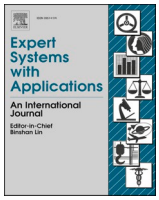
Expert Systems With Applications 237 (2024) 121433

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



TxLASM: A novel language agnostic summarization model for text documents

Ahmed Abdelfattah Saleh [a,*], Li Weigang [b]

[a] *PPMEC, Department of Mechanical Engineering, University of Brasilia, Brasilia, Brazil*
[b] *TransLab, Department of Computer Science, University of Brasilia, Brasilia, Brazil*

ARTICLE INFO

Keywords:
Language agnostic summarization
Extractive summarization model
Domain agnostic summarization
TxLASM
Text shape-encoding

ABSTRACT

In Natural Language Processing (NLP) domain, the majority of automatic text summarization approaches depend on a prior knowledge of the language and/or the domain of the text being summarized. Such approaches requires language dependent part-of-speech taggers, parsers, databases, pre-structured lexicons, etc. In this research, we propose a novel automatic text summarization model, *Text Documents - Language Agnostic Summarization Model* (TxLASM), which is able to perform extractive text summarization task in language/domain agnostic manner. *TxLASM* depends on specific characteristics of the major elements of the text being summarized rather than its domain, context, or language and thus rules out the need for language dependent pre-processing tools, taggers, parsers, lexicons or databases. Within *TxLASM*, we present a novel technique for encoding the shapes of major text elements (paragraphs, sentences, n-grams and words); moreover, we present language independent pre-processing algorithms to normalize words and perform relative stemming or lemmatization. Those algorithms and its *Shape-Coding* technique enable the *TxLASM* to extract intrinsic features of text elements and score them statistically, and subsequently extract a representative summary that is independent of the text language, domain and context. *TxLASM* was applied on an English and Portuguese benchmark datasets, and the results were compared to twelve state-of-the-art approaches presented in recent literature. In addition, the model was applied on French and Spanish news datasets, and the results were compared to those obtained by standard commercial summarization tools. *TxLASM* has outperformed all the SOTA approaches as well as the commercial tools in all four languages while maintaining its language and domain agnostic nature.

1. Introduction

The rapid development of the Internet and the massive exponential growth in web textual data has brought considerable challenges to tasks related to text management, classification and information retrieval. As such, Automatic text summarization (ATS) is becoming an extremely important means to solve this problem. ATS tends to mine the gist of the original text and then automatically generate a concise and readable summary that reflects the core important information in that text. Therefore, developing an efficient text summarization model is essential for information retrieval, knowledge inference, text processing, and dimensionality reduction for subsequent classification and understanding.

With the recent advances in computation, Natural Language Processing (NLP) field is gaining great advantage from adopting models and methodologies from Artificial Intelligence. In this study, we focus on developing language agnostic summarization model, aiming at improving the generalization performance in fields of NLP, by proposing a series of domain and language agnostic tools.

The present description of sentence processing in human cortex differentiates three linguistic processing phases (Friederici, 2002). The first processing phase is based on word category information on the sentence level. While the second phase computes the syntactic as well as the semantic relations in the sentence, which involves detecting of the relations between a verb and its arguments, and the subsequent assignment of thematic roles. Those steps lead to the third phase of compatible interpretation and comprehension (Friederici, 2011). Hence, and in order to achieve an efficient summarization task for a written text, first relevant words and sentences should be extracted and then related to topic comprehension or context in order to get a human like understanding. Words themselves should be categorized into stop words, *Named Entities* (nouns, concrete concepts, etc.) as well as verbs,

* Corresponding author.
E-mail address: ahmdsalh@gmail.com (A. Abdelfattah Saleh).

https://doi.org/10.1016/j.eswa.2023.121433
Received 19 June 2023; Received in revised form 10 August 2023; Accepted 1 September 2023
Available online 9 September 2023

prepositions, etc.

As such, the prior knowledge of language and/or domain of the text being summarized is a critical requirement by most ATS models. This can be limiting in cases where the language or domain of the text is unknown or rapidly changing.

In this research, we aim to achieve similar level of textual understanding in a language agnostic manner, avoiding the need to extract verbs, nouns or other syntactical relations that require a prior knowledge of language and/or its context. Rather, we extract prominent phrases to form an extractive summary using a novel and totally language/domain agnostic tools.

As will be discussed in details in the following section (Literature Review and Related Work), ATS can be performed using a broad range of approaches and techniques. The vast majority of which depend on pre-structured lexicons, databases, part of speech taggers and parsers, which are language dependent. In other words, such approaches require a former knowledge of the language of the text to be summarized and in some situations, its contextual domain. Such prerequisite might affect the generalization performance of the model in case it faces new language and/or domain. Moreover, efficient part of speech tagger or parsers are not always available for particular languages, in addition to the fact that that lexicons are mostly contextual, therefore, preparing and refining domain specific lexicons for all languages is considered a big challenge among linguistic researchers.

Apart from the language dependence of preprocessing tools and the contextual dependence of lexicons, obtaining an efficient representative summary might also require extracting or identifying Named Entities (NE's) and Concrete Concepts (CC's) due to their influence on the summarization quality. Such task is by nature heavily dependent on prior detection of the language and/or the context of the text to be summarized.

As such, the main objective of this research is to propose a model capable of performing efficient extractive text summarization in a language and domain independent manner. Therefore, we propose a novel extractive text summarization model, ***Text Documents - Language Agnostic Summarization Model*** (***TxLASM***), which is capable of performing extractive text summarization in a completely language and domain agnostic manner, and subsequently avoid the need of preparing language/domain specific tools and/or corpora.

The proposed model depends on specific characteristics of the major elements of the text being summarized rather than its domain, context, or language and thus rules out the need for language dependent preprocessing tools, taggers, parsers, lexicons or databases. Within *TxLASM*, we present a novel technique for encoding the shapes of major text elements (paragraphs, sentences, n-grams and words); moreover, we present language independent preprocessing algorithms to normalize words and perform relative stemming or lemmatization. Those algorithms and its *Shape-Coding* technique enable the *TxLASM* to extract intrinsic features of major text elements, score them statistically, and identify influential tokens (NE's and CC's) to extract a representative summary independent of the text language and/or its contextual domain.

In summary, the main contributions of this study are as follows: a) we propose a straightforward, yet efficient, Language and Domain Agnostic Summarization Model for Text Documents, named "*TxLASM*". b) *TxLASM* is an entirely unsupervised model, in terms of extracting influential tokens as NE's and CC's. c) We developed a novel *shape-coding technique* that encodes document elements into handful classes of distinct shapes, which in turn reflects their importance and influence on the generated summary. Moreover, d) we developed language agnostic pre-processing algorithms for stemming and stop words removal.

The rest of the paper is structured as follows, a literature review on ATS techniques and challenges is presented in section 2, and then we propose *TxLASM* in section 3, followed by section 4 that states the applied experiment, whose results are discussed in section 5. Finally, we conclude the paper and propose the future work in section 6.

2. Literature review and related work

Automatic Text Summarization (ATS) can be divided into three main approaches, *Extractive*, acts on extracting the most influential sentences of the text to be summarized (Rahimi, Mozhdehi, & Abdolahi, 2017); *Abstractive* depends on semantics to create new representative sentences made of new set of words (Alomari, Idris, Sabri, & Alsmadi, 2022); and a *Hybrid* approach (Hsu, et al., 2018).

Another way to look at the ATS is by considering the dimensionality of the text to be summarized. ATS could be applied for single document summarization, or multiple document summarization, which typically involves summarizing a set of documents belonging to the same topic while maintaining the relevancy and avoiding redundancy (Tomer & Kumar, 2022).

From the architecture viewpoint, El-Kassas, Salama, Rafea, & Mohamed (2021) has divided ATS into three distinct steps, *Pre-processing*, *Processing* and *Post-processing* as per Fig. 1. Where, pre-processing step (Smelyakov, et al., 2020) includes segmentation of sentences, tokenization, stemming, lemmatization (Bergmanis & Goldwater, 2018), tagging (Warjri, Pakray, Lyngdoh, & Maji, 2021), stop words removal (Kaur & Buttar, 2018), etc. while the processing step means applying the summarization technique itself, finally, the post-processing step focuses on refining the summary by solving problems and facing challenges. On the other hand, a generalized framework for abstractive ATS was also developed based on neural networks.

2.1. ATS preprocessing tools

Language summarization algorithms typically depends on feature extraction techniques, as stop words removal, stemming, lemmatization, POS tagging, etc. Such techniques are language dependent in nature, which requires the presence of lexicons, parsers and other language specific tools.

Stop words, for instance, are common words that are neither indexed nor searchable in search engine (Ladani & Desai, 2020), as in English languages, words like "is", "the", "in", and others, also, the words "في","كل","و"", etc. in Arabic language (Namly, Bouzoubaa, & Yousfi, 2019). Stop words impose noise to NLP models as such, their removal enhance the performance of NLP models significantly.

On the other hand, stemming was introduced by (Lovins, 1968), then it was developed through the years, and many algorithms have been developed for specific languages, as Nazief & Adriani stemmer for Indonesian language (Jumadi, Maylawati, Pratiwi, & Ramdhani, 2021), improved Arabic light-based stemmer (Alshalabi, Tiun, Omar, AL-Aswadi, & Alezabi, 2022), in addition to various specialized language dependent lemmatizers (Gupta & Jivani, 2022).

In addition, Part-of-Speech (POS) Tagging (Voutilainen, 2003), which is the process of annotation of tokens in a text, where a word is assigned to a speech class (noun, verb, subject, etc.), has gained growing attention and were implemented in various languages across the globe. As a language-dependent process, recent literature shows that intense work has been done for POS tagging of different languages using wide range of machine learning and deep learning models. For example, Bidirectional Encoder Representations from Transformers (BERT) model was used to build POS taggers for Arabic (Saidi, Jarray, & Mansour, 2021), Croatian (Vasić, et al., 2021) and even for ancient languages as Ancient and Byzantine Greek (Singh, Rutten, & Lefever, 2021). Moreover, POS taggers were built for indigenous languages as Khasi language spoken by indigenous people of the state of Meghalaya in India, where Conditional Random Field (CRF) method was used to build a Khasi POS Tagger (Warjri, Pakray, Lyngdoh, & Maji, 2021).

As deep learning techniques advances, many recent studies in literature have used supervised deep learning models to build language dependent POS taggers (Chiche & Yitagesu, 2022). For example, (Bahcevan, Kutlu, & Yildiz, 2018) has used deep learning networks, recurrent (RNN) and long-short term memory (LSTM) neural networks, to build a

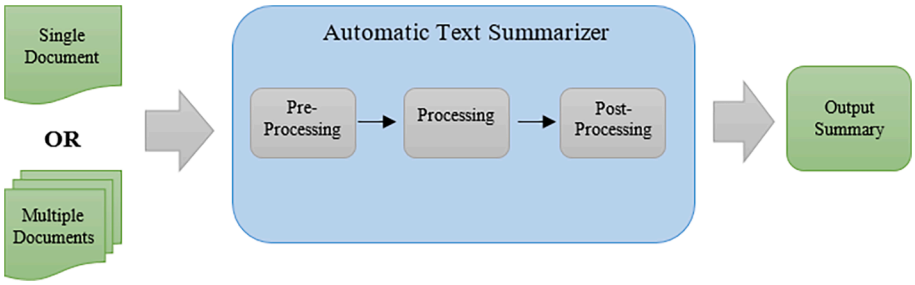


Fig. 1. Generalized architecture for automatic text summarizer for a single document or multiple documents (El-Kassas, Salama, Rafea, & Mohamed, 2021).

POS tagger for Turkish language, while (Rajani Shree & Shambhavi, 2021) has used RNN and LSTM as well to build POS tagger for one of the south Indian languages (Kannada). Moreover, many POS taggers were built based on deep learning models for local languages, as Malayalam (south Indian language) (Junaida & Babu, 2021), Maithili (Priyadarshi, 2022) in addition to national languages as Kazakh (Serek, Issabek, Akhmetov, & Sattarbek, 2021), Persian (Besharati, Veisi, Darzi, & Saravani, 2021), Thai (Chotirat & Meesad, 2021), and Mongolian (Lkhagvasuren, Rentsendorj, & Namsrai, 2021).

Semi-unsupervised approaches, as well, have benefited from deep learning and its deep neural networks to build POS taggers that can handle rare words, and out-of-vocabulary tokens (Alshemali & Kalita, 2020). For example, (Pota, Marulli, Esposito, De Pietro, & Fujita, 2019) has used semi-unsupervised deep learning based on word embedding representation to build POS taggers for Italian and English language.

2.2. Challenges in ATS

In general, ATS frameworks, whether extractive, abstractive or hybrid, are less biased and faster in processing than manually generated summarizes due to human bias. However, ATS has its own set of challenges as: a) Minimizing Redundancy, b) Maintaining diversity of topics in hybrid texts, as well as c) generating human readable summary (especially in abstractive ATS), and d) the challenge of Out-of-Vocabulary words (OOV) and repetition.

Many attempts were made to tackle those challenges. Kouris, Alexandridis, and Stafylopatis (2022) have proposed a framework for human readable abstractive summarization using knowledge-based content generalization and deep learning networks. Moreover, many deep learning approaches especially Long-Short Term Memory (LSTM) have been used to reduce redundancy while maintaining a readable human summary (Suleiman & Awajan, 2020), even in complex languages as Arabic language due to its high semantics, syntactical complexity and enormous word derivatives (Wazery, Saleh, Alharbi, & Ali, 2022). In general, common deep neural networks (DNNs) as recurrent neural networks (RNN), convolutional neural network (CNN), and graph neural network (GNN) are widely used in abstractive summarization to tackle some of the challenges mentioned above (Zhang, Zhou, Yu, Huang, & Liu, 2022).

Many frameworks, in the deep learning domain of abstractive text summarization, are used to tackle the challenge of understanding the text and generating human readable summaries. Such frameworks include, sequence-to-sequence framework (Cai, Shen, Peng, Jiang, & Dai, 2019; Dong, Shan, Liu, Qian, & Ma, 2021), as well as other encoder-decoder models, as encoder-decoder with basic attention mechanism (Chopra, Auli, & Rush, 2016; Qu, Lu, Wang, Yang, & Chen, 2022), Hierarchical Encoder-Decoder Models (Qi, Liu, Fu, & Liu, 2021), and CNN-Based Encoder-Decoder Models (Kumar & Gupta, 2021).

On the other hand, OOV words and repetition problems are handled in the abstractive summarization tasks through mixed approaches in the deep learning domain. Xu, Xiong, and Cheng (2021) have integrated core word information of the original vocabulary with the traditional attention mechanism to create FCWAM model, stands for Fusion Core Word Attention Mechanism Model, to tackle that problem. Others created datasets specific for particular languages as (Baykara & Güngör, 2022) did for Turkish and Hungarian languages.

The aforementioned techniques has somehow tackled the challenges related to structure and readability, however there is an important challenge that has not been addressed appropriately, which is the challenge of *topic bias*. Where the summary could be biased towards specific subtopics within a document (especially long ones) or in a set of grouped documents (multi-document summarization).

This tradeoff between readability and bias is more prominent in abstractive text summarization and to a lesser extend in extractive summarization. Recent work in literature has faced this problem thorough introducing an unsupervised component in the summarization model. In extractive summarization task, (Rani, 2021), (Zou, et al., 2021) and (Issam & Patel, 2020) have used topic modeling. While, (Ma, Zhang, Guo, Wang, & Sheng, 2020), (Alguliyev, Aliguliyev, Isazade, Abdi, & Idris, 2019) and (Akter, et al., 2017) have used unsupervised clustering, and (Alami, Meknassi, En-nahnahi, El Adlouni, & Ammor, 2021) has used a combination of both approaches to achieve a proper unbiased summaries.

2.3. Categories of extractive ATS

Various extractive summaries approaches were introduced in literature in the recent years, Gambhir and Gupta (2017) have divided extractive summarization approaches into five main categories according to the approach used in achieving the ATS task. Those categories are, a) statistical based, b) topic based, c) graph based, d) discourse based and e) machine learning based.

It is worth mentioning that most of these approaches are language dependent as they depend in one or more steps on a language dependent tool (taggers, lemmatizer, stemmers, etc.). Hereafter, we present the recent models introduced in literature in each of these five categories of extractive ATS.

2.3.1. Statistical based approaches

The extracted sentence depends on statistical features of the sentence itself and its containing document rather than its linguistic properties. However, those statistical methods might depend on one or more language dependent tool in the preprocessing steps, as taggers, parsers, lexicons, etc. Many statistical methods have been used for document element scoring, and the subsequent sentence/element selection and extraction (Deshpande & Kottawar, 2021).

Zhou et al. (Zhou, et al., 2018) have integrated the sentence selection and scoring routines into a single end-to-end neural network framework for extractive document summarization using hierarchical encoder.

Some methods utilize single word statistics while others utilize n-grams and other complex combinations of tokens. (Kumar & Rani, n.d.) used word frequency algorithms to extract the main features from paragraphs to achieve summarization on the paragraph level.

However, the work done by (AbdelFattah & Ren, 2009) is considered a strong base for statistical-based extractive text summarization, where they have applied multiple statistics, optimization and neural networks

techniques to score and extract sentence-level features such as sentence position, positive keywords, negative keywords, and more. Their work is extended in recent literature, (Joshi, Fidalgo, Alegre, & Alaiz-Rodriguez, 2022) has introduced Ranksum, an approach based on the rank fusion of sentence features that fused together using weighted scores of topic information, semantic content, significant keywords, and their positions in an unsupervised manner. While (Qaroush, Farha, Ghanem, Washaha, & Maali, 2021) has combined the statistical and semantic features with topic modeling for Arabic text summarization.

2.3.2. Topic-based approaches

This approach was first introduced by (Lin & Hovy, 2000), where they proposed to extract automatically sets of topic signatures of related words, and compute their associated weights as related to the head topics. This approach becomes later a base for a category of extractive text summarization task.

Belwal, Rai, and Gupta (2021b) used a mixed approach of topic-based modeling and the semantic measure within the vector space model to address the challenge of redundancy mentioned earlier. They aimed at extracting the strong sentences that represent the maximum of the embedded topics in the text to be summarized.

Srivastava, Singh, Rana, and Kumar (2022) has combined *Latent Dirichlet Allocation* (LDA) and *K-Medoids* clustering, the first is used to cluster sentences according to topics and the second to choose the most important sentences that form the summary in all subtopics. This model is language dependent as it depends on spaCy's POS lemmatizer (Lemmatizer, 2022). It is worth mentioning, that LDA was also used by (Ailem, Zhang, & Sha, 2019) for topic based approach text summarization but in the abstractive ATS tasks.

Moreover, (Belwal, Rai, & Gupta, 2022) has proposed a topic modeling approach that is applied on lower level entities inside a document, they modeled subtopics at clusters level in a single document, and then they addressed the limitations that might arise using an incorporated statement selection technique.

2.3.3. Graph-based approaches

Since the graph based approach LexRank was introduced by (Erkan & Radev, 2004), many methods have been presented in literature using the graph-based approach with different document elements graph representation.

Mallick, Das, Dutta, Das, and Sarkar (2019) have proposed a graph-based text summarization method using modified TextRank algorithm to constructs a graph with sentences as the nodes and compute their similarities to define the weights of the edges connecting them. It is worth mentioning that TextRank is a graph-based word-ranking model for keyword extraction, and widely used in text processing and summarization in particular (Zhang, Li, Yue, & Yang, 2020).

Mohamed and Oussalah (2019) have used a modified version of TextRank to build the graph-based text summarizer, where they computed the modified inverse sentence frequency-cosine similarity and used it to assign the weights for graph edges. Their approach differ from the typically used cosine similarity in that it gives different weightage to different words in the sentence, rather than the equal weights assigned by the traditional cosine similarity.

El-Kassas et al. have also introduced an extractive graph-based framework, named EdgeSumm (El-Kassas, Salama, Rafea, & Mohamed, 2020), that combines a set of four extractive algorithms, a) graph-based, b) statistical-based, c) semantic-based, and d) centrality-based methods) to benefit from their advantages and overcome their specific drawbacks.

Moreover, (Uçkan & Karcı, 2020) has proposed a graph-based method integrated with a text-processing tool that maintains semantic relation between sentences. On the other hand, (Belwal, Rai, & Gupta, 2021a) has introduced a mixed approach that integrates graphed-based approach with topic-based one, to create a model that uses the similarity between sentences and the document topic to assign the weight for the edges connecting individual sentences.

2.3.4. Machine learning based approaches

The machine learning based approach is the one that uses common machine learning algorithm, mostly classifiers, to achieve the summarization task through clustering or classifying the document elements into "*includeInSummary*" or "Not (*includeInSummary*)". The used machine-learning algorithms in this approach include Support Vector Machines (Mao, Yang, Huang, Liu, & Li, 2019), Naïve Bayes (Adhikari, 2020), Decision Trees (Nasar, Jaffry, & Malik, 2019), logistic regression (Mao et al., 2019), etc.

Moreover, deep learning networks have been applied under this approach to achieve the summarization task. For example, Bae, Kim, Kim, and Lee (2019) has used reinforcement learning through combining BERT based extractor and LSTM pointer network to achieve a hybrid extractive/abstractive summarization.

Ma et al. (Ma, et al., 2021) have incorporated BERT and LSTM with word embedding to build a hybrid model, *T-BERTSum*, which utilizes the topic-based and machine-learning-based approaches to generate a topic-aware extractive and abstractive summary. While, Grail et al. (Grail, 2021) have proposed a hierarchical propagation layer to overcome the limitations of BERT on summarizing long documents.

2.3.5. Discourse-based approaches

On the other hand, since the introduction of Rhetorical Structure Theory in the domain of computational linguistics in by Mann and Thompson in 1988 (Mann & Thompson, 1988), many Discourse-based applications in the field of computational linguistics have been introduced (Hou, Zhang, & Fei, 2020).

In the field of text summarization, Discourse-based summarizations models have been introduced in literature; such approaches represent the discourse in a document as a tree and focuses on the rhetorical connections between the text elements as in (Ishigaki, Kamigaito, Takamura, & Okumura, 2019), (Xu, Gan, Cheng, & Liu, 2019) and (Liu & Chen, 2021) for extractive summarization tasks, and (Feng, Feng, Qin, Geng, & Liu, 2021; Chen & Yang, 2021) in case of abstractive summarization.

3. Language agnostic summarization model for text (TxLASM)

The Semantically Annotated LaTeX project (*SALT*) (Groza, Handschuh, Möller, & Decker, 2007) has divided the semantic organization of a document, while preparing their sets of ontologies, into three layers: structural layer, rhetorical layer and finally the annotation layer that links the rhetorical characterizations with the structural components of the other two layers. While the rhetorical layer is based on the meaningful parts of a document (Brack, Hoppe, Stocker, Auer, & Ewerth, 2022), the structural layer is the one containing sentences, paragraphs, and other elements of a text document (Constantin, Peroni, Pettifer, Shotton, & Vitali, 2016).

In our study, we focused on the structural component of a document (or a piece of text), where a document can be seen as a hierarchal structure that consists of four elements: *paragraphs*, *sentences, n-grams* and *words* as seen in Fig. 2.

Throughout the study, the following notation is being used. Where, a document D^j is considered as a set of paragraphs $D^j = \left\{P^j_1, P^j_2, P^j_3, .., P^j_k\right\}$, where k is the total number of paragraphs in the document. On the other hand, a single paragraph P^j_k consists of a set of sentences $P^j_k = \left\{S^{j,k}_1, S^{j,k}_2, S^{j,k}_3, .., S^{j,k}_m\right\}$, where m is the total number of sentences in the paragraph. As the hierarchal relation goes deeper, a sentence $S^{j,k}_m$, in turn, consists of a set of words $S^{j,k}_m = \left\{W^{j,k,m}_1, W^{j,k,m}_2, .., W^{j,k,m}_n\right\}$, where n is the total number of words in a sentence.

A group of consecutive words in a sentence can be grouped together

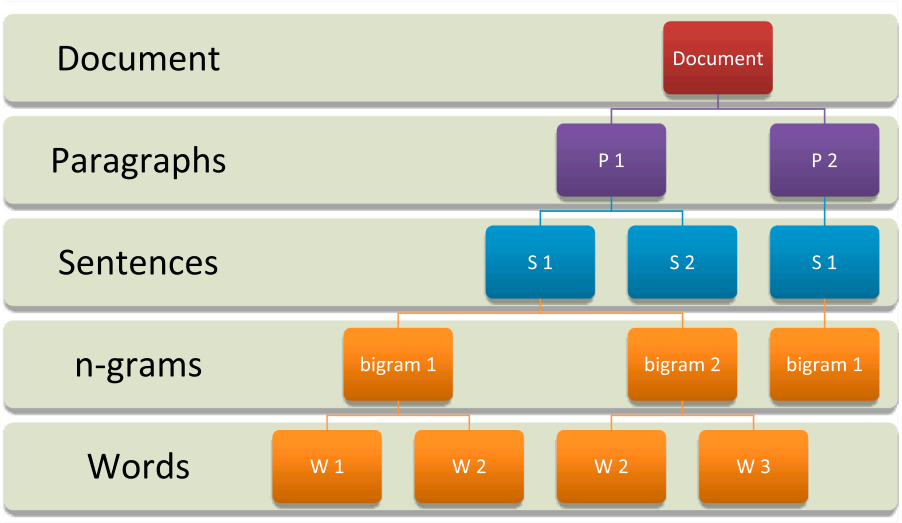


Fig. 2. The hierarchal relation between document elements.

to form what is called n-grams, where, a bigram is formed of two words $W_1^{j,k,m}, W_2^{j,k,m}$ or $W_2^{j,k,m}, W_3^{j,k,m}$, in addition a tri-gram consists of a sequence of three consecutive words $W_1^{j,k,m}, W_2^{j,k,m}, W_3^{j,k,m}$ or $W_2^{j,k,m}, W_3^{j,k,m}, W_4^{j,k,m}$. As such, a single sentence can also be seen as a collection of different size n-grams.

The *TxLASM* could be explained in four distinct steps: 1) the *Shape Coding* technique, 2) *Language Agnostic Pre-processing*, 3) *Elements Scoring*, and finally 4) *Summary Extraction*.

3.1. Shape coding

Shape-Coding is the main technique used in *TxLASM*. The idea of *shape coding* is to extract the main features of a document element and encode them in a simple way to reflect these features using a compact and intuitive sequence of letters. *Shape-coding* takes into consideration the nature of the characters forming a text element (letters or numbers), the case of that element (lowercase, uppercase or mixed) as well as its format (bold, italic, font size, etc.).

This proposed technique of *shape-coding* is the main part of the algorithm responsible for the language agnostic or language independent nature of the entire model.

With reference to the document elements mentioned earlier, three *shape-coding* techniques are proposed in this paper: a) *shape-coding* of words and n-grams, b) *shape-coding* of sentences, and c) *shape-coding* of paragraphs.

These *shape-coding* techniques extract the most powerful and influential words, n-grams, Named Entities (NE's), Concrete Concepts (CC's), key phrases and sentences. Where, combing the n-gram shapes with their frequency of occurrence can lead to the identification of NE's and CC's in a language and domain agnostic way. Those techniques are explained in details in the following sub-sections:

3.1.1. Shape-Coding of words

Words are the building blocks of a piece of text. As such, encoding the shape of a word will be reflected, directly, in encoding its parent *n-grams* and, indirectly, in encoding its parent sentences and paragraph. *Shape-coding* of words implies converting each character in a word into its corresponding code from a handful set of codes, in a process that results in a compact encoded word that reflects its important features.

In case of *shape-coding* of words, the *code set* consists of six elements (or letters) that are used to encode the word features. The elements of this *code set* are {X, x, C, c, N, n}. The indication of each code is explained in Table 1.

Table 1
Set of codes used in shape-coding of words and n-grams.

Code Element	*Indication*
X	Indicates a single uppercase letter.
x	Indicates a single lowercase letter.
C	Indicates 1 or more uppercase letters.
c	Indicates 1 or more lowercase letters.
N	Indicates a single numeric character.
n	Indicates 1 or more numeric characters.

Shape-coding of a word is done in four main steps: *i*) Remove all non-alphanumeric characters as: {., " / &;: @ etc.}. *ii*) Change all numeric characters to "N". *iii*) Change all letters to "X" and "x" for uppercase and lowercase letters respectively. And finally, *iv*) Group sequences of repeated codes using "C", "c" and "n" for repeated "X", "x" and "N" respectively. In other words, "C", "c" or "n" are used to replace a sequence of similar character shapes of length ≥ 1. Where, the first character code of the identical code sequence is kept unchanged while all the following similar characters codes are replaced with one of those three continuity codes, "C", "c" or "n", in order to encode the continuity of the same shape code. For example, "XXX" is grouped using C to be "XC", while "xxxx" and "NNNNNN" are grouped using "c" and "n" resulting in "xc" and "Nn" respectively. Table 2 shows some examples for encoding different word shapes.

As seen the examples in Table 2, *shape-coding* of words results in encoding those words into a small set of equivalent classes that represent their shapes. For example, while all numbers are normalized to "Nn", the vast majority of words in a piece of text are converted to "xc". On the other hand, names of cities, persons, etc. are converted to "Xxc", which is less common as compared to "xc".

Table 2
Examples for shape-coding of words.

Word	**Shape Coding Steps**	**Final Shape Code**
Egypt	"E" →"X", "gypt" →"xxxx" →"xc"	Xxc
mRNA	"m" →"x", "RNA" →"XXX" →"XC"	xXC
UnB	"U" →"X", "n" →"x" and "B" →"X"	XxX
1,027,708	"1,027,708" →"1027708" →"NNNNNNN" → "Nn"	Nn
game	game → xxxx → xc	xc
U.S.A.	U.S.A. → USA → XXX → XC	XC
2-way	2-way → 2way → Nxxx → Nxc	Nxc

In addition, the step of removing non-alphanumeric characters from a word, adds more power to the proposed model as it helps in normalizing similar words especially in case of abbreviations. For example, words like "U.S.A." or "USA" referring to the "United States of America" will be treated in our model in the same way, as both words are encoded into the same shape-code "XC".

The rareness of a word shape in a document reflects its importance. Using the same examples stated above, a capitalized word as "USA" has a word shape "XC". Moreover, a rare word like the name of our proposed tool "TxLASM" has even more rare shape-code, "XxXC". Both shape codes, "XC" and "XxXC", are rarer in a document than all other common words (verbs, nouns, adjectives, etc.) that will typically be encoded into the most abundant shape-code, "xc". As such, words encoded into rare shape-codes, are more likely to be a Named-Entity (name of a country, city, person, tool, abbreviations, etc.).

In conclusion, *shape-coding* of words results in: *i*) mapping words into a small set of equivalent classes. *ii*) Normalizing numbers and similar words into similar shape-codes. *iii*) identifies important words and NE's.

3.1.2. Shape-Coding of n-grams

Shape-coding of an n-gram is simply carried out by concatenating the encoded shapes of its constituent individual words. Where, a *bigram* is encoded by first encoding its two individual words and then concatenate those codes together using a space delimiter. See Table 3 for some examples of shape-coding of n-grams.

In addition, words or n-grams format can also be encoded easily using the same philosophy of the proposed shape-coding technique. Where different word formats can be encoded simply by applying the same format of the word (bold, italic, etc.) on the encoded shape.

For example, the word "***Brazil***" with bold and italic formatting will have a shape-code "***Xxc***", i.e. the shape-code is formatted in bold and italic as its parent word. As such, the rareness of formatted shape-codes reflects the importance or the influence of the original words.

3.1.3. Shape-Coding of sentences

Shape-coding of a sentence means converting its main features into a representative code, in a process resulting in a single fixed-length compact code that reflects the important features of that sentence.

Unlike the word *shape-coding* technique, the *code set* used for sentence *shape-coding* consists of only 2 elements that are used to encode the sentence features. This *code set* consists of {Z, z}, whose indications are explained in Table 4.

Moreover, the sentence's shape-code has a fixed length, formed of only three codes regardless of the length of the sentence or the shape of its parent paragraph or its constituent words.

Sentence shape-coding is carried out in three major steps: *i)* if the first letter in a sentence is uppercase, thus the code element "***Z***" will take the first spot of the 3-letters shape-code, while in rare situations where the first letter is lowercase then the first spot will be "***z***". *ii)* Then, the number of words starting with a lowercase letter (L), as well as those starting with an uppercase letter (U) are computed. *iii)* And finally, the ratio of words with initial uppercase letter to the total number of words in that sentence is computed to decide the second and third spots of the 3-letters *shape-code* as seen in equation (1).

$$Sentence\ Shape\ Code = \begin{cases} ZZ & if\ \dfrac{U}{L+U} \geq \omega_1 \\ Zz & if\ \dfrac{U}{L+U} < \omega_1\ \ and\ \ \geq \omega_2 \\ zz & if\ \dfrac{U}{L+U} < \omega_2 \end{cases} \tag{1}$$

Where, ω_1 and ω_2 are thresholds are decided by the model designer. The default values of these thresholds used in this paper are $\omega_1 = 0.7$, while $\omega_2 = 0.3$.

As such, and based on the first letter in the sentence and the ratio of $\frac{U}{L+U}$, a sentence could be encoded into: *i)* In case of uppercase initial letter: ***ZZZ*** (majority uppercase words, where the ratio of U is greater than or equal to ω_1); ***Zzz*** (majority lowercase words, where the ratio of U is below ω_2); ***ZZz*** (mixed case words, where the ratio of U lies between ω_1 and ω_2). Or, *ii)* ***zZZ***, ***zzz*** and ***zZz*** in case of lowercase initial letter.

As per the examples listed in Table 5, the process of sentence *shape-coding* results in encoding sentences into one of six encoded classes that represent their major features. The rareness of a sentence encoded-shape in a piece of text reflects its degree of importance. For example, sentences with encoded-shapes "***ZZZ***" or "***ZZz***" are rarer than those encoded into "***Zzz***", and thus should receive higher weights reflecting the relative importance or influence of those sentences.

3.1.4. Shape-Coding of paragraphs

Shape-coding of a paragraph means converting its main features into a representative code, in a process resulting in a single fixed-length compact code that reflects the important features of that paragraph. The *code set* for encoding paragraphs consists of 7 elements {B, N, O, S, M, P, p}, that are used to reflect the main features of a paragraph as per Table 6.

Unlike the variable length shape-codes used to encode words, paragraph's shape-code has a fixed five-code length, regardless of the shape or size of the paragraph. As in case of sentence shape-coding, this fixed five-length encoded shape has pre-defined fixed positions for each main

Table 3
Example for *shape-coding* different n-grams.

n-gram Class	*n-grams*	*n-grams Coded Shape*
Bigram	Middle East	Xxc Xxc
	school bus	xc xc
	100 mph	Nn xc
Trigrams	Republic of Ireland	Xxc xc Xxc
	189 square feet	Nn xc xc
	He plays football	Xx xc xc
4-grams	linking northeast Africa with	xc xc Xxc xc
	Arab Republic of Egypt	Xxc Xxc xc Xxc
	United States of America	Xxc Xxc xc Xxc
5-grams	the Federative Republic of Brazil	xc Xxc Xxc xc Xxc
	Egyptian Minister of Foreign Affairs	Xxc Xxc xc Xxc Xxc
	BFC bought 88% of BankAtlantic	XC xc Nn xc XxcXxc

Table 4
Sentence Code Set.

Code Element	Indication
Z	Indicates a sentence with an uppercase initial letter.
z	Indicates a sentence with an initial lowercase letter.

Table 5
Examples for shape-coding of sentences.

Sentence	**Shape Coding Steps**	**Final Shape Code**
Egypt is a country linking northeast Africa with the Middle East and it dates to the time of the pharaohs	***Egypt*** → ***Z*** $U = 4, L = 16$ $U/(U+L) = 0.2$, thus the sentence is considered to be mostly lowercase →"***zz***"	**Zzz**
Language Agnostic Summarization Model for Text Documents	***Language*** → ***Z*** $U = 6, L = 1$ $U/(U+L) = 0.86$, thus the sentence is considered to be mostly uppercase →"***ZZ***"	**ZZZ**
In this paper we propose Language Agnostic Summarization Model for extractive summarization	***In*** → ***Z*** $U = 5, L = 7$ $U/(U+L) = 0.42$, thus the sentence is considered to be a mixed case sentence →"***Zz***"	**ZZz**

Table 6
Paragraph Code Set.

Code Element	Indication
P	Indicates a paragraph with an uppercase initial letter.
p	Indicates a paragraph with an initial lowercase letter.
B	Indicates Bulleted Paragraph.
N	Indicates Numbered Paragraphs.
O	Indicates Ordinary Paragraph (neither bulleted nor numbered).
S	Indicates a paragraph consisting of Single Sentence.
M	Indicates paragraph consisting of Multiple Sentences.

feature in a paragraph. Where, the first spot is reserved for bullets and numbering, the second spot is for the number of sentences forming that paragraph, the third spot reflects whether the paragraph starts in an upper or lowercase letter, while the last two spots encode the ratio of words that starts with uppercase letter to the total number of words in the paragraph.

Paragraph's shape-coding is carried out as follows:

i. The first spot of the coded-shape is determined based on the type of the paragraph (*ordinary*, *numbered* or *bulleted*), with three possible values "***N***", "***B***" or "***O***" respectively. Where: a) The first spot takes the value "***N***" if the encoded paragraph starts with numbered lists as real numbers (1, 2, 3, etc.), letters (a, b, c, etc.) or roman numerals (i, ii, iii, etc.); b) "***B***" occupies the first spot in case of paragraphs starting with bulleted list (−, •, ✓, *, etc.), or c) "***O***" if the paragraph does not start with bulleted or numbered list.
ii. The second spot indicates the number of sentences forming the encoded paragraph, with two possible values "***S***" or "***M***" for single and multiple sentences paragraph respectively. Where a single sentence paragraph is usually titles, while multiple sentence paragraphs are usually the body of typical text documents.
iii. Moreover, the third spot takes the value "***P***" if the first word in the paragraph starts with an uppercase letter and "***p***" if it starts with a lowercase letter.
iv. The last two spots represent the ratio of words starting with uppercase letter (U) to the total number of words in the paragraph, regardless of the number of sentences forming the paragraph as per equation (2) below.

$$Paragraph\ Shape\ Code = \begin{cases} PP & if\ \dfrac{U}{L+U} \geq \omega_1 \\ Pp & if\ \dfrac{U}{L+U} < \omega_1\ \ and \geq \omega_2 \\ pp & if\ \dfrac{U}{L+U} < \omega_2 \end{cases} \quad (2)$$

Where, ω_1 and ω_2 are thresholds are decided by the model designer. The default values of these thresholds used in this paper are: $\omega_1 = 0.7$, while $\omega_2 = 0.4$.

As such, if the ratio of $\frac{U}{L+U}$ the ratio of U exceeds $\boldsymbol{\omega_1}$, then the last two spots will take the value ***PP***, and ***pp*** if the ratio is below $\boldsymbol{\omega_2}$, while if that ratio lies between $\boldsymbol{\omega_1}$ and $\boldsymbol{\omega_2}$, then the code will take the value ***Pp***.

It is worth mentioning that, the rareness of a paragraph encoded-shape in a piece of text reflects its importance. For example, "***OSPPP***" that encodes a single sentence paragraph with the majority of its words starting with uppercase letter, is probably representing a title, while the more abundant normal paragraphs are mostly encoded as "***OMPpp***". In other words, the rareness of "***OSPPP***" reflects its importance; as such, it should receive higher weight when computing the final score.

3.2. Language agnostic pre-processing

The aim of the preprocessing step is to normalize the shape and form of words in a text document. The pre-processing step will affect the score of the word and the subsequent n-grams and sentences.

The scoring of a particular word according to *TxLASM* is based on two assumptions: *i)* rare words in a text have more influence on the meaning of the text and should get heavier weights; and *ii)* rare or less frequent word shapes may indicate a NE that implies more influence in the text. In conclusion, the rarer the word form and shape are, the higher the word score and thus the more important it is for summarizing the text. As such, the pre-processing step should maintain the relative rareness among words in a particular piece of text.

Moreover, since the proposed model is based on a *language independent* approach, therefore the pre-processing step should not depend on any prior knowledge of the language of the text to be summarized with all of its etymology, grammatical, semantic or syntactic relations.

As presented in the literature review and related work section, most of text summarization techniques are language dependent especially in the preprocessing routines. Those language dependent preprocessing steps include stop words removal (Pant, Srinivasan, & Menczer, 2004), lemmatization or stemming (Khyani, Siddhartha, Niveditha, & Divya, 2021), Part-of-Speech (POS) Tagging (Martinez, 2012), and other language dependent preprocessing routines. These techniques require prior knowledge of the language of the text, to utilize the appropriate databases or lexicons (stop words, dictionary, etc.), knowledge bases and hand coded rules for stemming and lemmatization (Janaki Raman & Meenakshi, 2021), as well as parsers and taggers (POS taggers, etc.) (Moratanch, 2017).

TxLASM is an unsupervised text summarizer that is totally language agnostic and hence independent of any external databases or parsers (that by definition are language dependent). As such, one of the main contributions of this study is proposing a language agnostic stop words removal algorithm as well as a totally language independent stemmer.

Fig. 3 shows the pre-processing steps of words in a text document from the beginning of the preprocessing algorithm until the computation of *Word Score* (*WS*). Such steps can be explained as follows:

(i) *Tokenization*: A sentence is segmented into its individual tokens (words).
(ii) *Removal of special characters*: All none alphanumeric characters, as commas, hyphens, points, semicolons, etc., are removed from each individual token.
(iii) iii Normalize Numbers:

If the token is a number, then it is replaced by "####". This step of number normalization is crucial to avoid assigning false high weights for each number. Since numbers tend to be different in a document (i.e. it is rare that a single number is repeated many times in a single document especially in scientific texts), thus, without normalization, the model will consider every number as a rare influential word leading to an over estimated importance of the number and its containing sentence.

(iv) Normalize Word Forms:

Encoded shapes depend on the word form (its spelling) and its case (uppercase, lowercase or mixed). Since the rareness of a word shape reflects its importance (usually Named Entities are capitalized or start with uppercase letter), thus it is crucial to normalize word case such that uppercase words are those of NE's and not for words that accidently appear at the beginning of a sentence.

To achieve this normalization, each word that starts with a capital letter is searched in the whole document. If the word was found in any other position in the document with an initial lowercase letter, then, the word case is changed to lowercase.

(v) Combining Similar Words:

Some words are similar and should be treated as one when counting the frequency of occurrence of words. For example, in English language,

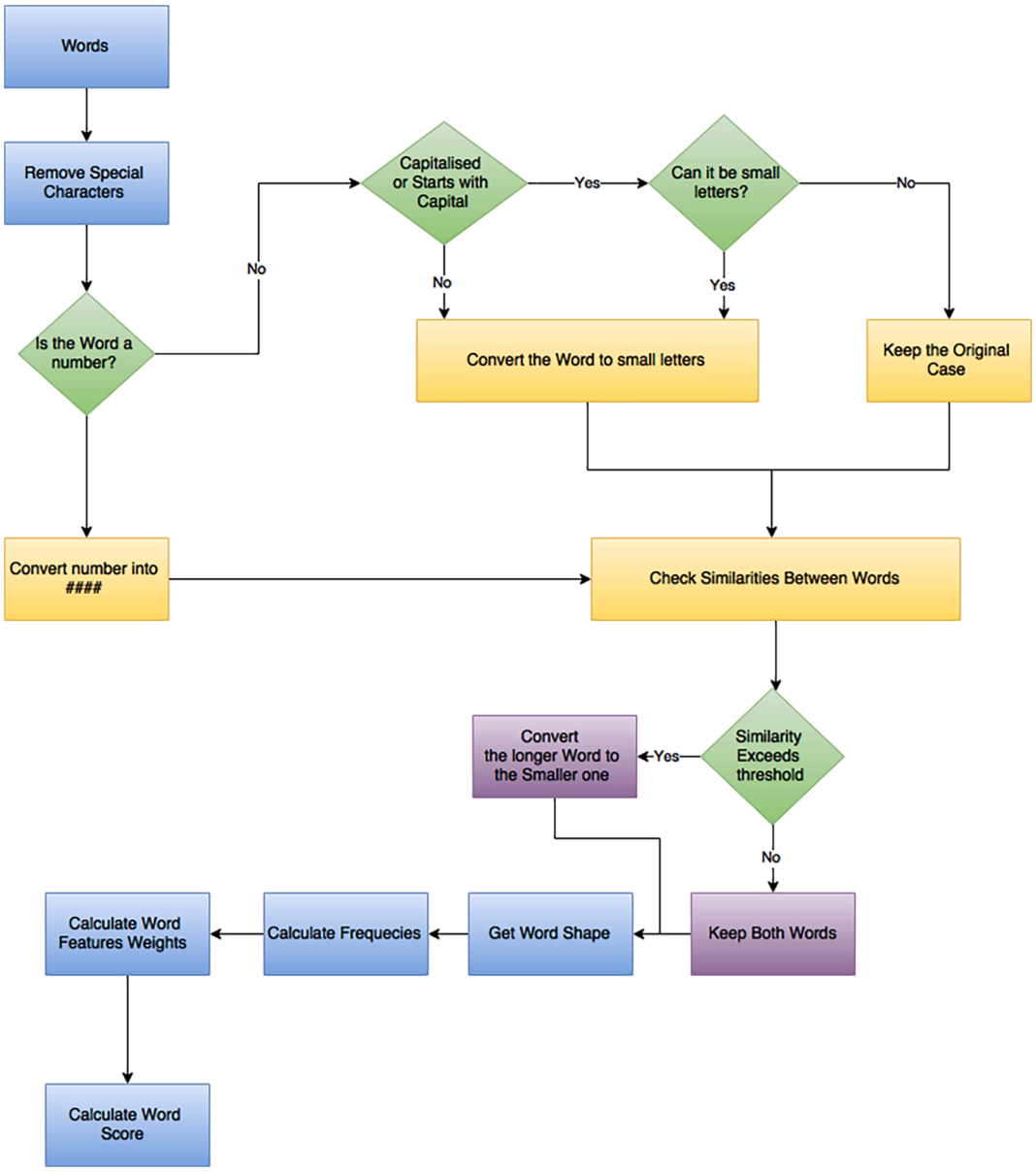


Fig. 3. Steps of words pre-processing.

words like "*year*" and "*years*"; "*automatic*" and "*automatically*"; or "*allow*" and "*allowed*"; and in Portuguese, "*embaixada*" and "*embaixadas*"; or "*quero*" and "*quer*".

Supervised text summarization models use *stemming* or *lemmatization* to revert a word back to its root/stem and hence combine similar words. This step, in case of supervised text summarization, is language dependent, due to the need of lexicons or language dependent hand coded rules. Since our prosed model is language agnostic, so it is important that we develop an algorithm that combines similar words depending on the word forms and the degree of similarities they have in common regardless the language of the original text and its words.

The proposed algorithm tends to calculate the number of similar letters between two words starting from the letter at position 1. In other words, the algorithm computes the length of sequence of common letters between two words, which is called *Degree of Similarity* or *DoS* for short.

The *Degree of Similarity* is computed using equation (3), and if that *DoS* value exceeds a predefined threshold, then the extra letters of the longer word are discarded, so that it is converted into the smaller one. (See examples in Table 7)

$$DoS = \frac{length\ of\ common\ letters\ sequence}{length\ of\ the\ longer\ word} \tag{3}$$

3.3. Elements scoring

TxLASM is a statistical extractive summarization model that depends on computing scores for each document element. The individual element score is then used to compute the *Overall Sentence Score* that is used to extract the most influential sentences that best represent the text to be summarized.

3.3.1. Word score (WS)

The *Word Score* (*WS*) of a token is simply done by combining the *Word form Weight* (*WfW*) and its *Word shape Weight* (*WsW*) as equation (4).

$$WS = (2-\alpha)WfW + \alpha WsW \tag{4}$$

Where, the *Word form Weight* (*WfW*) is based on the frequency of the word's form (its spelling). Since, the proposed model assumes that a rare

Table 7
Examples for combining similar words using a threshold of 0.68. Words with *DoS* < the threshold (colored in red) are maintained unchanged.

Word 1	Word 2	Number of Similar letters	*DoS*	Action Performed
Year	Years	4	4/5 = 0.8	"*Years*" is changed to "*Year*"
Automatic	Automatically	9	9/13 = 0.69	"*Automatically*" is changed to "*Automatic*"
An	And	2	2/3 = 0.66	Both words are maintained
Form	From	1	1/4 = 0.25	Both words are maintained
University	Universities	9	9/12 = 0.75	"*Universities*" is changed to "*University*"
Allow	Allowed	5	5/7 = 0.71	"*Allowed*" is changed to "*Allow*"
Quer	Quero	4	4/5 = 0.8	"*Quero*" is changed to "*Quer*"

word has more influence on the text than a more common word, thus as seen in equation (5) below, the *WfW* is computed by taking the reciprocal of the natural logarithm (log of base *e*) of the word count.

$$WfW = \frac{1}{\ln(count(word) + 1)} \tag{5}$$

The natural logarithm is inverted in order to give non-linear higher weights to rare words compared to abundant ones. While, the add 1 normalization is done to avoid dividing by zero in case of a word that was mentioned only once, as $\ln(1) = 0$.

On the other hand, the *Word shape Weight* (*WsW*) is based on the encoded shapes frequencies. Since, the proposed model assumes that a rare coded shape has more influence on the text than a more common shape, thus as seen in equation (6) below, the *WsW* is computed by taking the reciprocal of the logarithm (log of base 10) of the coded shape count.

$$WsW = \frac{1}{\log(count(shape) + 1)} \tag{6}$$

It is worth mentioning that logarithm of base 10 is used to compute *WsW* rather than the natural logarithm used for computing *WfW* since this technique will give much higher weights for encoded shapes in comparison to word forms (spelling). As such, if two words have equal frequency of occurrence, then the word with more rare encoded shape will get higher *Word Score* (*WS*). This technique of computing weights is useful in giving a relatively lower score for relatively unimportant words with very common shape "*xc*" that might appear, in rare situations, few times in the text to be summarized. For example, a word like "*for*" may exists only 3 times in a text, thus its common coded shape weight will pull it lower as compared to other words that has equal frequency but with rarer encoded shape.

Moreover, the α parameter used in equation (4) is a constant greater than or equal to 0 and less than or equal to 2. That constant is used to adjust the relative weights of each term of the *WS* computing equation. Usually α is set to 1 in order to maintain the relative weights of both terms of the equation imposed by the difference between log and ln. However, in certain situations where short texts are to be summarized, it is recommended to give larger weight for the shape term *WsW,* as in case of short text unimportant words might, accidently, occur in unusual low frequency. In such situation, higher α is used to give more weights for named entities with rarer encoded shapes.

For example, let us assume a text where common words like "*the*" has frequency of occurrence = 40, and its encoded shape "*xc*" has frequency = 300, while, the word "*UnB*" exists only 2 times and its coded shape "*XcX*" exits three times. In this case the *WS* for "*the*" $= 1 \times 0.2693 + 1 \times 0.4035 = 0.6728$, while $WS_{UnB} = 1 \times 0.9102 + 1 \times 1.6610 = 2.5712$. As such, the *WS* for both tokens reflect the importance of the token "*UnB*" as compared to the token "*the*".

3.3.2. n-Gram score (nGS)

In contrast to the word scoring assumption that, the rare words have more importance than the more abundant ones, in our proposed model the n-gram scoring is based on the exact opposite assumption. N-gram scoring approach assumes that probable *n-gram's*¸ in terms of form (spelling of its constituent words) and shape (shapes of its encoded words), have more influence on the meaning of the text than the less probable ones, and accordingly should get heavier weights.

This assumption aids in identifying and highlighting Named Entities and Concrete Concepts in a language agnostic manner. Moreover, this assumption and its role in identifying NE's and CC's can be explained in view of the fact that the overall score of an *n-gram* (*nGS*) is influenced by the scores of its constituent words. As such, in case of bigrams for instance, the more probable two rare words occurring together the higher the weight the bigram should have. For example, two rare words like "*United*" and "*States*" if they occur together in relatively high frequency this indicates that the bigram "*United States*" (with shape "*Xxc Xxc"*) is more likely to be a *Named Entity* or a *Concrete Concept*.

In conclusion, the more abundant the *n-gram* form and the more rare its individual words are, the higher its *nGS* and thus the more important it is for summarizing the text.

Computing the *n-gram score* (*nGS*) for the extracted bigrams, trigrams, 4-grams and 5-grams is done by multiplying their *n-Gram form Weight* (*nGfW*) and their *n-Gram shape Weight* (*nGsW*) as per equation (7).

$$nGS = nGfW \times nGsW \tag{7}$$

Where, the *nGfW* is computed by compute the Maximum Likelihood Estimate (MLE) of that n-gram's form as per the equations below.

$$nGfW = \left(\prod_{i=1}^{n} P(w_i|w_1,..,w_{i-1})\right) \times \sum_{i=1}^{n} WS(word_i) \tag{8}$$

$$Where, P(w_i|w_1,..,w_{i-1}) = \frac{count(w_1,..,w_i)}{count(w_1,..,w_{i-1})} \tag{9}$$

$P(w_i|w_1,..,w_{i-1})$ is the maximum likelihood estimate of an n-gram formed from *I* words, which is calculated as the probability of the word w_i given the sequence of words $\{w_1,..,w_{i-1}\}$, which is equal to the ratio of the frequency of occurrence of the whole sequence of words $\{w_1,..,w_i\}$ to the frequency of occurrence of the sequence formed from the words $\{w_1,..,w_{i-1}\}$ (Jurafsky & Martin, 2021). Where, w_i is the word at position *i* of an *n-gram*. While, $\sum_{i=1}^{n} WS(word_i)$ is the sum of words scores of all the words that form the n-gram. This term is included to lower the value of *nGfW* for probable n-grams that are made from weak words, and hence, give higher weights for n-grams made from stronger words.

On the other hand, for the *n-Gram shape Weight* (*nGsW*), the *n-gram* constituent encoded shapes frequencies are used to compute the MLE of that n-gram encoded shape as per the equations below.

$$nGsW = \left(\prod_{i=1}^{n} P(s_i|s_1,..,s_{i-1})\right) \tag{10}$$

$$Where, P(s_i|s_1,..,s_{i-1}) = \frac{count(s_1,..,s_i)}{count(s_1,..,s_{i-1})} \tag{11}$$

Where, s_i is the encoded shape of the word at position *i* of that *n-gram*, and $P(s_i|s_1,..,s_{i-1})$ is the maximum likelihood estimate of an n-gram formed from *i* encoded shapes. The MLE in this case is calculated as the probability of the shape s_i given the sequence of encoded shapes $\{s_1,..,s_{i-1}\}$, which is equal to the ratio of the frequency of occurrence of the whole sequence of encoded shapes $\{s_1,..,s_i\}$ to the frequency of occurrence of the sequence formed from the shapes $\{s_1,..,s_{i-1}\}$.

As discussed previously, the high probable n-grams form and shapes that are composed of rare words will get higher *nGS* than the less probable and/or weaker ones. For example in a text about astronomy, a relatively abundant bigram like "*Solar System*" will get higher *nGS* than less abundant one like "*the flare*".

However, in certain circumstances, a highly probable n-gram can receive lower *nGS* than less probable ones. This happens when the n-gram is composed of weak words with weak *WS*. For example, a probable and abundant bigram like "*in the*" with coded shape "*xc xc*" will have lower overall *nGS* than other less probable ones like "*United States*" due to the fact that "*in the*" is composed of weak words, while, "*United States*" is composed of strong words that intensify its overall *nGS*.

As such, *nGS* reflects the actual importance of an n-gram based on its probability of occurrence as well as the strength of its constituent words.

3.3.3. Paragraph score (PS)

The process of paragraph scoring involves only two steps, a) *shape-coding* as discussed in subsection 3.1.4, followed by b) counting the encoded shape frequencies, in order to get the final *Paragraph Score* (*PS*) as per equation **(12)** below.

$$PS = \frac{1}{\log(count(shape)+1)} \tag{12}$$

As in the case of word shape scoring, the logarithm is inverted in order to give non-linear higher weights to rare paragraph shapes compared to the abundant ones.

3.3.4. Sentence score (SC)

In extractive summarization task, sentences are the only output of any extractive summarization tool. As such, sentence scoring is the ultimate goal of the entire automatic summarization algorithm. After all the sentences in a text are scored, then, they are arranged in descending order according to their overall score *Sentence Score* (*SC*). Sentence Score is computed by combining the scores of all elements related to that sentence, the *WS*, *nGS* and *PS* in addition to the score of the sentence's encoded shape Weight (*SsW*) which is calculated as per equation **(13)**.

$$SsW = \frac{1}{\log(count(shape)+1)} \tag{13}$$

After computing all text element scores relevant to a sentence *j*, the overall sentence score for that sentence (SC^j) combines those terms as shown in equation (14)

$$SC^j = \eta.\left(\lambda_1 \sum_{i=1}^{N_w^j} \frac{WS_i}{\max(WS)} + \lambda_2 \sum_{i=1}^{N_{2g}^j} \frac{2GS_i}{\max(2GS)} + \lambda_3 \sum_{i=1}^{N_{3g}^j} \frac{3GS_i}{\max(3GS)} + \lambda_4 \sum_{i=1}^{N_{4g}^j} \frac{4GS_i}{\max(4GS)} + \lambda_5 \sum_{i=1}^{N_{5g}^j} \frac{5GS_i}{\max(5GS)}\right) + \beta.\frac{SsW}{max(SsW)} + \delta.\frac{PS}{max(PS)} \tag{14}$$

Where, N_w^j, N_{2g}^j, N_{3g}^j, N_{4g}^j and N_{5g}^j are the total number of words, bigrams, trigrams, 4 g and 5 g in sentence *j* respectively. While, max(..) means the maximum value of that score in the entire text (not only the sentence).

The weights λ_{1-5} are weights given to word and n-grams scores respectively, to help tweaking the sentence-scoring performance. It is worth mentioning that $\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 = 1$.

In addition, $\eta, \beta and \delta$ are weights ranging between 0 and 1 to adjust the relative importance of the sum of words and n-grams scores (*WS* and *nGS*), the Sentence Shape Weight (*SsW*), and the Paragraph Score (*PS*) respectively.

It is worth mentioning that all scores are normalized by dividing them by the maximum value of the score noticed in the entire text. This is done in order to keep the scores of all elements between 0 and 1 and prevent any unwanted effect of off-scale scores for some terms.

3.4. Summary extraction

After the text elements are scored properly, all sentences in the text are then arranged in a descending order of their overall score (*SC*). Then, the top *N* sentences are selected based on the desired *Degree of Compression (DoC)* of the summary.

Degree of Compression (DoC) can be set as: a) a percentage of sentences or words to be extracted from the original text, or b) the number of words that should appear in the summary.

For example: *DoC* = 150e means that the extractive summary should contain around 150 words, while *DoC* = 50% means that the summary should be compressed to have a size that is 50% from the original text.

As such, the top sentences (with the highest *SC*) are extracted and ordered according to their order in the original text to produce the required summary.

Fig. 4 shows a flowchart that summarizes all the major steps included in *TxLASM* from the step of tokenization until summary generation.

4. Experiments

4.1. Datasets

In order to assess the performance of the proposed model against state-of-the-art summarization models listed in literature, a text summarization benchmark dataset was selected. *DUC2002*, a benchmark dataset provided by the American National Institute of Standards NIST for the Document Understanding Conference (NIST, 2002).

DUC2002[1] is considered one of the most widely used benchmark dataset for English document summarization task. It consists of 59 sets of document, each of which contains around 5–10 English news article, i.e. a total of 567 news articles. The articles span 11 news categories; biography, politics, health, science, sports, natural disasters, society, business, culture, law and international.

Two assessors have manually summarized each article or set of articles, to be used as the evaluation standard. A Single article is summarized to a level of 100 words per summary. This rate of compression ranges from 40%, in case of documents with 250 words, down to 10% for documents with around 1000 words (which is considered a high compression rate).

Moreover, and for testing the ability of *TxLASM* to undergo an efficient text summarization in a language agnostic manner, three other datasets were used to assess the language agnostic capabilities of the proposed model. Datasets of text documents in Portuguese, Spanish and French languages were selected.

For Portuguese language, *TeMario*,[2] a Brazilian Portuguese text collection, is used. It consists of a collection of 100 news articles collected from the Brazilian journals "*Folha de São Paulo*" and "*Jornal do Brasil*". The news articles were divided into 5 categories, 20 articles each (political, international, social, opinion and world). The manually generated summaries are done with a degree of compression (*DoC*),

[1] https://www-nlpir.nist.gov/projects/duc/past_duc/duc2002/test.html.
[2] https://www.linguateca.pt/Repositorio/TeMario/.

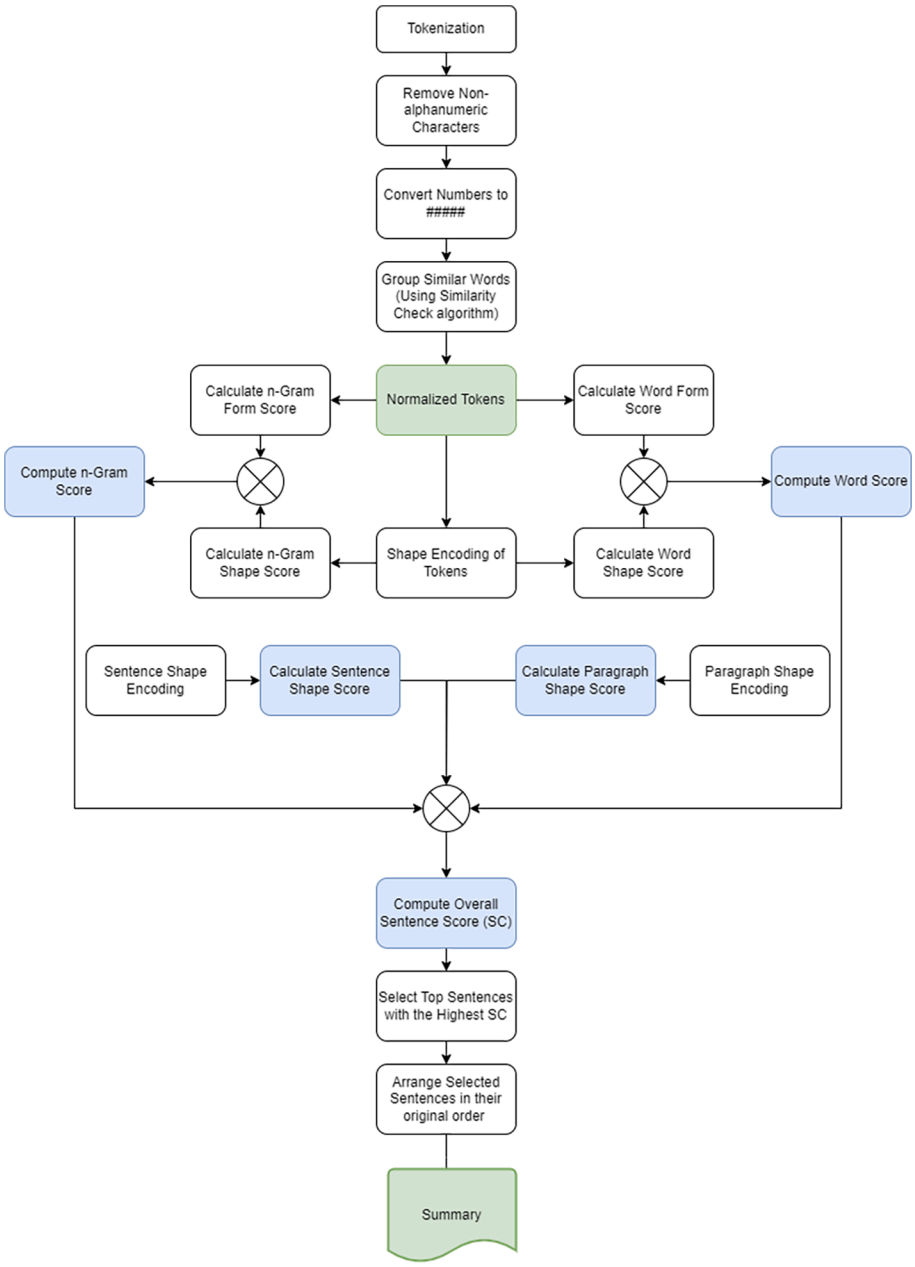


Fig. 4. Major steps to implement *TxLASM*.

ranging from 25% to 30% of the original documents.

While, in case of French and Spanish languages, two datasets 40 documents each, were obtained from French and Spanish news websites (*lemonde.fr* and *elpais.com* respectively). The news articles were chosen from 4 categories (sports, economy, politics and culture) 10 articles each. Each article has two manually generated summaries of 100 words length (i.e. $DoC = 100e$), made by two native French and Spanish speakers respectively.

4.2. Software

In this paper, a set of software tools and libraries were used for model building, analysis and data visualization. These tools are: *Jupytar Notebooks* (Kluyver, Ragan-Kelley, & Fernando, 2016) for Python, *Scikit-Learn* machine learning library (Pedregosa, et al., 2011), *MATLAB* R2015a (The MathWorks, 2015), *Voyant* visualization tools (Sinclair & Rockwell, 2022), and *VBA* of Microsoft Excel (Microsoft, 2016).

4.3. Parameter selection

The model parameters could have been adjusted by tenfold cross validation technique, using randomly selected articles. However, and in

Table 8
Value of the parameters used for document element scoring in TxLASM.

Parameter	Value	Applied to
DoS Threshold	0.65	*Threshold for the degree of similarity between words*
A	1	*Apply weight to WsW and WfW to calculate WS*
λ_1	0.2	*The weight of WS used in calculating the SC*
λ_2	0.2	*The weight of 2GS used in calculating the SC*
λ_3	0.2	*The weight of 3GS used in calculating the SC*
λ_4	0.2	*The weight of 4GS used in calculating the SC*
λ_5	0.2	*The weight of 5GS used in calculating the SC*
H	1	*Weight applied to the sum of WS and nGS in a sentence*
B	1	*Weight applied to the SsW to compute the SC*
δ	1	*Weight applied to the PS to compute the SC*

Table 9
Sentences of the article WSJ880912-0064.

Sentence ID	Sentence
S1	Hurricane Gilbert swept toward Jamaica yesterday with 100-mile-an-hour winds, and officials issued warnings to residents on the southern coasts of the Dominican Republic, Haiti and Cuba.
S2	The storm ripped the roofs off houses and caused coastal flooding in Puerto Rico.
S3	In the Dominican Republic, all domestic flights and flights to and from Puerto Rico and Miami were canceled.
S4	Forecasters said the hurricane was gaining strength as it passed over the ocean and would dump heavy rain on the Dominican Republic and Haiti as it moved south of Hispaniola, the Caribbean island they share, and headed west.
S5	"It's still gaining strength.
S6	It's certainly one of the larger systems we've seen in the Caribbean for a long time," said Hal Gerrish, forecaster at the National Hurricane Center in Coral Gables, Fla.
S7	At 3p.m. EDT, the center of the hurricane was about 100 miles south of the Dominican Republic and 425 miles east of Kingston, Jamaica.
S8	The hurricane was moving west at about 15 mph and was expected to continue this motion for the next 24 h.
S9	Forecasters said the hurricane's track would take it about 50 miles south of southwestern Haiti.
S10	The hurricane center said small craft in the Virgin Islands and Puerto Rico should remain in port until conditions improve.
S11	The forecasters said the Dominican Republic would get as much as 10 in. of rain yesterday, with similar amounts falling in Haiti last night and tonight.
S12	Hurricane warnings were issued for the south coast of Haiti and Cuba by their respective governments.
S13	In Jamaica, the government issued a hurricane watch for the entire island.
S14	Tropical Storm Gilbert formed in the eastern Caribbean and strengthened into a hurricane Saturday night.
S15	In Puerto Rico, besides tearing off several roofs, the storm caused coastal flooding and brought down power lines and trees along roads and highways in the west and southwestern regions.
S16	Three people were injured in Guayanilla, Puerto Rico, when a tree fell on their vehicle as they traveled along Route 97, police reported.
S17	Four policemen stationed on Mona Island, between Puerto Rico and the Dominican Republic, were stranded as a result of the weather.

order to maintain the unsupervised nature of the model, the parameters were initially set in a way that gives equal weights to all terms of the sentence scoring equation as per Table 8. Where, the λ weight parameter (λ_1,..., λ_5) was equally divided among the 5n-gram (*WS*, *2GS*,.., *5GS*) terms of the SC equation (1)/5 = 0.2), while all other parameters were set to 1 to maintain a balanced equation.

4.4. Evaluation metrics

The summarization results obtained are evaluated against the human generated summaries using, standard text summarization evaluation metrics known as ROUGE-1 and ROUGE-2 techniques (Rojas-Simon, Ledeneva, & Garcia-Hernandez, 2022).

The metric ROUGE stands for "*Recall-Oriented Understudy for Gisting Evaluation*". It is an evaluation metric for automatic text summarization task that does not require human annotation. ROUGE is considered the most commonly used intrinsic summarization evaluation metric, and was developed by Lin et al. (Lin & Hovy, 2003); Lin, 2004).

ROUGE is inspired by the BLEU metric, "*bilingual evaluation understudy*", used for evaluating machine translation output (Papineni, Roukos, Ward, & Zhu, 2002). ROUGE evaluates the candidate of computer-generated summaries by measuring the amount of *n*-grams overlap between the candidate and human-generated summaries. ROUGE can be applied to measure the overlap of any type of *n-grams*. As such, *ROUGE-1* is used to measure the overlapping *unigrams*, while *ROUGE-2* and *ROUGE-3* are used to measure the overlapping *bigrams* and *trigrams* respectively.

Table 10
Document WSJ880912-0064 Result Analysis.

Item	Value
Number of Sentences	17
Tokens	354
Unique Words	159
Unique word encoded shapes	4
Bigrams	278
Bigrams encoded shapes	12
Trigrams	301
Trigrams encoded shapes	20
4-grams	297
4-grams encoded shapes	32
5-grams	284
5-grams encoded shapes	47
Unique Sentence encoded shapes	1
Unique Paragraph encoded shapes	1

To calculate the ROUGE metric for a computer-generated summary for a document *D*, first, one or more human candidates should summarize that document. Then the amount of overlapping (matching) n-grams between the computer-generated summary and each of the human generated ones is calculated. As such, ROUGE-n metric is equal to the overall sum of the aforementioned overlaps. Equation (15) shows the calculation of ROUGE-2 metric using *bigrams* overlap.

$$ROUGE_2 = \frac{\sum_{S\in\{refsummaries\}}\sum_{bigrams\in S} Count_{match}(Bigrams)}{\sum_{S\in\{refsummaries\}}\sum_{bigrams\in S} Count(Bigrams)} \quad (15)$$

4.5. Performance assessment

In this study, the language agnostic summarization performance of the proposed TxLASM was assessed by comparing the obtained results to that obtained by commercial extractive summarization tools, as well as state-of-the-art extractive summarization approaches.

For the English and Portuguese benchmark datasets, the classification performance was assessed against state-of-the-art approaches that spans the five categories of the extractive summarization tasks as mentioned in subsection 2.3 *Categories of Extractive ATS*.

However, for the Spanish and French datasets, two commercial summarization tool were used, Apple's integrated summarizer within macOS 12 "***Monterey***" and Autosummarizer.com.

5. Results and discussions

In this section, we present in details the summarization of a single English document from DUC 2002 benchmark dataset. This document summarization case is used as a reference case study to be implemented on the entire 567 news articles, as well as the other three languages datasets in the following subsections.

5.1. Reference case study

We applied *TxLASM* to summarize English news article from the DUC 2002 datasets. Article WSJ880912-0064 was used in this case study (sentences are listed in Table 9). In addition, the same article was summarized using the Text Summarizer integrated within Apple's macOS 12 "***Monterey***". Both summaries were compared against human generated summaries using ROUGE-1 and ROUGE-2.

The summarization were done at *DoC = 100e*, meaning that the

Fig. 5. Unigram frequency graph for encoded word shapes.

summary should contain sentences with around 100 extracted words. The summaries are evaluated using ROUGE-1 and ROUGE-2 using the two human generated summaries, provided by DUC.

As seen in Table 10, by running the *TxLASM* on the selected article, the model has extracted 17 sentences, 354 words, 159 tokens (unique words) and 4 distinct word encoded shapes. The four distinct encoded shapes are = {Xxc, xc, Nn, Nnxc}, where:

a) "Xxc" was repeated 51 times, as this code represents 34 named entities as Dominican, Republic, Gilbert and 17 words that start each of the 17 sentences in the article.
b) "xc" was repeated 293 times, as this code represents the majority of words in the article, as such it gets less weight.
c) "Nn" was repeated 8 times, as this code represents the numbers that were stated in the article as {3, 425, 15, …}. This shape receives relatively higher weight when compared to the previous two shapes.
d) "Nnxc" a very rare shape code that occurred only once, encoding the word "100-mile-an-hour" that appeared in the first sentence, this shape code receives the highest shape score.

Fig. 5 shows the word frequency graph for the unigrams of the original article before summarization.

Moreover, hundreds of n-grams forms and encoded shapes were extracted. In case of bigrams, the most abundant shape was "xc xc" as it represents the majority of the text. However, Xxc Xxc was significant in identifying NE's as "Dominican Republic", "Hurricane Gilbert", "Coral Gables", etc. While in the case of trigrams, the shape "Xxc Xxc Xxc" was capable of identifying an important NE in this context, "National Hurricane Center".

It is worth mentioning that the number of unique shapes for sentences and paragraphs is only 1 for both, as all sentences start with

Fig. 6. Word frequency graph for article WSJ880912-0064.

Table 11
The top five Sentences of the Document.

ID	Score	Sentence
S1	0.959	Hurricane Gilbert swept toward Jamaica yesterday with 100-mile-an-hour winds, and officials issued warnings to residents on the southern coasts of the Dominican Republic, Haiti and Cuba.
S7	0.814	At 3p.m. EDT, the center of the hurricane was about 100 miles south of the Dominican Republic and 425 miles east of Kingston, Jamaica.
S6	0.189	It's certainly one of the larger systems we've seen in the Caribbean for a long time," said Hal Gerrish, forecaster at the National Hurricane Center in Coral Gables, Fla.
S16	0.159	Three people were injured in Guayanilla, Puerto Rico, when a tree fell on their vehicle as they traveled along Route 97, police reported.
S10	0.130	The hurricane center said small craft in the Virgin Islands and Puerto Rico should remain in port until conditions improve.

capital letter and dominated by lower case words, thus the only sentence shape in the text was "***Zzz***".

While, all paragraphs have the same shape "***OSPpp***", i.e. ordinary single sentence dominated by lower case letters despite a capital initial letter. The reason for obtaining this single sentence paragraph encoded shape is that the article was originally provided, from DUC, as segmented text with every sentence form its own paragraph.

In addition, Fig. 6 shows the unigram influence graph. It is clear that the most influential words are Hurricane (repeated 9 times with XC shape), Dominican (6), Republic (6), Puerto (6), Rico (6). This high influence is reflected when computing their individual scores, in terms of their word and encoded shape frequency weights.

As such, and after computing the overall sentence score for each of the 17 sentences, the top scored sentences are listed in Table 11 (in descending order of their score), where it is clear that *TxLASM* was capable of identifying the most influential NE's and CC's.

Since the summary is limited to 100e words only, thus only the top four sentences were selected and then ordered according to their original order in the article, to generate the summary.

As discussed above and shown in Fig. 6, the model was capable of identifying influential terms due to the rareness of their encoded shapes and forms. Among the top important terms/entities that the model has successfully identified during the scoring process, are: *Hurricane Gilbert*, *Dominican Republic*, *Hal Gerrish*, *Jamaica*, *Caribbean*, *Puerto Rico*, *Virgin Islands*, etc.

The generated summary by *TxLASM* is as follows:

"*Hurricane Gilbert swept toward Jamaica yesterday with 100-mile-an-hour winds, and officials issued warnings to residents on the southern coasts of the Dominican Republic, Haiti and Cuba. It's certainly one of the larger systems we've seen in the Caribbean for a long time," said Hal Gerrish, forecaster at the National Hurricane Center in Coral Gables, Fla. At 3p.m. EDT, the center of the hurricane was about 100 miles south of the Dominican Republic and 425 miles east of Kingston, Jamaica. Three people were injured in Guayanilla, Puerto Rico, when a tree fell on their vehicle as they traveled along Route 97, police reported.*"

The extracted summary was then evaluated using ROUGE-1 and ROUGE-2 against human-generated summaries and compared to the summaries done by Apple macOS 12 integrated summarizer. The results are shown in Table 12 below:

As seen from the results above, TxLASM was able to summarize the

Table 12
Evaluating the results of a single English document summarization.

Standard File	Tool	ROUGE 1	ROUGE-2
A	TxLASM	*61.2*	*36.3*
	macOS Summarizer	49.0	18.6
B	TxLASM	*51.0*	*18.0*
	macOS Summarizer	46.8	14.6
Total	TxLASM	*56.3*	*27.7*
	macOS Summarizer	47.9	16.8

Table 13
Comparing *TxLASM* against state-of-the-art methods applied on the entire 567 articles in the DUC2002 benchmark dataset for single document summarization task.

Category	*Model*	*ROUGE-1*	*ROUGE-2*	*REF*
Statistical-Based	***TxLASM*** (Language agnostic elements scoring)	***53.54***	*25.97*	–
Topic based	Topic modeled unsupervised clustering	49.35	**31.53**	(Srivastava, Singh, Rana, & Kumar, 2022)
	DeepSum (topic modeling and word embedding)	53.2	28.7	(Joshi, Fidalgo, Alegre, & Fernández-Robles, 2023)
Graph based	Topic Modeling based on weighted graph representation	48.10	23.3	(Parveen, Ramsl, & Strube, 2015)
	CoRank (word–sentence relationship and graph-based ranking model)	52.6	25.8	(Fang, Mu, Deng, & Wu, 2017)
Machine Learning based	BERT based extractor and LSTM pointer network	43.39	19.38	(Bae et al., 2019)
	Word2vector embedding	38.25	22.56	(Jain & Bhatia, 2017)
	SummCoder (deep auto-encoders)	51.7	27.5	(Joshi, Fidalgo, Alegre, & Fernández-Robles, 2019)
	SummaRuNNer(RNN-based sequence classifier)	47.4	24.0	(Nallapati, Zhai, & Zhou, 2017)
	HSSAS (Neural Network Classifier)	52.1	24.5	(Al-Sabahi, Zuping, & Nadher, 2018)
	TCNN (combines NN and LexRank)	44.3	19.68	(Mao et al., 2019)
Discourse-Based	FNARS (hierarchical Narrative Summaries – fully structured)	48.3	28.3	(Ghodratnama, Beheshti, Zakershahrak, & Sobhanmanesh, 2021)
	SNARS (hierarchical Narrative Summaries – semi structured)	52.9	24.8	(Ghodratnama, Beheshti, Zakershahrak, & Sobhanmanesh, 2021)

news article successfully, achieving 61% success when compared to human-generated summaries. TxLASM has outperformed Apple's macOS summarizer in both evaluation metrics, ROUGE-1 and ROUGE-2.

This performance is attributed to the ability of the model to identify influential terms and text elements based on the rareness of their encoded shapes and forms, regardless the text's language, domain, topic and/or subtopic.

5.2. Comparing to State-of-the-Art approaches

TxLASM was applied on two benchmark datasets, *DUC2002* for English text summarization, and *TeMario* for Portuguese text summarization. The obtained summaries were evaluated against human generated summaries using ROUGE-1 and ROUGE-2.

The results are then compared to state-of-the-art approaches reported in literature. The state-of-the-art methods were chosen to span the different categories of extractive summarization approaches discussed in chapter 2.

Table 13 and Table 14 compare the results of TxLASM to those reported in literature for DUC2002 and *TeMario* datasets respectively.

With respect to DUC 2002 benchmark dataset, and as seen from the Table 13, Fig. 7 and Fig. 8, the proposed TxLASM has outperformed all state-of-the-art models in terms of ROUGE-1 metric, and scored better performance than 69% of those models in terms of ROUGE-2 metric.

Due to the language and domain agnostic nature of *TxLASM*, the model was capable of identifying influential terms and text elements (n-grams, CC, NE, sentences) regardless the text's language, domain, topic, subtopic and/or sentence structure. As such, *TxLASM* has outperformed Topic and Graph based state-of-the-art approaches due to the diversity of topics/subtopics of News articles that are better summarized by topic agnostic unsupervised models as *TxLASM*.

Moreover, the diverse and irregular nature of news articles (in terms of structure, language and foreign terms, OOV, etc.) gives more advantage for unsupervised and domain/language agnostic models as TxLASM over structured ATS tools. As such, *TxLASM* outperformed Discourse-based techniques as SNARS and FNARS.

On the other hand, and with respect to Portuguese Language, TxLASM was applied to the *TeMario* benchmark dataset. The generated summaries were 25–30% of the size of the original documents.

The obtained results were then compared to that reported in literature (Aparício, et al., 2016), Table 14, Fig. 9 and Fig. 10 shows that the proposed TxLASM has scored better results than all state-of-the-art approaches reported in literature on the *TeMario* dataset.

5.3. Applying TxLASM on Spanish and French datasets

Regarding, Spanish and French Languages, TxLASM was applied to generate a summary of 100 words. Table 15 compares the summarization performance of TxLASM to commercial tools applied on the same datasets.

The depicted results show that our proposed model has achieved superior results when compared to those obtained by commercial tools, proving the efficiency of the language agnostic nature of the model.

6. Conclusion

In this paper, a novel ***Text Documents-Language Agnostic Summarization Model (TxLASM)*** is proposed to perform extractive text summarization in a language and domain agnostic manner. *TxLASM* generates an efficient language and domain independent extractive summary when evaluated against human generated summaries of the same text. *TxLASM* encodes and extracts specific features of major text elements (paragraphs, sentences, n-grams and words) using an innovative technique for encoding the shapes of those elements. *Shape-Coding* technique is performed by encoding text elements using a handful set of codes, and normalizing those shapes to fit into relatively small number of encoded classes. The abundance/rareness of those classes reflects the degree of importance of the encoded tokens.

The proposed model does not require any particular language-dependent preprocessing tools, due to its ability to neutralize the effect of stop words (unimportant words) without using stop words lexicons that are by definition language and/or context dependent. Moreover, the model includes a pre-processing algorithm that groups word derivatives together, in a step very similar to stemming, without using language dictionaries and/or hand-coded stemmer tools.

As such, *TxLASM* preserves the relative importance of potential text elements, with the ability to extract influential key phrases without any

Table 14
Comparing TxLASM to state-of-the-art methods applied to Temario Portuguese benchmark dataset for single document summarization task.

Model	**ROUGE-1**	**ROUGE-2**
TxLASM	***0.57***	***0.22***
MMR ($\lambda = 0.5$)	0.43	0.15
Support Sets (Manhattan Distance and Support set cardinality = 2)	0.52	0.19
KP-Centrality (10 Key Phrases)	0.54	0.20
LSA	0.56	0.20
GRASSHOPPER	0.54	0.19
LexRank	0.55	0.20

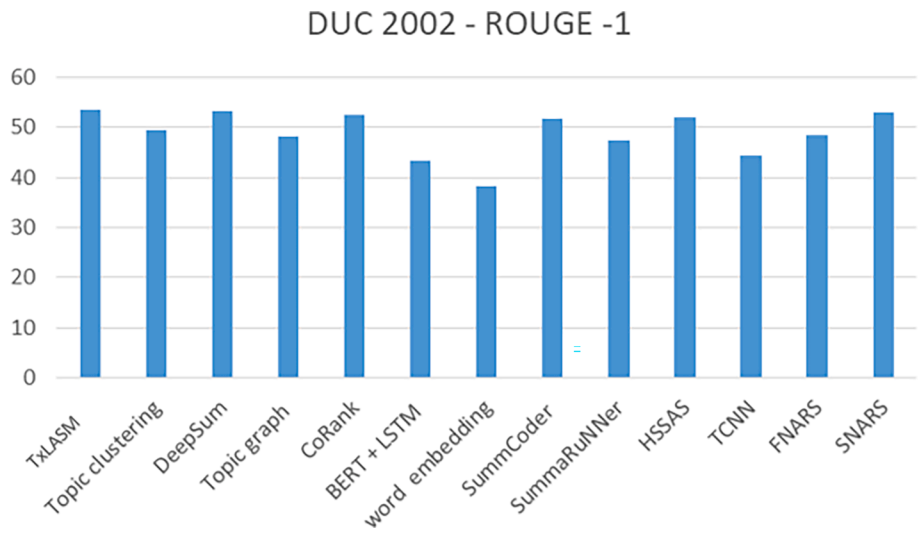


Fig. 7. ROUGE-1 results for TxLASM against state-of-the-art approaches applied on DUC 2002 benchmark dataset.

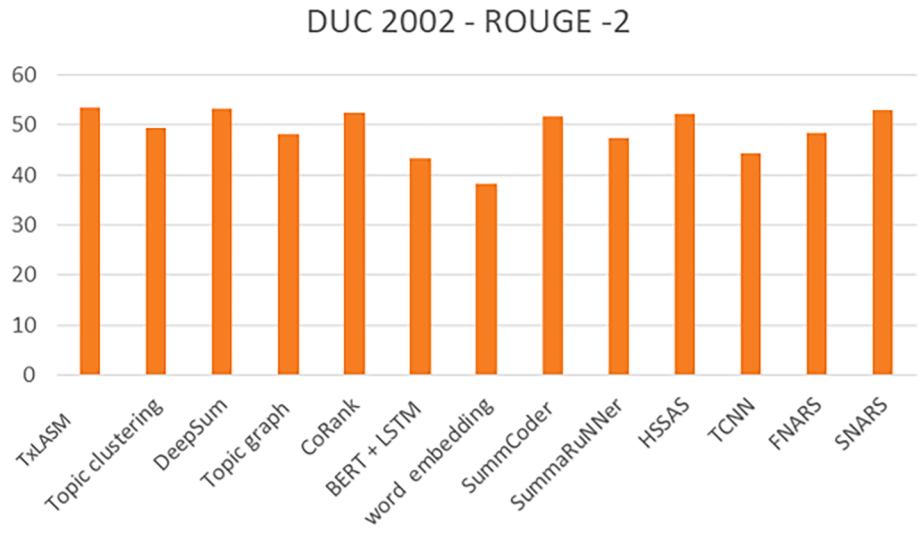


Fig. 8. ROUGE-2 results for TxLASM against state-of-the-art approaches applied on DUC 2002 benchmark dataset.

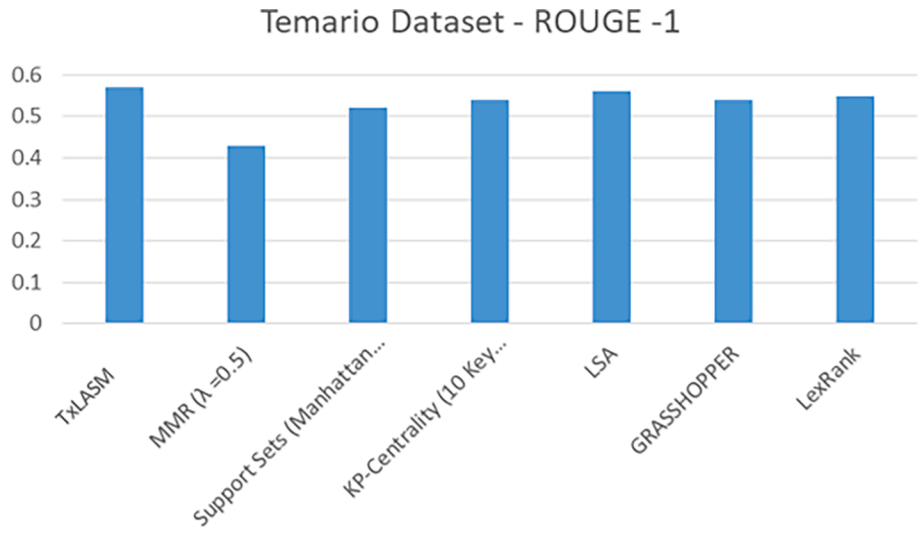


Fig. 9. ROUGE-1 results for TxLASM against state-of-the-art approaches applied on Temario Portuguese benchmark dataset.

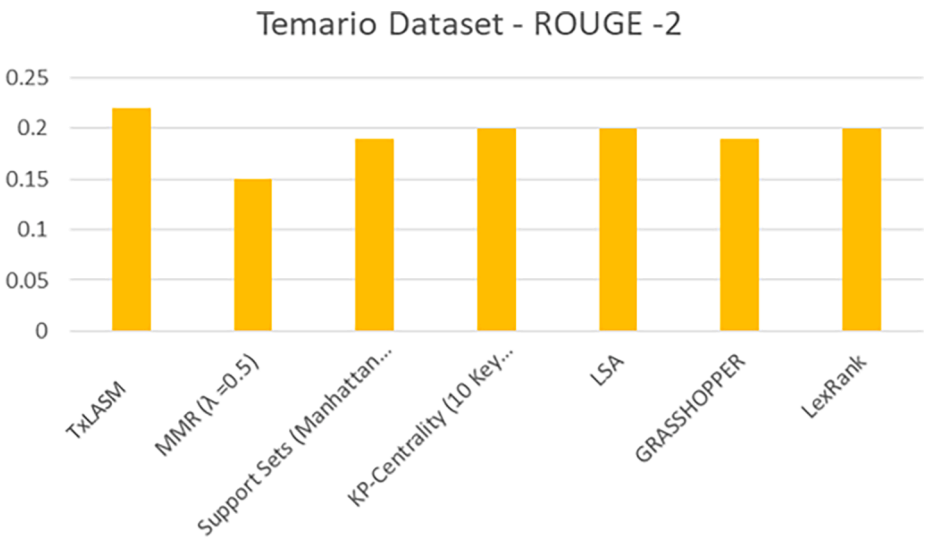


Fig. 10. ROUGE-2 results for TxLASM against state-of-the-art approaches applied on Temario Portuguese benchmark dataset.

Table 15
TxLASM summarization performance for Spanish and French datasets.

Dataset	Tool	ROUGE-1	ROUGE-2
Spanish News Dataset	*TxLASM*	*0.6778*	*0.4652*
	Apple macOS 12	0.5310	0.2451
	Autosummarizer	0.5377	0.2640
French News Dataset	*TxLASM*	*0.5762*	*0.3389*
	Apple macOS 12	0.5075	0.2760
	Autosummarizer	0.4577	0.2221

sort of dependency on language dependent databases or corpora.

TxLASM was tested on news datasets written in English, Portuguese, French and Spanish. The obtained results were evaluated against human-generated summaries using ROUGE-1 and ROUGE-2 metrics. In case of English and Portuguese, the results were compared to 18 state-of-the-art models and systems listed in recent literature, such models represent the five categories of the ATS task. While, the results of French and Spanish languages were compared to those obtained by Apple macOS 12 integrated summarizer as well as the online Automatic summarizer.

TxLASM achieved better performance over other tools in all of the four languages, without using any domain specific or language related lexicons, parsers or corpora, which proves the quality of the proposed contributions. This performance is attributed to the ability of the model to identify influential terms and text elements based on the rareness of their encoded shapes and forms, regardless the text's language, domain, topic and/or subtopic.

Future research using the *TxLASM* could tackle the following points: *i)* extending the boundaries of the model to solve multiple documents summarization tasks. *ii)* Multiple-document summarization could be extended and applied on long texts or sets of documents that contain mixed languages or context, as in the case of scientific papers and language books. In addition, c) expand the model's application domain to include oriental languages as Arabic, Persian, etc.

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.

Data availability

Data will be made available on request.

Acknowledgment

This work has been partially supported by Brazilian National Council for Scientific and Technological Development (CNPq), under the grant number 309545/2021-8.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.eswa.2023.121433.

References

AbdelFattah, M., & Ren, F. (2009). GA, MR, FFNN, PNN and GMM based models for automatic text summarization. *Computer Speech & Language, 23*(1), 126–144.
Adhikari, S. (2020). Nlp based machine learning approaches for text summarization. *Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, (pp. 535-538)). IEEE.
Ailem, M., Zhang, B., & Sha, F. (2019). Topic augmented generator for abstractive summarization. *arXiv preprint* .
Akter, S., Asa, A. S., Uddin, M. P., Hossain, M. D., Roy, S. K., & Afjal, M. I. (2017). An extractive text summarization technique for Bengali document (s) using K-means clustering algorithm. *IEEE International Conference on Imaging*.
Alami, N., Meknassi, M., En-nahnahi, N., El Adlouni, Y., & Ammor, O. (2021). Unsupervised neural networks for automatic Arabic text summarization using document clustering and topic modeling. *Expert Systems with Applications, 172*.
Alguliyev, R. M., Aliguliyev, R. M., Isazade, N. R., Abdi, A., & Idris, N. (2019). COSUM: Text summarization based on clustering and optimization. *Expert Systems, 36*(1).
Alomari, A., Idris, N., Sabri, A. Q., & Alsmadi, I. (2022). Deep reinforcement and transfer learning for abstractive text summarization: A review. *Computer Speech & Language, 71*.
Al-Sabahi, K., Zuping, Z., & Nadher, M. (2018). A hierarchical structured self-attentive model for extractive document summarization (HSSAS). *IEEE Access*, 24205–24212.
Alshalabi, H., Tiun, S., Omar, N., AL-Aswadi, F. N., & Alezabi, K. A. (2022). Arabic light-based stemmer using new rules. *Journal of King Saud University-Computer and Information Sciences., 34*(9), 6635–6642.
Alshemali, B., & Kalita, J. (2020). Improving the reliability of deep neural networks in NLP: A review. *Knowledge-Based Systems, 191*.
Aparício, M., Figueiredo, P., Raposo, F., de Matos, D. M., Ribeiro, R., & Marujo, L. (2016). Summarization of films and documentaries based on subtitles and scripts. *Pattern Recognition Letters, 73*, 7–12.
Bae, S., Kim, T., Kim, J., & Lee, S. G. (2019). Summary level training of sentence rewriting for abstractive summarization. *arXiv preprint* .
Bahcevan, C. A., Kutlu, E., & Yildiz, T. (2018). Deep neural network architecture for part-of-speech tagging for turkish language. In *3rd International Conference on Computer Science and Engineering (UBMK)* (pp. 235–238). Bosnia and Herzegovina.

Baykara, B., & Güngör, T. (2022). Abstractive text summarization and new large-scale datasets for agglutinative languages Turkish and Hungarian. *Language Resources and Evaluation*, 1–35.

Belwal, R. C., Rai, S., & Gupta, A. (2021a). A new graph-based extractive text summarization using keywords or topic modeling. *Journal of Ambient Intelligence and Humanized Computing, 12*(10), 8975–8990.

Belwal, R. C., Rai, S., & Gupta, A. (2021b). Text summarization using topic-based vector space model and semantic measure. *Information Processing & Management, 58*(3).

Belwal, R. C., Rai, S., & Gupta, A. (2022). Extractive text summarization using clustering-based topic modeling. *Soft Computing*, 1–18.

Bergmanis, T., & Goldwater, S. (2018). Context sensitive neural lemmatization with lematus. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (p. 1).

Besharati, S., Veisi, H., Darzi, A., & Saravani, S. H. (2021). A hybrid statistical and deep learning based technique for Persian part of speech tagging. *Iran Journal of Computer Science, 4*(1), 35–43.

Brack, A., Hoppe, A., Stocker, M., Auer, S., & Ewerth, R. (2022). Analysing the requirements for an Open Research Knowledge Graph: Use cases, quality requirements, and construction strategies. *International Journal on Digital Libraries, 23*(1), 33–55.

Cai, T., Shen, M., Peng, H., Jiang, L., & Dai, Q. (2019). Improving transformer with sequential context representations for abstractive text summarization. In *CCF International Conference on Natural Language Processing and Chinese Computing* (pp. 512–524). Cham: Springer.

Chen, J., & Yang, D. (2021). Structure-aware abstractive conversation summarization via discourse and action graphs. *arXiv preprint*.

Chiche, A., & Yitagesu, B. (2022). Part of speech tagging: A systematic review of deep learning and machine learning approaches. *Journal of Big Data, 9*(1), 1–25.

Chopra, S., Auli, M., & Rush, A. M. (2016). Abstractive sentence summarization with attentive recurrent neural networks. In *Proceeding of 2016 conference of the North American Chapter of the Association for Computational Linguistics* (pp. 93–98). California: Human Language Technologies.

Chotirat, S., & Meesad, P. (2021). Part-of-Speech tagging enhancement to natural language processing for Thai wh-question classification with deep learning. *Heliyon, 7*(10).

Constantin, A., Peroni, S., Pettifer, S., Shotton, D., & Vitali, F. (2016). The document components ontology (DoCO). *Semantic web, 7*(2), 167–181.

Deshpande, A. A., & Kottawar, V. G. (2021). Survey of Sentence Scoring Techniques for Extractive Text Summarization. In *Proceeding of International Conference on Computational Science and Applications* (pp. 65–77). Singapore: Springer.

Dong, T., Shan, S., Liu, Y., Qian, Y., & Ma, A. (2021). A Pointer-Generator Based Abstractive Summarization Model with Knowledge Distillation. In *International Conference on Neural Information Processing* (pp. 168–177). Cham: Springer.

El-Kassas, W. S., Salama, C. R., Rafea, A. A., & Mohamed, H. K. (2020). EdgeSumm: Graph-based framework for automatic text summarization. *Information Processing & Management, 57*(6).

El-Kassas, W. S., Salama, C. R., Rafea, A. A., & Mohamed, H. K. (2021). Automatic text summarization: A comprehensive survey. *Expert Systems with Applications, 165*.

Erkan, G., & Radev, D. (2004). LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research, 22*, 457–479.

Fang, C., Mu, D., Deng, Z., & Wu, Z. (2017). Word-sentence co-ranking for automatic extractive text summarization. *Expert Systems with Applications, 72*, 189–195.

Feng, X., Feng, X., Qin, B., Geng, X., & Liu, T. (2021). Dialogue discourse-aware graph convolutional networks for abstractive meeting summarization. *arXiv preprint*.

Friederici, A. D. (2002). Towards a neural basis of auditory sentence processing. *Trends in Cognitive Sciences, 6*, 78–84.

Friederici, A. D. (2011). The Brain Basis Of Language Processing: From Structure To Function. *Physiol, 91*, 1357–1392.

Gambhir, M., & Gupta, V. (2017). Recent automatic text summarization techniques: A survey. *Artificial Intelligence Review, 47*(1), 1–66.

Ghodratnama, S., Beheshti, A., Zakershahrak, M., & Sobhanmanesh, F. (2021). Intelligent narrative summaries: From indicative to informative summarization. *Big Data Research, 26*.

Grail, Q. P. (2021). Globalizing BERT-based transformer architectures for long document summarization. *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*, (p. Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volu).

Groza, T., Handschuh, S., Möller, K., & Decker, S. (2007). SALT-Semantically Annotated $mbox {LaTeX} $ for Scientific Publications. In *European Semantic Web Conference* (pp. 518–532). Berlin: Springer.

Gupta, R., & Jivani, A. G. (2022). LemmaQuest Lemmatizer: A Morphological Analyzer Handling Nominalization. *IETE Journal of Research*, 1–9.

Hou, S., Zhang, S., & Fei, C. (2020). Rhetorical structure theory: A comprehensive review of theory, parsing methods and applications. *Expert Systems with Applications, 157*.

Hsu, W. T., Lin, C. K., Lee, M. Y., Min, K., Tang, J., & Sun, M. (2018). A unified model for extractive and abstractive summarization using inconsistency loss. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. Melbourne, Australia.

Ishigaki, T., Kamigaito, H., Takamura, H., & Okumura, M. (2019). Discourse-aware hierarchical attention network for extractive single-document summarization. *Proceedings of the International Conference on Recent Advances in Natural Language Processing.*, (pp. 497-506).

Issam, K., & Patel, S. (2020). Topic modeling based extractive text summarization. *The International Journal of Innovative Technology and Exploring Engineering, 9*.

Jain, A., & Bhatia, D. T. (2017). Extractive text summarization using word vector embedding. In *International Conference on machine learning and data science (MLDS)* (pp. 51–55). IEEE.

Janaki Raman, K., & Meenakshi, K. (2021). *Automatic text summarization of article (NEWS) using lexical chains and wordnet—A review*. Singapore: Springer.

Joshi, A., Fidalgo, E., Alegre, E., & Alaiz-Rodriguez, R. (2022). RankSum—An unsupervised extractive text summarization based on rank fusion. *Expert Systems with Applications, 200*.

Joshi, A., Fidalgo, E., Alegre, E., & Fernández-Robles, L. (2019). SummCoder: An unsupervised framework for extractive text summarization based on deep auto-encoders. *Expert Systems with Applications, 129*, 200–215.

Joshi, A., Fidalgo, E., Alegre, E., & Fernández-Robles, L. (2023). DeepSumm: Exploiting topic models and sequence to sequence networks for extractive text summarization. *Expert Systems with Applications, 211*.

Jumadi, J., Maylawati, D. S., Pratiwi, L. D., & Ramdhani, M. A. (2021). Comparison of Nazief-Adriani and Paice-Husk algorithm for Indonesian text stemming process. *IOP Conference Series*. Materials Science and Engineering.

Junaida, M. K., & Babu, A. P. (2021). A Deep Learning Approach to Malayalam Parts of Speech Tagging. In *Second International Conference on Networks and Advances in Computational Technologies* (pp. 243–250). Cham: Springer.

Jurafsky, D., & Martin, J. (2021). *Speech and Language Processing, 3rd Edition* (2nd ed.). New Jersy: Pearson.

Kaur, J., & Buttar, P. K. (2018). A systematic review on stopword removal algorithms. *International Journal on Future Revolution in Computer Science & Communication Engineering, 4*(4), 207–210.

Khyani, D., Siddhartha, B. S., Niveditha, N. M., & Divya, B. M. (2021). An Interpretation of Lemmatization and Stemming in Natural Language Processing. *Journal of University of Shanghai for Science and Technology., 22*(10), 350–357.

Kluyver, T., Ragan-Kelley, B., & Fernando, P. (2016). Jupyter Notebooks – a publishing format for reproducible computational workflows. In *Positioning and Power in Academic Publishing: Players, Agents and Agendas* (pp. 87–90). F. Loizides and B. Schmidt.

Kouris, P., Alexandridis, G., & Stafylopatis, A. (2022). Abstractive text summarization: Enhancing sequence-to-sequence models using word sense disambiguation and semantic content generalization. *Computational Linguistics, 47*(4), 813–859.

Kumar, A., & Gupta, M. K. (2021). Abstractive Summarization System. *Journal of Electronics, 3*(4), 309–319.

Kumar, G. K., & Rani, D. M. (n.d.). Paragraph summarization based on word frequency using NLP techniques. *AIP conference proceedings. 2317*. AIP Publishing LLC.

Ladani, D. J., & Desai, N. P. (2020). Stopword identification and removal techniques on tc and ir applications: A survey. In *6Th International Conference on Advanced Computing and Communication Systems (ICACCS)* (pp. 466–472). IEEE.

Lemmatizer, S. (2022). *spaCy API Lemmatizer*. Retrieved 10 25, 2022, from https://spacy.io/api/.

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. *ACL - Workshop on Text Summarization Branches Out*.

Lin, C.-Y., & Hovy, E. (2000). The automated acquisition of topic signatures for text summarization. *Proceedings of the 18th conference on Computational linguistics*, (pp. 495-501).

Lin, C.-Y., & Hovy, E. H. (2003). *Automatic evaluation of summaries using N-gram co-occurrence statistics*. HLT NAACL.

Liu, Z., & Chen, N. (2021). Exploiting discourse-level segmentation for extractive summarization. *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, (pp. 116-121).

Lkhagvasuren, G., Rentsendorj, J., & Namsrai, O. E. (2021). Mongolian Part-of-Speech Tagging with Neural Networks. In *Advances in Intelligent Information Hiding and Multimedia Signal Processing* (pp. 109–115). Singapore: Springer.

Lovins, J. (1968). Development of a stemming algorithm. *Mech. Trans. Comput. Linguist., 11*(21), 22–31.

Ma, C., Zhang, W. E., Guo, M., Wang, H., & Sheng, Q. Z. (2020). Multi-document summarization via deep learning techniques: A survey. *ACM Computing Surveys (CSUR)*.

Ma, T., Pan, Q., Rong, H., Qian, Y., Tian, Y., & Al-Nabhan, N. (2021). T-bertsum: Topic-aware text summarization based on bert. *IEEE Transactions on Computational Social Systems, 9*(3), 879–890.

Mallick, C., Das, A. K., Dutta, M., Das, A. K., & Sarkar, A. (2019). Graph-based text summarization using modified TextRank. In *Soft computing in data analytics* (pp. 137–146). Singapore: Springer.

Mann, W., & Thompson, S. (1988). Rhetorical structure theory: Toward a functional theory of text organization. *Text, 8*, 243–281.

Mao, X., Yang, H., Huang, S., Liu, Y., & Li, R. (2019). Extractive summarization using supervised and unsupervised learning. *Expert systems with applications, 133*, 173–181.

Martinez, A. R. (2012). Part-of-speech tagging. *Wiley Interdisciplinary Reviews: Computational Statistics, 4*(1), 107–113.

Microsoft, C. (2016). *Microsoft Excel*. Seattle.

Mohamed, M., & Oussalah, M. (2019). SRL-ESA-TextSum: A text summarization approach based on semantic role labeling and explicit semantic analysis. *Information Processing & Management, 56*(4), 1356–1372.

Moratanch, N. &. (2017). A survey on extractive text summarization. *International conference on computer, communication and signal processing (ICCCSP)*, (pp. 1-6). Chennai.

Nallapati, R., Zhai, F., & Zhou, B. (2017). Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In. *Thirty-first AAAI conference on artificial intelligence*.

Namly, D., Bouzoubaa, K., & Yousfi, A. (2019). A bi-technical analysis for arabic stop-words detection. *Compusoft, 8*(5), 3126–3134.

Nasar, Z., Jaffry, S. W., & Malik, M. K. (2019). Textual keyword extraction and summarization: State-of-the-art. *Information Processing & Management, 56*(6).

NIST. (2002). *DUC 2002*. Retrieved Nov 1, 2015, from NIST: http://www-nlpir.nist.gov/projects/duc/past_duc/duc2002/test.html.

Pant, G., Srinivasan, P., & Menczer, F. (2004). Crawling the Web. In M. Levene, & A. Poulovassilis, *Web Dynamics: Adapting to Change in Content, Size, Topology and Use* (pp. 153-178).

Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002). BLEU: a method for automatic evaluation of machine translation. *40th Annual meeting of the Association for Computational Linguistics.*, (pp. 311–318). Philadelphia.

Parveen, D., Ramsl, H. M., & Strube, M. (2015). Topical coherence for graph-based extractive summarization. *Proceedings of the 2015 conference on empirical methods in natural language processing.*, (pp. 1949-1954).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830.

Pota, M., Marulli, F., Esposito, M., De Pietro, G., & Fujita, H. (2019). Multilingual POS tagging by a composite deep architecture based on character-level features and on-the-fly enriched Word Embeddings. *Knowledge-Based Systems, 164*, 309–323.

Priyadarshi, A. &. (2022). A study on the performance of Recurrent Neural Network based models in Maithili Part of Speech Tagging. . *Transactions on Asian and Low-Resource Language Information Processing*.

Qaroush, A., Farha, I. A., Ghanem, W., Washaha, M., & Maali, E. (2021). An efficient single document Arabic text summarization using a combination of statistical and semantic features. *Journal of King Saud University-Computer and Information Sciences, 33*(6), 677–692.

Qi, M., Liu, H., Fu, Y., & Liu, T. (2021). Improving Abstractive Dialogue Summarization with Hierarchical Pretraining and Topic Segment. *Findings of the Association for Computational Linguistics*, (pp. 1121-1130).

Qu, C., Lu, L., Wang, A., Yang, W., & Chen, Y. (2022). *Novel multi-domain attention for abstractive summarisation*. CAAI Transactions on Intelligence Technology.

Rahimi, S. R., Mozhdehi, A. T., & Abdolahi, M. (2017). An overview on extractive text summarization. In *IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI)* (pp. 54–62).

Rajani Shree, M., & Shambhavi, B. R. (2021). POS Tagger Model for South Indian Language Using a Deep Learning Approach. In *ICCCE* (pp. 155–168). Singapore: Springer.

Rani, R. (2021). An extractive text summarization approach using tagged-LDA based topic modeling. *Multimedia tools and applications, 80*(3), 3275–3305.

Rojas-Simon, J., Ledeneva, Y., & Garcia-Hernandez, R. (2022). State-of-the-art Automatic Evaluation Methods. In *Evaluation of Text Summaries Based on Linear Optimization of Content Metrics* (pp. 107–136). Cham: Sprinher.

Saidi, R., Jarray, F., & Mansour, M. (2021). A BERT based approach for Arabic POS tagging. In *International Work-Conference on Artificial Neural Networks* (pp. 311–321). Cham: Springer.

Serek, A., Issabek, A., Akhmetov, A., & Sattarbek, A. (2021). Part-of-speech tagging of Kazakh text via LSTM network with a bidirectional modifier. In *16Th International Conference on Electronics Computer and Computation (ICECCO)* (pp. 1–4).

Sinclair, S., & Rockwell, G. (2022). *Voyant Tool v 2.6.1*. https://voyant-tools.org/.

Singh, P., Rutten, G., & Lefever, E. (2021). A Pilot Study for BERT Language Modelling and Morphological Analysis for Ancient and Medieval Greek. In *The 5th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature* (pp. 128–137). Association for Computational Linguistics.

Smelyakov, K., Karachevtsev, D., Kulemza, D., Samoilenko, Y., Patlan, O., & Chupryna, A. (2020). Effectiveness of preprocessing algorithms for natural language processing applications. In *IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T)* (pp. 187–191). IEEE.

Srivastava, R., Singh, P., Rana, S., & Kumar, V. (2022). A topic modeled unsupervised approach to single document extractive text summarization. *Knowledge-Based Systems*, 246.

Suleiman, D., & Awajan, A. (2020). Deep learning based abstractive text summarization: Approaches, datasets, evaluation measures, and challenges. *Mathematical Problems in Engineering*.

The MathWorks, I. (2015). *MATLAB R2015a*. Massachusetts: Natick.

Tomer, M., & Kumar, M. (2022). Multi-document extractive text summarization based on firefly algorithm. *Journal of King Saud University-Computer and Information Sciences, 34*(8), 6057–6065.

Uçkan, T., & Karcı, A. (2020). Extractive multi-document text summarization based on graph independent sets. *Egyptian Informatics Journal, 21*(3), 145–157.

Vasić, D., Žitko, B., Grubišić, A., Stankov, S., Gašpar, A., Šarić-Grgić, I., & Markić-Vučić, M. (2021). Croatian POS Tagger as a Prerequisite for Knowledge Extraction in Intelligent Tutoring Systems. In *International Conference on Human-Computer Interaction* (pp. 334–345). Cham: Springer.

Voutilainen, A. (2003). Part-of-Speech Tagging. In R. Mitkov, *The Oxford Handbook of Computational Linguistics* (pp. 219-231).

Warjri, S., Pakray, P., Lyngdoh, S. A., & Maji, A. K. (2021). Part-of-speech (pos) tagging using conditional random field (crf) model for khasi corpora. *International Journal of Speech Technology, 24*(4), 853–864.

Wazery, Y. M., Saleh, M. E., Alharbi, A., & Ali, A. A. (2022). Abstractive Arabic Text Summarization Based on Deep Learning. *Computational Intelligence and Neuroscience*.

Xu, J., Gan, Z., Cheng, Y., & Liu, J. (2019). Discourse-aware neural extractive text summarization. *arXiv preprint*.

Xu, W., Xiong, C., & Cheng, H. (2021). Research on Chinese Text Summarization Based on Core Word Attention Mechanism. In 2021. *16th International Conference on Computer Science & Education (ICCSE)* , (pp. 859-863).

Zhang, M., Li, X., Yue, S., & Yang, L. (2020). An empirical study of TextRank for keyword extraction. *IEEE Access, 8*, 178849–178858.

Zhang, M., Zhou, G., Yu, W., Huang, N., & Liu, W. (2022). A Comprehensive Survey of Abstractive Text Summarization Based on Deep Learning. *Computational Intelligence and.* Neuroscience.

Zhou, Q., Yang, N., Wei, F., Huang, S., Zhou, M., & Zhao, T. (2018). Neural document summarization by jointly learning to score and select sentences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics* (pp. 654–663).

Zou, Y., Zhao, L., Kang, Y., Lin, J., Peng, M., Jiang, Z., & Liu, X. (2021). Topic-oriented spoken dialogue summarization for customer service with saliency-aware topic modeling. *Proceedings of the AAAI Conference on Artificial Intelligence, 35*, 14665–14673.