Scientometrics (2021) 126:135–156
https://doi.org/10.1007/s11192-020-03732-x

A comparative study of abstractive and extractive summarization techniques to label subgroups on patent dataset

Cinthia M. Souza[1] · Magali R. G. Meireles[1] · Paulo E. M. Almeida[2]

Received: 23 December 2019 / Published online: 17 October 2020

Abstract
Patents are an important source of information for measuring the technological advancement of a specific knowledge domain. To facilitate the search for information in patent datasets, classification systems separate documents into groups according to the area of knowledge, and designate names to define their content. The increase in the number of patented inventions leads to the need to subdivide these groups. Since these groups belong to a restricted knowledge domain, naming the generated subcategories can be extremely laborious. This work aims to compare the performance of abstractive and extractive summarization techniques in the task of generating sentences directly associated with the content of patents. The abstractive summarization model was composed by a Seq2Seq architecture and a LSTM network. The training was conducted with a dataset of patent titles and abstracts. The validation process was performed using the ROUGE set of metrics. The results obtained by the generated model were compared with the sentence resulting from an extractive summarization algorithm applied to the task of naming patent groups. The main idea was to help the specialist to name new patent groups created by the clustering systems. The naming experiments were performed on the dataset of abstracts of patent documents. Comparative experiments were conducted using four subgroups of the United States Patent and Trademark Office, which uses the Cooperative Patent Classification system.

Keywords Computational intelligence · Knowledge representation · Information systems · Automatic text summarization · Patent datasets

✉ Magali R. G. Meireles
magali@pucminas.br

[1] Pontifical Catholic University of Minas Gerais, Belo Horizonte, MG, Brazil

[2] Federal Center for Technological Education of Minas Gerais, Belo Horizonte, MG, Brazil

Introduction

Patents are an important knowledge source and, therefore, their analysis has been considered a useful tool for research and for management development. Patents are one of the most effective ways to protect an invention today (Wang et al. 2019). One of the objectives of granting patents is to facilitate the dissemination of scientific knowledge (Ouellette 2017). However, finding information in these documents is becoming an increasingly complex task due to the large number of patents in datasets (Sjögren et al. 2018). These documents have a complex language with excessive descriptive technical details and idiosyncrasies that report to the structure of the patent document and the length of the sentences. Thereafter, the retrieval process and analysis of these documents are time consuming and laborious (Codina-Filbà et al. 2017; Gomez 2019).

The efficient analysis of these documents allows for monitoring technological trends, defining business models, securing market share, decreasing time to develop new products and reducing possibility of patent infringement (Codina-Filbà et al. 2017; Kim and Lee 2015; Trappey et al. 2009). Camus and Brancaleon (2003) highlighted the importance of information contained in patent analysis, revealing risks and opportunities and gaining insight into business activities. However, in order to be useful for the decision-making task, the information contained in a patent dataset must be presented in an understandable format (Madani and Weber 2016).

The information contained in patents is distributed in sections, defined by the patent office. The formatting of a patent text is controlled by laws and regulations of the country or a patent authority in which the inventor applied for the patent. In general, patents have title, abstract, claims and description. The abstract is characterized by having complex syntactic constructs and a generic vocabulary. The claims section has a hierarchical structure, including independent and dependent claims. The independent claims present a general idea of the invention whereas the dependent claims present more specific information about the invention. Each claim is composed by a single sentence. This leads to the appearance of very long sentences and significant complexity. The description section is characterized by having distinctive information of the inventions (Codina-Filbà et al. 2017; Mille and Wanner 2008).

In order to take advantage of patent knowledge, it is essential to organize information in an accessible and simple format and to name groups provided by patent offices with sentences which truly represent them. Because these subgroups belong to a restricted knowledge domain, the naming task can be extremely laborious. In this context, it is necessary to look for techniques which facilitate this naming process, to assist the specialists in their task.

This work uses summarization techniques as an approach to name patents groups. In the work presented by Souza et al. (2019), the best performing methodology of extractive summarization was reached using Latent Semantic Analysis (LSA) algorithm applied to patent abstracts. In this work, we compare LSA algorithm with an abstractive summarization algorithm and evaluate if the use of an abstractive summarization algorithm achieves better performance in the task to name new patent groups.

This work is divided into 6 sections. First section presents the theoretical background and related works. "Proposed approach" section describes the abstractive summarization model, the used dataset and the methodological steps of the work. "Experiments" and "Final considerations" sections show the results, analysis and final considerations.

Theoretical background

In general, there are two main approaches to automatic summarization: extractive and abstractive. Extractive summarization selects the main sections of the original text to generate a summary. The extractive summarization systems are usually based on the sentence/topic extraction technique and attempt to identify a set of sentences that is most important for the general understanding of a particular document. In order to identify these sentences, many approaches use keywords as a criterion for choosing the sentences and, thus, extract the sentences that have the highest number of keywords (Wang et al. 2011). Abstractive summarization tries to develop an understanding of the main sections of the text and, from an internal semantic representation, expresses the knowledge obtained in natural language. For this, it uses linguistic methods to interpret and describe the text, generating a summary with the main information of this text (Wang et al. 2011). Because it requires extensive processing of natural language, abstractive summarization is more complex than extractive and therefore less explored (Gambhir and Gupta 2017).

Abstractive methods can be divided into two categories, syntactic and semantic. Syntactic methods verify the grammatical structure of the text and use the information obtained to generate a concise representation of the text. Semantic methods generate a summary of the text from its semantic representations, usually using ontologies. Approaches using semantic representations are considered more robust because the abstractive summarization needs a thorough analysis of the text (Khan et al. 2015). However, currently semantic analysis methods have not been performing well in texts considered simpler, nor in structurally more complex texts. This fact makes the summary generation task more challenging (Codina-Filbà et al. 2017).

This section is divided into five subsections in which the concepts related to the work are presented. "Seq2Seq model", "LSTM network" and "LSA algorithm" sections provide, respectively, a description of the Sequence to Sequence (Seq2Seq) model, the Long Short-Term Memory (LSTM) network, and the LSA extractive summarization algorithm. "Recall-oriented understudy for gisting evaluation" and "Analysis of semantic similarity" sections describe the used measurement metrics.

Seq2Seq model

Seq2Seq was first introduced by Cho et al. (2014) and Sutskever et al. (2014). The architecture of Seq2Seq model is divided into two parts, encoder and decoder. Each of these parts may be implemented by, for example, a Recurrent Neural Network (RNN). To perform the abstractive summarization task, a many-to-many Seq2Seq architecture is used, where the encoder has an artificial input Artificial Neural Network (ANN) that receives a sequence of words from the text $x = x_1, \dots, x_m$, and gets the corresponding hidden state $z = z_1, \dots, z_m$. The decoder receives as input z and outputs a sequence $h = h_1, \dots, h_t$ (Zhang et al. 2019). To determine when the decoder will start generating summaries, a symbol representing the end of the input is used. After the first output h_1 is generated, the decoder will produce a new hidden state along with a word representation vector. Each generated word is used as input for the next word generation.

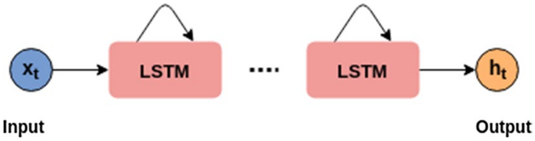


Fig. 1 LSTM network model scheme

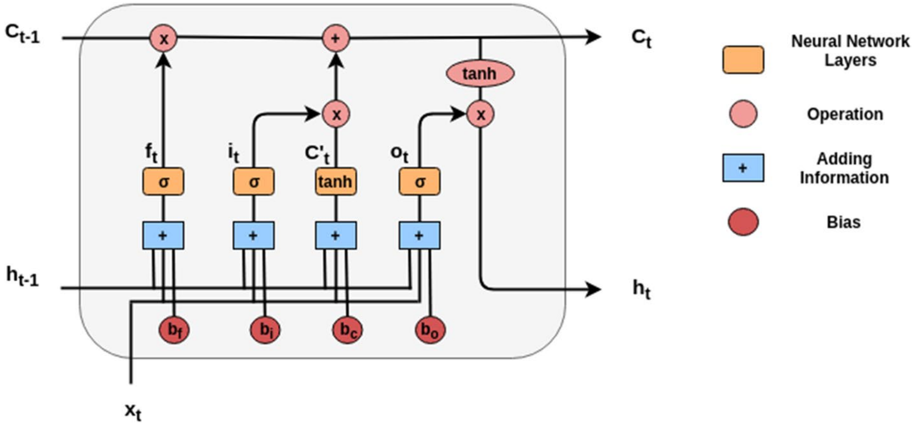


Fig. 2 LSTM detailed cell representation (Olah 2015)

LSTM network

LSTM networks consist of a set of recurrently connected blocks that are time-rolled (Greff et al. 2016). LTSM is a type of RNN. Standard RNN usually presents difficulties with long-term dependencies, so this network does not perform well on tasks that need a broader context (Greff et al. 2016). One alternative to this type of task is the use of LSTM. The LSTM network was introduced by Hochreiter and Schmidhuber (1997) and has since undergone several modifications. Currently, this network is mainly used in tasks that aim to solve sequential data learning problems (Greff et al. 2016), such as automatic text translation (Luong et al. 2015), automatic text summarization (Song et al. 2019) handwriting recognition (Paul et al. 2019), audio analysis (Bin et al. 2018) and video analysis (Abtahi et a l. 2018), among others.

In Fig. 1, we have a simplified model of a LSTM network consisting of an input x_t and an output h_t. Variable x_t entry goes through several layers of LSTM, and each cell has a loop. The function of the loop is to allow information to persist on the network for a certain time. In tasks that use sequential data, it is often necessary to look back to correctly predict the next state. In a basic RNN the amount of context available is smaller than in an LSTM network.

Each of the LSTM blocks has one or more memory cells and multiplicative units that are input gate (i_t), forget gate (f_t), output gate (o_t) and cell activation vectors (C_t). Basically, the input to the cells is multiplied by the activation gate, the output is multiplied by the output gate, and the previous cell values are multiplied by the forget gate (Graves and Schmidhuber 2005). The sigmoid layer (σ) outputs numbers between zero and one, and tanh layer creates a vector of new candidate values (C_t'). Figure 2 shows an internal diagram of a standart LSTM cell.

Table 1 LSTM network model variables descriptions

Variables	Descriptions
i	Input gate
f	Forget gate
o	Output gate
C	Cell activation vectors
σ	Sigmoid function
W_f, W_i, W_C, W_o	Weight matrices
b_f, b_i, b_C, b_o	Bias vector
h_t	Output of LSTM network

To map the input sequence x to an output sequence h, calculations are performed iteratively from $1 \rightarrow t$. In Fig. 2, each one of the paths presented in the LSTM cell is named. These paths are represented by the following equations:

$$f_t = \sigma(W_f \times [h_{t-1}, x_t] + b_f), \tag{1}$$

$$i_t = \sigma(W_i \times [h_{t-1}, x_t] + b_i), \tag{2}$$

$$C'_t = tanh(W_C \times [h_{t-1}, x_t] + b_C), \tag{3}$$

$$C_t = f_t \times C_{t-1} + i_t \times C'_t, \tag{4}$$

$$o_t = \sigma(W_o \times [h_{t-1}, x_t] + b_o), \tag{5}$$

$$h_t = o_t \times tanh(C_t). \tag{6}$$

Equation 1 has the role of deciding which information to forget. Equations 2 and 3 decide which information will be stored in the state of the cell. Equation 4 updates the state of the cell. Equations 5 and 6 decide which output will be produced. Table 1 defines the used variables.

LSA algorithm

Deerwester et al. (1990) described LSA as a method for information retrieval. However, Landauer et al. (1998) suggested using this method to find relationships between words. The main idea of this method is to reduce the number of dimensions, consequently reducing noise, and emphasizing strong indirect relationships between entities.

In this work, LSA is used to generate a summary of a document. The method was based on the work of Dokun and Celebi (2015) and consists of making an extractive summarization in which an algorithm extracts a single sentence from the document, identifying it as the sentence that best represents the document. For this, the algorithm receives as input a preprocessed document and generates a sentence-term matrix, usually sparse, in which a column vector represents the weighted frequency of the sentence in the document.

Table 2 ROUGE-N variables descriptions

Variables	Descriptions
R	Reference summary
C	Candidate summary
n	Length of n-grams
$Count_{match}(gram_n)$	Number of n-grams shared between R and C
$Count(gram_n)$	Number of n-grams in R

From the semantic point of view, Singular Value Decomposition (SVD), used by the algorithm, derives the latent semantic structure from the document represented by a matrix, reflecting a breakdown of the original document into linearly-independent base vectors or concepts. Each term and sentence from the document is jointly indexed by these base vectors/concepts. Beside this, if a word combination pattern is recurring in the document, this pattern will be represented by one of the singular vectors.

The magnitude of the corresponding singular value indicates the importance degree of this pattern within the document. Any sentences containing this word combination pattern will be projected along this singular vector, and the sentence that best represents this pattern will have the largest index value with this vector (Froud et al. 2013).

Recall-oriented understudy for gisting evaluation

Recall-Oriented Understudy for Gisting Evaluation[1] (ROUGE) is a set of metrics used to evaluate automatic text summarization and machine translation results. These metrics determine the similarity between a summary generated by a computational model and a summary generated by humans. One of the metrics of this set is ROUGE-N, which is a recall of N-grams between the candidate summaries and the reference summaries (Sanchez-Gomez et al. 2018). Thus, a ROUGE-1 score of 0.40 says that 40% of the content in the reference summary was captured by the summary generated by the model. ROUGE-N is calculated according to equation 7, proposed by Lin (2004). The used variables are described in Table 2.

$$ROUGE{-}N = \frac{\sum_{C\in R} \times \sum_{gram_n\in C} \times Count_{match}(gram_n)}{\sum_{C\in R} \times \sum_{gram_n\in C} \times Count(gram_n)}. \tag{7}$$

Another metric of this set is ROUGE-L, which evaluates correspondence between Longest Common Substring (LCS) shared by two sentences (Sanchez-Gomez et al. 2018). This metric assumes that the higher the LCS value of two R and C summaries, the more similar they are. Therefore, ROUGE-L will be 1.0 when both sequences are equal, and 0.0 when $LCS(R, C)$ is zero, indicating that there is no common sequence between R and C. To calculate this value, we use Eqs. 8, 9, 10 and 11 proposed by Lin (2004). The equations 8, 9 and 11 represent, respectively, the Recall, Precision and F-Measure of the LCS between R and C. The used variables are described in Table 3.

[1] Available at: https://github.com/chakki-works/sumeval.

Table 3 ROUGE-L variables descriptions

Variables	Descriptions
$LCS(R, C)$	Longest common substring between R and C
m	Summary size R
n	Summary size C

The precision metric checks how many of the values that were said to be positive are actually positive. The recall metric measures how many of the values that are positive were classified as positive. The F1-score metric combines the precision and recall values indicating the overall quality of the model.

$$R_{lcs} = \frac{LCS(R, C)}{m}, \tag{8}$$

$$P_{lcs} = \frac{LCS(R, C)}{n}, \tag{9}$$

$$\beta = \frac{P_{lcs}}{R_{lcs}}, \tag{10}$$

$$ROUGE{-}L = F_{lcs} = \frac{(1 + \beta^2) \times R_{lcs} \times P_{lcs}}{R_{lcs} + \beta^2 \times P_{lcs}}. \tag{11}$$

Analysis of semantic similarity

Semantic similarity is a measurement that verifies the similarity between sentences and texts, also defined as semantic entities. This similarity is measured using the distance between terms based on their meaning or semantic content. The semantic similarity index between the semantic entities is a numerical estimation obtained with the semantic information of the entities terms (Harispe et al. 2015).

In this work, the Semantic Similarity Estimator[2] (SenSim) method proposed by Al-Natsheh et al. (2017) was used. This method consists of two phases, the first is the extraction of characteristic pairs and the second is the regression estimation. For the extraction of feature pairs, the algorithm uses attributes such as Part-of-Speech (PoS), which is a category of words with similar lexical properties, Named-Entities (NE) such as people, organizations and sites, and the representation of sentences in Bag-of-Words (BoW), which is weighted by the TF-IDF algorithm. For regression estimation, the Random Forests (RF) method is used, which is a classifier that constructs decision trees during training. This method takes two sentences and assigns them a score between 0 and 5. A high score represents a large similarity between the sentences.

[2] Available at: https://github.com/natsheh/sensim.

Related works

Automatic text summarization aims to create a simple and descriptive summary of sections from the original text. Thus, the process identifies the significant aspects of one or more documents to represent them consistently (Allahyari et al. 2017). Abstractive summarization methods have still been less explored than extractive, as they require intense natural language processing (Gambhir and Gupta 2017). Works such as those published by Parmar et al. (2019), Zhang et al. (2019), Song et al. (2019), Yao et al. (2018) provide a perspective on how the abstractive summarization task is currently explored.

Parmar et al. (2019) evaluate in their work the performance of a Seq2Seq model and a bidirectional LSTM network. The used dataset was CNN/Daily Mail and Amazon reviews. The Seq2Seq model was validated in both datasets and the LSTM model only in the Amazon reviews. Both models were evaluated using ROUGE-1, ROUGE-2 and BLEU metrics. BLEU is a metric initially proposed for automatic text translation evaluation that uses a modified unigram precision. From the presented results, it was possible to verify that Seq2Seq model using the Amazon review dataset was the one that obtained the best result with BLEU metric, with a score of 26.25%, which indicated that it had the best accuracy among the three models under testing.

Zhang et al. (2019) presented in their work a generative model of abstractive text summarization using Convolutional Neural Network (CNN) and Seq2Seq. The proposed model had a copy mechanism for dealing with rare words and a hierarchical attention mechanism. According to the authors, the use of a CNN hierarchical structure was much more efficient than conventional models of the Seq2Seq RNN. The used datasets were GigaWord, DUC 2004 and CNN/Daily Mail. To evaluate the quality of the generated summaries, ROUGE-1, ROUGE-2 and ROUGE-L metrics were used. According to the authors, the proposed model had a good performance in relation to the state of the art.

Song et al. (2019) proposed in their work an abstractive summarization model based on LSTM-CNN. The proposed model consists of three steps, which are text pre-processing, sentence extraction and text summary generation. The used dataset was CNN/Daily Mail. The generated model was evaluated using ROUGE-1 and ROUGE-2 metrics. According to the authors, the results exceeded the existing models in terms of semantic and syntactic structure, combining extractive and abstractive summarization, and obtained competitive results in the manual assessment of linguistic quality.

Yao et al. (2018) presented in their work an abstractive summarization method that used a dual coding model. In the method presented by the authors, the primary encoder performed text encoding on a regular basis, while the secondary encoder modeled the importance of words in the text and generated a more accurate encoding of the text. For final summary generation, the two encodings were combined to generate a more diverse summary. The used dataset for the experiments were CNN/Daily Mail and DUC 2004. To evaluate the generated summaries, the metrics ROUGE-1, ROUGE-2 and ROUGE-L were used. According to the authors, the proposed method presented a good result in relation to the state of the art.

By analyzing these works, it is possible to verify that most of them evaluate their results using news datasets, not exploing domains such as patents. Works such as of Codina-Filbà et al. (2017), Mille and Wanner (2008), Trappey et al. (2009) highlight the importance of generating automatic summaries in patent documents. Therefore, it is necessary to evaluate the performance of these algorithms in this domain of knowledge.

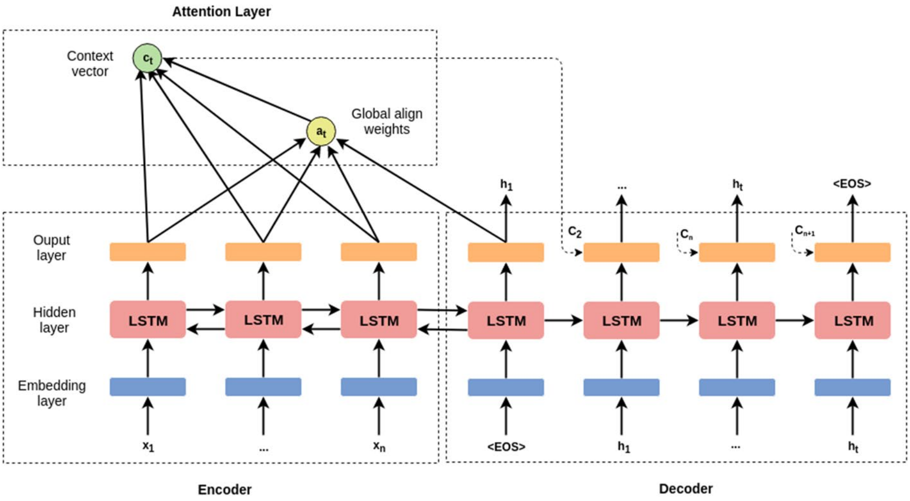


Fig. 3 Abstractive summarization model approach

Proposed approach

This section is divided into 3 subsections. The first one presents the abstractive summarization model used. The second one presents a description of the dataset used in experiments, while the third one presents the methodological steps taken during the practical experiments.

Abstractive summarization model

The model used in this work consists of two LSTM network architectures. The encoder used LSTM cell along with Stack Bidirectional Dynamic RNN, represented in Fig. 3 by the dotted box named Encoder. In this model, there is the stacking of several layers of two-way RNN, in which the combined outputs of the previous and subsequent layer are used as inputs to the next layer (Parmar et al. 2019). Using the bidirectional model has the advantage of being able to use past and future contextual information. The decoder uses LSTM BasicDecoder cells associated with the Beam Search Decoder. The decoder is represented in Fig. 3 by the dotted box named Decoder. Beam Search Decoder is a technique that allows you to find the best word combination for the output summary. According to Cohen and Beck (2019), this algorithm is one of the most commonly used in neural sequence models, as it performs non-greedy local searches that increase the chances of generating a sentence with a higher overall probability. For its use, it is necessary to set the parameter *beam_width*. In this work, *beam_width* is 10. The higher the value of *beam_width*, the better exploration of the search space and therefore the better the sentence should be. However, the computational cost is high.

In order for sequence *x* to be used as network input, it must pass through an embedding layer. In this work, the embedding layer uses the GloVe unsupervised learning algorithm. This algorithm generates vector representations for words, combining the advantages of global matrix factorization and local context window techniques. Model training is performed using the word co-occurrence information in a given corpus, and the resulting representations show linear substructures of the word vector space (Pennington et al. 2014). At the end, there are vector representations with the ability to highlight the semantic structure of words, allowing to capture the meaning and similarity between them.

One of the problems with this model is that the ANN needs to compress important sentence information into a fixed-length vector, called context vector (Parmar et al. 2019). This compression can lead to important information loss, especially when it comes to long sentences. To solve this problem, we use the Bahdanau attention mechanism (Bahdanau et al. 2014). The attention mechanism is represented inside the dotted box named Attention Layer. In addition, to avoid overfitting and improve model performance, the Dropout technique is used. This technique randomly drops network drives during training, along with their connections (Srivastava et al. 2014). For Dropout we use *keep_prob* = 0.8, which means that 20% of neurons can be dropped during training. Another problem presented by this network is that of exploding and vanishing gradients. An exploding gradient can occur when the gradient norm becomes too large, resulting in an unstable network. A vanishing gradient occurs when the gradient norm becomes too small, stopping the optimization process at a certain point. To avoid this problem, the clipping technique is used. This technique introduces a gradient threshold. The Gradient standards that exceed this limit are reduced to match the norm. The threshold value used is 5. The hyperparameters *keep_prob*, clipping threshold, number of LSTM layers and the dimensions of word embeddings were defined empirically.

Dataset

There are some classic datasets that are used for automatic text summarization task. These include CNN/Daily Mail, NYT, NEWSROOM, XSUM, ARXIV, PUBMED and Amazon Reviews datasets. Sharma et al. (2019) state that these datasets are not suitable for training abstractive summarization models, because the majority of the fragments used in the articles abstracts, in general, appear again in the text. The presence of the summary in the input text means that abstractive summarization do not have to generate a sentence, just extract the sentence from the input text. However, the goal of abstractive summarization is to build a model that can understand the content of the text and thus, subsequently, generate one or more sentences able to define the input text content. Thus, using texts that already have the summary content within the input text limits the learning process of the algorithm and makes the abstractive summarization more similar to the extractive than abstractive summarization algorithm. Because of this, Sharma et al. (2019) propose the use of patent documents to train abstractive summarization models, especifically, the description and abstract sections. These section, do not usually have fragments of the document text.

Therefore, to conduct the experiments following Sharma's suggestion, a dataset was created composed of abstracts and titles of patent documents provided by the United States Patent and Trademark Office (USPTO). Abstracts were used in the input model and titles were used as ground truth to compare them with the output model. USPTO uses the Cooperative Patent Classification (CPC) system, which classifies patents into sections, classes, subclasses, groups and subgroups as illustrated in Fig. 4. We chose to use titles and abstracts because the objective is to use the proposed approach to generate simple and descriptive sentences that are able to name, consistently, patent subgroups. As can be seen later, in Table 13, subgroups names are generally small.

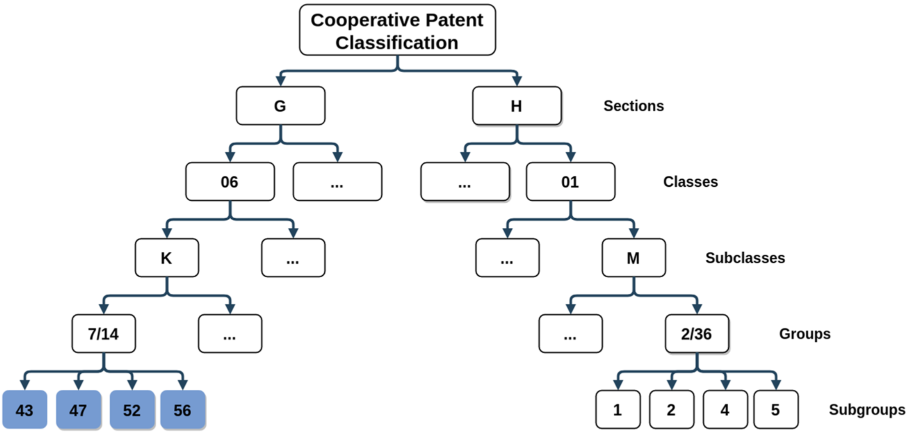


Fig. 4 Hierarchical organization of the CPC system

Two main datasets were created. The first was used in the training and validation of the abstractive summarization process. The second was used to compare the abstractive with the extractive summarization process used by Souza et al. (2019). To generate the first dataset, 7,000 documents were randomly selected from each of the nine sections of CPC. In the CPC system, documents can belong to more than one subgroup, so it was necessary to remove duplicate documents. In the end, we obtained a dataset composed of 41,527 patent documents, divided into a training dataset with 33,221 documents and a validation dataset with 8,306 documents. The dataset has an average compression ratio of 22.55, which represents the ratio between the number of words in the abstract and the number of words in the titles. Abstracts have an average of approximated 124 words and 6 sentences, and the titles have an average of approximately 8 words and 1 sentence.

Among the related work, Sharma et al. (2019) are the ones that evaluate the performance of abstractive summarization models in patent datasets. However, the authors propose a patent dataset composed of patent titles, abstracts and descriptions and evaluate performance using only abstracts and descriptions. Patent sections have different structure and language characteristics, which makes it impossible to compare the results of this work with those of Sharma et al. (2019). Beside this, the summaries generated in this work are more concise and use a smaller number of words.

To perform the naming task, as proposed by this work, we used a second dataset composed of four subgroups which have the following CPC codes: G06K 7/1443, G06K 7/1447, G06K 7/1452 and G06K 7/1456. From now on, the subgroups codes will be represented only by their suffixes 43, 47, 52 and 56. The second dataset is composed of 733 patents. Table 4 shows the distribution of patents in each of the subgroups.

Methodological steps

Initially, all used datasets were preprocessed. For all dataset files presented in "Dataset" section, special characters were removed, all texts were placed in lower case, all punctuations were separated from text, and periods were replaced by the # character.

Table 4 Second dataset patents distribution

Subgroups	Number of patents
43	348
47	198
52	68
56	119

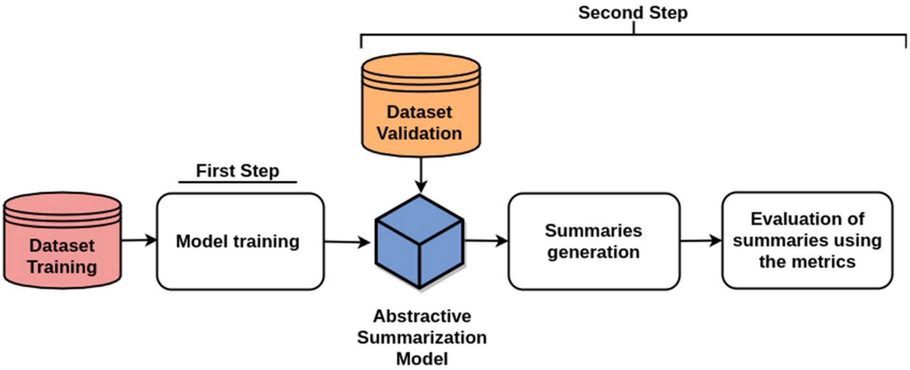


Fig. 5 First phase of the abstractive summarization process

The methodology used to find the sentence that best describes a group using abstractive summarization can be divided into two phases. The first phase is divided into two steps. Figure 5 presents a diagram representing the steps of the first phase. This phase was performed 30 times by initializing LSTM network weights randomly. It's necessary to execute these algorithms 30 times because they are stochastic, which means that different executions of the same algorithm using the same input data may return different results. Thus, the final performance of these algorithms is given by the average performance of 30 instances of their execution, ensuring statistical validation of the obtained performance.

The first step consists of training the model using the first dataset described in "Dataset" section. The model is trained with 2 layers with 150-dimensional hidden states and a pre-trained word vectors model 840b by 300-dimensional vectors[3] using Adam Optimizer. For training, patents abstracts were used as network input and document titles as outputs. In the second step, the validation of the model was performed. Validation was performed for each of the 30 generated instances. To verify the quality of the generated output, which we called "summaries", we used ROUGE-1, ROUGE-2 and ROUGE-L metrics, following Sharma et al. (2019) approach. In this work, the network outputs were compared with the document titles. For each of the 30 instances, the results were obtained for ROUGE-1, ROUGE-2 and ROUGE-L metrics. To calculate the average accuracy of the model, the metrics were averaged using the 8306 validation dataset records. This resulted in 30 values for each of the metrics. Afterwards, the average of each metric was calculated considering

[3] Available at: https://nlp.stanford.edu/projects/glove/.

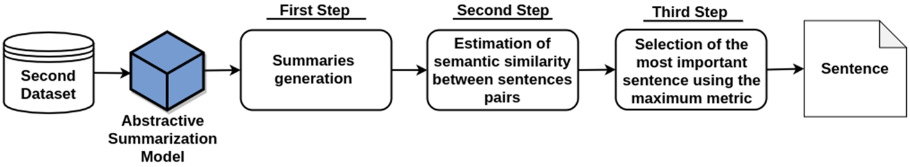

Second Dataset
Abstractive Summarization Model
First Step
Summaries generation
Second Step
Estimation of semantic similarity between sentences pairs
Third Step
Selection of the most important sentence using the maximum metric
Sentence

Fig. 6 Second phase of the abstractive summarization process

the 30 instances. The instance selected for the tests is the one that received the best general average with the three metrics.

The second phase is divided into three steps and consists of using the abstractive summarization model generated in the first phase for the task of automatically naming patent groups. Figure 6 presents a sequence diagram of the steps in this phase.

The first step consists of summarizing the abtracts of each document using the model generated in the first phase. For each document, only one sentence is generated. The second step of the process analyzes the similarity between the generated sentences, using the method proposed by Al-Natsheh et al. (2017). The semantic similarity of each sentence in relation to the other sentences of the subgroup is calculated, creating a list of sentence pairs, and their respective scores of similarity. In the third step, the maximum metric is used to select the most representative sentence of each subgroup. The maximum metric, used by Souza et al. (2019), selects the sentence that most frequently presents the highest similarity score.

Finally, the validation of the entire experiment is performed, using the second dataset, shown in Table 4, which already had its names designated by specialists. The selected sentences are quantitatively evaluated, analyzing the semantic similarity between the name of the subgroup and the chosen sentences as the most representative of each subgroup. In addition, a qualitative analysis is performed, with the name of the subgroup. The hypothesis is that if the selected sentence is semantically similar to the subgroup name, it will provide a meaningful description for the subgroup. From this analysis, it is possible to compare the results obtained using the abstractive summarization and the LSA extractive summarization, developed by Souza et al. (2019).

Experiments

Initially, when the training model is performed, we restrict the LSTM network input to 150 words and the output to 15 words. These values are chosen considering the average amount of words in the abstracts and in the titles of the patent documents. During the training, a dictionary with 48,083 words is generated. The model is trained using Google's Colab Notebooks with a Tesla K80 GPU. Each instance lasted 14h on average running 100 epochs. The average training and inference times of the 30 instances of the model, for each abstract, are approximately 1.5171 seconds and 0.0083 seconds, respectively. On average, the loss function value is of 0.1827 with a standard deviation of 0.4958. Figure 7a presents the histograms with the average distribution of the values of ROUGE-1. The average distribution of the values ROUGE-2 and ROUGE-L are presented in Fig. 7b and c for 30 instances.

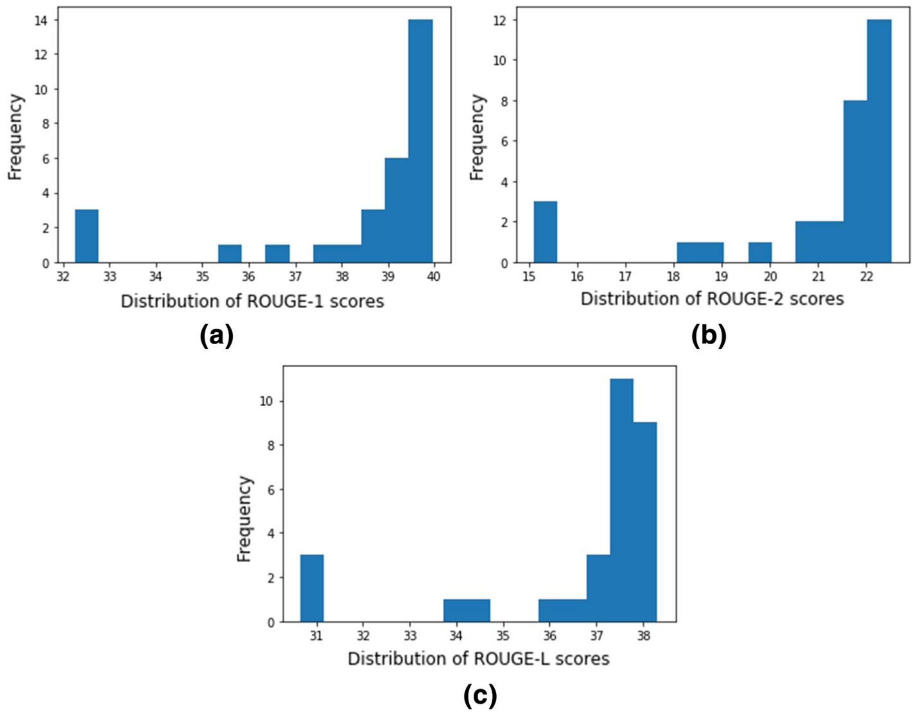


Fig. 7 Average distribution of ROUGE scores in 30 instances of execution

Table 5 ROUGE scores average

Metrics	Average (%)	Standard deviation (%)
ROUGE-1	38.39	2.21
ROUGE-2	20.97	2.16
ROUGE-L	36.68	2.16

Table 5 presents the average with their respective standard deviation values for ROUGE-1, ROUGE-2, and ROUGE-L metrics obtained for the 30 instances generated in this work. We chose to present only these three metrics because these are the ones used to evaluate the model performance of this work. Based on the presented data, it can be seen that most of the results of the discussed works do not perform well. This clearly shows that abstractive summarization still needs major development, both for general discourse texts and for patent documents, which are characterized by having structurally more complex texts.

Table 6 reproduces the results of ROUGE-1, ROUGE-2 and ROUGE-L metrics, in percentage, for each of the referred works. Based on the presented data, it can be seen that most of the results of those reports do not perform well. This clearly shows that abstractive summarization still needs major development, both for general discourse texts and for patent documents, which are characterized by having structurally more complex texts. The general average of the metrics used to evaluated the model match some results present in the literature, as shown in Table 6.

Table 6 ROUGE scores for the discussed works

Author (year) Strategy	Datasets	Metrics (%)		
		ROUGE-1	ROUGE-2	ROUGE-L
Parmar et al. (2019) Seq2Seq	Amazon reviews	42.53	16.33	X
	CNN/Daily Mail	58.21	20.57	X
Parmar et al. (2019) Bi-LSTM	Amazon reviews	21	1	X
Zhang et al. (2019) CNN-Seq2Seq	GigaWord	37.95	18.64	35.11
	DUC corpus	29.74	9.85	25.81
	CNN/Daily Mail	42.04	19.77	39.42
Song et al. (2019) LSTM-CNN	CNN/Daily Mail	34.9	17.8	X
Yao et al. (2018) Seq2Seq extend	CNN/Daily Mail	40.85	18.08	37.13
	DUC 2004	29.91	9.61	25.95

Table 7 Results of abstractive summarization for P_1

Abstract	the invention relates to a cutting tool having twisted edge in a blade part , having a sintered compact with higher hardness and higher wear resistance than a base sintered body buried and affixed along the twisted edge , and its manufacturing method # the blade part of the cutting tool comprises the base sintered body having a twisted groove in the position of forming the twisted edge on the outer circumference , and the sintered compact of high hardness and high wear resistance applied and buried in the twisted groove and affixed to the base sintered body by sintering , and the twisted edge is formed on the sintered compact of high hardness and high wear resistance # its manufacturing method comprises a step of forming a presintered or a sintered base material of the blade part with material powder , a step of forming a twisted groove on the outer circumferenced of the base material , a step of filling the twisted groove with material powder of sinter of high hardness and high wear resistance , a step of heating and pressurizing the base material with the material powder , and sintering and affixing the material powder to the base material , and a step of machining thus sintered and united base material to form a twisted edge on the sintered compact of high hardness and high wear resistance #
Title	**cutting tool with twisted edge and manufacturing method thereof**
Summary	**cutting tool with twisted edge and manufacturing method thereof**

To select the instance to be used in the second phase, the three metrics were averaged for each one. We then averaged these values. The global average was 32.01% with standard deviation equal to 2.18%. From this value, we selected the instance that had an average value closer to this global average. In Tables 7, 8, 9 and 10, some of the obtained results with the second dataset are presented. In each table, the first line is the patent abstract, the second, the patent title and the third, the generated summary, which is automatically generated as a label for each patent, by the proposed approach. In bold, the words that appear in

Table 8 Results of abstractive summarization for P_2

Abstract	the present invention provides a novel method to produce grade road base material using recycled oilfield waste , called oil and gas waste , more specifically , drilling waste and aggregate and a novel road base material # hydration and mixing of the waste materials along with a binder , will achieve an irreversible pozzolanic chemical reaction necessary for stabilization into a road base # an asphalt emulsifier may be included in the binder to manufacture asphalt stabilized road base # the entire method is a cold batch process #
Title	**method for making a road base material using treated oil and gas waste** material
Summary	**method for making a road base material using treated oil and gas waste**

Table 9 Results of abstractive summarization for P_3

Abstract	exemplary embodiments are directed to wireless power transfer # a transmitting device or a receiving device for use in a wireless transfer system may be equipment or a household appliance # the transmitting device includes a transmit antenna to wirelessly transfer power to a receive antenna by generating a near field radiation within a coupling-mode region # an amplifier applies an rf signal to the transmit antenna # a presence detector detects a presence of a receiver device within the coupling-mode region # a controller adjusts a power output of the amplifier responsive to the presence of a receiver device # the presence detector may also detect a human presence # the power output may be adjusted at or below the regulatory level when the presence signal indicates human presence and above a regulatory level when the presence signal indicates human absence #
Title	**wireless power transfer for** appliances and equipments
Summary	**wireless power transfer for** communication

Table 10 Results of abstractive summarization for P_4

Abstract	a technique is disclosed for a system and method for combined production of power and hydrogen utilizing the heat from a first working fluid heated by a geothermal energy source using a steam generator and an electrolyzer designed to receive the steam produced by the steam generator for the production of hydrogen and oxygen using electrolysis #
Title	system and **method** for production of hydrogen
Summary	power generation and **method** for providing energy

Table 11 ROUGE scores for patents P_1, P_2, P_3 and P_4

Patents	ROUGE-1 (%)	ROUGE-2 (%)	ROUGE-L (%)
P_1	100	100	100
P_2	94.74	94.11	94.74
P_3	66.67	57.14	66.67
P_4	22.22	0.0	22.22

Table 12 ROUGE scores distribution for 8,306 Patents

ROUGES	ROUGE-1	ROUGE-2	ROUGE-L
75–100%	1176	731	1081
50–75%	1756	774	1603
25–50 %	2341	1277	2373
0–25%	3033	5524	3249

the document title and the generated summary were highlighted. The patents were selected from four different CPC sections. The selected patents were identified as P_1, P_2, P_3 and P_4, each belonging, respectively, to section B, E, G and F of the CPC system.

Table 11 shows the resulting metrics for patents P_1, P_2, P_3 and P_4. According to the presented results, we can verify that the used model has promising results, especially when compared to the examples of summaries generated by abstractive models of other works, as presented in “Introduction” section.

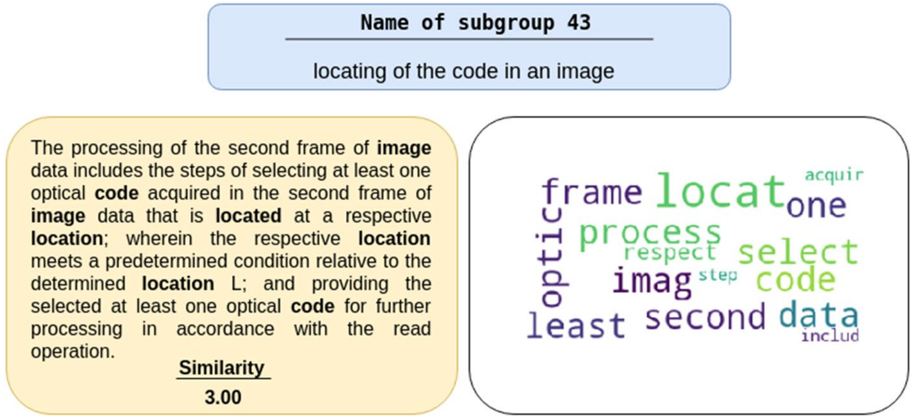
Overall, out of 8,306 documents of the validation dataset, 543 simultaneously obtained the maximum value for the three metrics, such as $P1$ patent shown in Table 7. Table 12 shows the number of patents in each percentage range for ROUGE-1, ROUGE-2, and ROUGE-L metrics. By analyzing the characteristics of these texts, we conclued that most of them have texts in the training dataset, considering that there are many documents related to the same topic. This shows that the generated model was able to identify this relationship. In some cases, it was noted, by a qualitative comparison, that the generated summary had the same semantic content as the input, but the generated summary did not have all the words of the reference summary. In these cases, the metric rated it with a very low score. There are also cases in which the summaries differ in some words, such as the P_3 patent shown in Table 9. In this case, the score was also severely penalized. Therefore, we conclude that the used metrics do not perform well to evaluate abstractive summaries, because unlike the extractive summaries that always have the same words as the input texts, the abstractive summaries have more freedom to generate sentences. This makes it possible to generate sentences semantically similar to the input text, consistent with the text content, but which do not have exactly the same words, such as P_4 patent shown in Table 10.

Moreover, by analyzing all the results obtained with the proposed approach, it was possible to realize that, in many cases, we obtained significant results. The results usually presented in the literature were trained with larger dataset, general speech texts and more intense training. Therefore, we believe that the results obtained in this work are promising, since we apply abstractive summarization in patent texts which have a more complex language and structure, as systematically described in the literature.

After the training, validation, and analysis of the results obtained by the abstractive summarization model, we used the instance selected in the group naming task. A sentence was generated for each of the four subgroups presented in Table 4. The idea is that the generated sentence should be able to describe the content of each one of the subgroups. The generated sentences are presented in the second column of Table 13. The third column presents the names given by the specialists. The word cloud, shown in Fig. 8, is a graphical representation that helps evaluating the existing similarity between the original text and the summarized sentence obtained in the extractive summarization by Souza et al. (2019).

Table 13 Generated sentences using abstractive summarization

Subgroups	Generated sentences	Subgroups names
43	bar code reading system	locating of the code in an image
47	authenticating system and printing methods adjusted to transmitting barcode on a user with plurality	extracting optical codes from image or text carrying said optical code
52	optical screen method and apparatus	detecting bar code edges
56	system and method for printed marks	determining the orientation of the optical code with respect to the reader and correcting therefore


Name of subgroup 43

locating of the code in an image

The processing of the second frame of **image** data includes the steps of selecting at least one optical **code** acquired in the second frame of **image** data that is **located** at a respective **location**; wherein the respective **location** meets a predetermined condition relative to the determined **location** L; and providing the selected at least one optical **code** for further processing in accordance with the read operation.

Similarity

3.00

Fig. 8 Sentence extracted using LSA and metric **maximum** for the subgroup 43 (Souza et al. 2019)

Table 14 Semantic similarity scores comparisons

Subgroups	Abstractive summarization	Extractive summarization (Souza et al. 2019)
43	1.16	**3.00**
47	**1.70**	1.66
52	1.98	**3.09**
56	1.03	**3.07**

By comparing the scores of semantic similarity between the generated sentences and the subgroups names, it is possible to see that extractive summarization has superior results. Only one of the results with the abstractive summarization presented a higher score than the extractive summarization. However, sentences obtained using extractive summarization are not able to name a group but rather provide a sentence to help the specialist to define the name of a group. On the other hand, the expectation is that the abstractive summarization could name the group without the intervention of the specialist. Given this, it can be concluded that abstractive summarization has great potential for this task, however, the techniques still need to be improved. Table 14 presents the results of semantic similarity between sentences generated by abstractive and extractive summarization and the subgroup names defined by the specialists. The best results were highlighted in bold. These results vary between 0 and 5.

Final considerations

The main contribution of this work is to propose an approach for automatic generation of patent group names, using summarization techniques. An abstractive summarization model was compared to the performance of an extractive summarization algorithm applied to the sentence generation task, capable of assisting the specialists when naming new groups/subgroups. The experiments were performed using a modern abstractive summarization strategy that uses a Seq2Seq architecture and LSTM networks applied to an area of interest of the academic and industrial community. The task of generating abstractive summaries of patent documents is still little explored. Therefore, we hope to contribute to the study of these techniques in patent datasets. Based on the experiments performed, it was possible to verify that the abstractive summarization model used has promising results for the patent domain. Although the performance of extractive summarization had a better result than the abstractive one, in the task of group naming, it was possible to identify advantages associated with the use of abstractive summarization. Therefore, a proposal to continue this work is with the expansion of the training dataset, the training with a larger number of epochs, the comparison of the approach presented here with other variations and the analysis of other techniques to evaluate the performance of the models, such as the validation by specialists.

Acknowledgements The authors would like to thank the financial support of the Pontifical Catholic University of Minas Gerais (PUC Minas), the Federal Center for Technological Education of Minas Gerais (CEFET-MG), the National Council for Scientific and Technological Development (CNPq, Grant 429144/2016-4) and the Foundation for Research Support of the State of Minas Gerais (FAPEMIG, Grant APQ 01454-17).

References

Abtahi, F., Ro, T., Li, W., Zhu, Z. (2018). Emotion analysis using audio/video, emg and eeg: A dataset and comparison study. In 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, (pp. 10–19).

Al-Natsheh, H. T., Martinet, L., Muhlenbach, F., Zighed, D. A. (2017). UdL at SemEval-2017 task 1: Semantic textual similarity estimation of English sentence pairs using regression model over pairwise features. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Association for Computational Linguistics, Vancouver, Canada, (pp 115–119).

Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., Kochut, K. (2017). A brief survey of text mining: Classification, clustering and extraction techniques. arXiv preprint arXiv:170702919.

Bahdanau, D., Cho, K., Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *CoRR*arXiv:1409.0473.

Bin, Y., Yang, Y., Shen, F., Xie, N., Shen, H. T., & Li, X. (2018). Describing video with attention-based bidirectional lstm. *IEEE Transactions on Cybernetics*, *49*(7), 2631–2641.

Camus, C., & Brancaleon, R. (2003). Intellectual assets management: From patents to knowledge. *World Patent Information*, *25*(2), 155–159.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), (pp. 1724–1734).

Codina-Filbà, J., Bouayad-Agha, N., Burga, A., Casamayor, G., Mille, S., Müller, A., et al. (2017). Using genre-specific features for patent summaries. *Information Processing & Management*, *53*(1), 151–174.

Cohen, E., Beck, C. (2019). Empirical analysis of beam search performance degradation in neural sequence models. In International Conference on Machine Learning, (pp. 1290–1299).

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, *41*(6), 391–407.

Dokun, O., & Celebi, E. (2015). Single-document summarization using latent semantic analysis. *International Journal of Scientific Research in Information Systems and Engineering (IJSRISE)*, *1*(2), 57–64.

Froud, H., Lachkar, A., & Ouatik, S. A. (2013). Arabic text summarization based on latent semantic analysis to enhance arabic documents clustering. *International Journal of Data Mining & Knowledge Management Process (IJDKP)*, *3*(1), 79–95.

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

Gomez, J. C. (2019). Analysis of the effect of data properties in automated patent classification. *Scientometrics*.

Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, *18*(5–6), 602–610.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2016). Lstm: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(10), 2222–2232.

Harispe, S., Ranwez, S., Janaqi, S., & Montmain, J. (2015). Semantic similarity from natural language and ontology analysis. *Synthesis Lectures on Human Language Technologies*, *8*(1), 1–254.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Khan, A., Salim, N., & Kumar, Y. J. (2015). A framework for multi-document abstractive summarization based on semantic role labelling. *Applied Soft Computing*, *30*, 737–747.

Kim, J., & Lee, S. (2015). Patent databases for innovation studies: A comparative analysis of uspto, epo, jpo and kipo. *Technological Forecasting and Social Change*, *92*, 332–345.

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes*, *25*(2–3), 259–284.

Lin, C. Y. (2004). Rouge: A package for automatic evaluation of summaries. In Text summarization branches out, (pp. 74–81).

Luong, M. T., Sutskever, I., Le, Q. V., Vinyals, O., Zaremba, W. (2015). Addressing the rare word problem in neural machine translation. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics, (pp. 11–19).

Madani, F., & Weber, C. (2016). The evolution of patent mining: Applying bibliometrics analysis and keyword network analysis. *World Patent Information*, *46*, 32–48.

Mille, S., Wanner, L. (2008). Multilingual summarization in practice: The case of patent claims. In Proceedings of the 12th European association of machine translation conference, (pp. 120–129).

Olah, C. (2015). Understanding lstm networks.

Ouellette, L. L. (2017). Who reads patents? *Nature Biotechnology*, *35*(5), 421.

Parmar, C., Chaubey, R., Bhatt, K. (2019). Abstractive text summarization using artificial intelligence. Available at SSRN 3370795.

Paul, I. J. L., Sasirekha, S., Vishnu, D. R., Surya, K. (2019). Recognition of handwritten text using long short term memory (lstm) recurrent neural network (rnn). In AIP Conference Proceedings, AIP Publishing, 2095, (pp. 030011).

Pennington, J., Socher, R., Manning, C. (2014). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), (pp. 1532–1543).

Sanchez-Gomez, J. M., Vega-Rodríguez, M. A., & Pérez, C. J. (2018). Extractive multi-document text summarization using a multi-objective artificial bee colony optimization approach. *Knowledge-Based Systems*, *159*, 1–8.

Sharma, E., Li, C., Wang, L. (2019). Bigpatent: A large-scale dataset for abstractive and coherent summarization. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, (pp. 2204–213).

Sjögren, R., Stridh, K., Skotare, T., Trygg, J. (2018). Multivariate patent analysis-using chemometrics to analyze collections of chemical and pharmaceutical patents. *Journal of Chemometrics*, (pp. e3041).

Song, S., Huang, H., & Ruan, T. (2019). Abstractive text summarization using lstm-cnn based deep learning. *Multimedia Tools and Applications*, *78*(1), 857–875.

Souza, C. M., Santos, M. E., Meireles, M. R., Almeida, P. E. (2019). Using summarization techniques on patent database through computational intelligence. In EPIA Conference on Artificial Intelligence, Springer, (pp. 508–519)

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929–1958.

Sutskever, I., Vinyals, O., Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Advances in neural information processing systems, (pp. 3104–3112).

Trappey, A. J., Trappey, C. V., & Wu, C. Y. (2009). Automatic patent document summarization for collaborative knowledge systems and services. *Journal of Systems Science and Systems Engineering*, *18*(1), 71–94.

Wang, D., Zhu, S., Li, T., Chi, Y., & Gong, Y. (2011). Integrating document clustering and multidocument summarization. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *5*(3), 14.

Wang, X., Ren, H., Chen, Y., Liu, Y., Qiao, Y., & Huang, Y. (2019). Measuring patent similarity with sao semantic analysis. *Scientometrics*, *121*(1), 1–23.

Yao, K., Zhang, L., Du, D., Luo, T., Tao, L., Wu, Y. (2018). Dual encoding for abstractive text summarization. *IEEE transactions on cybernetics*.

Zhang, Y., Li, D., Wang, Y., Fang, Y., & Xiao, W. (2019). Abstract text summarization with a convolutional seq2seq model. *Applied Sciences*, *9*(8), 1665.