ISSN 2007-9737

Learning Relevant Models using Symbolic Regression for Automatic Text Summarization

Eder Vázquez, Yulia Ledeneva, René Arnulfo García-Hernández

Autonomous University of the State of Mexico,
Instituto Literario,
Mexico

{eder2v, renearnulfo}@hotmail.com, yledeneva@yahoo.com

Abstract. Natural Language Processing (NLP) methods allow us to understand and manipulate natural language text or speech to do useful things. There are several specific techniques in this area, and although new approaches to solving the problems arise, its evaluation remains similar. NLP methods are regularly evaluated by a gold standard, which contains the correct results which must be obtained by a method. In this situation, it is desirable that NLP methods can close as possible to the results of the gold standard being evaluated. One of the most outstanding NLP task is the Automatic Text Summarization (ATS). ATS task consists in reducing the size of a text while preserving their information content. In this paper, a method for describing the ideal behavior (gold standard) of an ATS system, is proposed. The proposed method can obtain models that describe the ideal behavior which is described by the topline. In this work, eight models for ATS are obtained. These models generate better results than other models used in the state-of-the-art on ATS task.

Keywords. Natural language processing, gold standard, topline, symbolic regression, data modeling, automatic text summarization task.

1 Introduction

Natural Language Processing (NLP) is a sub-discipline in artificial intelligence and computational linguistics, which tries to extract (automatically or semi-automatically) the most complete meaningful representation of a text documents using computational models [27]. The main reason for the importance of NLP techniques is that allows transform information described in natural language into a format understandable by a computer [18, 27]. NLP techniques have created tools and systems that easily access and handling information. Some of these tools are: knowledge management, automatic translation, text mining, authorship identification, information retrieval, document classification, automatic summarization, or recent applications on identification of suicidal ideation [11].

All or most systems of NLP shared some characteristics in common [8]: (1) require a text documents representation [24, 37]; (2) require an association function (or similarity) between two text documents [29, 46]; and (3) require an evaluation paradigm [10].

Text documents are usually represented using the vector space model [58, 61, 62]. In this representation each text document is expressed as a weighted high-dimensional vector, the dimensions correspond to individual features such as words, concepts or numbers values [29]. This representation allows be effectively processes by computers.

Similarity metric is a property that measures the degree of similarity between two text (o documents), that is, similarity between them quantifies the dependency or independence between two texts [23]. On one hand, compute the similarity metric between sets of text documents have great importance in many of text analysis task [29], especially in task related to information retrieval [17], document clustering [41], plagiarism detection [46], recommendation systems [69], automatic classification [30], etc. On the other hand, the definition of a similarity measure is useful in several application areas such as multimedia search, medical imaging, molecular biology, assisted engineering computers, marketing assistance, etc. [12].

Like other informatics programs, NLP systems may be evaluated bellow standard method. There are several methods to evaluate NLP systems [10], but the classical distinction was proposed by [32]: intrinsic and extrinsic evaluation. Intrinsic evaluation is based on measuring the performance of a NLP system and this type of evaluation is characterized its performance mainly with respect to a gold standard result predefined by the evaluators. In extrinsic evaluation, the system output is assessed whit an external task to the system, it considers the system in a complex setting or in a task of final user [32].

Several NLP task uses intrinsic evaluation, specifically evaluation based on gold standard [15]. Gold standard are manually annotated collection of text, i.e., tag documents, sentences or words with a predefined set of categories [25, 68]. Gold standard evaluation is used because provides basis for the comparison of several systems against the same set of data in a certain task. This allows the evolution of performance results outputs in NLP tasks, besides is an invaluable resource for evaluation [50].

As a gold standard corpus directly impact the development of NLP systems [68], researchers want to develop systems to improve performance of the results compared to corpus of correct answers and thus know what combinations, modifications o components are the most optimal [33].

One of the most important NLP tasks is the Automatic Text Summarization (ATS). A summary is a set of phrases or sentences that best covers the relevant concepts of documents [22]. Specifically, it is a reductive transformation of the content of a input document by the selection or generalization of the most important information in the document [62].

An approach to solve ATS task is generate extractive summaries, which only select the most important words, sentences or paragraphs from the source document to conform the output [28, 52]. To select the most import sentences, several models that describe the importance of sentence position have been proposed [4, 20, 47]. These models have been obtained competitive results, but there is no description of the ideal behavior of an ATS system.

In this paper, a method to learning relevant models that describes the ideal behavior of an ATS system are proposed.

2 Background and Related Work

The models are theoretical schemes, generally, in mathematical form, of a system or a complex reality, which are made to facilitate understanding and the study. A model is an explicit representation of reality, as people who will use the model to understand a part of reality [49]. Notably, the models are only representations, indicating that all models are wrong, or have errors, but can be useful [5].

In real problems, it is common to find situations where we must estimate or model the behavior of an output variable based on one or more input variables. Traditionally, these problems have been resolved with statistical regression models. Also, there are situations where statistical and mathematical models cannot provide a good solution and it is necessary to develop tools and methods that allow us achieve our goals [54].

There are two ways to solved this problem: inferential and inductive [45]. The inferential process is based on the application of physical laws that are accepted as hypothesis, based on the choice of a probability distribution model of observed data. The inductive process performs the estimation models from data analysis, which emphasizes algorithmic approaches. Inductive analysis allows commenting on phenomena, and from these observations making inferences from the entire problem. Unlike, deductive analysis is a method of facts to draw conclusions, *i.e.*, can be deduced solutions from theory phenomena observation. Inductive process may be called learning process [9].

In the first group (inferential process), regression analysis techniques are the most relevant (lineal, no lineal and logistic regression), which study the relationship between dependent variables and independent variables [48]. In second group (inductive process), connectionist models (neural networks) and induction models (evolutionary algorithms and genetic programming) which can be considered learning algorithms [2, 9] are included.

The main problem using regression analysis techniques, is that, is necessary select the model to be used (lineal, logistic) before at variables selection and this difficult the generation of models [1]. Also, problems where analysis contains more than one variable, process become more complicated. Therefore, analysis regression for large data sets are realized by computers [43].

One of the most used techniques to analyze large data sets is symbolic regression. It's an application of genetic programming, also known as a technique of function identification, since it involves finding a mathematical expression, in symbolic form, which describes the relationship between dependent variable and independent variables as accurately as possible [35]. As working directly with genetic programming, symbolic regression is responsible for evolve mathematical functions in order to estimate the behavior of a data set [7].

Symbolic Regression is considered a supervised learning task, as it has a training sample, under which a model should be adjusted in order to reduce their error [34]. For this reason, it stands as a viable approach to the problem of data modeling, and does not assume the response of a structure, but discovers as it evolves.

In this paper, symbolic regression technique based on genetic programming to modeling data gold standard of NLP tasks are used.

2.1 Related Work

There are works to compute short text similarity that used external resources (knowledge-based) such as WordNet or British National Corpus, although these methods are good, they are no useful to work with languages that have not these linguistic resources. In this situation, a corpus-based method to find the short text similarity are presented in [59], specifically, paraphrase detection. This work, taken the problem of short similarity texts as an unsupervised classification problem, *i.e*., it doesn't use external resources to compute text similarity. Shrestha [59] indicates that best classification methods in text similarity are the vector space model [57]. This method is based on the vector space model but is different in having feature vector are created: the vector of terms is like the *word space model*, in which only the term distribution is stored. This is based on his assumption that the sentences are independent of each other, and each sentence represents a single idea, therefore, each term within the sentence must be related to this idea. The method is called Short text-based Vector Space Model (SVSM) and used the Microsoft Research Paraphrase Corpus (MSRPC) to evaluate. This corpus consists of 5,801 pairs of sentences, in which gold standard indicates what pair of sentences have a paraphrase equivalence.

According to [43], Vector Space Model which used cosine similarity measure is the baseline for most of the similarity studies, so that, in order to perform comparison to the method, Shrestha [59] performed similarity measure using classical vector space model with cosine similarity, in addition performed the Dice and Jaccard similarity measures. The proposed method was compared results to other methods which are tested on the same corpus. The obtained results indicate that their proposed vector space model (SVSM) improvement over the baseline (vector space model and cosine similarity) but are not better than other methods.

The growth of subjective information stored on Internet (mainly in social networks) has given rise interest in detecting sentiment, emotions (polarity) or opinions on different topics of such subjective information. In [58], two models to discover the polarity of messages extracted from twitter are presented. The importance of having models of polarity analysis lies in the possibility of polarity evaluation expressed by users about a topic. The polarity in NLP can be understood as the presence or absence of grammatical particles that define whether a sentence is positive or negative. In the work of [58], two models to solve this problem are proposed.

The first one is based on a lexical-syntactic method, in which certain lexical and syntactic characteristics are stored in a one-dimensional vector, indicating value that each characteristic has within a certain text (tweet). The second one is based on a graph method, in which a graph of co-occurrences between all the words contained in a corpus is created. The models are proposed for SemEval-2014 conference corpus, which consists of 6364 annotated (gold standard) in five levels of polarity (positive, negative, neutral, objective and

objective/neutral). For the first model, features such as emoticons, acronyms, URL's, hashtags, among others, were extracted and stored under the vector space model. For the second model, a co-occurrence graph was created with all corpus terms used, after which, four centrality algorithms were applied; of each algorithm the 300 most *central words* were obtained and these words were taken as the characteristics for the vector in each tweet. In the training phase, Support Vector Machines (SVM) was used in each of the proposed models, and as test, data of the corpus dedicated to this purpose were taken. The results show that the model based on the lexical-syntactic characteristics is better than the model based on co-occurrence graphs. In this work, two models of text representation were used together, a vector-space model and a graph-based representation model.

In [51], a model to determine the degree of relevance between two graph-based representations is proposed. The problem to solve is determinate the degree of relevance of reviews given by a reviewer to an author of a paper. The text-based reviews provided by a reviewer should be evaluated to ensure that it provides an accurate assessment of the work. One way in which quality of a review can be measured is to check if the reviews have semantic and syntactic similarity to the work. Ramachandran [51] proposes to use a graph-based representation (since this representation captures both the semantic and syntactic information of the information) to solve that problem. To calculate the relevance of the review, the *k-nn* algorithm was used. To calculate the similarity between two-text base-review, the text review should be transformed into graphs from POS (Part-Of-Speech) information such as nouns, verbs, adjectives, adverbs.

In addition, the similarity between the graphs was calculated by five different forms: Matching graph edges, same-syntax matching, different-syntax matching, double edge matching and complete vector. To evaluate his model, made use of a corpus obtained through Expertiza [21], which are composed by review-review pairs. This corpus consists of three data sets. As results, the best mean of similarity between graphs was double edge. To better understand the behavior of the model, it compares the results obtained using a standard similarity measure: cosine similarity. Because of this comparison, the cosine measure exceeded on average (from the similarities of graphs mentioned) to sets 1 and 2 of the corpus used, whereas in set 3, the proposed model clearly outperforms the cosine measure. For this work, text representation method used was graph representation, whereas that the similarity measures used were five graph-based metrics. Finally, the gold standard corpus used was composed on pairs of review-review text.

Then, some works that used data modeling techniques to improve results in NLP are showed. These works use several data modeling techniques to get their results.

In [63], an approach to learn similarity measures of high knowledge (knowledge-intensive similarity measure) for the Case-Based Reasoning task is presented. Case-Based Reasoning systems have been become a popular tool for development of knowledge-based systems. These systems record data (called cases) representing information about problems that were resolved in the past, where the idea is re-use this knowledge in solving new problems [13, 53], the problem of these systems is identifying when two problems are similar. In his work, Stahl develops a methodological framework that formalizes a necessary domain knowledge for defining optimal similarity measures, implementing machine learning strategies, that allows extract knowledge of data set training and after applied the induced similarity metric in real case-based reasoning. His method has evaluated on computer data recommendation and car recommendations.

In [17] shows the results of the application of a genetic program to evolve features ranking in the Information Retrieval task, where it is necessary to have a ranking function that allows order the documents retrieved according to degree of relevance to a query entered. In this work, is proposed a novel algorithm that introduces genetic programming into boosting procedure technique (this technique is usually utilized to improve model's performance). It indicates that the functions obtained in their proposed method, produce better results than several algorithms specified in other jobs.

In [3], a learning algorithm is defined, which uses various similarity measures to blend using a

linear regression algorithm to obtain the degree of semantic similarity between pairs of phrases. It presents the UKP system. Their system uses a log-linear regression model, which is trained on the data set train, to combine various text similarity measures of several complexities and forms. UKP participated in the pilot Semantic Textual Similarity (STS) in SemEval-2012 test and it obtained the three best performance on official metrics evaluation. As conclusion, author proposed a research walk to inspect the performance of a system that combine the output of several systems that participle in a single linear model.

A particular problem of detecting similarity in texts, it is when texts are of short length, this is due to the little information that it has to carry out that task [39]. In [8], this issue is addressed, mainly, in the detection paraphrase short documents, where paraphrasing is understood as an intellectual activity that consists of transferring own words the ideas expressed above [55]. He uses various ways to detect the similarity of short texts, those based on Overlap of words (Dice, Jaccard, and Cosine), sequence alignment (the longest common sub-sequence and Levenshtein algorithm), and those based on semantic information (Levenshtein, Neddleman-Wunsch and the Smith-Waterman algorithm with semantic information). Conceptual graphs are used for generation of summaries in [44, 45].

The aim in Carmona's thesis, was that the results of the similarity measures were combined to get a new similarity measure (by an optimization algorithm) to improve textual similarity. Optimization algorithm used in his work was genetic programming, with which it was combine basic steps to obtain the best results. The results of applying the genetic program are superior to the results of the baseline in both sets of data. The author concludes that their proposed approaches maintain competitiveness in the task detection and detection paraphrases reuse.

Also, [69] worked with text similarity of short length in a general query suggestion scenario (methods usually focus on recommending queries that are the most relevant or similar to the input query [31]). Firstly, a new similarity measure was proposed: Web-relevance similarity measure. This measure uses a keyword extraction system as the weighting function in a vector space model representation. On the other hand, two learning approaches were used to leverage the strengths of different similarity measures (including Dice, Jaccard, Cosine, KL-divergence [42], Web-based kernel [59] and their Web relevance similarity) because no reason to believe that any measure is ideal for all applications, and all similarity measures have different coverage. In first approach, once obtained several similarity scores (of different similarity measures), a regression function was used to learn a similarity metric (Metric learning), where the goal was to learn a function in two segments q and s $f_m(q_i, s_i) > f_m(q_j, s_j)$ such that output measure indicated that question q_i and s_i were more alike compared to q_j and s_j. In second approach preference ordering was learned (preference $a \succ b$ can mean that a is an algorithm that out performance b on a certain problem [6]). Then, given the labeled preference for each pair of questions can be train a binary classifier to predict if a question q_i is preferable than q_j.

Test data set was created by taking a random sample of 365 thousand queries from the top 1 million of most frequencies queries in 2005. Their results compare the quality of similarity metrics mentioned before as well as the learned similarity functions of two approaches. Their Web relevance similarity is better than methods existing in the state-of-art, and both Metric learning and Preference learning showed best results than other similarity metrics.

Support vector machine use a kernel which is responsible to data transformation and so allow an appropriate classification. Details with this technique are precisely the kernel, since the definition of an adequate kernel is not trivial. Although exists many methods for this definition, each problem requires a different kernel in specific objective, which limits definition to domain experts.

In [64], using a genetic program (specifically symbolic regression) in order to evolve the kernel in SVM and their parameters to improve results in classification task. His method is called KGP and was evaluated with several data sets from UCI repository [65]. His proposed method uses at terminal nodes (in syntactic trees used in genetic

programming) the basic kernels of SVM (polynomial, Gaussian or Sigmoidal). Symbolic regression evolves their kernels in an iterative process. The proposed method was compared to another similar model (SVM-GRID). SVM-GRID found the optimum parameters of a Gaussian kernel. The results show that KGP gets an aptitude upper of 95% in three data sets, on the other and, SVM-GRID indicates best results in other five data sets.

As we can see, exist several data modeling techniques that can be used to improve results in NLP task, but any of the works showed proposed a general method that solves different NLP task. From these works, we proposed a method that is used in many NLP task, which allows improve results when this task use a gold standard corpus evaluation.

3 Proposed Method

In this paper, a liable method to improve the performance of ATS systems which are evaluated by a gold standard is proposed. Then, the proposed method stages are described. In section 5, we describe the application of the proposed method on ATS task where the method has been successfully applied.

3.1 Main Stages

The proposed method includes the following: (1) corpus of a specific ATS task with their respective gold standard evaluation; (2) a model of representation that allows depict the corpus which provide a solution to ATS task; (3) the topline for the corpus used; and (4) a data modeling technique that allows approximate the topline values. Then these elements are described.

Corpus and Gold Standard. The corpus used must be of a specific ATS task. Moreover, this corpus must have a gold standard which contains the results of the test set and annotations that allows evaluate such task.

Representation Model. To facilitate the relevance of estimation process, corpus of text documents need to be transformed into a form that can be effectively processed by computers [16]. One of the most used representation model is the Vector Space Model [58, 61, 62], in this representation the dimensions of each text document corresponding to individual features such as words, concepts, collocations, word senses or others. Other representation models can be trees or graphs.

Topline. The best result obtained for the given collection, is called topline [36]. Topline describes the ideal behavior of an ATS system. This heuristic can be obtained of different forms, but it also finds the best combination of the sentences, words, concepts, collocations or other text document representation that obtained the best result in the ATS system evaluated with their respective gold standard. For this work, the topline should be describe the ideal behavior of an ATS task.

Data Modeling Technique. This technique should allow approximate or find the best model that describes the data topline, that is, enable modeling the best combination of features (sentences, words or others) who obtained the best results in ATS task evaluated. Data model techniques can be: regression analysis techniques (lineal, no lineal and logistic regression), neural networks, symbolic regression or others. The only restriction is that the used technique must return a model (mathematical, algorithmic, physical or otherwise) which would give solution to ATS task.

3.2 Algorithm

Regardless of the representation model, the method to extract the topline, and data modeling technique, the application of the proposed method consists of the following main steps:

1. Depict the corpus under a model of appropriate representation for the ATS task.
2. From selected representation model, test all or most of the possible combinations to find the best result for each element of the gold standard being evaluated. This way obtaining the set of characteristics that were selected for the best results. This data set is the topline.
3. Once obtained the topline, it can form a set of related data (c, f), in which c indicates the characteristic that was used to obtain the most

Table 1. Models used to measure the relevance sentence in a document

Work	Feature	Model
[47]	Word position	$f(i) = \frac{n-i+1}{n}$ (1)
		$f(i) = 1/i$ (2)
		$f(i) = \left(\frac{1}{2}\right)^{i-1}$ (3)
[4]	Sentence position	$f(i) = \sqrt{(1/i)}$ (4)
[20]	Sentence position	$f(i) = t(i-x)+x$ $x = 1 + \frac{(n-1)}{2}$ (5)

similar to the gold standard results and f indicates the frequency with which this feature it was used throughout the evaluation.

4. The data set (c, f) is the input set for the data modeling technique. The selected approach should find a model that describes the set (c, f) obtained from the topline. The output of this step is a model.

Model obtained in previous step is the best approach with respect to the topline obtained from the gold standard, therefore, this model must return the best results evaluating all data in the ATS task. To verify this, the model obtained should be evaluated with the truly data set of the ATS task.

4 Proposed Method Applied to ATS

ATS consists in reducing the size of a text while preserving their information content [40]. There is abstractive and extractive text summarization [28, 62, 67]. An abstractive summary consists of novel phrasings describing the content of the original (which might be paraphrased), and an extractive summary only contains whole of literal portions extracted from the original [28]. In this paper, extractive ATS, is used.

ATS task requires to determinate what features must be considered to meet the desired quality. Traditional methods focus on sentences and define scores according to their meaning. The features includes keywords, position in the sentence, and certain linguistic information [26].

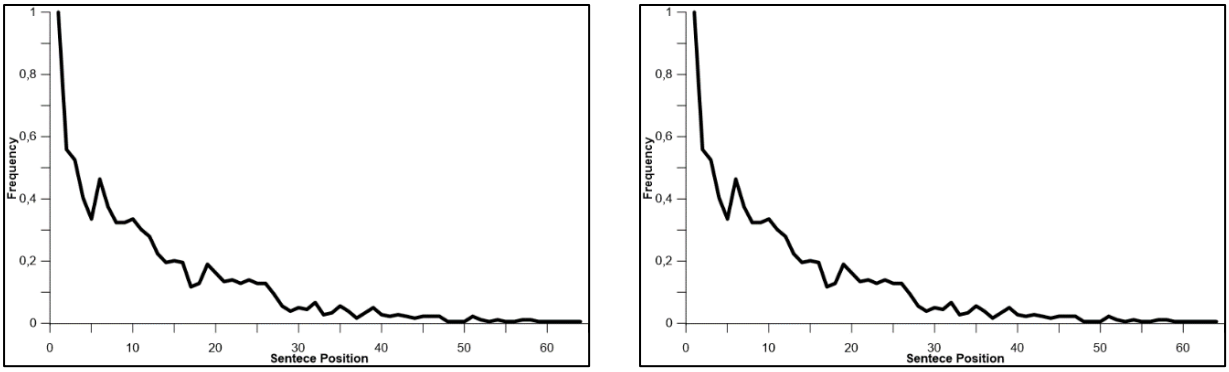
One of the most used features in automatic extractive summaries has been the position of the sentence within a document [14]. Many studies tried to model summaries made by humans, using the position of sentences within documents [4, 14, 19, 26, 47, 61, 66]. According to [14], the sentence relevance is assigned by ordinal position within the text, giving more weight to the first sentence of that document to the last sentence thereof. Several models that have been used to measure the relevance sentence in a document are showed in Table 1.

Although these models have been successful in their experiments, they do not reflect how a human makes a summary. Therefore, the first approximation of the proposed method in this paper is to find a model that describes the relevance sentence position in a document and their frequency to be part of a summary.

The stages of proposed method for ATS are described:

Corpus with Gold Standard: To prove the proposed method in automatic extractive text summarization, we use the DUC01 and DUC02 collection. DUC01 contains 309 news articles in English, where each one has the golden summaries created by two different people. DUC02 contains 567 news articles in English of different lengths and different topics. Also, the two gold standard summaries were created by two human experts.

Representation Model: The representation model selected is the vector space model, where



a) b)

Fig. 1.a. Representation of data set: sentence position – frequency (Topline) for DUC01 collection)
Fig. 1.b. Representation of data set: sentence position – frequency (Topline) for DUC02 collection

in each element represent a sentence from the document to summarize.

Topline: In work of [60], topline was obtained trying all combinations of sentences in one document. The best combination of sentences, which has the highest score is chosen as topline result for a given document. The topline was obtained for 309 documents of DUC01 collection, and 567 documents of DUC02 collection.

Data modeling technique: The selected technique to model topline data is symbolic regression. We use symbolic regression because is a viable technique to the problem of data modeling, and does not assume the answer to structure.

From topline obtained by [60] can be get the set of related data (c, f) where c indicates the position of sentence and f indicates the sentence frequency that was selected to be part of a summary in the DUC01 and DUC02 collection. The data set (c, f) can be graphically represented, see Fig. 1, for topline of DUC01 and, see Fig. 2, for topline of DUC02 collection. Topline results are normalized to 1.

The fourth step of the algorithm indicates that set (c, f) can be modeled by a data modeling technique. As mentioned above, data modeling technique selected was symbolic regression. To obtaining the models that describes data showed in Fig. 1and Fig. 2. Several experiments were realized used symbolic regression.

On one hand, to describe the topline of DUC01 collection, four models were obtained. These models are showed in Eqs. (6, 7, 8, 9), where X indicates the position of the sentence inside of the document.

$$f(i) = \left(16.72 + \frac{7587}{X^3} - (5.9 * 10^{-7})X^4 - \frac{19430956}{2.74^{X^2} - 26X^9}\right)^{-0.03443X}, \quad (6)$$

$$f(i) = \frac{7}{8}X^{\frac{1}{2}\left(-\frac{374236X}{-(-22)^{7-X}+439\sqrt{X}+6039598} - \frac{1}{58}X\sqrt{\frac{7}{2}-\frac{4}{7}}\right)}, \quad (7)$$

$$f(i) = \frac{12371X^{\frac{5}{2}} + 101671}{-474X^{\frac{7}{2}} + X^6 + 12371X^3 + 101671X}, \quad (8)$$

$$f(i) = \frac{114X - 233X^2}{166 - 192X^2\left(\sqrt{X + 103^{\frac{1}{25}X}}\right)}. \quad (9)$$

On the other hand, four models were obtained to describe the topline of DUC02 collection. These models are showed in Eqs. (10, 11, 12,13).

$$f(i) = \left(\frac{4X}{-X^2 + X - 1883.84}\right) + \left(\frac{3X}{X^2 + 2X}\right), \quad (10)$$

$$f(i) = \left|\frac{99.95 - |X| + \frac{X+97.94}{X^2} - X}{|X^2| - 2X + 197.89}\right|, \quad (11)$$

$$f(i) = \frac{\frac{X-1}{1.04464 - 0.478469X} + \frac{0.957266}{X^3} - 1.47847X + 95.48}{(X - 1.09)(2X - 1.04464) + \left(\frac{95.48}{X} - 0.47847X\right)\left(2X - \frac{1}{X}\right)}, \quad (12)$$

Table 2. Model error obtained by evaluated topline of DUC01

Work	Model	Absolute Error
[47]	Eq. (1)	9.4203
	Eq. (2)	4.0000
	Eq. (3)	5.3125
[4]	Eq. (4)	2.1687
[20]	Eq. (5)	8.1873
Proposed	Eq. (6)	1.6018
	Eq. (7)	1.2505
	Eq. (8)	1.0209
	Eq. (9)	1.0049

Table 3. Model error obtained by evaluated topline of DUC01

Work	Model	Absolute Error
[47]	Eq. (1)	11.3669
	Eq. (2)	3.7368
	Eq. (3)	5.2327
[4]	Eq. (4)	1.9976
[20]	Eq. (5)	9.9515
Proposed	Eq. (10)	1.3828
	Eq. (11)	0.6780
	Eq. (12)	0.5693
	Eq. (13)	0.5463

$$f(i) = \frac{41.3294}{15.53+X^2-2X+\frac{704.55}{X}} + \frac{X^2+4X-663.61}{-2X^3-665.61X-\frac{31.06}{X}}, \quad (13)$$

To know if the obtained models by symbolic regression describe the behavior of topline, it was evaluated against same topline.

In addition, the models were evaluated, shown in Table 1. The fitness function used to evaluate all models was the *absolute error metric*, to obtain the true error and facilitate comparison to other models.

The absolute errors obtained from evaluating the models described in Table 1and the models described in Eqs. (6, 7, 8, 9) with the topline of DUC01 collection, are shown in Table 2. In Table 3 are shown the absolute errors obtained from the evaluation of the models described in Table 1 and the models described by the Eqs. (10, 11, 12, 13).

On one hand, models obtained by the proposed method gets lowest error in both, topline of DUC01 and topline of DUC02. On the other hand, of the models described in the state-of-the-art, the model of [4], described by Eq. (4) is the one that gets the lowest absolute error for topline of DUC01 and DUC02.

The models obtained by symbolic regression used topline of DUC01 and DUC02 as input data set obtains the lowest absolute errors respectively.

The last step in algorithm of proposed method, indicates that the model obtained in fourth step it must be evaluated with actual data from ATS task. Because the models shown above were obtained from DUC01 and DUC02, it is necessary to test each model with the data corpus for which it was obtained.

It is necessary apply the models obtained to a method of ATS task to evaluate the models. In this work, the method selected to test each model is the method proposed by [19], which is based on a genetic algorithm. In method of [19], all parameters that the genetic algorithm use are calculated automatically considering the structure of the source text (considering the number of sentences that the document contains).

Basically, the genetic algorithm of [19] selected the sentences that generates a good summary. The main of the genetic algorithm, is the *fitness function* used to guide the evolution, which is composed by two elements: coverage and sentence position.

Coverage measure the similarity between resultant summary and the source document which implies that a best summary should contains the most frequent words (or text unit) with respect to the original text, in addition to including the most important information (non-redundant).

Sentence position is the model used to calculate the importance of a sentence to be part of the summary, it based on their position. Fitness function used in [19] :

Table 4. Model error obtained by evaluated topline of DUC01

Text unit / Model	1-gram	2-gram	3-gram
Eq. (1)	0.44615	0.44392	0.44548
Eq. (2)	0.44335	0.44460	0.44337
Eq. (3)	0.42884	0.43118	0.42604
Eq. (4)	0.44805	0.44559	0.44287
Eq. (5)	0.44482	0.44885	0.44549
Eq. (6)	0.41589	0.41703	0.41562
Eq. (7)	0.41815	0.41829	0.41770
Eq. (8)	0.44855	0.44495	0.44384
Eq. (9)	0.44947	0.44876	0.44534

Table 5. ROUGE-2 results of models applied to DUC01 collection

Text unit / Model	1-gram	2-gram	3-gram
Eq. (1)	0.19142	0.19668	0.19660
Eq. (2)	0.19235	0.19606	0.19551
Eq. (3)	0.17304	0.17630	0.17418
Eq. (4)	0.19611	0.19536	0.19620
Eq. (5)	0.18811	0.19454	0.19441
Eq. (6)	0.16237	0.16210	0.16329
Eq. (7)	0.16338	0.16426	0.16222
Eq. (8)	0.19830	0.19581	0.19662
Eq. (9)	0.19450	0.19834	0.19713

$$fitness = \beta \times \delta, \tag{14}$$

where δ is the model of sentence position indicated in Eq. (5), and β is the coverage feature described by Eq. (15).

$$\beta = \frac{\sum_{p=\{word \in S\}}^{m} frequency(p,T)}{\sum_{q=\{word \in T\}}^{m} frequency(q,T)}. \tag{15}$$

4.1 Experimental Results

Genetic algorithm of [19] is the base to test the models obtained by topline of DUC01 and topline of DUC02. In fitness function showed in Eq. (14), we replaced the sentence position model (β) used by [19], for each of the models obtained in this work. And the coverage feature (δ), that [19] use words as text units, we use different text units based on *n-grams*, specifically: 1-grams, 2-grams, and 3-grams in word level.

The ROUGE evaluator is used to evaluate the experiment realized in each DUC collection. The ROUGE evaluation toolkit is used to evaluate our results because it has a highly correlation with human judgments [38]. It compares the summaries generated by a system to the human-generated (gold-standard) summaries. For comparison, it uses n-gram statistics. Our evaluation is done using n-gram (1, 1) setting of ROUGE, which was found to have the highest correlation with human judgments, namely, at a confidence level of 95%. ROUGE evaluates the f-measure that is a balance (not an average) of recall and precision results. The results are presented for ROUGE-1 and ROUGE-2 metrics to 100 words.

Table 4 shows the ROUGE-1 results obtained by the models applied to DUC01 collection, and Table 5 shows the ROUGE-2 for the same collection. Table 6 and Table 7 shows the ROUGE-1 and ROUGE-2 applied to DUC02 collection respectively.

The results obtained by applying the different models that describe the importance of the sentences position to generate a text summary, both the models of the state-of-the-art, as well as the models obtained by the method proposed in this work, it can be observed that, the less the absolute error when comparing the model used with the topline of dataset, better results are obtained in the evaluation.

Because the models obtained and described by the Eqs. (6, 7, 8, 9) for DUC01, and by the Eqs. (10, 11, 12, 13) for DUC02, were obtained by trying to reduce their error compared to the topline of each dataset, that is, by decreasing their absolute error, it is to be expected that the more accurate the model, the better results it would obtain.

In Table 8, sorted results of ROUGE-1 and ROUGE-2 evaluations applying the method of [19]

Table 6. ROUGE-1 results of models applied to DUC02 collection

Text unit			
Model	**1-gram**	**2-gram**	**3-gram**
Eq. (1)	0.47864	0.48184	0.47977
Eq. (2)	0.47964	0.48099	0.47982
Eq. (3)	0.45795	0.46371	0.45980
Eq. (4)	0.48312	0.48295	0.48038
Eq. (5)	0.47576	0.47779	0.47870
Eq. (10)	0.47289	0.47253	0.47114
Eq. (11)	0.48281	0.48309	0.48157
Eq. (12)	0.47911	0.48269	0.47860
Eq. (13)	0.48470	0.48273	0.48088

Table 7. ROUGE-1 results of models applied to DUC02 collection

Text unit			
Model	**1-gram**	**2-gram**	**3-gram**
Eq. (1)	0.22040	0.22762	0.22666
Eq. (2)	0.22466	0.22684	0.22667
Eq. (3)	0.19636	0.20677	0.20444
Eq. (4)	0.22651	0.22894	0.22678
Eq. (5)	0.21631	0.22460	0.22531
Eq. (10)	0.21617	0.21950	0.21794
Eq. (11)	0.22651	0.22935	0.22794
Eq. (12)	0.22087	0.22794	0.22577
Eq. (13)	0.22792	0.22872	0.22759

using the models described above for the DUC01 collection, are showed. The best result for ROUGE-1 evaluation are obtained by the Eq. (9), which it is the model with least absolute error. For ROUGE-2, the best result is obtained by Eq. (8), which is the second-best model to DUC01.

In Table 8, sorted results of ROUGE-1 and ROUGE-2 evaluations applying the method of [19] using the models described above for the DUC01 collection, are showed. The best result for ROUGE-1 evaluation are obtained by the Eq. (9), which it is the model with least absolute error. For ROUGE-2, the best result is obtained by Eq. (8), which is the second-best model to DUC01.

Table 8. Sorted results of ROUGE-1 and ROUGE-2 evaluation of models applied to DUC01 collection

Work	Model	ROUGE-1	ROUGE-2
Proposed	Eq. (9)	**0.44947**	0.19450
[20]	Eq. (5)	0.44885	0.19454
Proposed	Eq. (8)	0.44855	**0.19830**
[4]	Eq. (4)	0.44805	0.19611
[47]	Eq. (1)	0.44615	0.19142
[47]	Eq. (2)	0.44460	0.19606
[47]	Eq. (3)	0.43118	0.17630
Proposed	Eq. (7)	0.41829	0.16426
Proposed	Eq. (6)	0.41703	0.16210

Table 9. Sorted results of ROUGE-1 and ROUGE-2 evaluation of models applied to DUC02 collection

Work	Model	ROUGE-1	ROUGE-2
Proposed	Eq. (13)	0.**48470**	0.22792
[4]	Eq. (4)	0.48312	0.22651
Proposed	Eq. (11)	0.48309	0.**22935**
Proposed	Eq. (12)	0.48269	0.22794
[47]	Eq. (1)	0.48184	0.22762
[47]	Eq. (2)	0.48099	0.22684
[20]	Eq. (5)	0.47870	0.22531
Proposed	Eq. (10)	0.47289	0.21617
[47]	Eq. (3)	0.46371	0.20677

In Table 9, sorted results of ROUGE evaluations, are showed. For this case, the best result for ROUGE-1 are obtained by the model of Eq. (13), which it is a model obtained by the proposed method and it has the least absolute error. The best result for ROUGE-2 are obtained by the Eq. (11), which it is the third best model for DUC02 topline.

5 Conclusions

In this paper, a novel process based on symbolic regression, which makes it possible to obtain models that describes the importance of a sentence within the source document, according to their position in the document, to be used in the generation of an extractive text summary.

First, the four main elements that the method needs in general are described, and each of them is described in detail. Afterwards, and once explained the elements that compose the proposed method, it is shown to application of the method to automatic extractive text summarization task.

As standard, DUC collections are taken as database datasets to test automatic text summarization systems, and based on this, it was decided to use the DUC01 and DUC02 datasets to evaluate and compare the proposed method.

Because the proposed method obtains models that indicate the importance of sentences based on its position, and because there are several models with the same purpose in the state-of-the-art, it was decided to use method of [19] to compare the models described in the state-of-the-art-against the models obtained in this work.

Applying the method of [19] and using each of the models obtained, for DUC01 and DUC02 respectively, it was determined that, in general terms whereas more accurate is the model compared to the data sets being modeled (in this case, the topline for each data set), better results in evaluation are obtained.

The work of [19] only uses two sentence features, coverage and sentence position. Coverage feature are not modified in this work, only the sentence position. Modifying the sentence position, that is, replacing the model used by [19], the models obtained by the proposed method and the models used in state-of-the-art, are evaluated. It indicates that only with two sentence features, coverage and sentence position, is enough to obtain competitive results.

As main contribution, a method to describe the behavior of a set of numerical values (as a topline), which describes the best result of a natural language processing task, are proposed. The proposed method was applied to automatic text summarization task, and results obtained indicates that the model obtained by the proposed method obtain best results that model used in state-of-the-art. Whereas more accurately is the model obtained by the proposed method, best result is obtained in final evaluation.

References

1. **Banzhaf, W., Francone, F. D., Keller, R. E., & Nordin, P. (1998).** *Genetic programming: an introduction: on the automatic evolution of computer programs and its applications.* Morgan Kaufmann Publishers Inc.
2. **Bär, D., Biemann, C., Gurevych, I., & Zesch, T. (2012).** UKP: computing semantic textual similarity by combining multiple content similarity measures, *Proceedings of the First Joint Conference on Lexical and Computational Semantics*, pp. 435– 440.
3. **Bossard, A., Genereux, M., & Poibeau, T. (2008).** Description of the LIPN systems at TAC 2008: Summarizing information and opinions. *Text Analisys Conference.*
4. **Box, G. E. P. & Draper, N. R. (1987).** *Empirical model-building and response surface*. John Wiley & Sons, Inc.
5. **Hüllermeier, E., Fúrnkranz, J., Cheng, W., & Brinker, K. (2008).** Label ranking by learning pairwise preferences, *Artificial Intelligence* Vol. 172, pp. 1897–1916.
6. **Can, B. & Heavey, C. (2011)** Comparison of experimental designs for simulation-based symbolic regression of manufacturing systems. *Computers and Industrial Engineering,* Vol. 61, pp. 447–462. DOI: 10.1016/j.artint.2008.08.002.
7. **Álvarez, M. Á. (2014).** *Detección de similitud semántica en textos cortos*. Tesis: Instituto Nacional de Astrofísica, Optica y Electrónica, INAOE.
8. **Ciampi, A. & Lechevallier, Y. (2007).** Statistical Models and Artificial Neural Networks: Supervised Classification and Prediction Via Soft Trees. *Advances in Statistical Methods for the Health Sciences*, pp. 239–261. DOI: 10.1007/978-0-8176-4542-7_16
9. **Clark, A., Fox, C., & Lappin, S. (2013).** *The Handbook of Computational Linguistics and Natural Language Processing*, Wiley.
10. **Cook, B. L., Progovac, A. M., Chen, P., Mullin, B., Hou, S., & Baca-Garcia, E. (2016).** Novel Use of Natural Language Processing (NLP) to Predict Suicidal Ideation and Psychiatric Symptoms in a Text-Based Mental Health Intervention in Madrid.

Computational and Mathematical Methods in Medicine.

11. **Chen, S., Ma, B., & Zhang, K. (2009).** On the similarity metric and the distance metric. *Theoretical Computer Science,* Vol. 410, pp. 2365–2376. DOI: 10.1016/j.tcs.2009.02.023.
12. **Cheng, W. & Hüllermeier, E. (2008).** Learning Similarity Functions from Qualitative Feedback. *Advances in Case-Based Reasoning,* pp. 120–134.
13. **Edmundson, H. P. (1969).** New Methods in Automatic Extracting. *Journal of the ACM (JACM)* Vol. 16, pp. 264–285. DOI: 10.1145/321510. 321519.
14. **Elliott, D., Hartley, A., & Atwell, E. (2003).** Rationale for a multilingual corpus for machine translation evaluation. *Proceedings of CL2003: International Conference on Corpus Linguistics*, Lancaster University, pp. 191–200.
15. **Fan, W., Gordon, M. D., & Pathak, P.** A generic ranking function discovery framework by genetic programming for information retrieval. *Information Processing & Management,* Vol. 40, pp. 587–602. DOI: 10.1016/j.ipm.2003.08.001
16. **Feng, W. & Xinshun, X. (2010).** AdaGP-Rank: Applying boosting technique to genetic programming for learning to rank. *IEEE Youth Conference on Information Computing and Telecommunications (YC-ICT)*, pp. 259–262.
17. **Friedman, C., Rindflesch, T.C., & Corn, M. (2013).** Natural language processing: State of the art and prospects for significant progress, a workshop sponsored by the National Library of Medicine. *Journal of Biomedical Informatics,* Vol. 46, pp. 765–773. DOI: 10.1016/j.jbi.2013.06 .004.
18. **García-Hernández, R. & Ledeneva, Y. (2013).** Single Extractive Text Summarization Based on a Genetic Algorithm. *Pattern Recognition,* pp. 374–383.
19. **García-Hernández, R., Montiel, R., Ledeneva, Y., Rendón, E., Gelbukh, A., & Cruz, R. (2008).** Text Summarization by Sentence Extraction Using Unsupervised Learning. (*MICAI´08) Advances in Artificial Intelligence,* **Gelbukh, A. & Morales, E., eds.,** Springer Berlin Heidelberg, pp. 133–143. DOI: 10.1007/3-540-45571-X_52.
20. **Gehringer, E., Ehresman, L. M., Conger, S. G., & Wagle, P. A. (2006).** Reusable learning objects through peer review: The Expertiza approach. *Innovate: Journal of Online Education,* Vol. 3.
21. **Gillick, D., Favre, B., Hakkani-Tür, D., Bohnet, B., Liu, Y., & Xie, S. (2009).** The ICSI/UTD Summarization System at TAC´09.
22. **Goshtasby, A. A. (2012).** *Similarity and Dissimilarity Measures.* Image Registration, Springer London, pp. 7–66.
23. **Hartigan, J. A. (1975).** *Clustering Algorithms.* John Wiley & Sons, Inc.
24. **Hinze, A., Heese, R., Luczak-Rösch, M., & Paschke, A. (2012).** Semantic Enrichment by Non-experts: Usability of Manual Annotation Tools. *The Semantic Web – (ISWC´12) 11th International Semantic Web Conference*, Springer Berlin Heidelberg, pp. 165–181.
25. **Hirao, T., Isozaki, H., Maeda, E., & Matsumoto, Y. (2002).** Extracting important sentences with support vector machines. *Proceedings of the 19th international Conference on Computational linguistics,* Vol. 1, pp. 1–7. DOI: 10.3115/1072228. 1072281.
26. **Hogenboom, F., Frasincar, F., & Kaymak, U. (2010).** An Overview of Approaches to Extract Information from Natural Language Corpora. *Tenth Dutch-Belgian Information Retrieval Workshop (DIR´10),* **Heijden, M.v.d.,. Hinne, M, Kraaij, W., Kuppeveld, M.v., Verberne, S., & Weide, T.v.d., eds.,** Nijmegen, The Netherlands, pp. 69–70.
27. **Hovy, E. & Lin, C.Y. (1998).** Automated text summarization and the SUMMARIST system. *Proceedings of a workshop on held at Baltimore*, Association for Computational Linguistics, pp. 197–214. DOI: 10.3115/1119089.1119121.
28. **Huang, L., Milne, D. N., Frank, E., & Witten, I. H. (2012).** Learning a concept-based document similarity measure. *Journal of the American Society for Information Science and Technology,* Vol. 63, No. 8, pp. 1593–1608. DOI: 10.1002/ asi.22689.
29. **Islam, Z. & Mehler, A. (2013).** Automatic readability classification of crowd-sourced data based on linguistic and information-theoretic features. *Computación y Sistemas,* Vol. 17, No. 2. DOI: 10.13053/cys-17-2-1516.
30. **Jiang, D., Leung, K. W. T., Vosecky, J., & Ng, W. (2014).** Personalized Query Suggestion With Diversity Awareness. *IEEE 30th International Conference on Data Engineering*, pp. 400–411. DOI: 10.1016/j.knosys.2015.09.003.
31. **Jones, K. S. & Galliers, J. R. (1996).** *Evaluating Natural Language Processing Systems: An Analysis and Review*. Springer-Verlag New York, Inc.
32. **Kilgariff, A. (1998).** Senseval: An exercise in evaluating word sense disambiguation programs. *Proceedings of the first international conference on language resources and evaluation*, Liège, Belgium, pp. 581–588.

33. **Kommenda, M., Affenzeller, M., Burlacu, B., Kronberger, G., & Winkler, S. M. (2014)** Genetic programming with data migration for symbolic regression. *Proceedings of the Conference companion on Genetic and evolutionary computation companion*, ACM, pp. 1361–1366. DOI: 10.1145/2598394.2609857.
34. **Koza, J. R. (1992).** Genetic programming: on the programming of computers by means of natural selection*. Statics and Computing*, Vol. 4, No. 2, pp. 87–112. DOI: 10.1007/BF00175355.
35. **Ledeneva, Y.N. (2008).** *Automatic language-independent detection of multiword descriptions for text summarization*, Doctorado en Ciencias de la Computación Thesis, Instituto Politécnico Nacional.
36. **Lee, M. C., Chang, J. W., & Hsieh, T. C. (2014)** A Grammar-Based Semantic Similarity Algorithm for Natural Language Sentences. *The Scientific World Journal*, Vol. 2014. DOI: 10.1155/2014/437162.
37. **Lin, C.Y. (2004).** Rouge: A package for automatic evaluation of summaries. *Text summarization branches out: Proceedings of the ACL-04 workshop*.
38. **Liu, X., Zhou, Y., & Zheng, R. (2007).** Sentence Similarity based on Dynamic Time Warping. *International Conference on Semantic Computing (ICSC´07)*, pp. 250–256. DOI: 10.1109/ICSC. 2007.48.
39. **Luhn, H. P. (1958).** The automatic creation of literature abstracts. *IBM Journal of Research and Development,* Vol. 2, No. 2, pp. 159–165. DOI:10.1147/rd.22.0159.
40. **Magdaleno, D., Fuentes, I. E., & García, M. M. (2015).** Clustering XML Documents Using Structure and Content based on a New Similarity Function OverallSimSUX. *Computación y Sistemas,* Vol. 19, No. 1, pp. 151–161. DOI: 10.13053/CyS-19-1-1922.
41. **Metzler, D., Dumais, S., & Meek, C. (2007).** Similarity measures for short segments of text. *Proceedings of the 29th European conference on IR research*, Springer-Verlag, pp. 16–27. DOI: 10.1007/978-3-540-71496-5_5.
42. **Mihalcea, R. & Corley, C. (2006).** Corpus-based and knowledge-based measures of text semantic similarity. *AAAI´06*, pp. 775–780. DOI: 10.1.1.232.208.
43. **Milton, J. S. (2007).** *Estadística para Biología y Ciencias de la Salud*.
44. **Miranda, S., Gelbukh, A., & Sidorov, G. (2014).** Generating summaries by means of synthesis of conceptual graphs. [in Spanish] Revista Signos 47(86), 463–485.
45. **Miranda, S., Gelbukh, A., & Sidorov, G. (2014).** Summarizing conceptual graphs for automatic summarization task. *International Conference on Conceptual Structures*, pp. 245–253.
46. **Newman, G.D. (2006).** El razonamiento inductivo y deductivo dentro del proceso investigativo en ciencias experimentales y sociales. *Revista de educación,* Vol. 12, pp. 180–205.
47. **Niewiadomski, A. & Akinwale, A. (2015).** Efficient Similarity Measures for Texts Matching, *Journal of Applied Computer Science,* Vol. 23, No. 1, pp. 7–28.
48. **Ouyang, Y., Li, W., Lu, Q., & Zhang, R. (2010).** A study on position information in document summarization. *Proceedings of the 23rd International Conference on Computational Linguistics: Posters, Association for Computational Linguistics,* pp. 919–927.
49. **Palacios-Cruz, L., Pérez, M., Rivas-Ruiz, R., & Talavera, J. O. (2013).** Investigación clínica XVIII. Del juicio clínico al modelo de regresión lineal. *Revista Médica del Instituto Mexicano del Seguro Social,* Vol. 51, No. 6, pp. 656–661.
50. **Pidd, M. (2009).** *Tools for Thinking: Modelling in Management Science*.
51. **Poibeau, T. & Messiant, C. (2008).** Do we still Need Gold Standards for Evaluation?. *Language Resource and Evaluation Conference*, Morocco.
52. **Ramachandran, L. & Gehringer, E. F. (2011).** Determining Degree of Relevance of Reviews Using a Graph-Based Text Representation. *Proceedings of the 2011 IEEE 23rd International Conference on Tools with Artificial Intelligence*, IEEE Computer Society, pp. 442–445. DOI: 10.1109/ICTAI.2011.72.
53. **Reddy, P. & Balabantaray, R. C. (2012).** Improvisation of the Document Summarization by Combining the IR Techniques with “Code-Quantity Memory and Attention” Linguistic Principles. *Procedia Technology,* Vol. 6, pp. 118–125. DOI: 10.1016/j.protcy.2012.10.015.
54. **Riesbeck, C. K. & Schank, R. C. (1989).** *Inside Case-Based Reasoning*. L. Erlbaum Associates Inc.
55. **Roy, J. F., Pitarque, A., & Ruiz, J. C. (1998).** Redes Neurales Vs Modelos Estadisticos Simulaciones Tareas Prediccion Clasificacion. *Psicologia*, Vol. 19, pp. 387–400.
56. **Rus, V., Banjade, R., & Lintean, M. (2014).** On Paraphrase Identification Corpora. *Proceedings of the Ninth International Conference on Language,* Resources and Evaluation.
57. **Sahami, M. & Heilman, T. D. (2006).** A web-based kernel function for measuring the similarity of short text snippets. *Proceedings of the 15th international*

conference on World Wide Web, ACM, pp. 377–386. DOI: 10.1145/1135777.1135834.

58. **Salton, G., Wong, A., & Yang, C. S. (1975).** A vector space model for automatic indexing. *Commun. (ACM´18)*, pp. 613–620. DOI: 10.1145/361219.361220.
59. **Sanzón, Y. M., Vilariño, D., Somodevilla, M. J., Zepeda, C., & Tovar, M. (2015).** Modelos para detectar la polaridad de los mensajes en redes sociales. *Research in Computing Science* 99, pp. 29–42.
60. **Shrestha, P. (2011).** Corpus-Based methods for Short Text Similarity. *Rencontre des Étudiants Chercheurs en Informatique pour le Traitement automatique des Langues*, pp. 297.
61. **Sidorov, G. (2013).** *Non-linear construction of n-grams in computational linguistics: syntactic, filtered, and generalized n-grams*. [in Spanish] SMIA, p. 166.
62. **Sidorov, G. (2019).** *Syntactic n-grams in computational linguistics*. Springer, p. 125.
63. **Simón, J. R. (2017).** *Calculo de topline para la generación de resúmenes usando algoritmos genéticos*. Bachelor thesis, Universidad Autónoma del Estado de México.
64. **Simon, J. R., Ledeneva, Y., & García-Hernández, R. (2018).** Calculating the Significance of Automatic Extractive Text Summarization using a Genetic Algorithm. *Journal of Intelligent & Fuzzy Systems*, Vol. 35, pp. 1–12.
65. **Sparck, K. (1998).** Automatic Summarising: Factors and Directions. *Advances in Automatic Text Summarization*, pp. 1–12.
66. **Stahl, A. (2004).** *Learning of Knowledge-Intensive Similarity Measures in Case-Based Reasoning*. PHD-Thesis, Technische Universität Kaiserslautern.
67. **Sullivan, K. M. & Luke, S. (2007).** Evolving kernels for support vector machine classification. *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, pp. 1702–1707. DOI:10.1145/1276958.1277292.
68. **UCI (2015).** *UCI Machine Learning Repository*.
69. **Vazquez, E., García-Hernández, R., & Ledeneva, Y. (2018).** Sentence Features Relevance for Extractive Text Summarization using Genetic Algorithms. *Journal of Intelligent & Fuzzy Systems*. Vol. 35, No. 1, pp. 353–365. DOI: 10.3233/JIFS-169594.
70. **Verma, R. & Lee, D. (2017).** Extractive Summarization: Limits, Compression, Generalized Model and Heuristics. *Revista Computación y Sistemas,* Vol. 21, No. 4, pp. 787–798. DOI: 10.13053/cys-21-4-2855.
71. **Wissler, L., Almashraee, M., Díaz, D. M., & Paschke, A. (2014).** The Gold Standard in Corpus Annotation. *IEEE GSC.*
72. **Yih, W. T. & Meek, C. (2007).** Improving similarity measures for short segments of text. *Proceedings of the 22nd national conference on Artificial intelligence,* Vol. 2, pp. 1489–1494.

Article received on 22/03/2018; accepted on 23/05/2018.
Corresponding author is Yulia Ledeneva.