary and reference summary is 64%.

Similar to the work (Burney et al., 2012), Muhammad, Jazeb, Martinez-Enriquez, & Sikander (2018) also generates summary using sentence weight algorithm, but results are validated on the Urdu Sumary Corpus (Humayoun et al., 2016). The average $F_1$ score using ROUGE-N (with $n = 1$) measure is 0.5941 and using ROUGE-N (with $n = 2$) is 0.4075, when stopwords are removed. Another similar study is (Bhatti & Aslam, 2019), though its contribution is unclear. These three studies have a limited scope as compared to our work.

A recent study (Farooq, Batool, & Noreen, 2021) reports analysis based on eight unsupervised extractive summarization algorithms[9] This study also relies on Urdu *abstracts* (Humayoun et al., 2016) (Only 14 articles are used in the evaluation). Evaluation is performed using the $F_1$ ROUGE-2 scores. The highest score achieved is 0.71 on preprocessed input. In preprocessing phase, stopwords are removed, and stemming is

[6] Nallapati et al. (2016a) uses CNN/Daily Mail corpus only.

[7] Urdu uses a Perso-Arabic script, but Hindi uses Devanagari script.

[8] Urdu vocabulary is heavily influenced by Persian, and somewhat influenced by Arabic (Humayoun, Hammarström, & Ranta, 2007b). In contrast, Hindi vocabulary is heavily influenced by Sanskrit.

[9] These summarizers are: Reduction Summarizer, Luhn Summarizer, Kullback-Lieber sum algorithm, Sum Basics Summarizer, Edmundson Summarizer, TextRank, LSA, and LexRank.

applied. However, the details such as the source of the stopword list and the stemmer are not reported.

Nawaz et al. (2020) produces Urdu extracts using two basic approaches namely sentence weight method (Burney et al., 2012) and weighted term frequency method (Rakesh, Sahoo, Sahoo, & Swain, 2012). Three algorithms are used as baseline, namely, the *de facto* Vector Space Model (Mohd, Jan, & Shah, 2020), TextRank (Mihalcea & Tarau, 2004), and the word embedding based technique (Noor, Bakhtyar, & Baber, 2019). This study also relies on Urdu *abstracts* provided by Humayoun et al. (2016). However, in addition to *abstracts*, the corresponding extractive summaries are generated for Urdu Summary Corpus in this study. The extractive summaries range from 33% to 40% of the given articles. Three experts created three summary candidates for an article. All those sentences are considered as summaries where at least two of the experts agreed. No further details are reported for the creation of reference extracts. Furthermore, the inter-rater agreement between the reference extracts is also not reported. The reference Urdu Summary Corpus (USC) *extracts* developed by this study are available on request. The results are reported for both: abstracts and extracts. The $F_1$ score with ROUGE-2 for the sentence weight method on Urdu Summary Abstracts and extracts is reported to be 0.21 and 0.53 respectively. Whereas, the $F_1$ score with ROUGE-2 for the weighted term frequency method on Urdu Summary Abstracts and extracts is reported to be 0.26 and 0.65 respectively.

In contrast to the studies (Farooq et al., 2021; Humayoun & Yu, 2016; Nawaz et al., 2020; Verma et al., 2019), we have developed a benchmark dataset of Urdu *extracts* from scratch consisting of 151 news articles. The compression rate of our extracts is strictly 40%. We have given a detailed account of the reference summaries creation and the inter-rator agreement among the summaries. In addition, we also evaluate the effect of detailed preprocessing settings on summarization task, and the range and depth of preprocessing settings in our work surpasses the studies of Humayoun & Yu (2016), Nawaz et al. (2020), and Farooq et al. (2021). Furthermore, the related studies mainly evaluate the unsupervised methods, whereas we apply supervised learning classifiers. In addition, we also analyze the role of each feature in the feature vector in detail. Preparing such a feature vector is a preliminary step for the training of a supervised learning algorithm. We also evaluate the effect of fixed-length token segmentation as words. See Section 1.1 for details. In terms of results, we achieved the highest $F_1$ score for the experimental setting E.48 (fixed-length word segmentation of length 6) for Logistic Regression (Sentence Match $F_1$: 0.5702, and ROUGE-2 $F_1$: 0.644). When words are segmented on space, we achieved the highest $F_1$ score for experimental setting E.04 (no stopword removal but ultra stemming applied $L = 1$) for the Logistic Regression algorithm (Sentence Match $F_1$: 0.5612, and ROUGE-2 $F_1$: 0.6364).

### 2.2. Benchmark corpus for text summarization

The creation of language resources is a laborious yet essential task. Language resources play a significant role in pushing the state of the art for natural language processing and its applications. For instance, English NLP has been improved significantly due to the language resources made available with great efforts over the last few decades. A non exhaustive list of such significant resources are: English Penn Treebank (Marcus, Marcinkiewicz, & Santorini, 1993), WordNet (Miller, 1995), PropBank (Palmer, Gildea, & Kingsbury, 2005) and FrameNet (Fillmore, 1982). In the context of automatic text summarization, the Document Understanding Conferences (DUC, from 2001 to 2006) was the main forum releasing the datasets. This forum allowed researchers to compare methods and results on standard test sets. Numerous summarization tasks were covered over the years. The manually generated extracts have been presented in DUC 2001 and DUC 2002.

#### 2.2.1. DUC (Document understanding conference) datasets

The DUC 2001 covers both single and multi-document summarization of newswire genre and provides data sets of 60 document sets with 10 documents in each set (Harman & Over, 2004). It was required to generate a summary of 100 words for each document automatically for an intrinsic evaluation. In addition, four generic summaries of the entire set were also required with a fixed target length of approximately 50, 100, 200, and 400 words. For the creation of reference summaries, each contributer selected six document sets. They produced a 100 word manual abstract for each document to be used in single document summarization task. Then, they produced manual abstracts for the entire document set at the word lengths of 50, 100, 200, and 400 for the multi-document task.

The DUC 2002 followed the footsteps of DUC2001 and provided: (*i*) Abstracts of single documents and document sets of fixed lengths of 10, 50, 100, and 200 words, and (*ii*) Extracts of document sets of fixed target lengths of 200 and 400 words. The corpus size is 60 document sets with roughly 10 documents in each set.

The task in DUC 2003 was changed a bit and it provided: (*i*) Abstracts of single documents with very short (10-word) summaries, and (*ii*) Multi-document summaries focused by TDT (topic detection and tracking technology) event topics, viewpoints, question topics of fixed target length of 100 words. The corpus size for single document summarization is 60 document sets with 10 documents in each set and for muli-document summarization track is 30 document sets with 25 documents in each set.

The DUC 2004 provided: (*i*) Abstracts of single documents with very short (10-word) summaries, and (*ii*) Multi-document summaries focused by event and 'who is' question topics of fixed target length of 100 words. The corpus size is 100 document sets with 10 documents in each set.

DUC 2005–2006 provided a complex multi-document summarization task to produce a question-focused abstractive summary. The corpus size for DUC 2005 is 50 document sets with 32 documents in each set, and for DUC 2006 is 50 document sets with 25 documents in each set.

It is argued by studies (Chopra, Auli, & Rush, 2016; Grusky, Naaman, & Artzi, 2018; Nallapati, Zhai, & Zhou, 2017) that the DUC datasets are small and cannot be used for training models with a large number of parameters and therefore should be used in conjunction with other datasets.

#### 2.2.2. TAC (text analysis conference) datasets

In 2008, DUC became part of the Text Analysis Conference (TAC) under the summarization track, which has happened five times (in 2008, 2009, 2010, 2011, and 2014). TAC 2008 (Dang & Owczarzak, 2009) covered multi-document summarization by offering two tasks: The first task is to write a short (100-word) *initial* summary (called Summary A) of a set of 10 newswire articles about a particular topic. The second task is to write an *updated* summary (called Summary B) of a subsequent set of 10 newswire articles for the same topic. It is assumed that the user has already read the first ten articles and, therefore, should be given new information in the generated summary. English newswire articles of 907K were collected, and 48 topics were developed by 8 NIST assessors. A set of 20 documents for each topic were also selected. The documents were ordered chronologically and divided into two sets of 10 documents each, such that Set B followed Set A in a temporal order.

TAC 2009 covered the same first task as TAC 2008. The second task was on evaluation (Automatically Evaluating Summaries Of Peers – AESOP). TAC 2010 introduced 'guided summarization' requiring a deeper linguistic analysis. The task is to write a 100-word summary of a set of 10 newswire articles for a given topic from a predefined category. A list of aspects for each category is already provided, and a summary must include all aspects found for its category. The second task is the same as TAC 2009, i.e., AESOP. TAC 2011 had three tasks. The first two tasks are the same as of TAC 2010, whereas the third task is Multiling

Pilot. It is a multi-document summarization task to produce a summary of 250 words. An important aspect of this task is that the original corpus was translated into six languages[10] to give a multilingual test collection. It was expected that the output summary should be of the same language as its source documents. The dataset has 100 documents assigned to ten reference sets, one set discussing the same topic. Finally, TAC 2014 introduced the biomedical summarization track.

#### 2.2.3. Some very large datasets

The New York Times Annotated Corpus (NYTAC) (Sandhaus, 2008) is an extensive collection of newspaper articles. Out of approximately 1.8 million news articles, 650,000 articles have been summarized by library scientists (single-document *abstracts*). Large Scale Chinese Short Text Summarization Dataset (LCSTS) (Hu, Chen, & Zhu, 2015) contains over 2 million short texts with short summaries (*abstracts*) given by the writer of each text. It is constructed from the Chinese microblogging website SinaWeibo.

A recent work by the researchers at DeepMind (Hermann et al., 2015) released large corpora for the task of reading comprehension and Q&A. It is based on the fact that news articles in two newspapers CNN and Daily Mail, contain important points as bullets in the same article. So the original articles can be used as documents and the bullet points in the same document as a reference summary (*extract*). Two large datasets containing a vast collection of documents (90k and 197k each) are released. A thorough examination of these datasets on reading comprehension task is reported here (Chen, Bolton, & Manning, 2016). Unfortunately, we do not see any Urdu newspaper that follows this format.

Newsroom (Grusky et al., 2018), a summarization dataset comprises 1.3 million articles and summaries written by authors and editors in the newsrooms of 38 major news publications. Newsroom editors and journalists wrote these extracted summaries to appear in social media distribution and search results. If multiple summaries were available for an article, they used the first available. Those articles were excluded that have no summary text. They also removed article-summary pairs with a high amount of precisely overlapping text to remove the rule-based generated summaries fully copied from the article (e.g., the first paragraph).

In the context of Urdu corpora development, the following studies are related: For the task of Word Sense Disambiguation (WSD), we found two studies. The study (Saeed, Nawab, Stevenson, & Rayson, 2018) provides a small corpus containing 50 target words (30 nouns, 11 adjectives, and 9 verbs). The work (Saeed, Nawab, Stevenson, & Rayson, 2019) provides a corpus containing 5K words of Urdu running text in which all ambiguous words (856 instances) are manually tagged with senses. COUNTER (Sharjeel, Nawab, & Rayson, 2016) is a corpus of Urdu news text reuse containing 1200 documents. Each document is manually annotated with three levels of reuse: wholly derived, partially derived, and non derived. The corpus can used for the task of plagiarism detection. CLE Urdu Digest Corpus (Urooj, Hussain, Adeeba, Jabeen, & Perveen, 2012) is a large Urdu corpus collected from an Urdu magazine Urdu Digest published ranging from 2003 to 2011. Finally, a corpus related to this work is discussed below in a separate subsection.

#### 2.2.4. Urdu summary corpus (USC)

Urdu Summary Corpus (USC) (Humayoun et al., 2016) is a benchmark corpus for Urdu summaries (*abstracts*). It provides 50 Urdu articles and their abstractive summaries. The articles are collected from various online sources, mainly news websites and blogs. With the corpus, a software package is also provided consisting: a script normalizing utility, a POS tagger, a table-lookup based morphological analyzer & lemmatizer, and a stemmer. This corpus has been used in an extensive study (Humayoun & Yu, 2016) analyzing in detail the effect of basic preprocessing settings for Urdu using graph-based methods. This corpus is also used in the following studies: (Bhatti & Aslam, 2019; Farooq et al., 2021; Muhammad et al., 2018; Nawaz et al., 2020). All in all, it is an excellent resource for the resource-poor Urdu language.

**Table 1**
Statistics for urdu summary corpus.

| Compression | Summaries |
|---|---|
| >= 60 | 2 |
| 50 to 59.9 | 7 |
| 40 to 49.9 | 19 |
| 30 to 39.9 | 20 |
| 20 to 29.9 | 2 |

It is worth noting that the compression rate of the reference summaries in the Urdu Summary Corpus is of varying lengths (see Table 1 for details). The authors asked the writers to produce good summaries, and it should not matter "if a summary is large, medium or small in size" (Humayoun et al., 2016). It might be motivated by the fact that summary authors tend to differ on the suitable length of an sammary, specifically more for abstracts (Jing, Barzilay, McKeown, & Elhadad, 2000). The qualitative analysis of Urdu Summary Corpus (USC) is shown in Fig. 1.

We did not use USC because of the following reasons. First, the resource's size is small and may not be enough to effectively train the supervised learning classifiers. Second, a supervised learning algorithm expects labels (the right answers). In the case of *extracts*, the labels are readily available in the dataset. However, in the case of *abstracts*, additional work is required to define a label because a summary sentence may not entirely be present in the document. Third, we wanted to contribute another resource for Urdu NLP. At the time of our summary corpus development, no extractive summary corpus existed. It motivated us to take up this task.

## 3. Benchmark corpus for urdu extractive summaries

Many resources and natural language processing tools are available for different English, French, and German languages. Many languages suffer from a lack of resources; Urdu is one of them. We have created a resource for Urdu single document summarization along with their reference summaries (CORPURES). The Urdu language experts produce the reference extractive summaries. It is a general practice to produce language resources manually by the experts, having a high quality but costly to produce. In recent times, crowdsourcing has emerged as an alternative that requires less cost and resources are of reasonable quality (El-Haj, Kruschwitz, & Fox, 2015). To reduce the laborious work without compromising on quality, we tried to use a simpler version of crowdsourcing in which the workers were sitting in the same classroom. These workers were: 1) volunteer students[11] of Urdu literature programme with adequate knowledge of summarization task, and 2) university teachers[12] teaching Urdu literature to the students mentioned above. The task was conducted in two sessions (1.5 h per session) with around 50 volunteers in each session. Another session of one hour with faculty members (around 7) was also conducted. It resulted in four hours of crowdsourcing work performed by ~107 contributors.

### 3.1. The document collection

The Corpus for Urdu Extractive summaries contains 161 Urdu news

[10] Arabic, Hindi, French, Czech, Greek, and Hebrew

[11] We did not have any funds for this project. The authors conveyed the importance and a possible impact of this work. Those who got convinced participated in this work free of cost. Though the authors distributed simple refereshments in these sessions.

[12] Urdu Literature department, University of Education Lahore, Pakistan. http://ue.edu.pk

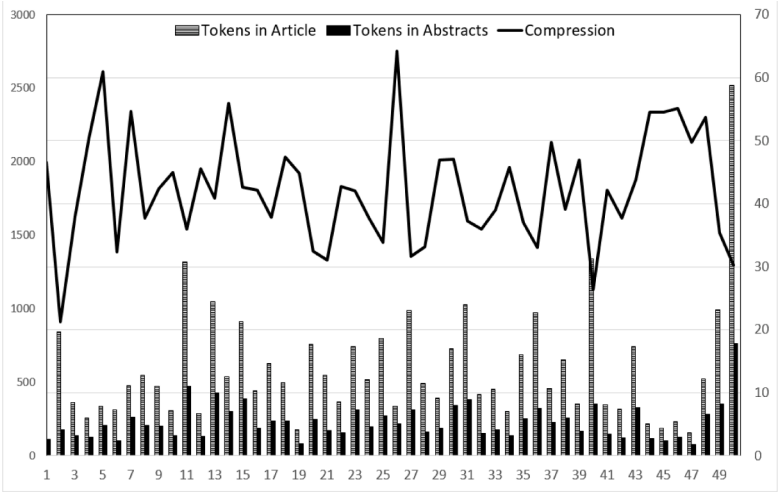
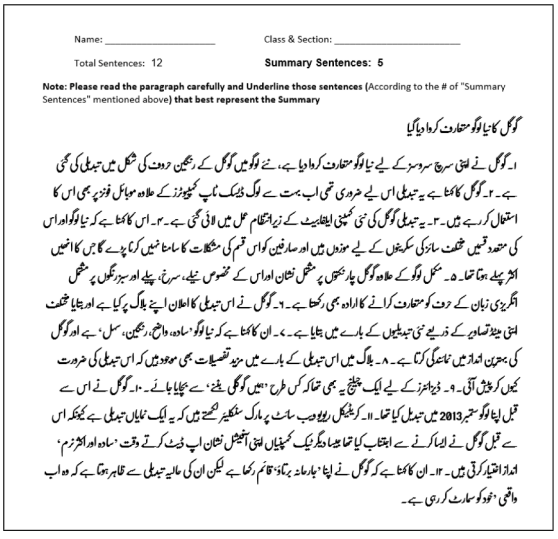



**Fig. 1.** Qualitative analysis of Urdu Summary Corpus (USC). The bar with horizontal pattern indicates the word count in each article, the bars with solid black indicates the word count in the extracts, and the black line shows the ratio of summary with given article in percentage ranging from 21.3 to 64.2 with an average of 42.2 compression rate in percentage.

**Table 2**
Statistics for CORPURES.

| | |
|---|---|
| Documents | 161 |
| Sentences | 2649 |
| Words | 72,537 |
| Summaries per document | 04 (03 by contributers, 01 by majority voting) |



Name: _____ Class & Section: _____

Total Sentences: 12 **Summary Sentences: 5**

**Note: Please read the paragraph carefully and Underline those sentences** (According to the # of "Summary Sentences" mentioned above) **that best represent the Summary**

**گوگل کا نیا لوگو متعارف کروا دیا گیا**

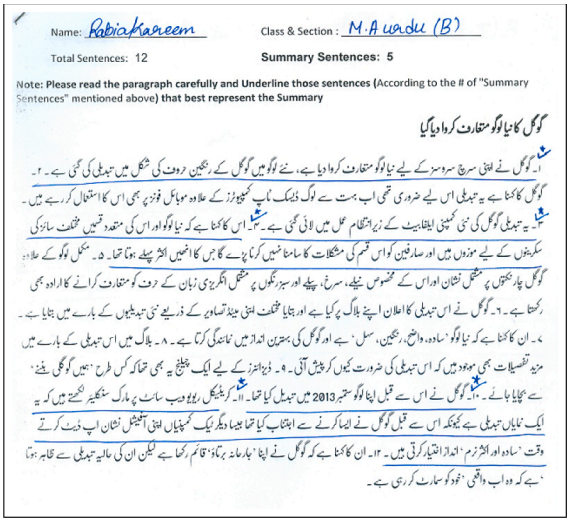
۱۔ گوگل نے اپنی سرچ سروسز کے لیے نیا لوگو متعارف کروا دیا ہے، نئے لوگو میں گوگل کے رنگین حروف کی شکل میں تبدیلی کی گئی ہے۔ ۲۔ گوگل کا کہنا ہے یہ تبدیلی اس لیے ضروری تھی اب بہت سے لوگ ڈیسک ٹاپ کمپیوٹرز کے علاوہ موبائل فونز پر بھی اس کا استعمال کر رہے ہیں۔ ۳۔ یہ تبدیلی گوگل کی نئی کمپنی ایلفابیٹ کے زیر اہتمام عمل میں لائی گئی ہے۔ ۴۔ اس کا کہنا ہے کہ نیا لوگو اور اس کی متعدد قسمیں مختلف سائز کی سکرینوں کے لیے موزوں ہیں اور صارفین کو اس قسم کی مشکلات کا سامنا نہیں کرنا پڑے گا جس کا انھیں اکثر پہلے ہوتا تھا۔ ۵۔ مکمل لوگو کے علاوہ گوگل چار نکتوں پر مشتمل نشان اور اس کے مخصوص نیلے، سرخ، پیلے اور سبز رنگوں پر مشتمل انگریزی زبان کے حرف کو متعارف کرانے کا ارادہ بھی رکھتا ہے۔ ۶۔ گوگل نے اس تبدیلی کا اعلان اپنے بلاگ پر کیا ہے اور بتایا مختلف اپنی میڈ تصاویر کے ذریعے نئی تبدیلیوں کے بارے میں بتایا ہے۔ ۷۔ ان کا کہنا ہے کہ نیا لوگو 'سادہ، واضح، رنگین، سہل' ہے اور گوگل کی بہترین انداز میں نمائندگی کرتا ہے۔ ۸۔ بلاگ میں اس تبدیلی کے بارے میں مزید تفصیلات بھی موجود ہیں کہ اس تبدیلی کی ضرورت کیوں کر پیش آئی۔ ۹۔ ڈیزائنرز کے لیے ایک چیلنج یہ بھی تھا کہ کس طرح 'ہیں گوگل بیٹے' سے بچایا جائے۔ ۱۰۔ گوگل نے اس سے قبل اپنا لوگو ستمبر 2013 میں تبدیل کیا تھا۔ ۱۱۔ کرسٹیکل ریویو ویب سائٹ پر مارک سنکلیئر لکھتے ہیں کہ یہ ایک خیالی تبدیلی ہے کیونکہ اس سے قبل گوگل نے ایسا کرنے سے اجتناب کیا تھا جبکہ دیگر ٹیک کمپنیاں اپنی آفیشل نشان اپ ڈیٹ کرتے وقت 'سادہ اور اثر انداز' انداز اختیار کرتی ہیں۔ ۱۲۔ ان کا کہنا ہے کہ گوگل نے اپنا 'بارہ مہ' برقرار رکھا ہے لیکن ان کی مالیہ تبدیلی سے ظاہر ہوتا ہے کہ وہ اب واقعی 'خود کو سمارٹ کر رہی ہے۔

**Fig. 2.** Snapshot of an article from CORPURES.

articles along with their manually created summaries. These summaries are extraction-based – the language experts select significant sentences from the given document to form reference summaries. The news articles were mainly collected from the BBC Urdu website. The news articles cover a wide range of topics from different areas such as current affairs, entertainment, health, politics, sports, science & technology, and are written by the native speakers of Urdu. The general statistics are shown in Table 2.

### 3.2. The reference summaries

The contributors were asked to read the text and select sentences for the summary. The authors went to the classrooms and distributed the



Name: Rabia Kareem Class & Section: M.A urdu (B)

Total Sentences: 12 **Summary Sentences: 5**

**Note: Please read the paragraph carefully and Underline those sentences** (According to the # of "Summary Sentences" mentioned above) **that best represent the Summary**

**گوگل کا نیا لوگو متعارف کروا دیا گیا**

۱۔ گوگل نے اپنی سرچ سروسز کے لیے نیا لوگو متعارف کروا دیا ہے، نئے لوگو میں گوگل کے رنگین حروف کی شکل میں تبدیلی کی گئی ہے۔ ۲۔ گوگل کا کہنا ہے یہ تبدیلی اس لیے ضروری تھی اب بہت سے لوگ ڈیسک ٹاپ کمپیوٹرز کے علاوہ موبائل فونز پر بھی اس کا استعمال کر رہے ہیں۔ ۳۔ یہ تبدیلی گوگل کی نئی کمپنی ایلفابیٹ کے زیر اہتمام عمل میں لائی گئی ہے۔ ۴۔ اس کا کہنا ہے کہ نیا لوگو اور اس کی متعدد قسمیں مختلف سائز کی سکرینوں کے لیے موزوں ہیں اور صارفین کو اس قسم کی مشکلات کا سامنا نہیں کرنا پڑے گا جس کا انھیں اکثر پہلے ہوتا تھا۔ ۵۔ مکمل لوگو کے علاوہ گوگل چار نکتوں پر مشتمل نشان اور اس کے مخصوص نیلے، سرخ، پیلے اور سبز رنگوں پر مشتمل انگریزی زبان کے حرف کو متعارف کرانے کا ارادہ بھی رکھتا ہے۔ ۶۔ گوگل نے اس تبدیلی کا اعلان اپنے بلاگ پر کیا ہے اور بتایا مختلف اپنی میڈ تصاویر کے ذریعے نئی تبدیلیوں کے بارے میں بتایا ہے۔ ۷۔ ان کا کہنا ہے کہ نیا لوگو 'سادہ، واضح، رنگین، سہل' ہے اور گوگل کی بہترین انداز میں نمائندگی کرتا ہے۔ ۸۔ بلاگ میں اس تبدیلی کے بارے میں مزید تفصیلات بھی موجود ہیں کہ اس تبدیلی کی ضرورت کیوں کر پیش آئی۔ ۹۔ ڈیزائنرز کے لیے ایک چیلنج یہ بھی تھا کہ کس طرح 'ہیں گوگل بیٹے' سے بچایا جائے۔ ۱۰۔ گوگل نے اس سے قبل اپنا لوگو ستمبر 2013 میں تبدیل کیا تھا۔ ۱۱۔ کرسٹیکل ریویو ویب سائٹ پر مارک سنکلیئر لکھتے ہیں کہ یہ ایک خیالی تبدیلی ہے کیونکہ اس سے قبل گوگل نے ایسا کرنے سے اجتناب کیا تھا جبکہ دیگر ٹیک کمپنیاں اپنی آفیشل نشان اپ ڈیٹ کرتے وقت 'سادہ اور اثر انداز' انداز اختیار کرتی ہیں۔ ۱۲۔ ان کا کہنا ہے کہ گوگل نے اپنا 'بارہ مہ' برقرار رکھا ہے لیکن ان کی مالیہ تبدیلی سے ظاہر ہوتا ہے کہ وہ اب واقعی 'خود کو سمارٹ کر رہی ہے۔

**Fig. 3.** Snapshot of a tagged summary for the article in Fig. 2, The underlined sentences forms the extract.

printed articles among students. At the start of each session, the authors explained 'what is an extractive summary' by giving examples on multimedia display. The four dimensions of a good reference summary (Coherence, Consistency, Fluency, and, Relevance) mentioned by Kryscinski, McCann, Xiong, & Socher (2020) were quickly explained. The authors stressed selecting those sentences that pass these guidelines (especially the Coherence and Relevance, as these two seem more appropriate for the extracts). The number of sentences required for a summary was mentioned on each document. For this corpus, a compression rate of 40% is decided as the summary size. We picked this compression rate because: (1) It is also suggested by others, for instance, (Chuang & Yang, 2000; Kupiec et al., 1995), (2) By using the contributors' voting, the compression rate can be changed to the other values such as, 30%, 20% or 10%. However, this will require an extra contributor (veto contributor) to make a final decision if a tie occurs. A Snapshot of a sample document, its summary and English translation are shown in Figs. 2–4.

We cannot have a single reference summary for a given document due to unavoidable variation in human judgment (Harman & Over, 2004; Rath, Resnick, & Savage, 1961). To reduce the impact of

**Google's new logo has been introduced.**

1. Google has introduced a new logo for its search services, the shape of the Google's colorful characters in the new logo has been changed.
2. Google says that this change was important because many people are using it on mobile phones besides desktop computers.
3. This change has been brought under the control of Google's new company alphabet.
4. It says that the new logo and its many types are suitable for different sizes of screens and users will not have to face the difficulties they often had.
5. In addition to the full logo, Google intends to introduce the English language vocabulary containing four distinctive symbols and distinctive blue, red, yellow, and green colors.
6. Google has announced this change on its blog and has told about new changes through various animated images.
7. They say the new logo is 'simple, clear, colorful, comfortable' and represents Google in the best way.
8. There are more details about the change in the blog that why this change is needed.
9. There was another challenge for designers regarding how to prevent us from 'becoming Googly'.
10. Google has changed its logo earlier in September 2013.
11. At the Critical Review website, Mark Sinclair writes that it is a significant change because before Google had avoided doing such things, as other tech companies take the 'simple and often soft' style while updating their official marks.
12. They say that Google has maintained its "aggressive behavior" but its recent change shows that it really is becoming "smart".

**Fig. 4.** English translation of the article in Fig. 2.

**Table 3**
Snapshot of summary selection table for the article in Fig. 3.

| CurrentAfr01, select sentences = 05 | | | | | |
|---|---|---|---|---|---|
| Sentence no. | C1 | C2 | C3 | Weight | Selection |
| **1** | 0 | 0 | 0 | **0** | **0** |
| **2** | 1 | 0 | 1 | **2** | **1** |
| **3** | 0 | 1 | 1 | **2** | **1** |
| **4** | 1 | 0 | 0 | **1** | **0** |
| **5** | 0 | 0 | 1 | **1** | **0** |
| **6** | 0 | 0 | 0 | **0** | **0** |
| **7** | 0 | 0 | 0 | **0** | **0** |
| **8** | 1 | 1 | 1 | **3** | **1** |
| **9** | 1 | 1 | 0 | **2** | **1** |
| **10** | 0 | 0 | 0 | **0** | **0** |
| **11** | 0 | 1 | 1 | **2** | **0** |
| **12** | 1 | 1 | 0 | **2** | **1** |

**Table 4**
Details of the dataset.

| Category | Documents | Sentences | Words | Summary sentences (40%) |
|---|---|---|---|---|
| Current Affairs | 14 | 225 | 5396 | 90 |
| Entertainment | 28 | 443 | 12,153 | 179 |
| International | 52 | 775 | 21,953 | 308 |
| Political | 19 | 241 | 7198 | 98 |
| Science & Tech | 19 | 393 | 8915 | 157 |
| Sports | 22 | 406 | 12,319 | 161 |
| Health | 07 | 166 | 4593 | 67 |
| Total | 161 | 2649 | 72,587 | 1060 |

individual bias in the summaries, the reference summaries for each document must be more than one, each created by a different human contributor (El-Haj et al., 2015). Because of that, we produced three different versions of a summary. In this way, each summary of a given document was generated by three different contributors. There are 483 summaries of 161 documents (three reference summaries per document). The details of the dataset is given in Table 4.

From three reference summaries per document, we make a final reference summary produced by majority voting. Each sentence in the three summaries (for each document) is given a score of "1" if the human contributor selected it to include in the final reference summary, or "0" otherwise. By adding the scores of each sentence, "weight" was calculated. The sentence weight helps to select the candidate sentences for the final reference summary. Table 3 shows a snapshot of the summary selection.

Only those sentences of a document were considered in the final reference summary that are selected by two or more than two contributors. If there was a tie or ambiguity in selecting a sentence, the authors acted as the fourth contributor and took the final decision carefully after reading the article (Jing et al., 2000; Parthasarathy & Hasan, 2015). Note that we also kept the three summaries per document (in addition to the final reference summary) in our dataset that can be used if needed.

**Table 5**
Agreement between three summaries for each document.

| Contributors' selection | Sentence percentage |
|---|---|
| One contributor picked a sentence | 31.86% |
| Two contributors picked a sentence | 28.01% |
| *a*. No contributor picked a sentence | 28.39% |
| *b*. All the contributors (three) picked a sentence | 11.74% |
| Complete agreement ($a + b$) | 40.13% |
| **Method** | Agreement between three summaries per document |
| Inter-rater percentage | 60.07% |
| Fleiss' Kappa score | 0.17 (Slight) |

The number of sentences in the document collection is between 7 to 70, and the average length of a document is 39. The number of sentences in the reference summaries is between 3 to 28, with an average of 16 sentences per document.

### 3.3. Inter-rater agreement

The agreement between the three summaries of each article has been evaluated and shown in Table 5. The inter-rater agreement between three summaries per document is 60.07%, whereas, Fleiss' Kappa score is 0.17 indicating 'slight' agreement. Note that calculating how many sentences are common in two reference summaries is a strict measure for extractive summaries. Therefore, it is possible that for a reference extract, two contributor may select two different senteces which are quite close to each other in meaning. Given that, human judgment tends to vary on a reference summary (Harman & Over, 2004; Mitra, Singhal, & Buckley, 1997; Moratanch & Chitrakala, 2017; Verberne, Krahmer, Hendrickx, Wubben, & van den Bosch, 2018), 'slight' agreement seems reasonable. This view is supported by Moratanch & Chitrakala (2017) who claims that the agreement between human summarizers is generally low in a number of studies that they surveyed. For instance Mitra et al. (1997) reports a low level of agreement between the two human subjects (to be more precise 46% overlap on average). For DUC 2001 to 2003, Harman & Over (2004) concluded that "despite large variations in the human-generated model summaries and large variations in human judgments of single-model coverage, the ranking of the systems remained comparatively constant [... ]". The studies (Mitra et al., 1997;

**Table 6**
Effect of Majority Voting. **C1**: Summary by the first contributor, **C2**: Summary by the second contributor, **C3**: Summary by the third contributor, **FS**: Final summary by the authors using majority vote.

| | Cohen's kappa |
|---|---|
| C1 vs. FS | 0.54 (Moderate) |
| C2 vs. FS | 0.6 (Moderate) |
| C3 vs. FS | 0.51 (Moderate) |
| All: C1, C2, C3 and FS | 0.36 (Fair) |

Verberne et al., 2018) argue that a supervised model trained on reference summaries with low inter-rater aggreement can still produce sensible summaries given that the versions of a reference summary are combined into one reference summary Jing et al. (2000), for instance, by majority voting Parthasarathy & Hasan (2015).

It is worth noting that three summaries per document is used by the authors to form the final reference summary using majority vote. Gillick & Liu (2010) argue that the summary produced by the experts is generally better than the crowdsourced reference summary. Therefore, due to the authors better understanding of the Urdu language and the task, we hope that the final summary is better than the three counterparts. In order to see the effect of majority voting, we compute the inter-rater agreement between the final reference summary and the three manually created summaries as shown in Table 6. It can be observed that the agreement between the final summary and the three independent summaries is 'Moderate' on Cohen's Kappa scale. The Fleiss' Kappa between four summaries per document is 'Fair'[13]

## 4. Supervised learning framework for extractive text summarization

In supervised learning, the task of extractive text summarization can be modeled as a binary classification problem (Neto et al., 2002). A supervised learning algorithm known as classifier is trained on a collection of training documents and their reference summaries. If the sentence is present in the reference extractive summary then it may be labeled something like ***Keep***, otherwise ***Reject***. Once training is completed, the classifier can take a document as an input, and as an output, partition document's sentences into summary and non summary sentences with the labels *keep* and *Reject* respectively. It is worth noting that the framework returns a continuous value between 0 to 1, which allowed us to pick up the top-ranked sentences needed for the required compression rate (in the same order as they appear in the article).

An extractive summarization framework consists of five steps: preprocessing (Section 4.1), feature extraction (Section 4.2), training classifiers (Section 4.3), producing machine generated extractive summaries (Section 5), and finally, evaluation on reference summaries (Sections 4.5, 5).

### 4.1. The preprocessing

Preprocessing is the first step in NLP and IR, including summarization. A collection of documents is given as input in preprocessing step, and output is a collection of processed documents where each sentence of a document consists of a set of processed words.

#### 4.1.1. Standard preprocessing settings

The standard preprocessing settings discussed below are applied to all experiments.

[13] We know that the final reference summary is not completely independent of the others. However, as the voting process is more sophisticated than blind voting, we think that it is justified.

**Table 7**
The accuracy of the tagger.

| Part of speech | Coverage | Correctly tagged |
|---|---|---|
| Noun | 27.36% (997 counts) | 84.65% (844 counts) |
| Proper Noun | 8.81% (321 counts) | 74.45% (239 counts) |
| Pronoun | 3.21% (117 counts) | 98.29% (115 counts) |
| Numerical values | 1.4% (47 counts) | 92.16% (47 counts) |

*Diacritic removal* Diacritic marks are not consistently used in Urdu texts both on electronic or printed medium (Hussain & Afzal, 2001). To ensure the consistency of data, removing all the diacritics (i.e. zer, zabar, pesh) is a common practice (Humayoun, 2021; Humayoun et al., 2016; Humayoun & Yu, 2016) followed by Urdu related NLP tools.[14] It means that diacritic removal makes the text more suitable to the existing tools.

*Text normalization* It is a process of transforming multiple canonical representations of a character into a single form. Urdu script inherits some characters from Persian and Arabic scripts. Because of this, characters that visually look similar (or semantically similar) have different Unicode, resulting in orthographic variations. It is necessary to convert all these forms into one standard form (Mukund, Srihari, & Peterson, 2010). We normalize all such variations using a utility (Gulzar, 2007) in Normalization Form C (NFC) as proposed by Unicode Normalization standard (Davis & Whistler, 2014).

*Part of speech tagger* A stand-alone tagger for Urdu with a reported accuracy of 88.74% is (Jawaid, Kamran, & Bojar, 2014), though the tool is not available publicly. However, a large tagged corpus of Urdu is released publicly. We used a large fragment of this tagged corpus to train a model on *Stanford POS tagger*.[15] POS tagging is needed to calculate features 5, 6, and 7 (see Section 4.2.4). *The accuracy of the tagger* We only need to count the occurrences of nouns, proper nouns, pronouns, and numerical values. We randomly selected seven documents from the CORPURES (one document from each category) and calculated the mistakes made by the POS tagger. There are 3644 tokens in total, and 40.78% (1,486 counts) of the tags belong to the needed part of speech. The POS resutls are shown in Table 7.

#### 4.1.2. Word segmentation

Urdu faces word segmentation problem, meaning that space marker is a weak indicator of a word boundary. Urdu faces the problem of space insertion and space omission. It is mainly because Urdu writers tend to simply produce the correct shape of a word (with or without space). Due to the lack of readily available NLP resources, it is a common practice to segment word tokens on space (Bin Zia, Raza, & Athar, 2018; Durrani & Hussain, 2010; Humayoun, Hammarström, & Ranta, 2007b; Humayoun, Nawab, Uzair, Aslam, & Farzand, 2016; Humayoun & Yu, 2016; Mukund, Srihari, & Peterson, 2010). The study (Humayoun & Yu, 2016) concludes with experiments that proper segmentation of words does not add any significant improvement for the extractive text summarization. It might be due to the high error rate and low coverage of existing Urdu tools for the tasks of stemming, morphological analysis, and POS tagging.

There are two Urdu word segmentation studies worth mentioning (Bin Zia et al., 2018; Durrani & Hussain, 2010). The study (Bin Zia et al., 2018) has published their tagged corpus (of 4325 sentences – 111,000

[14] For instance,

حَاصِل

(Hasil: to get) which contains 'zabar' and 'zer' is converted to its non-diacretic form

حاصل

[15] http://nlp.stanford.edu/software/tagger.shtml

tokens approximately) and a Conditional Random Field (CRF) word segmentation utility on Github.[16] It is reported that the trained CRF model achieved $F_1$ score of 0.97 and 0.85 for word and sub-word prediction tasks, respectively. First, the model removes all the spaces, and then, the prediction of word and sub-word is performed character by character. Using the CRF word segmentation utility, we automatically created another version of CORPURES, i.e., *Proper segmented* document collection where tokens are separated as words on actual word boundaries.

*Error rate on CORPURES* The same document set from CORPURES used in POS tagger accuracy is used in this experiment. To find the error rate, we manually tagged them as a reference, for word and sub-word boundaries. There are 3644 tokens in the original 7 documents. It is manually calculated that there are a total 34 space omission and 126 space insertion mistakes. in the original 7 documents, making the total mistakes to be only 4.39%.

However, the CRF model achieved a $F_1$ score of 0.93 and 0.51 for word and sub-word prediction tasks, respectively. The low $F_1$ score in the sub-word prediction task is due to the fundamental design limitation of the word segmentation pipeline of the study (Bin Zia et al., 2018). First, it removes all the spaces from the input and then predicts actual word boundaries, leaving more mistakes as compared to the space segmented text. Another reason for low $F_1$ scores could be the unknown named entities that could be more than usual in the CORPURES document collection because it is collected from the news articles. For the experiments reported in Section 5.1.2, we have created a Properly Segmented version of the CORPURES dataset using this tool.

#### 4.1.3. Stopword removal

The words that are very common and have a little meaning are known as stopwords (Lo, He, & Ounis, 2005). There is no standard stopword list for Urdu. The stoplist we used is provided by Humayoun et al. (2016); Humayoun & Yu (2016), and it contains 500 words where Nouns and Named entities are not included. It is built by calculating the term frequency on a large Urdu corpus – UrMonoCorp (Jawaid et al., 2014).

#### 4.1.4. Lemmatization, stemming, and similarity measures

These preprocessing settings affect three features out of ten: Number of Title words, Sentence cohesion, and TF-ISF (Features: 4, 9, 10; Section 4.2).

*Humayoun's lemmatizer* We used Urdu Morphological Analyzer (Humayoun, Hammarström, & Ranta, 2007b) to convert all inflected surface forms of a word to its lemma or root. This tool covers approximately 5000 words, capable of handling 140,000 word forms.

*Assas-band stemmer* Stemming is also reducing the inflected surface forms of a word back to its stem or root. Assas-band (Akram, Naseer, & Hussain, 2009) is a rule-based Urdu Stemmer. It uses affix based exception lists instead of the conventional lexical methods used for developing stemmers for other languages.

*Fixed length stemming* This radical approach is, in fact, a crude chopping. In general, we keep the first $n$ letters and discard the rest. In the case of length $n = 1$, we keep only the first letter. For example, if we have the following surface forms: car, cars, car's, can, cart, care, …, all of these forms will be replaced by "c". It is called ultra stemming. Torres-Moreno (2012) argues that ultra stemming improves the score of a summarization system by reducing the size of matrix representation. It also reduced the computational processing time required to generate a summary.

*Similarity measures* Similarity measures discussed below are used as an alternate to lemmatization and stemming.

- *Levenshtein distance* measures the minimum edit distance (Levenshtein, 1966) between two strings using three operations: *insert, delete* and *substitute*. Such a similarity measure could be a naive way of identifying different surface forms of a word. For example, it has been applied for stemming (Garabk, 2006; Lyras, Sgarbas, & Fakotakis, 2007) and for preprocessing before sentiment analysis (Elshakankery & Farouk, 2019).
- *Word embeddings* captures the semantic properties of words and is considered to be a breakthrough in recognizing similar words, entities, or concepts (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013). In addition to identifying different surface forms of a word, it can recognize that words such as 'king', 'queen', and 'car', 'automobile' are similar concepts. There are three established techniques for generating word embeddings: Word2vec, Glove, and fastText. We trained a word embedding model for Urdu using Word2vec on a large monolingual Urdu corpus – UrMonoCorp (Jawaid et al., 2014).

#### 4.1.5. Fixed-length word segmentation

Fixed-length word segmentation is a radical approach inspired by the N-Gram model that can be used as an alternative to stemming (Hammarström & Borin, 2011; Mayfield & McNamee, 2003; Xu & Croft, 1998). It can be defined as follows: *a*) take a text as input, *b*) remove all the spaces between words (but keep the sentences separate from each other), and finally, *c*) divide the text into fixed-length segments (keep the leftover chunks at the end of a sentence whose length is less than the fixed-length parameter). We have created five versions of the CORPURES dataset for the experiments reported in Section 5.2, with the word segmentation of fixed-length of 2, 3, 4, 5, and 6. Fixed-length word segmentation is only used in Experiment III 5.2.

#### 4.1.6. Artificial data generation

Synthetic Minority Oversampling Technique (SMOTE) (Chawla, Bowyer, Hall, & Kegelmeyer, 2002) is used to generate the synthetic datasets. Note that there is a minor imbalance in CORPURES dataset as the compression rate for each summary is 40%. Thus, there are 40% *Keep* labels and 60% *Reject* labels. SMOTE over-samples the minority class by generating synthetic examples. The synthetic examples are added "along the line segments joining any/all of the $k$ minority class nearest neighbors" (Chawla et al., 2002). Thus, synthetic examples help a classifier to generalize better with a larger and less specific decision region.

For the experiments reported in Section 5.4, we have created three balanced synthetic datasets from CORPURES. For the first dataset (call it Resampled dataset 1), the minority class (i.e., *Keep* labels) is synthetically added by SMOTE with a 50% increase of synthetic instances. The objective is to observe the effect of a balanced dataset on the extractive summarization task. However, in the second synthetic dataset (call it Resampled dataset 2), instances are added with a 450% increase in both classes to make it larger. The objective is to observe the effect of a large synthetic dataset on the extractive summarization task. In the third dataset (call it Resampled dataset 3) the majority class is reduced by undersampling,[17] making the dataset balanced. Artificial data is generated and used only in Experiment V reported in Section 5.4.

### 4.2. Features extraction

Features extraction is a necessary preprocessing task for supervised learning. High accuracy of summarization can be obtained by increasing the number of features (Kianmehr et al., 2009). Each sentence is represented as a feature vector. As suggested by the literature reported in Section 2.1, we selected ten features as shown below:

[16] https://github.com/harisbinzia/Urdu-Word-Segmentation

[17] Using SpreadSubsample filter of WEKA

**Table 8**
*Experimental Setup I* – $F_1$ results for **space segmented** dataset. The top results are **boldfaced and underlined**; the second best results are only **boldfaced**; the third best are only underlined. Only Sentence Match $F_1$ scores are discussed, whereas ROUGE-2 scores are omitted from the discussion.

| Experiment setup (E.) | Baseline lead | C4.5 | Random forest | Naive bayes | Logistic regression | SVM Poly | SVM Puk | Multilayer perceptron | AVG. |
|---|---|---|---|---|---|---|---|---|---|
| **E.01** no stopword removal and no stemming | | | | | | | | | |
| Sentence Match F1 | 0.4553 | 0.4806 | 0.521 | 0.554 | 0.5575 | 0.4779 | 0.4929 | 0.5308 | 0.5164 |
| ROUGE-2 F1 | 0.5210 | 0.5424 | 0.5907 | 0.6261 | 0.6341 | 0.5545 | 0.5469 | 0.6068 | 0.5859 |
| **E.02** no stopword removal and AssasBand stemming applied | | | | | | | | | |
| Sentence Match F1 | 0.4553 | 0.4857 | 0.5213 | 0.5546 | 0.5575 | 0.4835 | 0.4945 | 0.5262 | 0.5176 |
| ROUGE-2 F1 | 0.521 | 0.5453 | 0.5956 | 0.6269 | 0.6352 | 0.5538 | 0.5494 | 0.6023 | 0.5869 |
| **E.03** no stopword removal but lemmatization applied | | | | | | | | | |
| Sentence Match F1 | 0.4553 | 0.4885 | 0.5173 | 0.5534 | 0.5554 | 0.4843 | 0.4903 | 0.5125 | 0.5145 |
| ROUGE-2 F1 | 0.521 | 0.5487 | 0.5878 | 0.6251 | 0.632 | 0.5548 | 0.5442 | 0.5902 | 0.5833 |
| **E.04** no stopword removal but ultra stemming applied $L = 1$ | | | | | | | | | |
| Sentence Match F1 | 0.4553 | **0.5** | 0.534 | **0.5564** | **0.5612** | 0.4896 | 0.4893 | 0.5214 | **0.5217** |
| ROUGE-2 F1 | 0.521 | 0.5589 | 0.6016 | 0.6276 | 0.6364 | 0.5565 | 0.5442 | 0.5943 | 0.5885 |
| **E.05** no stopword removal but ultra stemming applied $L = 2$ | | | | | | | | | |
| Sentence Match F1 | 0.4553 | 0.4772 | **0.5449** | 0.5506 | 0.5525 | 0.4805 | 0.4919 | 0.5283 | 0.518 |
| ROUGE-2 F1 | 0.521 | 0.5394 | 0.61 | 0.6207 | 0.628 | 0.5476 | 0.5462 | 0.6021 | 0.5849 |
| **E.06** no stopword removal but ultra stemming applied $L = 3$ | | | | | | | | | |
| Sentence Match F1 | 0.4553 | 0.4871 | 0.5361 | 0.5459 | 0.5498 | 0.4828 | 0.4916 | 0.5279 | 0.5173 |
| ROUGE-2 F1 | 0.521 | 0.5461 | 0.607 | 0.6196 | 0.6273 | 0.5525 | 0.5466 | 0.61 | 0.587 |
| **E.07** no stopword removal but ultra stemming applied $L = 4$ | | | | | | | | | |
| Sentence Match F1 | 0.4553 | 0.4829 | 0.523 | 0.5537 | 0.5563 | 0.4821 | 0.4929 | 0.523 | 0.5163 |
| ROUGE-2 F1 | 0.521 | 0.5414 | 0.5945 | 0.6264 | 0.6332 | 0.56 | 0.5479 | 0.5974 | 0.5858 |
| **E.08** no stopword removal but ultra stemming applied $L = 5$ | | | | | | | | | |
| Sentence Match F1 | 0.4553 | 0.4838 | 0.5302 | 0.5537 | 0.5559 | 0.4814 | 0.4929 | 0.5232 | 0.5173 |
| ROUGE-2 F1 | 0.521 | 0.5433 | 0.6024 | 0.6264 | 0.6332 | 0.5591 | 0.5469 | 0.6034 | 0.5878 |
| **E.09** no stopword removal but ultra stemming applied $L = 6$ | | | | | | | | | |
| Sentence Match F1 | 0.4553 | 0.4847 | 0.5218 | 0.554 | 0.5559 | 0.481 | 0.4929 | 0.5268 | 0.5167 |
| ROUGE-2 F1 | 0.521 | 0.5465 | 0.5915 | 0.6266 | 0.6334 | 0.5572 | 0.5469 | 0.6043 | 0.5866 |
| **E.10** no stopword removal but ultra stemming applied $L = 7$ | | | | | | | | | |
| Sentence Match F1 | 0.4553 | 0.4806 | 0.5195 | **0.5549** | 0.5559 | 0.4865 | 0.4913 | 0.5188 | 0.5154 |
| ROUGE-2 F1 | 0.521 | 0.5424 | 0.593 | 0.6269 | 0.6334 | 0.5598 | 0.545 | 0.594 | 0.5849 |
| **E.11** no stopword removal but word embeddings applied, threshold 0.7 | | | | | | | | | |
| Sentence Match F1 | 0.4553 | 0.478 | 0.5137 | 0.5441 | 0.5499 | 0.4926 | 0.4896 | 0.5242 | 0.5132 |
| ROUGE-2 F1 | 0.521 | 0.5388 | 0.5832 | 0.6229 | 0.6277 | 0.5649 | 0.5439 | 0.5988 | 0.5829 |
| **E.12** no stopword removal but word embeddings applied, threshold 0.8 | | | | | | | | | |
| Sentence Match F1 | 0.4553 | 0.4859 | 0.5106 | 0.5441 | 0.5499 | 0.4938 | 0.4896 | 0.5242 | 0.514 |
| ROUGE-2 F1 | 0.521 | 0.5498 | 0.583 | 0.6229 | 0.6277 | 0.5668 | 0.5439 | 0.5972 | 0.5845 |
| **E.13** no stopword removal but word embeddings applied, threshold 0.9 | | | | | | | | | |
| Sentence Match F1 | 0.4553 | 0.4871 | 0.5104 | 0.5441 | 0.5499 | 0.4933 | 0.4896 | 0.5242 | 0.5141 |
| ROUGE-2 F1 | 0.521 | 0.5515 | 0.5812 | 0.6229 | 0.6277 | 0.5638 | 0.5439 | 0.5968 | 0.584 |
| **E.14** no stopword removal but Levenshtein distance 1 applied | | | | | | | | | |
| Sentence Match F1 | 0.4553 | 0.4711 | 0.5232 | 0.5404 | 0.5475 | 0.4772 | 0.4924 | 0.5287 | 0.5115 |
| ROUGE-2 F1 | 0.521 | 0.5327 | 0.5931 | 0.6187 | 0.6262 | 0.5563 | 0.5469 | 0.6055 | 0.5828 |
| **E.15** no stopword removal but Levenshtein distance 2 applied | | | | | | | | | |
| Sentence Match F1 | 0.4553 | 0.4784 | 0.5135 | 0.5442 | 0.5487 | **0.5059** | 0.4913 | 0.5291 | 0.5159 |
| ROUGE-2 F1 | 0.521 | 0.5393 | 0.5831 | 0.6219 | 0.6261 | 0.5724 | 0.5469 | 0.6003 | 0.5843 |
| **E.16** no stopword removal but Levenshtein distance 3 applied | | | | | | | | | |
| Sentence Match F1 | 0.4553 | 0.4759 | 0.5195 | 0.544 | 0.5476 | 0.4868 | 0.4912 | 0.5256 | 0.5129 |
| ROUGE-2 F1 | 0.521 | 0.5366 | 0.5886 | 0.6225 | 0.626 | 0.563 | 0.5457 | 0.6005 | 0.5833 |
| **E.17** no stopword removal but Levenshtein distance 4 applied | | | | | | | | | |
| Sentence Match F1 | 0.4553 | 0.4732 | 0.5228 | 0.5446 | 0.5465 | 0.4992 | 0.4917 | 0.5391 | 0.5167 |
| ROUGE-2 F1 | 0.521 | 0.5332 | 0.5872 | 0.6236 | 0.6248 | 0.5699 | 0.5458 | 0.6159 | 0.5858 |
| **E.18** stopwords removed but nothing else | | | | | | | | | |
| Sentence Match F1 | 0.4553 | 0.4854 | 0.5177 | 0.5428 | 0.5421 | 0.4946 | 0.5007 | 0.5336 | 0.5167 |
| ROUGE-2 F1 | 0.521 | 0.5475 | 0.5926 | 0.6151 | 0.6208 | 0.5665 | 0.5623 | 0.6102 | 0.5879 |
| **E.19** stopword removed and AssasBand stemming applied | | | | | | | | | |
| Sentence Match F1 | 0.4553 | 0.4856 | 0.517 | 0.5403 | 0.5418 | 0.479 | 0.4974 | 0.5217 | 0.5118 |
| ROUGE-2 F1 | 0.521 | 0.5467 | 0.5894 | 0.6128 | 0.6203 | 0.5551 | 0.5585 | 0.5987 | 0.5831 |
| **E.20** stopwords removed and lemmatization applied | | | | | | | | | |
| Sentence Match F1 | 0.4553 | 0.4887 | 0.5094 | 0.5366 | 0.5433 | 0.4875 | 0.497 | 0.5253 | 0.5125 |
| ROUGE-2 F1 | 0.521 | 0.5522 | 0.5843 | 0.6104 | 0.6206 | 0.5617 | 0.5577 | 0.6037 | 0.5844 |
| **E.21** stopwords removed and ultra stemming applied $L = 1$ | | | | | | | | | |
| Sentence Match F1 | 0.4553 | 0.4859 | 0.5262 | 0.5534 | **0.5599** | 0.4961 | 0.4901 | 0.5208 | 0.5189 |
| ROUGE-2 F1 | 0.521 | 0.5464 | 0.599 | 0.6265 | 0.636 | 0.5628 | 0.5477 | 0.5975 | 0.588 |
| **E.22** stopwords removed and ultra stemming applied $L = 2$ | | | | | | | | | |
| Sentence Match F1 | 0.4553 | **0.4929** | **0.5469** | 0.5518 | 0.5597 | 0.4878 | 0.4919 | 0.5381 | **0.5242** |
| ROUGE-2 F1 | 0.521 | 0.5537 | 0.6139 | 0.6249 | 0.6359 | 0.5594 | 0.5479 | 0.6112 | 0.5924 |
| **E.23** stopwords removed and ultra stemming applied $L = 3$ | | | | | | | | | |
| Sentence Match F1 | 0.4553 | 0.4853 | 0.5337 | 0.5413 | 0.5555 | 0.4717 | 0.4885 | 0.5284 | 0.5149 |
| ROUGE-2 F1 | 0.521 | 0.5463 | 0.6008 | 0.6158 | 0.6343 | 0.5501 | 0.5441 | 0.6063 | 0.5854 |
| **E.24** stopwords removed and ultra stemming applied $L = 4$ | | | | | | | | | |
| Sentence Match F1 | 0.4553 | 0.4898 | 0.5189 | 0.5461 | 0.5447 | 0.4978 | 0.4974 | 0.5283 | 0.5176 |
| ROUGE-2 F1 | 0.521 | 0.5535 | 0.5907 | 0.617 | 0.6228 | 0.5689 | 0.5586 | 0.6056 | 0.5882 |

(*continued on next page*)

**Table 8** (*continued*)

| Experiment setup (E.) | Baseline lead | C4.5 | Random forest | Naive bayes | Logistic regression | SVM Poly | SVM Puk | Multilayer perceptron | AVG. |
|---|---|---|---|---|---|---|---|---|---|
| **E.25** stopwords removed and ultra stemming applied $L = 5$ | | | | | | | | | |
| Sentence Match F1 | 0.4553 | 0.4895 | 0.5131 | 0.5412 | 0.5467 | 0.4992 | 0.4979 | **0.5392** | 0.5181 |
| ROUGE-2 F1 | 0.521 | 0.5526 | 0.5902 | 0.6138 | 0.625 | 0.5673 | 0.5583 | 0.6139 | 0.5887 |
| **E.26** stopwords removed and ultra stemming applied $L = 6$ | | | | | | | | | |
| Sentence Match F1 | 0.4553 | 0.4842 | 0.5209 | 0.5427 | 0.5459 | 0.4963 | 0.5014 | 0.5213 | 0.5161 |
| ROUGE-2 F1 | 0.521 | 0.5473 | 0.5958 | 0.615 | 0.6245 | 0.5649 | 0.5637 | 0.597 | 0.5869 |
| **E.27** stopwords removed and ultra stemming applied $L = 7$ | | | | | | | | | |
| Sentence Match F1 | 0.4553 | 0.4854 | 0.5129 | 0.541 | 0.5439 | **0.5052** | 0.4976 | 0.5273 | 0.5162 |
| ROUGE-2 F1 | 0.521 | 0.5485 | 0.5856 | 0.6133 | 0.6226 | 0.5735 | 0.558 | 0.6035 | 0.5864 |
| **E.28** stopwords removed and embeddings applied, thd. 0.7 | | | | | | | | | |
| Sentence Match F1 | 0.4553 | 0.4701 | 0.5191 | 0.5425 | 0.5439 | 0.4976 | 0.4963 | 0.5197 | 0.5127 |
| ROUGE-2 F1 | 0.521 | 0.5327 | 0.5857 | 0.6216 | 0.6242 | 0.5635 | 0.5571 | 0.5977 | 0.5832 |
| **E.29** stopwords removed and embeddings applied, thd. 0.8 | | | | | | | | | |
| Sentence Match F1 | 0.4553 | 0.4712 | 0.5336 | 0.5425 | 0.5439 | 0.4932 | 0.4963 | 0.5181 | 0.5141 |
| ROUGE-2 F1 | 0.521 | 0.5342 | 0.6047 | 0.6216 | 0.6242 | 0.5576 | 0.5571 | 0.597 | 0.5852 |
| **E.30** stopwords removed and embeddings applied, thd. 0.9 | | | | | | | | | |
| Sentence Match F1 | 0.4553 | 0.4767 | 0.5161 | 0.5425 | 0.5439 | 0.4921 | 0.4963 | 0.5177 | 0.5122 |
| ROUGE-2 F1 | 0.521 | 0.5419 | 0.5865 | 0.6216 | 0.6242 | 0.5642 | 0.5571 | 0.597 | 0.5846 |
| **E.31** stopword removed and Levenshtein distance 1 applied | | | | | | | | | |
| Sentence Match F1 | 0.4553 | 0.478 | 0.5058 | 0.5368 | 0.539 | 0.494 | 0.5024 | 0.536 | 0.5131 |
| ROUGE-2 F1 | 0.521 | 0.5392 | 0.5832 | 0.6153 | 0.6214 | 0.5644 | 0.5645 | 0.61 | |
| **E.32** stopword removed and Levenshtein distance 2 applied | | | | | | | | | |
| Sentence Match F1 | 0.4553 | 0.4772 | 0.5152 | 0.536 | 0.5438 | 0.4831 | 0.5013 | 0.519 | 0.5108 |
| ROUGE-2 F1 | 0.521 | 0.5379 | 0.5903 | 0.6141 | 0.6224 | 0.5552 | 0.5642 | 0.5987 | 0.5833 |
| **E.33** stopword removed and Levenshtein distance 3 applied | | | | | | | | | |
| Sentence Match F1 | 0.4553 | 0.4736 | 0.5218 | 0.5388 | 0.55 | 0.4954 | **0.5055** | 0.5319 | 0.5167 |
| ROUGE-2 F1 | 0.521 | 0.5344 | 0.5922 | 0.6172 | 0.6293 | 0.5672 | 0.5674 | 0.6144 | 0.5889 |
| **E.34** stopword removed and Levenshtein distance 4 applied | | | | | | | | | |
| Sentence Match F1 | 0.4553 | 0.4779 | 0.5082 | 0.5387 | 0.5456 | 0.4975 | **0.5041** | **0.5484** | 0.5172 |
| ROUGE-2 F1 | 0.521 | 0.5378 | 0.5841 | 0.6186 | 0.6259 | 0.5686 | 0.5654 | 0.6278 | 0.5897 |
| **Max S. Match F1** | | 0.5 | 0.5469 | 0.5564 | 0.5612 | 0.5059 | 0.5055 | 0.5484 | 0.5242 |
| **Max ROUGE-2 F1** | | 0.5589 | 0.6139 | 0.6276 | 0.6364 | 0.5735 | 0.5674 | 0.6278 | 0.5924 |
| **Min S. Match F1** | | 0.4701 | 0.5058 | 0.5360 | 0.539 | 0.4717 | 0.4885 | 0.5125 | 0.5108 |
| **Min ROUGE-2 F1** | | 0.5327 | 0.5812 | 0.6104 | 0.6203 | 0.5476 | 0.5439 | 0.5902 | 0.5828 |

#### 4.2.1. Sentence location

Typically, the first and last sentences of the source document are considered important (Nenkova & McKeown, 2012). It is considered the most significant feature in many studies because it represents an overview of the document (Fattah & Ren, 2008; Yeh, Ke, & Yang, 2008). We used the following two features to properly represent it ($S$ is the current sentence). *Normalized sentence location in a document*

$$\text{Score}_{\text{Feature1}}(S) = \frac{\text{position of } S}{\text{total number of sentences}} \tag{1}$$

*Sentence location in a document*

$$\text{Score}_{\text{Feature2}}(S) = \begin{cases} \text{ONE} & \text{first sentence,} \\ \text{TWO} & \text{second sentence,} \\ \text{LAST} & \text{last sentence,} \\ \text{REST} & \text{otherwise.} \end{cases} \tag{2}$$

#### 4.2.2. Length of sentence

The length of a sentence can also be an indicator. Usually, very long and tiny sentences are not included in the summary.

$$\text{Score}_{\text{Feature3}}(S) = \frac{\text{number of words in sentence } S}{\text{total words in the document}} \tag{3}$$

The denominator in Eq. (3) and all the forthcoming equations below is used for normalization.

#### 4.2.3. Number of title words

It is the number of words in the title that are present in a sentence. The sentences containing the words that occurred in the title indicate the source document's central theme, so they have a greater chance to be added in summary (Fattah, 2014). Repetition of title words in the sentences is also counted.

$$\text{Score}_{\text{Feature4}}(S) = \frac{\text{number of title words in } S}{\text{total words in } S} \tag{4}$$

#### 4.2.4. Number of nouns, proper nouns, pronouns and numerical values

The sentences containing nouns and proper nouns have more chance to be included in the summary (Fattah & Ren, 2008).

$$\text{Score}_{\text{Feature5}}(S) = \frac{\text{number of nouns in } S}{\text{total words in } S} \tag{5}$$

$$\text{Score}_{\text{Feature6}}(S) = \frac{\text{number of proper nouns in } S}{\text{total words in } S} \tag{6}$$

The sentences containing pronouns are not considered important and are usually left behind from the summary.

$$\text{Score}_{\text{Feature7}}(S) = \frac{\text{number of pronouns in } S}{\text{total words in } S} \tag{7}$$

The sentences containing numerical values are considered important and have more chances to be included in the summary. Numbers written in words are not considered for the moment.

$$\text{Score}_{\text{Feature8}}(S) = \frac{\text{number of numerical values in } S}{\text{total words in } S} \tag{8}$$

#### 4.2.5. Sentence cohesion

It is simply to compute the vocabulary overlap between a sentence and other sentences of the source document and then sum up those similarity values:

$$\text{Score}_{\text{Feature9}}(S) = \frac{\text{Keywords in } S \bigcap \text{Keywords in other sentences}}{max(\text{Keywords in } S_i \bigcap \text{Keywords in other sentences})} \tag{9}$$

where $S$ represents the sentence under consideration and $S_i$ is the sentence number $i$ in the document. Keywords in a document are thematic words that contain important information.

**Table 9**
*Experimental Setup III* – $F_1$ results for **fixed-length word segmented** dataset. The top results are boldfaced and underlined, whereas, the second and third best are only boldfaced.

| Experiment settings (E.) | Baseline: lead | C4.5 | Random forest | Naive bayes | Logistic regression | SVM Poly | SVM Puk | Multilayer perceptron | AVG. |
|---|---|---|---|---|---|---|---|---|---|
| **E.35** Fixed-length word segmentation of length 2 | | | | | | | | | |
| Sentence Match F1 | 0.4553 | 0.5167 | **0.5602** | **0.5568** | 0.5576 | 0.5318 | 0.5032 | **0.5479** | 0.5392 |
| ROUGE-2 F1 | 0.521 | 0.594 | 0.6311 | 0.6372 | 0.6335 | 0.6088 | 0.5695 | 0.6161 | 0.6129 |
| **E.36** Fixed-length word segmentation of length 3 | | | | | | | | | |
| Sentence Match F1 | 0.4553 | **0.5314** | **0.5552** | **0.5572** | 0.5565 | **0.534** | **0.5157** | 0.5437 | **0.542** |
| ROUGE-2 F1 | 0.521 | 0.6092 | 0.6256 | 0.6378 | 0.6334 | 0.6121 | 0.5903 | 0.6209 | 0.6185 |
| **E.37** Fixed-length word segmentation of length 4 | | | | | | | | | |
| Sentence Match F1 | 0.4553 | **0.5294** | 0.5578 | 0.5506 | **0.564** | **0.5381** | **0.5155** | **0.552** | **0.5439** |
| ROUGE-2 F1 | 0.521 | 0.6136 | 0.6299 | 0.6324 | 0.6396 | 0.6198 | 0.5835 | 0.6335 | 0.6218 |
| **E.38** Fixed-length word segmentation of length 5 | | | | | | | | | |
| Sentence Match F1 | 0.4553 | **0.5553** | **0.5502** | **0.559** | **0.5682** | **0.5501** | **0.5153** | **0.5479** | **0.5494** |
| ROUGE-2 F1 | 0.521 | 0.6336 | 0.6249 | 0.6406 | 0.6459 | 0.629 | 0.591 | 0.6226 | 0.6268 |
| **E.39** Fixed-length word segmentation of length 6 | | | | | | | | | |
| Sentence Match F1 | 0.4553 | 0.5145 | 0.5431 | 0.5564 | **0.5702** | 0.533 | 0.5146 | **0.5485** | 0.54 |
| ROUGE-2 F1 | 0.521 | 0.6054 | 0.6111 | 0.6383 | 0.644 | 0.6109 | 0.5856 | 0.6249 | 0.6172 |
| **Max S. Match F1** | | 0.5553 | 0.5602 | 0.559 | 0.5702 | 0.5501 | 0.5157 | 0.552 | 0.5494 |
| **Max ROUGE-2 F1** | | 0.6336 | 0.6311 | 0.6406 | 0.6459 | 0.629 | 0.591 | 0.6335 | 0.6268 |
| **Min S. Match F1** | | 0.5145 | 0.5431 | 0.5506 | 0.5565 | 0.5318 | 0.5032 | 0.5437 | 0.5392 |
| **Max ROUGE-2 F1** | | 0.594 | 0.6111 | 0.6324 | 0.6334 | 0.6088 | 0.5695 | 0.6161 | 0.6129 |

#### 4.2.6. Term frequency-inverse sentence frequency (TF-ISF)

TF-ISF can calculate keywords. The computation of TF-ISF is similar to TF-IDF, but a sentence replaces the notion of a document.

$$\text{Score}_{\text{Feature10}}(S) = \text{TF}(w, S) \times \text{ISF}(w) \tag{10}$$

where term frequency $\text{TF}(w, S)$ represents the number of times a word $w$ occurs in a sentence $S$, and inverse sentence frequency $\text{ISF}(w)$ is calculated as:

$$\text{ISF}(w) = \log\left(\frac{\text{number of words in } S}{\text{SF}(w)}\right) \tag{11}$$

where $\text{SF}(w)$ represents the number of sentences in which the word appears.

### 4.3. Classification algorithms

To build a text summarization framework, we employed six classification algorithms, namely, C4.5 decision tree (Quinlan, 1993), Random Forest (Breiman, 2001), Naive Bayes (John & Langley, 1995), Logistic regression (le Cessie & van Houwelingen, 1992), Artificial Neural Network – Multilayer Perceptron (Witten, Frank, Hall, & Pal, 2016), and Support Vector Machines – SVM (Platt, 1998). For SVM, we have used two kernel functions: (1) Polynomial and (2) Pearson VII Universal Kernel (PUK) (Üstün, Melssen, & Buydens, 2006), as these two kernels performed better in our initial experiments, making seven classification algorithms in total. We used the open-source WEKA workbench implementations (Witten et al., 2016) of these algorithms.

### 4.4. Machine generated extracts

Machine Generated (MG) extractive summaries are produced by the corresponding models trained on six classification algorithms (Section 4.3). Since the reference extracts provided by CORPURES have a compression rate of 40%, the same compression rate for the MG summaries is kept in all the experiments. From an implementation point of view, it is worth mentioning that WEKA implementations of the selected six classification algorithms can return a continuous value between 0 to 1, which allowed us to apply the needed compression rate. For every experiment, the MG summaries are generated again after applying the settings needed for the experiment. The MG summaries are: (7 models × 161 documents × number of experiments).

*Baseline: lead based* For a comparison between supervised learning algorithms, Lead based method is used as a baseline, where the first N-sentences of a document are taken as a summary. The value of N depends upon the compression rate which is 40% in our case.

### 4.5. Evaluation methodology

A Machine Generated (MG) summary produced by a classification algorithm can be compared with the corresponding Reference Summary (RS) using different evaluation methods. The following two are the most established methodologies based on the usage in studies over the past ten years (Widyassari et al., 2020).

Comparing how many sentences as a whole match between MG and RS using evaluation measures Precision, Recall, and $F_1$, (Neto et al., 2002; Verberne et al., 2018). It is a strict method as it makes a true/false judgment on the selection of sentences. This methodology is more prevalent in studies related to extracts (Widyassari et al., 2020). In this paper, we have used this measure to report the results. The results in Section 5 are discussed with $F_1$ only due to space restrictions. We call it Sentence Match $F_1$ in Tables 8 and 9. A more flexible approach counts the overlapping textual units (such as n-grams, word sequences, and word pairs) using ROUGE (Lin, 2004; Liu & Liu, 2008). Then the evaluation measures Precision, Recall, and $F_1$ are calculated. This approach is suitable for both extracts and abstracts and considers to be the most widely used method (Widyassari et al., 2020). Therefore, we also report results using ROUGE-2 $F_1$, though we did not discuss and explain these scores due to space restrictions in our experiments in Section 5). ROUGE-2 uses bigram overlapping of the words.

Since, we have seven models, seven comparisons were performed for each summary. Subsequently, confusion matrices are generated from the responses of the algorithms. The main elements of confusion matrices are True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). For each MG summary, TP represents the sum of correctly identified summary units, FP represents the sum of incorrectly classified summary units, TN represents the sum of correctly identified non-summary units, and FN represents the sum of incorrectly classified non-summary units. The unit is deifned as a sentence for the first evaluation metric, and bigram for the second evaluation metric. The Precision (P) is the percentage of the chosen units that are found in the reference summary. The Recall (R) is the percentage of the reference summary units that were chosen by a machine generated summary. Finally, the $F_1$ measure is the harmonic mean of precision and recall:

$$P = \frac{TP}{TP + FP} \qquad R = \frac{TP}{TP + FN} \qquad F_1 = 2 \times \frac{P \times R}{P + R} \tag{12}$$

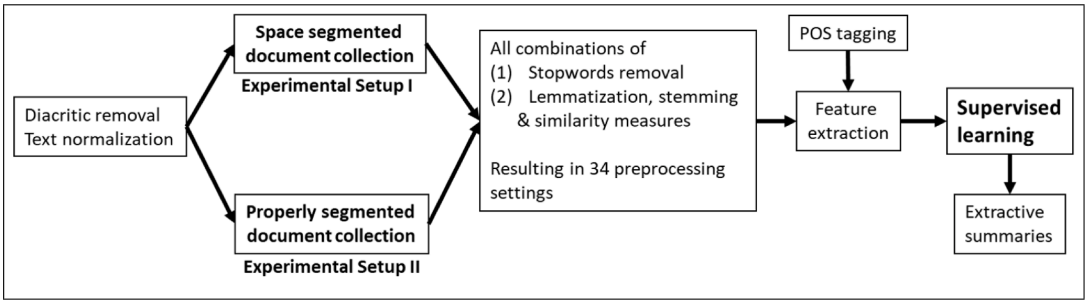



**Fig. 5.** A diagram explaining experimental setup I and II.

#### 4.5.1. Evaluations on reference summaries

The classification algorithms are evaluated using 10-fold cross-validation (Arlot & Celisse, 2010; Borra & Ciaccio, 2010; Kohavi, 1995) in all the experiments. In supervised learning, we usually have one *right* answer for each instance. The summarization task is different because an instance is a whole document that needs to be summarized. Inside a document, we have the *right* answer for each sentence. Due to this reason, the k-fold implementation of WEKA needed to be adapted for our specific needs (call it document wise 10-fold cross validation). In each fold, 145 (90%) documents of 161 are taken for a train set and 16 documents (10%) taken as a test set. A random order for documents in the dataset was fixed.

## 5. Benchmark experiments

Experiments aim to evaluate the effect of a number of constraints when producing extractive summaries using supervised learning. These constraints are (1) the effect of preprocessing settings on two versions of the document collection CORPURES: tokenized on space and properly segmented on actual word boundaries (Section 5.1) (2) the effect of fixed-length word segmentation (Section 5.2) (3) the effect of features (Section 5.3), and (4) the effect of artificially generated datasets (Section 5.4).

As a starting point, the common preprocessing settings (diacritic removal, text normalization, POS tagging) are performed on the CORPURES document collection. This step is performed for all experiments. Second, one of the experimental setups is picked to produce a specific version of CORPURES, and features reported in Section 4.2 are produced. Third, models are trained using supervised learning algorithms (Section 4.3). Fourth, machine generated extractive summaries are produced (Section 4.4). Finally, evaluations of the models are performed (Section 4.5).

### 5.1. The effect of preprocessing settings

These experiments aim to determine the effect of various preprocessing settings on space segmented and properly segmented versions of the CORPURES document collection. More precisely, we performed the experiments with 34 preprocessing settings on:

1. Space segmented version of CORPURES reported in Experimental Setup I (Section 5.1.1)
2. Properly segmented version of CORPURES reported in Experimental Setup II (Section 5.1.2)

For both experimental setups, we essentially produce all the combinations of two preprocessing settings:

1. stopword removal
2. lemmatization, stemming & similarity measures (to reduce the word surface forms to their stem) for three features: Number of Title words, Sentence cohesion, and TF-ISF (Features: 4, 9, 10; Section 4.2). For instance, Ultra stemming with length 1 to 7 is used, word embedding with threshold 0.7 to 0.9 is used, and Levenshtein distance 1 to 4 is used.

An overall picture explaining experimental setup I and II is shown in Fig. 5. These two experimental setups allow us to determine what combination of preprocessing settings is more suitable for Urdu extractive summarization task both on space segmented and properly segmented document collection. The $F_1$ results on space segmented document collection are shown in Table 8, though only Sentence Match

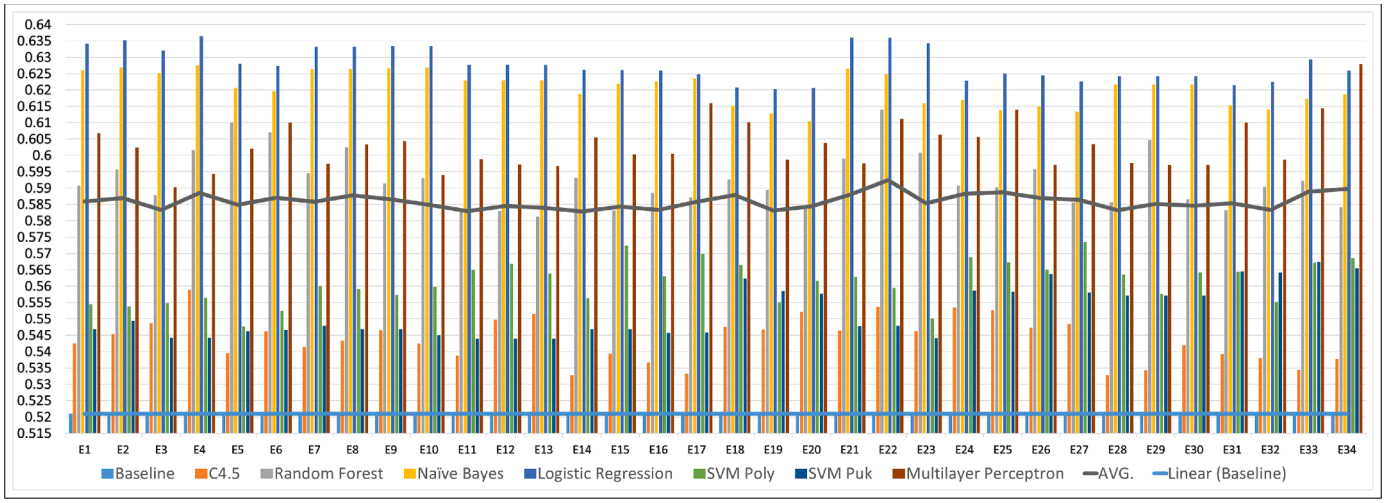



**Fig. 6.** *Experimental Setup I* – A bar chart of ROUGE-2 $F_1$ results for **space segmented** dataset.

$F_1$ scores are discussed, ROUGE-2 scores are omitted from discussion (A bar chart is made using ROUGE-2 $F_1$ in Fig. 6). All the $F_1$ scores discussed in the forthcoming sections corresponds to Sentence Match $F_1$. The results for properly segmented document collection are omitted due to space limitations. See Section 5.1.2 for more details. The top results are **boldfaced and underlined**; the second best results are only **boldfaced**; the third best are only underlined.

#### 5.1.1. Experimental setup I: result and analysis on space segmented document collection

This experiment aims to determine the effect of various preprocessing settings for Urdu extractive summarization task when words are tokenized on space in the document collection. See results in Table 8, and ROUGE-2 $F_1$ in Fig. 6.

First, it can be seen that all the $F_1$ scores in the table outperform the baseline. Only Sentence Match $F_1$ scores are discussed, whereas ROUGE-2 scores are omitted from the discussion. With regards to evaluating the scores of learning algorithms, the best performing algorithms on average for all the preprocessing settings are Logistic Regression (best), Naive Bayes (second best), and Multilayer Perceptron (third best). One reason for this performance could be the fact that many features including weak ones, lead to a better result with maximum entropy classifiers (Logistic Regression). Similarly, the conditional independence in Naive Bayes limits the effect of weak features on good features. In contrast, the least performing algorithms are C4.5 (last), Support Vector Machines with Polynomial kernel – SVM Poly (second last), and Support Vector Machines with Pearson VII Universal Kernel (SVM Puk) (third last).

Individually, the highest $F_1$ score in Table 8 is **0.5612**, achieved in E.04 – no stopword removal but ultra stemming applied $L = 1$, for Logistic Regression algorithm. If we take the average for all the algorithms (the last column in Table 8), the best three preprocessing settings are when ultra stemming is applied ($L = 1, 2$), with or without stopwords removal. It is good news for a low-resource language such as Urdu, as applying ultra stemming is trivial and inexpensive.

- E.22 – stopwords removed and ultra stemming applied $L = 2$ (*best*)
- E.04 – no stopword removal but ultra stemming applied $L = 1$ (*second best*)
- E.21 – stopwords removed and ultra stemming applied $L = 1$ (*third best*)
- E.05 – no stopword removal but ultra stemming applied $L = 2$ (*fourth best but close to the third best*)

Which preprocessing settings outperform experiment E.01 (when no preprocessing is applied; $F_1 = 0.5164$)? Our findings are the following: When the stopwords only are removed (i.e., E.18), the average $F_1$ score negligibly increases (0.5167). When stemming is applied without stopword removal (i.e., E.02), the score further increases slightly (0.5176). However, the stemming and stopword removal together (i.e., E.19) decrease the score (0.5118). In contrast, when Humayoun's lemmatizer is applied with or without stopword removal (i.e., E.03 and E.20), the score slightly decreases as compared to E.01. To summarize, stemming & lemmatization and stopword removal together are not good.

When the Levenshtein distance is applied with or without stopword removal, only the following settings beat the $F_1$ score of the experiment E.01. It seems that the Levenshtein distance 2, 3, 4 is able to capture a few similar word forms.

- E.34 stopword removal and Levenshtein distance 4 applied (0.5172)
- E.17 no stopword removal but Levenshtein distance 4 applied (0.5167)
- E.33 stopword removal and Levenshtein distance 3 applied (0.5167)
- E.15 no stopword removal but Levenshtein distance 2 applied (0.5159)

Word embeddings are an excellent way to combine the surface forms with corresponding terms (or concepts). Therefore, we expected that its use as preprocessing might outperform. However, it turns out that none of these settings outperformed E.01. It might be due to the overfitting, i.e., many different words playing different roles in the text got converted to the same concepts. However, the highest scores among this setting are when word embeddings with thresholds 0.9 and 0.8 are applied.

#### 5.1.2. Experimental setup II: result and analysis on properly segmented document collection

This experiment aims is to determine the effect of various preprocessing settings for Urdu extractive summarization task when words are properly segmented in the document collection. Similar to space segmented document collection, all the $F_1$ results outperform the baseline. However, all the $F_1$ scores are lower than the corresponding $F_1$ scores for space segmented corpus. Despite the lower $F_1$ scores, the results are comparable to the space segmented corpus with a few variations. As discussed in Section 4.1.2, this might be due to the following reasons:

1. The mistakes in identifying proper word segments.
2. The limitations of existing resources (stemming, lemmatization, and POS tagging) for the properly segmented words.

Due to space limitations, we decided to omit these results from the paper.

### 5.2. Experimental setup III: result and analysis on fixed-length word segmented document collection

To measure the effect of fixed-length word segmentation for the Urdu extractive summarization task, we performed five experiments using CORPURES document collection. In these experiments, we remove spaces from the text. Then, it is divided into fixed-length chunks of 2 to 6, keeping the sentences intact. Note that in this experiment, only common preprocessing settings are applied (See Section 4.1.1). Other preprocessing settings: stopword removal and lemmatization, stemming, and similarity measures are not applied. It is because, with fixed-length word segmentation, tokens are no longer identifiable words, and therefore, there is no point in applying the settings mentioned above. The $F_1$ scores are shown in Table 9. Only Sentence Match $F_1$ scores are discussed, whereas ROUGE-2 scores are omitted from the discussion.

It is surprising to note that, on average $F_1$ score for all the algorithms (the last column in Table 9), the fixed-length word segmentation with length 3, 4, 5 outperforms all the preprocessing settings, i.e., the scores from space segmented corpus and properly segmented corpus (E1 – E34). The fixed-length word segmentation with length 5 gives the highest $F_1$ score on average for all the algorithms. It is good news for a low-resource language such as Urdu, as the preparation of fixed-length corpus is trivial and inexpensive.

With regards to evaluating the results of learning algorithms, the trend matches with other preprocessing settings (i.e., E1 – E34). For example, the best performing algorithms on average for all the preprocessing settings are Logistic Regression (best), Naive Bayes (second best), and Multilayer Perceptron (third best), whereas the least performing algorithms are SVM Puk (last), C4.5 (second last) and SVM Poly (third last).

Individually, the highest $F_1$ score in Table 9 is **0.5702**, achieved in the setting: E.48 (fixed-length word segmentation of length 6) for Logistic Regression.

### 5.3. Experimental setup IV: the effect of features

This experiment aims to determine the role of each feature (attribute) for the extractive summarization task. The preprocessing settings in previous experiments (i.e., E.01 to E.39) affect the values for various

**Table 10**
*Experimental Setup IV*: The effect of features on three attribute selection algorithms: (1) Correlation based Feature Selection, (2) Classifier Subset Evaluator, (3) Wrapper Subset Evaluator. Results between 0–10 (0:least significant, 10:most significant).

| | CfsSubsetEval | ClassifierSubsetEval | | WrapperSubsetEval | | AVG. |
|---|---|---|---|---|---|---|
| | | NaiveBayes | Logistic | NaiveBayes | Logistic | |
| | 10 | 10 | 10 | 10 | 10 | 10 |
| **E.01a: no preprocessing settings applied** | | | | | | |
| $F_1$: Sentence Number | 10 | 4 | 4 | 1 | 5 | **4.8** |
| $F_2$: Sentence Position | 10 | 10 | 10 | 10 | 10 | **10** |
| $F_3$: Sentence Length | 6 | 5 | 8 | 0 | 6 | **5** |
| $F_4$: Title Word | 8 | 0 | 2 | 0 | 1 | 2.2 |
| $F_5$: Noun | 0 | 5 | 3 | 9 | 3 | 4 |
| $F_6$: Numeric | 2 | 3 | 3 | 2 | 2 | 2.4 |
| $F_7$: Proper Noun | 0 | 0 | 2 | 1 | 1 | 0.8 |
| $F_8$: Pronoun | 0 | 2 | 3 | 10 | 0 | 3 |
| $F_9$: Sentence Cohesion | 0 | 4 | 3 | 1 | 3 | 2.2 |
| $F_{10}$: TF-ISF | 0 | 0 | 3 | 0 | 1 | 0.8 |
| **E.04a: stopwords not removed but ultra stemming applied** $L = 1$ | | | | | | |
| $F_1$: Sentence Number | 10 | 4 | 5 | 0 | 4 | **4.6** |
| $F_2$: Sentence Position | 10 | 10 | 10 | 10 | 10 | **10** |
| $F_3$: Sentence Length | 6 | 5 | 6 | 2 | 5 | **4.8** |
| $F_4$: Title Word | 8 | 1 | 4 | 0 | 1 | 2.8 |
| $F_5$: Noun | 0 | 5 | 2 | 7 | 3 | 3.4 |
| $F_6$: Numeric | 2 | 3 | 3 | 1 | 1 | 2 |
| $F_7$: Proper Noun | 0 | 1 | 2 | 2 | 0 | 1 |
| $F_8$: Pronoun | 0 | 3 | 1 | 8 | 1 | 2.6 |
| $F_9$: Sentence Cohesion | 0 | 6 | 4 | 2 | 3 | 3 |
| $F_{10}$: TF-ISF | 0 | 2 | 3 | 0 | 2 | 1.4 |
| **E.26a: stopwords removed and ultra stemming applied** $L = 1$ | | | | | | |
| $F_1$: Sentence Number | 10 | 4 | 5 | 1 | 4 | **4.8** |
| $F_2$: Sentence Position | 10 | 10 | 10 | 10 | 10 | **10** |
| $F_3$: Sentence Length | 6 | 5 | 8 | 2 | 6 | **5.4** |
| $F_4$: Title Word | 8 | 0 | 5 | 0 | 1 | 2.8 |
| $F_5$: Noun | 0 | 4 | 3 | 7 | 4 | 3.6 |
| $F_6$: Numeric | 2 | 2 | 4 | 2 | 1 | 2.2 |
| $F_7$: Proper Noun | 0 | 0 | 0 | 1 | 2 | 0.6 |
| $F_8$: Pronoun | 0 | 2 | 4 | 8 | 1 | 3 |
| $F_9$: Sentence Cohesion | 0 | 6 | 7 | 3 | 5 | 4.2 |
| $F_{10}$: TF-ISF | 0 | 1 | 2 | 0 | 1 | 0.8 |
| **E.47: fixed-length word segmentation of length 5** | | | | | | |
| $F_1$: Sentence Number | 10 | 1 | 5 | 0 | 5 | **4.2** |
| $F_2$: Sentence Position | 10 | 10 | 10 | 10 | 10 | **10** |
| $F_3$: Sentence Length | 0 | 0 | 9 | 0 | 7 | 3.2 |
| $F_4$: Title Word | 1 | 0 | 0 | 1 | 0 | 0.4 |
| $F_5$: Noun | 0 | 1 | 5 | 0 | 2 | 1.6 |
| $F_6$: Numeric | 0 | 5 | 2 | 1 | 0 | 1.6 |
| $F_7$: Proper Noun | 0 | 1 | 4 | 0 | 5 | 2 |
| $F_8$: Pronoun | 0 | 0 | 1 | 0 | 2 | 0.6 |
| $F_9$: Sentence Cohesion | 10 | 10 | 10 | 10 | 10 | **10** |
| $F_10$: TF-ISF | 9 | 0 | 6 | 0 | 3 | 3.6 |

features, and therefore, the effect of features on the task will justify the corresponding $F_1$ scores. We picked the following preprocessing settings for this experiment:

- E.01 – no preprocessing settings applied (*representing weak scores*)
- E.04 – stopwords not removed but ultra stemming applied $L = 1$ (*representing the good scores*)
- E.21 – stopwords removed and ultra stemming applied $L = 1$ (*representing the good scores*)
- E.38 – fixed-length word segmentation of length 5 (*representing the best scores*)

We used three feature selection algorithms: Correlation based Feature Selection (CFS) (Hall, 1998), Classifier Subset Evaluator (Hall et al., 2009), Wrapper Subset Evaluator (Kohavi & John, 1997). The feature selection techniques search the subset of feature space, evaluating each one on some criteria. Many techniques achieve it in two steps: a feature subset evaluator to measure the efficacy, and a search method to find good candidates from the search space. In these feature selection algorithms, we have used BestFirst as search method with bi-directional tracking. CFS evaluates the efficacy based on the idea that "a good feature subset has features which are highly correlated to the output class yet unrelated to each other" (Hall, 1998). In contrast, Classifier Subset Evaluator and Wrapper Subset Evaluator are learner based feature selection algorithms that use a classifier as an evaluator. Thus, they are relatively slow. However, the results are better in terms of classification accuracy as each feature is judged with respect to learning. However, the results may be biased by the learning algorithm. To overcome these issues, we used two classifiers: Logistic Regression and Naive Bayes because (1) they outperformed in other experiments, (2) relatively inexpensive computationally, and (3) fundamentally different from each other. A 10-fold cross validation is used in this experiment.

The results are shown in Table 10. The results are between 0 - 10; 0 being least significant, whereas 10 being most significant. It can be seen that the most effective feature for all the experiments is $F_2$: Sentence Position, whereas, $F_1$: Sentence Number (normalized) and $F_3$: Sentence Length are also reliable features. Note the evolution for two features $F_9$ and $F_{10}$ (Sentence cohesion and TF-ISF, respectively). For instance, it seems that E.38 (the fixed-length word segmentation of length 5) can find the common fragments of subwords (simulating

**Table 11**
*Experimental Setup V*: Results for the effect of synthetic dataset. Only Sentence match F1 scores are shown.

| CORPURES Dataset | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| (a) Original dataset details | | | | | | | | | | | |
| Total instances | 2649 | Percentage | | | | | | | | | |
| Reject class instances | 1589 | 58.98% | | | | | | | | | |
| Keep class instances | 1060 | 40.02% | | | | | | | | | |
| **Resampled Dataset 1** | | | | | | | | | | | |
| | **Fold-1** | **Fold-2** | **Fold-3** | **Fold-4** | **Fold-5** | **Fold-6** | **Fold-7** | **Fold-8** | **Fold-9** | **Fold-10** | **AVG.** |
| (b) Synthetic datasets' details for each fold | | | | | | | | | | | |
| **Total Instances** | 2926 | 2807 | 2843 | 2849 | 2919 | 2828 | 2845 | 2933 | 2777 | 2890 | **2861.7** |
| **Reject class** | 1646 | 1405 | 1418 | 1426 | 1460 | 1414 | 1420 | 1466 | 1388 | 1444 | **1448.7** |
| **Keep class** | 1462 | 1402 | 1425 | 1423 | 1459 | 1414 | 1425 | 1467 | 1389 | 1446 | **1431.2** |
| **Resampled Dataset 2** | | | | | | | | | | | |
| **Total Instances** | 11,720 | 11,228 | 11,372 | 11,396 | 11,676 | 11,312 | 11,380 | 11,732 | 11,108 | 11,560 | **11448.4** |
| **Reject class** | 5856 | 5620 | 5672 | 5704 | 5840 | 5656 | 5680 | 5864 | 5552 | 5776 | **5722** |
| **Keep class** | 5854 | 5608 | 5700 | 5692 | 5835 | 5656 | 5700 | 5868 | 5556 | 5784 | **5725.3** |
| **Resampled Dataset 3 – undersampled** | | | | | | | | | | | |
| **Total Instances** | 1950 | 1870 | 1900 | 1898 | 1946 | 1886 | 1900 | 1956 | 1852 | 1928 | **1908.6** |
| **Reject class** | 975 | 935 | 950 | 949 | 973 | 943 | 950 | 978 | 926 | 964 | **954.3** |
| **Keep class** | 975 | 935 | 950 | 949 | 973 | 943 | 950 | 978 | 926 | 964 | **954.3** |

| **Resampled Dataset 1, Size: 2862** | | | | | | |
|---|---|---|---|---|---|---|
| | **E.01a: no preprocessing applied** | | | **E.38: fixed-length word seg. length 5** | | |
| | **Sampled** | **Original** | **Improvement** | **Sampled** | **Original** | **Improvement** |
| (c) Sentence match F1 scores for each classifier on Resampled Dataset 1 | | | | | | |
| **NB** | 0.5507 | 0.554 | − 0.0033 | 0.5517 | 0.559 | − 0.0073 |
| **Logistic** | 0.5525 | 0.5575 | − 0.005 | 0.5718 | 0.5682 | 0.0036 |
| **ANN** | 0.5268 | 0.5308 | − 0.004 | 0.5584 | 0.5479 | 0.0105 |
| **C4.5** | 0.5161 | 0.4806 | 0.0355 | 0.5423 | 0.5553 | − 0.013 |
| **RandomForest** | 0.5208 | 0.521 | − 0.0002 | 0.5491 | 0.5502 | − 0.0011 |
| **SVM Poly** | 0.4914 | 0.4779 | 0.0135 | 0.554 | 0.5501 | 0.0039 |
| **SVM Puk** | 0.4862 | 0.4929 | − 0.0067 | 0.5264 | 0.5153 | 0.0111 |
| AVG | **0.5206** | **0.5164** | **0.0043** | **0.5505** | **0.5494** | **0.0011** |

| **Resampled Dataset 2, Size: 11,448** | | | | | | |
|---|---|---|---|---|---|---|
| | **E.01a: no preprocessing applied** | | | **E.38: fixed-length word seg. length 5** | | |
| | **Sampled** | **Original** | **Improvement** | **Sampled** | **Original** | **Improvement** |
| (d) Sentence match F1 scores for each classifier on Resampled Dataset 2 | | | | | | |
| **NB** | 0.5485 | 0.554 | − 0.0055 | 0.5544 | 0.559 | − 0.0046 |
| **Logistic** | 0.5493 | 0.5575 | − 0.0082 | 0.5616 | 0.5682 | − 0.0066 |
| **ANN** | 0.5203 | 0.5308 | − 0.0105 | 0.561 | 0.5479 | 0.0131 |
| **C4.5** | 0.4894 | 0.4806 | 0.0088 | 0.4564 | 0.5553 | − 0.0989 |
| **RandomForest** | 0.5317 | 0.521 | 0.0107 | 0.5345 | 0.5502 | − 0.0157 |
| **SVM Poly** | 0.4865 | 0.4779 | 0.0086 | 0.5013 | 0.5501 | − 0.0488 |
| **SVM Puk** | 0.4907 | 0.4929 | − 0.0022 | 0.5139 | 0.5153 | − 0.0014 |
| **AVG** | **0.5166** | **0.5164** | **0.0002** | **0.5262** | **0.5494** | **− 0.0233** |

| **Resampled Dataset 3, Size: 1,909** | | | | | | |
|---|---|---|---|---|---|---|
| | **E.01a: no preprocessing applied** | | | **E.38: fixed-length word seg. length 5** | | |
| | **Sampled** | **Original** | **Improvement** | **Sampled** | **Original** | **Improvement** |
| (e) Sentence match F1 scores for each classifier on Resampled Dataset 3 | | | | | | |
| **NB** | 0.5466 | 0.554 | − 0.0074 | 0.5589 | 0.559 | − 0.0001 |
| **Logistic** | 0.549 | 0.5575 | − 0.0085 | 0.5674 | 0.5682 | − 0.0008 |
| **ANN** | 0.5427 | 0.5308 | 0.0119 | 0.5568 | 0.5479 | 0.0089 |
| **C4.5** | 0.5135 | 0.4806 | 0.0329 | 0.5238 | 0.5553 | − 0.0315 |
| **RandomForest** | 0.517 | 0.521 | − 0.004 | 0.549 | 0.5502 | − 0.0012 |
| **SVM Poly** | 0.508 | 0.4779 | 0.0301 | 0.56 | 0.5501 | 0.0099 |
| **SVM Puk** | 0.5016 | 0.4929 | 0.0087 | 0.5317 | 0.5153 | 0.0164 |
| **AVG** | **0.5255** | **0.5164** | **0.0091** | **0.5497** | **0.5494** | **0.0002** |

proper stemming), and these features started to play a significant role.

When comparing E.04 and E.21 (ultra stemming applied $L = 1$ with/ without stopwords), it can be observed from Table 10 that the stopword removal has a positive effect[18] In addition to sentence cohesion, it probably also has increased the accuracy for POS tagging as stopwords are removed and cannot be misclassified anymore by the tagger. The results in Table 10 also concur the following about preprocessing settings:

1. Space and proper word segmentation leave the mistakes behind, affecting features like sentence cohesion and TF-ISF. The mistakes in proper word segmentation must be greater than the space segmented corpus explaining the lower $F_1$ scores in Experiment II.
2. The AssasBand stemming and Humayoun's lemmatizer are making mistakes probably due to over stemming and low coverage, respectively.
3. Similarly, for some settings in the Levenshtein distance ($L = 2$ to 5) we have somewhat better results probably when the parameters are able to identify similar terms a bit better. The same is happening for word embeddings.

[18] $F_1$: Sentence number, 4.6 to 4.8; $F_3$: Sentence length, 4.8 to 5.4; $F_5$: Noun, 3.4 to 3.6; $F_6$: Numeric, 3 to 2.2; $F_8$: Pronoun, 2.6 to 3; $F_9$: Sentence cohesion, 3 to 4.2; $F_{10}$: TF-ISF, 1.4 to 0.8.

### 5.4. Experimental setup V: the effect of synthetic dataset

This experiment aims to assess the impact of the artificially generated dataset for the extractive text summarization task. We have three balanced synthetic datasets created from the CORPURES by SMOTE. In the first dataset (Resampled dataset 1), the classes have been balanced by synthetically adding the minority class with a 50% increase. The second synthetic dataset (Resampled dataset 2) is big and synthetic instances are added with a 450% increase. In contrast, in the third dataset (Resampled dataset 3), the classes have been balanced by undersampling the majority class. The details of the original dataset are shown in Table 11(a), whereas the details of synthetic datasets for each fold are shown in Table 11(b).

We picked two preprocessing settings for this experiment: **E.01**: no preprocessing applied and **E.38**: fixed-length word segmentation of length 5, mainly because of the outperformance in other experiments. First, these two preprocessing settings are applied on Resampled dataset 1, 2 and 3, and then, the Sentence Match $F_1$ scores for each classifier are shown in Table 11(c)–(e) respectively. Similar to other experiments, a document wise 10-fold cross validation (see Section 4.5) is applied. To be more precise, the same training sets for E.01 and E.38 are taken from each fold, and SMOTE forms corresponding synthetic datasets. However, the test sets from the folds remain the same to have comparable results.

It is generally considered that a dataset prepared with oversampling can produce better classification results than the original dataset. However, the $F_1$ scores reported in Table 11(c)–(e), show that the synthetic datasets were unable to show a significant improvement. The $F_1$ scores with Resampled dataset 2 are especially mediocre as it was more expensive to train due to its larger size.The scores for E.38 are specifically not good.

## 6. Conclusion

Text summarization has attracted the increasing attention of researchers for more than three decades now. The lack of standardized resources with human-written summaries is a major obstacle in analyzing and developing text summarization systems for any language. This paper makes three key contributions to the advancement of Urdu text summarization research. Firstly, we have developed and publicly released first Urdu extractive summary corpus (CORPURES). It contains 161 documents with manually written extractive summaries from newswire domain. To develop a summary corpus, we applied a simpler yet robust crowdsourcing version, producing high-quality summaries. We hope that the released corpus provides the necessary basis for the future development and the evaluation of extractive text summarization techniques for an important low-resource language.

Secondly, we develop a custom-built supervised learning framework for Urdu extractive summarization. This framework employed six classifiers and evaluated their effectiveness for the task at hand. The highest F1 score is acehieved by Logistic Regression is 57% (ROUGE-2 $F_1$ = 64.4%) when employed fixed-length word segmentation of length 6. In fact, Logistic Regression performs best for most of the preprocessing settings in the experiments.

Thirdly, we have performed a number of detailed benchmark experiments on the custom-built supervised learning framework. In addition, the benchmark experiments evaluate the effectiveness of several settings on the task, such as *a*) the effectiveness of basic preprocessing settings such as stopword removal, lemmatization, stemming, similarity measures, *b*) the effectiveness of space segmentation, proper word segmentation and fixed-length segmentation, *c*) the role of each feature, and finally, *d*) the effectiveness of synthetic datasets. For Urdu summarization, this is a first-ever attempt to thoroughly evaluate the performance of supervised learning algorithms with an in-depth evaluation. For the reliability of results, document-wise 10-fold cross validation of experiments was performed. From the results of the experiments, we conclude that:

1. The classifiers such as Naive Bayes, Logistic Regression, and Multilayer Perceptron are good candidates when supervised learning techniques are employed.
2. Until the proper word segmentation tools for Urdu are mature enough, words tokenized on space is a reliable approach.
3. A radical stemming such as Ultra stemming with length equal to 1, 2 works better than the existing stemming and lemmatization tools for Urdu.
4. A radical word segmentation technique such as fixed-length outperform all other settings for L= 3 to 6.
5. The use of word embeddings as similarity measure does not improve results.
6. The artificially generated datasets do not show a significant difference as compared to the original data.
7. CORPURES is a small yet valuable resource for Urdu extractive summarization.

The future work includes building a large summary corpus with limited intervention and using deep learning techniques on such a corpus instead of traditional supervised learning techniques.

## CRediT authorship contribution statement

**Muhammad Humayoun:** Conceptualization, Methodology, Software, Writing – review & editing. **Naheed Akhtar:** Software, Writing – original draft, Data curation, Validation.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

We sincerely thank the anonymous reviewers for their insightful comments and suggestions.

## References

Akram, Q.-u.-A., Naseer, A., & Hussain, S. (2009). Assas-band, an affix-exception-list based urdu stemmer. *Proceedings of the 7th workshop on Asian language resources (ALR7)* (pp. 40–47). Association for Computational Linguistics.

Aone, C., Okurowski, M. E., & Gorlinsky, J. (1998). Trainable, scalable summarization using robust NLP and machine learning, *vol. 1. 36th annual meeting of the association for computational linguistics and 17th international conference on computational linguistics* (pp. 62–66). Montreal, Quebec, Canada: Association for Computational Linguistics. https://doi.org/10.3115/980845.980856

Arlot, S., & Celisse, A. (2010). A survey of cross-validation procedures for model selection. *Statistics Surveys, 4*, 40–79. https://doi.org/10.1214/09-SS054

Baker, P., & McEnery, A. (1999). Needs of language-engineering communities; corpus building and translation resources. *Technical Report*. Lancaster University. In Technical report, MILLE working paper 7

Baxendale, P. B. (1958). Machine-made index for technical literature: An experiment. *IBM Journal of Research and Development, 2*(4), 354–361. https://doi.org/10.1147/rd.24.0354

Becker, D., & Riaz, K. (2002). A study in urdu corpus construction, *vol. 12. In proceedings of the 3rd workshop on Asian language resources and international standardization* (pp. 1–5). Association for Computational Linguistics Association for Computational Linguistics.

Bhatti, M. W., & Aslam, M. (2019). ISUTD: Intelligent system for urdu text de-summarization. *2019 international conference on engineering and emerging technologies (ICEET)* (pp. 1–5). https://doi.org/10.1109/CEET1.2019.8711842

Bin Zia, H., Raza, A. A., & Athar, A. (2018). Urdu word segmentation using conditional random fields (CRFs). *Proceedings of the 27th international conference on computational linguistics* (pp. 2562–2569). Association for Computational Linguistics.http://aclweb.org/anthology/C18-1217

Borra, S., & Ciaccio, A. D. (2010). Measuring the prediction error. A comparison of cross-validation, bootstrap and covariance penalty methods. *Computational Statistics & Data Analysis, 54*(12), 2976–2989. https://doi.org/10.1016/j.csda.2010.03.004

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32. https://doi.org/10.1080/01969722.2015.1012892

Burney, S., Sami, B., Mahmood, N., Abbas, Z., & Rizwan, K. (2012). Urdu text summarizer using sentence weight algorithm for word processors. *International Journal of Computer Applications, 46*, 38–43.

le Cessie, S., & van Houwelingen, J. (1992). Ridge estimators in logistic regression. *Applied Statistics, 41*(1), 191–201.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*(1), 321–357.

Chen, D., Bolton, J., & Manning, C. D. (2016). A thorough examination of the CNN/daily mail reading comprehension task. *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 2358–2367). Berlin, Germany: Association for Computational Linguistics. https://doi.org/10.18653/v1/P16-1223

Chopra, S., Auli, M., & Rush, A. M. (2016). Abstractive sentence summarization with attentive recurrent neural networks. *Proceedings of the 2016 conference of the north American chapter of the association for computational linguistics: Human language technologies* (pp. 93–98). San Diego, California: Association for Computational Linguistics. https://doi.org/10.18653/v1/N16-1012

Chuang, W. T., & Yang, J. (2000). Extracting sentence segments for text summarization: A machine learning approach. *Proceedings of the 23rd annual international ACM SIGIR conference on research and development in information retrieval, SIGIR '00* (pp. 152–159). New York, NY, USA: ACM. https://doi.org/10.1145/345508.345566

Collins, E., Augenstein, I., & Riedel, S. (2017). A supervised approach to extractive summarisation of scientific papers. CoRR, abs/1706.03946http://arxiv.org/abs/1706.03946.

Dang, H., & Owczarzak, K. (2009). Overview of the tac 2008 update summarization task. https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=903465.

Das, D., & Martins, A. (2007). A survey on automatic text summarization. *Language, 4*, 1–31.

Davis, M., & Whistler, K. (2014). Unicode normalization forms, unicode standard annex #15. *Technical Report.* The Unicode Consortium.

Durrani, N., & Hussain, S. (2010). Urdu word segmentation. *Human language technologies: Conference of the North American chapter of the association of computational linguistics, proceedings, June 2–4, 2010, Los Angeles, California, USA* (pp. 528–536).

Edmundson, H. P. (1969). New methods in automatic extracting. *Journal of ACM, 16*(2), 264–285.

El-Haj, M., Kruschwitz, U., & Fox, C. (2015). Creating language resources for under-resourced languages: Methodologies, and experiments with arabic. *Language Resources and Evaluation, 49*(3), 549–580. https://doi.org/10.1007/s10579-014-9274-3

El-Shishtawy, T., & El-Ghannam, F. (2012). Keyphrase based arabic summarizer (KPAS). CoRR, abs/1206.5384http://arxiv.org/abs/1206.5384.

Elshakankery, K., & Farouk, M. (2019). Hilatsa: A hybrid incremental learning approach for arabic tweets sentiment analysis. *Egyptian Informatics Journal*. https://doi.org/10.1016/j.eij.2019.03.002

Erkan, G., & Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial intelligence research (JAIR), 22*(1), 457–479.

Fang, C., Mu, D., Deng, Z., & Wu, Z. (2016). Word-sentence co-ranking for automatic extractive text summarization. *Expert Systems with Applications, 72*. https://doi.org/10.1016/j.eswa.2016.12.021

Farooq, A., Batool, S., & Noreen, Z. (2021). Comparing different techniques of urdu text summarization. *2021 Mohammad Ali Jinnah university international conference on computing (MAJICC)* (pp. 1–6). https://doi.org/10.1109/MAJICC53071.2021.9526246

Fattah, M., & Ren, F. (2008). Automatic text summarization. *Proceedings of World Academy of Science, Engineering and Technology, 27*, 192–195. https://doi.org/10.1016/j.csl.2008.04.002

Fattah, M. A. (2014). A hybrid machine learning model for multi-document summarization. *Applied Intelligence, 40*(4), 592–600. https://doi.org/10.1007/s10489-013-0490-0

Fillmore, C. J. (1982). *Frame semantics* (pp. 111–137)). Seoul, South Korea: Hanshin Publishing Co..

Garabk, R. (2006). Slovak morphology analyzer based on levenshtein edit operations, *vol. 62. In proceedings of the WIKT'06 conference* (pp. 2–5). Walter de Gruyter GmbH. https://doi.org/10.2478/v10113-011-0002-x

Gillick, D., & Favre, B. (2009). A scalable global model for summarization. *Proceedings of the workshop on integer linear programming for natural language processing* (pp. 10–18). Boulder, Colorado: Association for Computational Linguistics.https://www.aclweb.org/anthology/W09-1802

Gillick, D., & Liu, Y. (2010). Non-expert evaluation of summarization systems is risky. *Proceedings of the NAACL HLT 2010 workshop on creating speech and language data with Amazon's mechanical turk* (pp. 148–151). Los Angeles: Association for Computational Linguistics.https://aclanthology.org/W10-0722

Grimes, B. F. (2021). *Ethnologue: Languages of the world* (24th ed.). Ethnologue.

Grusky, M., Naaman, M., & Artzi, Y. (2018). Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. *Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: Human language technologies, volume 1 (long papers)* (pp. 708–719). New Orleans, Louisiana: Association for Computational Linguistics. https://doi.org/10.18653/v1/N18-1065

Gulzar, A. (2007). Urdu normalization utility v1.0. *Technical Report.* Center for Language Engineering, Al-kwarzimi Institute of Computer Science (KICS), University of Engineering, Lahore, Pakistan. http://www.cle.org.pk/software/langproc/urdunormalization.htm

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: An update. *SIGKDD Explorations Newsletter, 11*(1), 10–18. https://doi.org/10.1145/1656274.1656278

Hall, M. A. (1998). *Correlation-based feature subset selection for machine learning.* University of Waikato, Hamilton, New Zealand. Ph.D. thesis.

Hammarström, H., & Borin, L. (2011). Unsupervised learning of morphology. *Computational Linguistics, 37*(2), 309–350. https://doi.org/10.1162/COLI_a_00050

Harman, D., & Over, P. (2004). The effects of human variation in DUC summarization evaluation. *Text summarization branches out* (pp. 10–17). Barcelona, Spain: Association for Computational Linguistics.https://www.aclweb.org/anthology/W04-1003

Hassel, M., & Mazdak, N. (2004). Farsisum - a persian text summarizer. *Proceedings of the workshop on computational approaches to Arabic script-based languages* (pp. 82–84). Geneva, Switzerland: COLING.https://aclanthology.org/W04-1615

Hermann, K. M., Kociský, T., Grefenstette, E., Espeholt, L., Kay, W., & Suleyman, M. et al. (2015). Teaching machines to read and comprehend. CoRR, abs/1506.03340http://arxiv.org/abs/1506.03340.

Hu, B., Chen, Q., & Zhu, F. (2015). LCSTS: A large scale Chinese short text summarization dataset. *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 1967–1972). Lisbon, Portugal: Association for Computational Linguistics. https://doi.org/10.18653/v1/D15-1229

Humayoun, M. (2021). The 2021 Urdu fake news detection task using supervised machine learning and feature combinations. *Forum for information retrieval evaluation, December 13–17, 2021, India (fire 2021). Cicling 2021 track. International conference on computational linguistics and intelligent text processing. http://ceur-ws.org/Vol-3159/* (pp. 1156–1161). Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation.

Humayoun, M., Hammarström, H., & Ranta, A. (2007a). Implementing urdu grammar as open source software. *Conference on language and technology, university of Peshawar*.

Humayoun, M., Hammarström, H., & Ranta, A. (2007b). *Urdu morphology, orthography and lexicon extraction. CAASL-2: The second workshop on computational approaches to arabic script-based languages, LSA linguistic institute* (pp. 21–22). California, USA: Stanford university.https://arxiv.org/abs/2204.03071

Humayoun, M., Nawab, R. M. A., Uzair, M., Aslam, S., & Farzand, O. (2016). Urdu Summary Corpus. *Proceedings of the tenth international conference on language resources and evaluation (LREC 2016)* (pp. 796–800). Portorož, Slovenia: European Language Resources Association (ELRA).https://www.aclweb.org/anthology/L16-1128

Humayoun, M., & Yu, H. (2016). Analyzing pre-processing settings for urdu single-document extractive summarization. *Proceedings of the tenth international conference on language resources and evaluation (LREC 2016)* (pp. 3686–3693). Portorož, Slovenia: European Language Resources Association (ELRA).

Hussain, S., & Afzal, M. (2001). S.: Urdu computing standards: Urdu zabta takhti (uzt) 1.01. *In proceedings of the IEEE INMIC, Lahore, Pakistan*.

Jawaid, B., Kamran, A., & Bojar, O. (2014). A tagged corpus and a tagger for urdu. In N. C. C. Chair), K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, & S. Piperidis (Eds.), *Proceedings of the ninth international conference on language resources and evaluation (LREC'14)*. Reykjavik, Iceland: European Language Resources Association (ELRA).

Jing, H., Barzilay, R., McKeown, K., & Elhadad, M. (2000). Summarization evaluation methods: Experiments and analysis. AAAI symposium on intelligent summarization.

John, G. H., & Langley, P. (1995). Estimating continuous distributions in Bayesian classifiers. *Eleventh conference on uncertainty in artificial intelligence* (pp. 338–345). San Mateo: Morgan Kaufmann.

Kaikhah, K. (2004). Automatic text summarization with neural networks, *vol. 1. 2nd international ieee conference on 'intelligent systems'. proceedings (IEEE cat. no.04ex791)* (pp. 40–44). https://doi.org/10.1109/IS.2004.1344634Print isbn: 0-7803-8278-1

Kianmehr, K., Gao, S., Attari, J., Rahman, M. M., Akomeah, K., Alhajj, R., et al. (2009). Text summarization techniques: SVM versus neural networks. *iiwas'2009 - the eleventh international conference on information integration and web-based applications and services, 14–16 December 2009, Kuala Lumpur, Malaysia* (pp. 487–491). https://doi.org/10.1145/1806338.1806429

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection, . In *IJCAI'95: vol. 2. Proceedings of the 14th international joint conference on artificial intelligence* (pp. 1137–1143). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc..http://dl.acm.org/citation.cfm?id=1643031.1643047

Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence, 97*(1–2), 273–324.Special issue on relevance

Kryscinski, W., McCann, B., Xiong, C., & Socher, R. (2020). Evaluating the factual consistency of abstractive text summarization. *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)* (pp. 9332–9346). Online: Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.750

Kupiec, J., Pedersen, J., & Chen, F. (1995). A trainable document summarizer. *Proceedings of the 18th annual international ACM SIGIR conference on research and development in information retrieval, SIGIR '95* (pp. 68–73). New York, NY, USA: ACM. https://doi.org/10.1145/215206.215333

Kutlu, M., Cigir, C., & Cicekli, I. (2010). Generic text summarization for turkish. *The Computer Journal, 53*(8), 1315–1323. https://doi.org/10.1093/comjnl/bxp124

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Cybernetics and Control Theory, 10*(8), 707–710.Original in *Doklady Akademii Nauk SSSR* 163(4): 845–848 (1965).

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In S. S. Marie-Francine Moens (Ed.), *Text summarization branches out: Proceedings of the ACL-04 workshop* (pp. 74–81). Barcelona, Spain: Association for Computational Linguistics.

Liu, F., & Liu, Y. (2008). Correlation between ROUGE and human evaluation of extractive meeting summaries. *Proceedings of the 46th annual meeting of the association for computational linguistics on human language technologies: Short papers, HLT-Short '08* (pp. 201–204). Stroudsburg, PA, USA: Association for Computational Linguistics.

Lo, R. T., He, B., & Ounis, I. (2005). Automatically building a stopword list for an information retrieval system. *Journal of Digital Information Management (JDIM), 3*(1), 3–8.http://www.dirf.org/jdim/abstractv3i1.htm#01

Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of Research and Development, 2*(2), 159–165.

Lyras, D. P., Sgarbas, K. N., & Fakotakis, N. D. (2007). Using the levenshtein edit distance for automatic lemmatization: A case study for modern Greek and English, *vol. 2. 19th IEEE international conference on tools with artificial intelligence(ICTAI 2007)* (pp. 428–435). https://doi.org/10.1109/ICTAI.2007.41

Marcus, M. P., Marcinkiewicz, M. A., & Santorini, B. (1993). Building a large annotated corpus of English: The penn treebank. *Computational Linguistics, 19*(2), 313–330.http://dl.acm.org/citation.cfm?id=972470.972475

Mayfield, J., & McNamee, P. (2003). Single n-gram stemming. *Proceedings of the 26th annual international ACM SIGIR conference on research and development in informaion retrieval, SIGIR '03* (pp. 415–416). New York, NY, USA: ACM. https://doi.org/10.1145/860435.860528

McEnery, T., Baker, P., & Burnard, L. (2000). Corpus resources and minority language engineering. *Proceedings of the second international conference on language resources and evaluation (LREC'00)*. Athens, Greece: European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2000/pdf/187.pdf

Mihalcea, R., & Tarau, P. (2004). Textrank: Bringing order into texts. *Proceedings of EMNLP-04 and the 2004 conference on empirical methods in natural language processing*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Q. Weinberger (Eds.), *vol. 52. Advances in neural information processing systems 26* (pp. 3111–3119). Curran Associates, Inc.. https://doi.org/10.1111/jssr.12042

Miller, G. A. (1995). Wordnet: A lexical database for english. *Communications of the ACM, 38*(11), 39–41. https://doi.org/10.1145/219717.219748

Mitra, M., Singhal, A., & Buckley, C. (1997). Automatic text summarization by paragraph extraction. *Intelligent scalable text summarization*.https://aclanthology.org/W97-0707

Mohd, M., Jan, R., & Shah, M. (2020). Text document summarization using word embedding. *Expert Systems with Applications, 143*, 112958. https://doi.org/10.1016/j.eswa.2019.112958

Moratanch, N., & Chitrakala, S. (2017). A survey on extractive text summarization. *2017 international conference on computer, communication and signal processing (ICCCSP)* (pp. 1–6). https://doi.org/10.1109/ICCCSP.2017.7944061

Muhammad, A., Jazeb, N., Martinez-Enriquez, A., & Sikander, A. (2018). EUTS: Extractive urdu text summarizer. *2018 seventeenth mexican international conference on artificial intelligence (MICAI)* (pp. 39–44). Los Alamitos, CA, USA: IEEE Computer Society. https://doi.org/10.1109/MICAI46078.2018.00014

Mukund, S., Srihari, R., & Peterson, E. (2010). An information-extraction system for urdu—A resource-poor language. *ACM Transactions on Asian Language Information Processing, 9*(4). https://doi.org/10.1145/1838751.1838754

Murray, G., Renals, S., & Carletta, J. (2005). Extractive summarization of meeting recordings. *Interspeech 2005 - eurospeech, 9th european conference on speech communication and technology*.

Najibullah, A. (2015). Indonesian text summarization based on Naïve Bayes method. *Proceeding of the international seminar and conference 2015: The golden triangle (Indonesia-India-Tiongkok) interrelations in religion, science, culture, and economic, Semarang, Indonesia* (p. 12).

Nallapati, R., Zhai, F., & Zhou, B. (2016a). Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. CoRR, abs/1611.04230http://arxiv.org/abs/1611.04230.

Nallapati, R., Zhai, F., & Zhou, B. (2017). Summarunner: A recurrent neural network based sequence model for extractive summarization of documents, . In *AAAI'17Proceedings of the thirty-first aaai conference on artificial intelligence* (pp. 3075–3081). AAAI Press.

Nallapati, R., Zhou, B., & Ma, M. (2016b). Classify or select: Neural architectures for extractive document summarization. CoRR, abs/1611.04244http://arxiv.org/abs/1611.04244.

Narayan, S., Papasarantopoulos, N., Lapata, M., & Cohen, S. B. (2017). Neural extractive summarization with side information. CoRR, abs/1704.04530http://arxiv.org/abs/1704.04530.

Nawaz, A., Bakhtyar, M., Baber, J., Ullah, I., Noor, W., & Basit, A. (2020). Extractive text summarization models for urdu language. *Information Processing & Management, 57* (6), Article 102383. https://doi.org/10.1016/j.ipm.2020.102383

Nenkova, A., & McKeown, K. (2012). A survey of text summarization techniques. In C. C. Aggarwal, C. Zhai, & blubberdiblubb (Eds.), *Mining text data* (pp. 43–76). Springer.

Neto, J. L., Freitas, A. A., & Kaestner, C. A. A. (2002). Automatic text summarization using a machine learning approach. In G. Bittencourt, & G. L. Ramalho (Eds.), *Advances in artificial intelligence* (pp. 205–215). Berlin, Heidelberg: Springer Berlin Heidelberg.

Noor, F., Bakhtyar, M., & Baber, J. (2019). Sentiment analysis in E-commerce using SVM on roman urdu text. In M. H. Miraz, P. S. Excell, A. Ware, S. Soomro, & M. Ali (Eds.), *Emerging technologies in computing* (pp. 213–222). Cham: Springer International Publishing.

Palmer, M., Gildea, D., & Kingsbury, P. (2005). The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics, 31*(1), 71–106. https://doi.org/10.1162/0891201053630264

Parthasarathy, S., & Hasan, T. (2015). Automatic broadcast news summarization via rank classifiers and crowdsourced annotation. 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP), (pp. 5256–5260).

Platt, J. (1998). Fast training of support vector machines using sequential minimal optimization. In B. Schoelkopf, C. Burges, & A. Smola (Eds.), *Advances in kernel methods - support vector learning*. MIT Press. http://research.microsoft.com/~jplatt/smo.html

Qaroush, A., Abu Farha, I., Ghanem, W., Washha, M., & Maali, E. (2019). An efficient single document arabic text summarization using a combination of statistical and semantic features. *Journal of King Saud University - Computer and Information Sciences*. https://doi.org/10.1016/j.jksuci.2019.03.010

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Radev, D. R., Hovy, E., & McKeown, K. (2002). Introduction to the special issue on summarization. *Computational Linguistics, 28*(4), 399–408.

Rakesh, C., Sahoo, D., Sahoo, B., & Swain, M. (2012). Text summarization using term weights. *International Journal of Computer Applications, 38*, 10–14. https://doi.org/10.5120/4570-6731

Rath, G. J., Resnick, A., & Savage, T. R. (1961). *The formation of abstracts by the selection of sentences. Part I. Sentence selection by men and machines*. American Documentation, Wiley Online Library.

Saeed, A., Nawab, R., Stevenson, M., & Rayson, P. (2018). A word sense disambiguation corpus for urdu. *Language Resources and Evaluation, 53*. https://doi.org/10.1007/s10579-018-9438-7

Saeed, A., Nawab, R., Stevenson, R., & Rayson, P. (2019). A sense annotated corpus for all-words urdu word sense disambiguation. *ACM Transactions on Asian and Low-Resource Language Information Processing, 18*. https://doi.org/10.1145/3314940

Sandhaus, E. (2008). The new york times annotated corpus. Linguistic Data Consortium. https://catalog.ldc.upenn.edu/LDC2008T19.

Sharjeel, M., Nawab, R., & Rayson, P. (2016). Counter: Corpus of urdu news text reuse. *Language Resources and Evaluation, 51*. https://doi.org/10.1007/s10579-016-9367-2

Steinberger, J., & Jezek, K. (2004). Using latent semantic analysis in text summarization and summary evaluation. *Proceedings of the 5th international conference on information systems implementation and modelling. Marq Ostrava, isbn 80-85988-99-2.* (pp. 93–100).

Steinberger, J., & Ježek, K. (2009). Update summarization based on latent semantic analysis. In V. Matoušek, & P. Mautner (Eds.), *Text, speech and dialogue* (pp. 77–84). Berlin, Heidelberg: Springer Berlin Heidelberg.

Steinberger, J., & Ježek, K. (2012). Evaluation measures for text summarization. *Computing and Informatics, 28*(2), 251–275.

Torres-Moreno, J. (2012). Beyond stemming and lemmatization: Ultra-stemming to improve automatic text summarization. *ACM Computing Research Repository (CoRR)*. abs/1209.3126

Urooj, S., Hussain, S., Adeeba, F., Jabeen, F., & Perveen, R. (2012). Cle urdu digest corpus. *Conference on language and technology, Lahore, Pakistan*.

Üstün, B., Melssen, W. J., & Buydens, L. M. C. (2006). Facilitating the application of support vector regression by using a universal Pearson VII function based kernel, *vol. 81. Chemometrics and intelligent laboratory systems* (pp. 29–40). https://doi.org/10.1016/j.chemolab.2005.09.003

Verberne, S., Krahmer, E., Hendrickx, I., Wubben, S., & van den Bosch, A. (2018). Creating a reference data set for the summarization of discussion forum threads. *Language Resources and Evaluation, 52*, 461–483. https://doi.org/10.1007/s10579-017-9389-4

Verma, P., Pal, S., & Om, H. (2019). A comparative analysis on hindi and english extractive text summarization. *ACM Transactions on Asian and Low-Resource Language Information Processing, 18*, 1–39. https://doi.org/10.1145/3308754

Virk, S. M., Humayoun, M., & Ranta, A. (2010). An open source urdu resource grammar. *Proceedings of the eighth workshop on Asian language resouces* (pp. 153–160). Beijing, China: Coling 2010 Organizing Committee.https://aclanthology.org/W10-3220

Widyassari, A. P., Rustad, S., Shidik, G. F., Noersasongko, E., Syukur, A., Affandy, A., et al. (2020). Review of automatic text summarization techniques & methods. *Journal of King Saud University - Computer and Information Sciences*. https://doi.org/10.1016/j.jksuci.2020.05.006

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data mining, fourth edition: Practical machine learning tools and techniques* (4th ed.). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Xu, J., & Croft, W. B. (1998). Corpus-based stemming using cooccurrence of word variants. *ACM Transactions on Information Systems, 16*(1), 61–81. https://doi.org/10.1145/267954.267957

Ye, S., Chua, T.-S., Kan, M.-Y., & Qiu, L. (2007). Document concept lattice for text understanding and summarization. *Information Processing and Management, 43*(6), 1643–1662.

Yeh, J.-Y., Ke, H.-R., & Yang, W.-P. (2008). Ispreadrank: Ranking sentences for extraction-based summarization using feature weight propagation in the sentence similarity network. *Expert Systems with Applications, 35*, 711–715. https://doi.org/10.1007/978-981-13-1217-5_71