ISSN 2007-9737

The Impact of Key Ideas on Automatic Deception Detection in Text

Ángel Hernández Castañeda [1,2], René Arnulfo García Hernández[2], Yulia Ledeneva[2], Christian Eduardo Millán Hernández[2]

[1] Cátedras CONACYT, Mexico

[2] Universidad Autónoma del Estado de México, Mexico

angelhc2305@gmail.com, renearnulfo@hotmail.com, yledeneva@yahoo.com

Abstract. In recent years, with the rise of the Internet, the automatic deception detection in text is an important task to recognize those of documents that try to make people believe in something false. Current studies in this field assume that the entire document contains cues to identify deception; however, as demonstrated in this work, some irrelevant ideas in text could affect the performance of the classification. Therefore, this research proposes an approach for deception detection in text that identifies, in the first instance, key ideas in a document based on a topic modeling algorithm and a proposed automatic extractive text summarization method, to produce a synthesized document that avoids secondary ideas. The experimental results of this study indicate that the proposed method outperform previous methods with standard collections.

Keywords. Clustering algorithms, topic modeling, genetic algorithms, deep learning.

1 Introduction

In recent years, a large amount of data on the Internet is causing a lack of control over what information is true. In addition, people could easily be deceived since humans are not particulary effective in detecting deception [15, 16] (especially in texts); for example, a politician could be discredit by spreading misleading texts online that could cause low popularity in campaign [2].

Deceptive texts might not only stand for negative opinions, but also might stand for positive ones; for instance, a seller could increase sales of a product if the reviews about it are mostly positives, even if the product is not good.

This fact allows to make profitable the business of fake reviews, on the one hand, by spreading positive reviews to improve sales and, on the other hand, by spreading negative reviews to discredit a competitive product or service.

According to the above, humans could have two main problems in the task of detecting deception: the large amount of data that hinders the task of analyzing each opinion and the poor human ability to detect deception. Consequently, because the memory and data processing capabilities of machines exceed those of humans, deception detection in text has been tackled by combining artificial intelligence subfields such as: pattern recognition, machine learning and natural language processing (NLP).

Several works have applied supervised approaches to automatically detect deceptive texts [9, 8, 19] by extracting features, which construct numeric vectors and then pass them through a classification algorithm.

Commonly, each complete document is processed using different feature generation methods to extract information at different language levels, for example: lexical, syntactic and semantic. However, taking the entire document into account could lead a biased representation of it because, as this research has shown, not all the text content is useful in providing signs of deception.

Therefore, this study proposes an approach that attempts to keep only the main ideas in the documents, since the inference is that the secondary ideas are not focused on producing deceptive text.

For this purpose, on the one hand, this work proposal implements an algorithm that identifies the main ideas in text through the following general steps. First, documents are processed at sentence level, as this unit is considered to contain a full meaning of an author's idea. Next, an evolutionary algorithm organizes each document into sentence groups based on a partitional clustering and selects the best groups configuration according to an aptitude function. Finally, a topic modeling algorithm is applied to select the most representative sentences (that will take part of the new document) of each cluster formed.

This general process allows the proposed method to classify only the key sentences of each text and determine if a text is deceptive or truthful. The research results show that the cues to deception are mainly found in the key ideas of a text.

The rest of the paper is organized as follows. In Section 2, some approaches for the deception detection task are discussed. The proposed approach, for the detection of deception, is described in Section 3, and the experimental results are presented in Section 4. Finally, the conclusions are drawn in Section 5.

2 Related Work

Process of lying is very complex, both for humans and for machines, since it implies a cognitive process; ordinary people, for example, have shown about 50% accuracy in detecting lies, while some deception detection experts reach about 89%, but only in specific contexts [21].

This limited human ability to detect deception has motivated the development of computational models to automatically detect a lie. These models can be generated using different sources, such as videos, speeches, facial expressions; however, the difficulty increases when only texts are the source of information due to the lack of features related to the liar, that is, the words in text are the only information available.

In addition, unlike other classification problems such as polarity detection, the deception detection task lacks explicit cues or words that define a deceptive or truthful text. For example, in a sentiment analysis task, finding the word "smile" in a text indicates that its polarity could be positive; on the other hand, there is no explicit set of words that give clues to deceptive text (at least not in a general context).

The main purpose of a deceptive text is to change the people's opinion about someone or something, making them believe something that a writer does not believe. So detecting deception in text is a task of great importance for many applications, for example, detecting if a review of a product or service is written to deceive people, or if a fake news published on social networks is trying of discrediting a candidate in a political campaign.

Several works have been dedicated to the problem of detecting deception in text and in each one of them different methods have been implemented to generate features. Each method might be successful to some degree depending on the kind of information that is extracted from text. For instance, in various works, that reports competitive results, the Linguistic Inquiry and Word Count (LIWC) tool has been applied for extracting psycholinguistic-based features [23, 22], [15]; other common methods used to extract lexical information are n-grams [4] and character n-grams [5]. In addition, syntactic information has also been shown to provide information relevant to deception detection [3, 24].

NLP researchers have proposed several models to increase the accuracy of deception detection. For example, in the work of Pérez-Rosas et al. [17], different features of texts are obtained in different domains to detect deception in news. The authors obtain features based on different language levels, such as lexical, syntactic and semantic, trying to discover which of them increase the classification performance. Further experiments also combine domains, for example, using texts about technology to train the model and using texts about sports to test it.

On the other hand, some studies have focused on finding a set of universal cues to identify deceptive text; however, the signs of deception appear to change depending on the domain of documents. For example, in some texts on controversial topics some self-references are truthful text indicators [13], while in some interview texts they are neutral [11], or even, in another domain, they are distinctive of deceptive texts [15]. Studies looking for universal features of deception to produce domain-independent approaches generally fail because the evidence so far shows that there is no such set. As a result, commonly proposed approaches create models that evaluate text in specific domains.

It can be seen that several works, which tackle the detection of deception in the texts, apply three general steps: 1) text preprocessing, 2) selection of methods for the generation of features that generates a vector space model (VSM), and 3) classification of the VSM's patterns to discover which texts are deceptive or truthful.

The problem with common approaches is that they consider tagged full text to train models; however, documents may contain irrelevant information that could affect the performance of the classification, since our inference is that liars focus on the key ideas in text to deceive the reader and use secondary ideas only as a complement.

3 Proposed Approach for Deception Detection

This study proposes an approach to deception detection (see Figure 1) that attempts to capture the key ideas in a document to avoid secondary information that could affect the search for cues to deception. To achieve this, a clustering representation is chosen to identify key ideas and form a new collection of synthesized documents which should avoid secondary ideas as far as possible. The synthesized documents are then classified to identify truthful and deceptive texts. This process, as can be seen in Section 4, improves results in the deception detection task by avoiding the conservation of unimportant features. Below is a detailed description of each step of the proposed approach.

The first step in this proposal is to remove the secondary ideas from the original texts of the collections. Then, each document is processed by splitting each one into sentences, because sentence is considered the minimum unit with a complete meaning.

Sentences in the splitted documents are then organized by a partitional clustering representation (detailed in Section 3.2). That is, grouping objects according to their proximity by mapping the problem to a vector space model, where sentences stand for the objects in this model. Therefore, each sentence in the source document is mapped to a numerical vector using a TF–IDF scheme. As a result, this step allows generating groups of sentences according to their relevance in the document and obtaining an overview of it.

The main problem of the partitional clustering algorithms, such as k-means or k-medoids, is to determine the number of groups to be formed, since testing each grouping becomes a combinatorial problem. Therefore, in this work implements a genetic algorithm (GA) to find the best clustering (detailed in Section 3.3) using the silhouette index as an fitness function.

When the GA finds the best clustering configuration, the LDA algorithm is responsible for selecting the key ideas of each cluster (see Section 3.4). This is possible because LDA can obtain word and topic distribution of a source document. This information is used to select the most important sentences, as the most likely topics and words are the main components of the document.

As a result, the new synthesized documents, made up of the sentences selected according to the LDA algorithm, contain only the potential key ideas. Finally, these documents are mapped using different feature generation methods, such as: OHE, D2V, LDA and TF–IDF, and they are then classified as truthful or deceptive using a back-propagation neural network.

3.1 Feature Extraction Methods

As stated above, the first step of this study requires mapping texts to numeric vectors. Thus, there are various feature extraction methods that obtain information at different language levels.

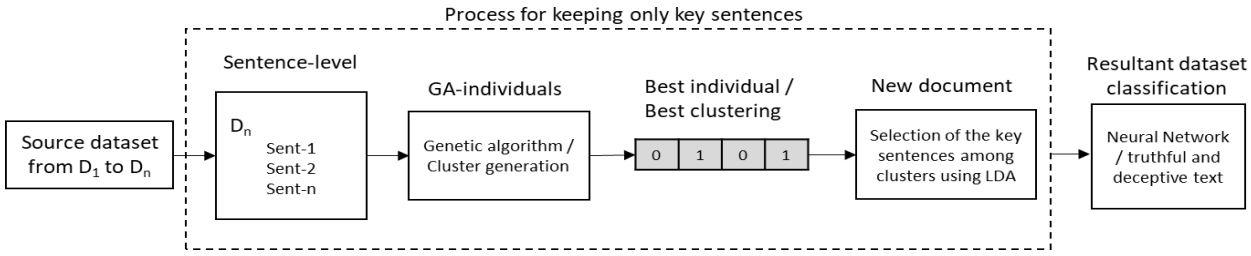

Fig. 1. Detailed steps for detecting deception in documents

In this study, various methods have been used to measure the importance of features at lexical and semantic level. These methods are briefly described below.

To build one-hot vectors, we simply **obtain an OHE representation**, in which a list of all the words W_1, W_2, ..., W_n in the dataset is made. Then, we analyze each document to determine whether W_n exists in the current text. If so, feature n (F_n) is set to 1 or to 0 otherwise.

TF–IDF reflects the importance of a word in a document and, in turn, in a dataset. This feature may be useful in the information-retrieval task of searching for similar documents; however, in the proposed framework, the relevance of the words in the document can be useful for determining whether the sentence is relevant.

Latent Dirichlet Allocation (LDA) [1] is a probabilistic generative model for discrete data collections such as text collections. It represents documents as a mixture of different topics, where each topic consists of a set of words that have a link between them. Words, in turn, are chosen on the basis of probability. The process of selecting topics and words is repeated to generate a document or a set of documents. As a result, each generated document is based on different topics.

Doc2Vec [10] is an unsupervised algorithm that generates fixed-length numeric vectors by processing a document; it was inspired by Word2Vec [14]. The difference between the two algorithms is that the former builds a fixed-length vector representation of a variable-length text, whereas the latter builds a vector for each word in the text.

In contrast to the bag-of-words approach, Doc2Vec can consider the ordering and semantics of the words. In addition, this algorithm avoids sparsity and high dimensionality, in contrast to OHE.

3.2 Clustering Representation

The clustering representation was implemented with the purpose of obtaining a general structure of the document. This step, detailed below, allowed generating groups with sentences that are similar to each other based on the TF–IDF representation.

Following the human behavior wherein people create summaries by choosing the most important sentences in a document, we attempted to capture the key sentences, in the source document, by considering that they are surrounded by other similar ideas, just as a centroid is surrounded by attracted patterns. Therefore, this clustering representation involves two aspects: (1) the generation of the word space model (WSM) and (2) the selection of proximity measures.

1. The TF-IDF scheme was selected to the clustering representation for mapping each sentence to a numeric vector. This representation with the clustering algorithm are combined to organize the documents according to the relevance of word in the sentences.

2. To obtain proximity between objects two measures were combined: Euclidean and cosine, since these combined measures proved to outperform results in clustering problems [6], and they empirically proven to obtain better results in this study.

Because the cosine measure represents similarity and the Euclidean measure represents the distance between objects, we turn the Euclidean distance measure into a similarity measure by using the following adequacy: $modifyEuclidean = \frac{1}{Euclidean+1}$; *similarityEuclidean* obtains values in the range $(0, 1]$, where 1 means that the objects are the same and values close to 0 means that the objects are highly dissimilar. The cosine measure was modified by simply adding a unit to obtain only positive values: $modifiedCosine = cosine + 1$ in the range $[1, 2]$. Finally, the similarity between two objects is given by $modifiedEuclidean * modifyCosine$.

To generate the groups of similar objects, we use the basics of partitional clustering algorithms, i.e., assigning each object (sentence) to the closer centroid. Therefore, if there are n-centroids, then n-groups should be created.

However, determination of the number of groups to be generated to find the best solution becomes a combinatorial problem; that is, partitional algorithms may organize a set of sentences into K clusters. Therefore, given a set of sentences $x_i \in \mathcal{R}^d, i = 1, ..., N$, it is possible to enumerate all possibilities to determine the best solution. However, this brute-force approach is infeasible because it becomes a problem that is extremely expensive computationally, as suggested in [25].

3.3 Finding the Optimal Number of Clusters

A GA representation is proposed to ensure the covering of sentences in the synthesized document, being this document that which contain only potential key sentences. After this process is applied, each sentence in a document will belong to some cluster based on its word relevance. This process is detailed below.

Individuals are configured as follows: the number of genes in each individual is equal to the number of sentences in the document to be synthesized. In turn, the individual codification is binary, and, thus, each gene may be set to 1 or 0, where 1 means that the sentence is a centroid and 0 means otherwise.

The initial population is generated by assigning a random value to each gene. That is, given the individual $P = \{g_1, g_2, ..., g_n\}$, where n is the total number of sentences in the document, each $g_i = Random[0, 1]$.

The activated genes ($g_n = 1$) act as attractors to the closer sentences. Thus, an individual formed of n-centroids would form n-clusters. Finally, the centroids of the groups are considered the main topics of the document, whereas the sentences attracted by the centroid are considered ideas that are close to the main topic.

The selection process over the population was addressed by selecting the Silhouette index as a fitness function due to its correlation with the selection of main sentences [7]. In addition, according to [12, 20], dedicated measuring the efficiency of the validation indexes, the Silhouette index performed better than other ones. This index considers a range of real values between 1 and -1, where the values closest to 1 represent a better clustering; therefore, those fitness values closest to 1 represent the best individuals of the population.

The principle of evolution suggests that the recombination of good solutions tends to provide outperforming solutions. However, their diversity is also important. Thus, the parents' selection process is performed by using a roulette operator that provides a high likelihood that the best solutions are selected; however, it does not completely discriminate against the bad solutions. In this study, other methods were also applied, such as random, rank and tournament selection; however, they proved to get lower performance.

To generate the offspring, we applied a technique of one-point crossover, and a standard mutation wherein the operator inverts the binary value of a selected gene.

3.4 Selection of Main Sentences using LDA

Sentence selection can be performed by selecting the centroids of the formed clusters because the inference is that the centroid sentences are the main ideas of the document, whereas the remaining sentences are secondary ideas; however, this assumption is not quite true.

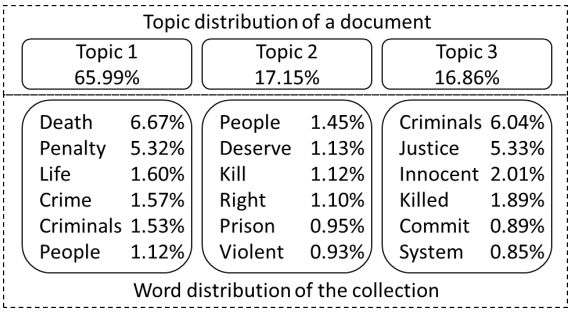

Fig. 2. Example of distribution of words and themes of a document on the death penalty

For example, if the clustering is built using TF–IDF as the mapping method, then the best configuration will guarantee that the centroids represent the sentences that are dissimilar among them, with respect to the word relevance in the document. This representation could provide centroid sentences with relevant words, but also the opposite, i.e., sentences with few relevant words, because the centroids should meet the separation property. Given this premise, the selection of key sentences could be incorrect.

Therefore, in this study, we propose the creation of a vector space model by adding the semantic information obtained using an LDA model. An example of the sentence distribution obtained with the LDA model is shown in the Figure 2; as can be seen in the figure, LDA reports distribution at the word and sentence levels. That is, given a sentence, topic 1 has a 65.99% probability of being part of it, and, in turn, the word "Death" has a probability of 6.67% of being selected into Topic 1.

The LDA model is configured to generate three topics in the experiments of this study because this configuration has been empirically proven to yield the best results. This model is applied in two steps of the proposed approach: (1) in the process of mapping text to numeric vectors, and (2) in the selection of the key sentences. Both procedures are discussed in detail below.

1. Sentence distribution, represented by the topic distribution (see Figure 2), is used to generate numeric vectors of the each document in the collection after the elimination of secondary ideas, i.e., for the classification step. This process allows to known how informative is the topic distribution in the task of detection of deception.

2. The clustering of sentences does not yet provide information about the key sentences in the document. Therefore, with the aim of identifying these sentences, the 10 most representative words of 3 topics are selected as keywords. Thus, the selection of key sentences for each cluster was conducted as follows. Given each probability pT_i associated with each topic T_i, and each probability p_i associated with each word w_i in the keywords, each word w_s in the candidate sentence was compared with w_i. If w_i was equal to w_s, $p_i * pT_i$ was accumulated in $pTotal$. The sentence that reached the maximum value of $pTotal$ was selected as a main sentence.

   If there is no $w_i = w_s$ in any sentence of the cluster, the centroid is selected to be part of the synthesized document. This guarantee that at least one of the sentence of each cluster will take part of the new document.

   The key sentence selection process ends when a candidate sentence was extracted from each cluster. Therefore, the clustering scheme allows to cover the topics of the document, whereas the LDA's word distribution allows to select the most promising key ideas in each cluster.

3.5 Classification of the Synthesized Documents

Deep learning has proven to be more efficient, on the one hand, because the computational power has considerable increased and, on the other hand, because these algorithms can generate more complex functions than other classifiers such as naïve Bayes (NB), support vector machine (SVM) or random forest.

This research implements an artificial neural network (ANN) with backpropagation to carry out the classification process. ANN generate nonlinear functions that through several hidden layers become more robust.

This process provides more efficient models at the expense of a computational cost.

The specific parameters used in this study were the following: L-BFGS was selected as activation function, the number of hidden layers was set to 1000, and the activation function used was ReLU.

Once the new documents that avoid secondary ideas have been generated, the next step was to classify them into deceptive and truthful texts. Two common methods for mapping texts to numeric vectors were used to obtain the word space model (WSM) representation: TF–IDF and OHE. In addition, we proposed building LDA and Doc2Vec models to add semantic information to the feature vectors. Thus, the next step were to measure distance between vectors. Therefore, four methods for feature generation were applied to the reduced texts obtaining numeric vectors to feed the classifier. In this step, the documents are not divided into sentences, but are completely processed using each mapping method.

The five-fold cross validation technique was used to validate the performance of the classification. This evaluation also allows comparing the results of this work with those of others who classified the same corpora.

The SVM and NB classifiers were also applied in this study and showed an error of 11.4% and 12.47% in deception detection. Instead, the ANN showed a 9% error. Therefore, the latter achieved the best results.

3.6 Datasets

Several datasets were collected in different domains to obtain reliable results in the experiments. Each one deals with a certain topic and was labeled following certain strategies described below.

DeRev dataset (DEception in REViews) [4] is a corpus composed of deceptive and truthful opinions obtained from the Amazon website. This corpus includes opinions about books. This gold standard corpus contains 236 texts, of which 118 are truthful and 118 are deceptive.

OpSpam dataset (Opinion Spam) [16] is a corpus composed of fake and genuine opinions about different hotels. It was also collected from the Amazon website. This corpus contains 800 texts, of which 400 are truthful and 400 are deceptive.

Opinions dataset [18] is a corpus composed of opinions on three controversial topics: abortion, best friend and death penalty. It consists of 100 deceptive texts and 100 truthful texts.

4 Results

As mentioned above, the first step in this approach is to remove secondary ideas for analyzing and generating features only considering the main information in documents. This process involves a clustering representation that considers sentences as objects to find those that could be a centroid. These centroids are the sentences that represent the partitions generated by means of a GA algorithm. As a result, each sentence should belong to a certain group after applying the clustering process. Finally, the word distribution of the document provided by LDA helps to select, from the groups or partitions, the sentences that will form the new text.

Table 1 shows the statistics of each corpus, analyzed in this study, before and after being processed and synthesized. The displayed values are tokens and types, that is, the total number of words and the number of words without repetition in the corpus, respectively; average tokens per document is also displayed. As can be seen from the table, the number of tokens decreases after the source documents are processed and thus the percentage of words that were removed from each corpus is displayed. This percentage varies depending on the LDA selection process detailed in Section 3.4.

The next general step is classify the documents from selected datasets into deceptive or truthful texts. The corpora analyzed in this study address two general domains: controversial topics (abortion, best friend and death penalty) and reviews of services or products (OpSpam, DeRev). All corpora were classified using the same configuration detailed above. Table 2 shows the classification results of controversial topics. This table shows the methods to generate features and the values of precision (P), recall (R), and

Table 1. Corpora's types and tokens before and after deleting secondary ideas. ATD=Average tokens per document

Dataset	# docs	Source documents			Reducted documents			% deleted words
		Tokens	Types	ATD	Tokens	Types	ATD	
Abortion	200	18,098	1,939	90	11,490	1,434	57	27%
Best friend	200	11,717	1,718	59	10,055	1,375	50	14%
Death penalty	200	15,615	2,034	78	11,487	1,443	57	27%
DeRev	236	29,990	5,162	127	16,696	2,989	70	44%
OpSpam	800	96,793	6,469	121	59,418	4,006	74	39%

Table 2. Classification results, in controvertial topics, obtained before and after eliminating secondary ideas. Values of precision (P), recall (R), and F-measure (F) are shown

Abortion						
Method	Keeping everything			Keeping only key ideas		
	P	R	F	P	R	F
TF–IDF	78.56	74.0	76.93	80.37	78.0	78.18
LDA	50.85	56.0	49.88	50.79	51.0	50.94
D2V	84.08	74.0	76.93	67.44	62.0	64.9
OHE	84.08	85.0	83.95	91.25	92.0	91.46
Best Friend						
Method	Keeping everything			Keeping only key ideas		
	P	R	F	P	R	F
TF–IDF	82.85	83.0	82.99	83.68	85.0	83.94
LDA	51.73	54.0	50.95	53.54	53.0	52.38
D2V	74.2	77.0	74.88	78.8	80.0	78.95
OHE	96.39	71.0	83.44	91.28	94.0	92.5
Death Penalty						
Method	Keeping everything			Keeping only key ideas		
	P	R	F	P	R	F
TF–IDF	65.68	66.0	65.41	64.45	67.0	64.96
LDA	50.33	49.0	49.14	52.96	52.0	52.0
D2V	63.13	64.0	62.91	55.88	57.0	55.86
OHE	99.0	72.0	85.64	92.5	84.0	88.41

F-measure (F), before and after removing the secondary ideas. As can be seen, the reduction of data in the documents affects the TF–IDF, LDA, and D2V methods in some cases; for example, the F-measure decreases on the subject of abortion after keeping only the key sentences when applying the TF–IDF method. However, the F-measure increases in all cases when applying the OHE method. Similar behavior is shown in the Table 3, where reviews are classified.

For the proposed framework, the methods that extract lexical information from documents (TF–IDF and OHE) show the best results, while those that provides semantic information (LDA and D2V) are greatly affected by the sentence elimination process.

The method that provides the best performance for all datasets is OHE, as can be seen in the Table 2 and the Table 3; these tables show the results of the classification on controversial topics and reviews, respectively.

Finally, Table 4 shows the comparison between the performance of the proposed approach and the current studies on deception detection, which analyze the same datasets.

Table 3. Classification results, in reviews, obtained before and after eliminating secondary ideas. Values of precision (P), recall (R), and F-measure (F) are shown

DeRev						
Method	Keeping everything			Keeping only key ideas		
	P	R	F	P	R	F
TF–IDF	86.81	87.43	86.88	89.18	97.43	92.77
LDA	57.37	54.89	53.1	56.19	58.73	55.86
D2V	88.27	91.59	89.45	82.61	84.78	83.49
OHE	90.12	99.17	94.07	92.44	100.0	95.72
OpSpam						
Method	Keeping everything			Keeping only key ideas		
	P	R	F	P	R	F
TF–IDF	86.09	86.0	86.0	82.73	82.75	82.57
LDA	56.82	68.5	57.31	49.43	51.0	49.56
D2V	81.23	86.0	86.0	79.6	76.5	78.33
OHE	90.87	84.0	87.72	92.11	92.0	91.99

Table 4. Comparison of our proposal with other approaches on the same corpora. Stat.Sig. = Statistical Significance

Corpus	Approach	F-measure	Z-score	Stat.Sig.
Abortion	Our proposal	**91.4**	3.21	yes
	Pérez-Rosas and Mihalcea [18]	80.3		
	Feng et al. [3]	77.0		
Best friend	Our proposal	**92.5**	2.38	yes
	Pérez-Rosas and Mihalcea [18]	75.9		
	Feng et al. [3]	85.0		
Death penalty	Our proposal	**88.4**	4.14	yes
	Pérez-Rosas and Mihalcea [18]	77.2		
	Feng et al. [3]	71.5		
OpSpam	Our proposal	**91.9**	0.50	no
	Ott et al. [16]	89.8		
	Feng et al. [3]	91.2		
	Fusilier et al. [5]	90.2		
DeRev	Our proposal	**95.7**	6.31	yes
	Fornaciari and Poesio [4]	76.3		

Furthermore, the statistical significance between the most competitive approach and the best result of this work is provided by calculating the z-score.

To determine statistical significance (SS), a 95% confidence level was selected. Because the critical value of this confidence interval is in a range of -1.96 to +1.96, each value outside this range means that there is a SS.

Therefore, as can be seen in Table 4, the proposal achieves SS in four of the five datasets evaluated.

5 Conclusions

The proposal of this study is an approach to the detection of deception.

Our inference was that the cues to deception stand out in the main ideas of a text. Because the basic unit with a full meaning of a writer's idea is the sentence, the detection of key ideas was addressed by trying to find the most important sentences in each document.

Different methods were proposed for the generation of features to extract information at the lexical and semantic level. The research results showed that the lexical level was the most appropriate to detect deception for this proposed framework.

In several studies, lexical information provide the most relevant features for detecting deception; however, other language levels have been shown to complement those features and increase the classification performance. Thus, method assembly will be considered future work.

In both kind of domains: controversial topics and reviews, (see Table 2 and Table 3) the results for the OHE method showed an increasing f-score when secondary ideas were discarded. This result shows the correlation between the key ideas and their impact as a source of information for the detection of deception.

Acknowledgments

The authors wish to thank the Mexican Government (Cátedras CONACYT, SNI, Universidad Autónoma del Estado de México) for its support.

References

1. **Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003).** Latent dirichlet allocation. *Journal of machine Learning research*, Vol. 3, No. Jan, pp. 993–1022.
2. **Clementson, D. E. (2018).** Truth bias and partisan bias in political deception detection. *Journal of Language and Social Psychology*, Vol. 37, No. 4, pp. 407–430.
3. **Feng, S., Banerjee, R., & Choi, Y. (2012).** Syntactic stylometry for deception detection. *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, Association for Computational Linguistics, pp. 171–175.
4. **Fornaciari, T. & Poesio, M. (2014).** Identifying fake amazon reviews as learning from crowds. Association for Computational Linguistics, pp. .
5. **Fusilier, D. H., Montes-y Gómez, M., Rosso, P., & Cabrera, R. G. (2015).** Detection of opinion spam with character n-grams. *International Conference on Intelligent Text Processing and Computational Linguistics*, Springer, pp. 285–294.
6. **Gu, X., Angelov, P. P., Kangin, D., & Principe, J. C. (2017).** A new type of distance metric and its use for clustering. *Evolving Systems*, Vol. 8, No. 3, pp. 167–177.
7. **Hernández Castañeda, N., García Hernández, R. A., Ledeneva, Y., & Hernández Castañeda, Á. (2020).** Evolutionary automatic text summarization using cluster validation indexes. *Computación y Sistemas*, Vol. 24, No. 2.
8. **Hobson, J. L., Mayew, W. J., Peecher, M. E., & Venkatachalam, M. (2017).** Improving experienced auditors' detection of deception in ceo narratives. *Journal of Accounting Research*, Vol. 55, No. 5, pp. 1137–1166.
9. **Krishnamurthy, G., Majumder, N., Poria, S., & Cambria, E. (2018).** A deep learning approach for multimodal deception detection. *arXiv preprint arXiv:1803.00344*.
10. **Le, Q. & Mikolov, T. (2014).** Distributed representations of sentences and documents. *International conference on machine learning*, pp. 1188–1196.
11. **Levitan, S. I., Maredia, A., & Hirschberg, J. (2018).** Linguistic cues to deception and perceived deception in interview dialogues. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1941–1950.
12. **Liu, Y., Li, Z., Xiong, H., Gao, X., & Wu, J. (2010).** Understanding of internal clustering validation measures. *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, IEEE, pp. 911–916.
13. **Mihalcea, R. & Strapparava, C. (2009).** The lie detector: Explorations in the automatic recognition of deceptive language. *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, Association for Computational Linguistics, pp. 309–312.
14. **Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013).** Distributed representations of words and phrases and their

compositionality. *Advances in neural information processing systems*, pp. 3111–3119.

15. **Newman, M. L., Pennebaker, J. W., Berry, D. S., & Richards, J. M. (2003).** Lying words: Predicting deception from linguistic styles. *Personality and social psychology bulletin*, Vol. 29, No. 5, pp. 665–675.
16. **Ott, M., Choi, Y., Cardie, C., & Hancock, J. T. (2011).** Finding deceptive opinion spam by any stretch of the imagination. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, Association for Computational Linguistics, pp. 309–319.
17. **Pérez-Rosas, V., Kleinberg, B., Lefevre, A., & Mihalcea, R. (2017).** Automatic detection of fake news. *arXiv preprint arXiv:1708.07104*.
18. **Pérez-Rosas, V. & Mihalcea, R. (2014).** Cross-cultural deception detection. *ACL (2)*, pp. 440–445.
19. **Rangel, F., Rosso, P., Charfi, A., Zaghouani, W., Ghanem, B., & Snchez-Junquera, J. (2019).** Overview of the track on author profiling and deception detection in arabic. *Working Notes of the Forum for Information Retrieval Evaluation (FIRE 2019). CEUR Workshop Proceedings. In: CEUR-WS. org, Kolkata, India*, pp. .
20. **Rendón, E., Abundez, I., Arizmendi, A., & Quiroz, E. M. (2011).** Internal versus external cluster validation indexes. *International Journal of computers and communications*, Vol. 5, No. 1, pp. 27–34.
21. **Rosso, P. & Cagnina, L. C. (2017).** Deception detection and opinion spam. In *A Practical Guide to Sentiment Analysis*. Springer, pp. 155–171.
22. **Schelleman-Offermans, K. & Merckelbach, H. (2010).** Fantasy proneness as a confounder of verbal lie detection tools. *Journal of Investigative Psychology and Offender Profiling*, Vol. 7, No. 3, pp. 247–260.
23. **Toma, C. L. & Hancock, J. T. (2012).** What lies beneath: The linguistic traces of deception in online dating profiles. *Journal of Communication*, Vol. 62, No. 1, pp. 78–97.
24. **Xu, Q. & Zhao, H. (2012).** Using deep linguistic features for finding deceptive opinion spam. *COLING (Posters)*, pp. 1341–1350.
25. **Xu, R. & Wunsch, D. (2008).** *Clustering*, volume 10. John Wiley & Sons.

Article received on 16/06/2020; accepted on 21/07/2020.
Corresponding authors are René Arnulfo García Hernández and Yulia Ledeneva.