Towards automated fact-checking: An exploratory study on the detection of checkable statements in Spanish

Joaquín Saralegui
Facultad de Ciencias Exactas, UNICEN
Email: jsaralegui@alumnos.exa.unicen.edu.ar

Antonela Tommasel
ISISTAN, CONICET-UNICEN
Email: antonela.tommasel@isistan.unicen.edu.ar

***Abstract*—Nowadays, digital media allows important facts to be turned and twisted both knowingly and unknowingly, leading to the spread of misinformation to a continuously growing audience with little or no consequences. Thus, it has become crucial to verify (or fact-check) the authenticity of the shared information. One of the critical tasks in such verification process is determining which statements could be fact-checked. The automatization of this task would reduce human bias in the selection of checkable statements and save valuable time, thus helping to increase the coverage of the verification, and, potentially, their effectiveness. Although detecting checkable statements has received attention in the literature, most techniques have focused only on English. In this context, this study evaluates the performance of different approaches for detecting checkable statements in Spanish. Experimental evaluation showed promising results, achieving similar performance to the state-of-the-art techniques for the English language.**

***Index Terms*—Fact-checking, claim detection, Natural Language Processing, Text classification**

I. Introduction

For a few years now, society has been dealing with an unprecedented volume of fake news. Conspiracy theories, fake news, rumours, and hoaxes find in digital and social media a fast and effective way to easily and quickly reach a continuously growing audience [22, 23]. Moreover, important facts can be turned and twisted both knowingly and unknowingly leading to the spread of misinformation with little or nonexistent consequences [21], thus raising concerns regarding the manipulation of public opinions [6, 26]. When the actors spreading incorrect and twisted facts are the official government agencies and politicians, concerns are also expressed in terms of growing distrust in politicians and institutions. In the medium to long term, distrust can foster political apathy and low political participation that can directly impact democracies [16, 20, 29]. In this context, checking the factuality and authenticity of the shared information becomes crucial[1].

In the last decades, to aid in the fact-checking efforts, several non-profit organizations were created, being PolitiFact[2], FactCheck[3] and FullFact[4] among the three most popular initiatives. In Latin America, Chequeado[5], the first fact-checking organization, was created in Argentina in 2010. These organizations aim to provide users, and more generally, citizens, with the necessary mechanisms to evaluate the veracity of the statements made by public figures (e.g., politicians), the media or experts [25], thus empowering users to make informed decisions. Fact-checking has proved to be a highly demanding manual task, resulting in that only a reduced number of facts can be checked. Then, with the large volumes of information that are daily shared on the news, social media, and government organizations, and the short time for double-checking information, it is essential to automate the fact-checking process [17]. In addition to claim verification, one of the key tasks in such verification process is determining which statements can be fact-checked [3]. Automating this task would reduce human bias in selecting worthy phrases and save valuable time, allowing organizations to dedicate scarce resources to the most relevant tasks, thus increasing the coverage of verification and, potentially, their effectiveness.

Even though detecting fact-checkable statements has received attention in the literature, most techniques have focused only on the English language. In this context, in this study, we evaluate the performance of different approaches for identifying checkable statements in the Spanish language. Experimental evaluation performed over a Spanish data collection[6] showed promising results, achieving similar performance to the techniques available for the English language. In addition, the resulting technique was successfully integrated into the Chequeado ecosystem with satisfactory results.

The remainder of this paper is organized as follows. Section II presents related works. Section III describes the study methodology, including the labelled data collection, the extracted features and the classification task. Section IV presents the obtained results. Finally, Section V presents the conclusions of the study and future perspectives on the task.

This study was carried out while the first author was interim at Chequeado.

[1]Elisabeth, J. (2014). Who are you calling a fact checker? https://www.americanpressinstitute.org/fact-checking-project/fact-checker-definition/.

[2]https://www.politifact.com/

[3]https://www.factcheck.org/

[4]https://fullfact.org/

[5]https://chequeado.com/

[6]The collection focused on the Rioplatense Spanish variety, which is mostly spoken in Argentina and Uruguay.

978-1-6654-5674-6/22/$31.00

II. Related Works

The process of detecting checkable statements is not simple, as checking each phrase might require considerable time and resources, which are generally scarce in fact-checking organizations. Fact-checkers choose which statement to check by considering both the effort of checking the statement and the consequences of not checking it, including the damage it can cause in terms of health risks, worsening emergencies that are already dangerous (e.g., responses to natural disasters) or even compromising democratic processes. In addition, although a statement could be factually checked, the information to check it might not be directly available, and consulting with specialists or alternative sources can sometimes take several days and still yield unsuccessful results [12].

Given the unprecedented volume of information shared on multiple platforms, manually monitoring all content, news and public figures and their statements is not feasible. Then, keeping a record of all the checkable statements requires an enormous human effort and resources, and even in such cases, there is no certainty that relevant statements are not missed. On the other hand, if fact-checkers wait for statements to go viral before verifying them, their damage has likely already been done. Fact checks are estimated to go viral six times slower than fake news [27]. Therefore, the rapid detection of checkable statements as early as possible provides an advantage against the spread of false information.

Closely related to this study are ClaimBuster [8], ClaimRank [9] and the Full Fact detection module [11]. ClaimBuster [8] focused on differentiating between non-factual sentences (such as opinions, predictions or questions), unimportant factual sentences (phrases that, despite expressing a fact, are irrelevant, for example, "it rained yesterday"), and checkable factual sentences. These categories have been criticized as the category of a phrase might depend on the context, and a currently irrelevant phrase could be relevant in the future. The approach was based on over 6k features, including sentiment, number of words, TF-IDF weighting of terms, frequency of POS tags, and frequency of named entities. Evaluation was based on US presidential debates between 1998 and 2012. Precision results ranged between 0.69 and 0.813, while recall ranged between 0.745 and 0.827. The best results were obtained when training an SVM classifier with POS frequencies and TF-IDF weighting of terms.

ClaimRank [9] is an extension of [7] and proposed to rank phrases according to their "check-worthiness", i.e., their relevance for the fact-checking task. The authors extended ClaimBuster by including lexical features (e.g., words classified as biases, sentiment, subjectivity) and structural features, such as the position of a sentence within a discourse. Ranking was based on a deep learning model. Evaluation was also based on phrases extracted from US presidential debates. There was no indication regarding whether the data was balanced. Precision when considering the top-k elements, where k is the number of checkable statements ranged between 0.33 and 0.38. The model was also evaluated for Arabic data, which was obtained by translating the English dataset. According to the authors, the lower results obtained for the Arabic data could be caused by the lack of adequate NLP libraries for such language. ClaimRank and ClaimBuster aimed at ranking statements according to their worth, a concept that depends on the context and thus might change over time. In this sense, the output of both approaches, by definition, would be outdated as it depends on the originally (and perhaps static) labelled data.

Unlike ClaimBuster and ClaimRank, Full Fact [11] tried to define context-independent categories for classifying statements, including personal experiences, quantities, comparisons, laws, correlations or causations, predictions and non-verifiable phrases. In this case, phrases were represented based on POS and named entity frequencies combined with InferSent [5] word embeddings. The best results were obtained when training a Logistic Regressor classifier only considering the embedding representation, achieving a precision and recall of 0.88 and 0.8, respectively. The authors replicated the best performing alternatives of ClaimBuster and ClaimRank, showing that their approach outperformed both.

Finally, Chequeado conducted two studies to classify Spanish phrases. First, they conducted a study with over $3,500$ volunteers [14] who had to identify checkable and non-checkable statements from a fictional political speech containing 8 phrases (4 checkable and 4 non-checkable). The goal of the study was to determine how the demography of participants could affect their ability to detect the checkable statements. Results showed that volunteers were able to identify checkable phrases with a precision of 0.69. The highest scores were achieved by young and university-educated male volunteers. Results also showed that it was easier for volunteers to correctly identify stand-alone statements than statements that were part of a full text. Second, Chequeado proposed a Naïve Bayes classification model based on POS tag frequency, lemmatization and 3-grams to classify Spanish phrases[7]. Unfortunately, the implementation details and documentation were insufficient to replicate the solution. The model was trained on a small dataset, and performance results have not been made public[8].

III. Study description

This study aims to identify checkable statements. To this end, we defined a binary classification task where each statement could either be checkable or not. We extend the work of Chequeado by providing an open-source model that improves the feature extraction process and evaluates a large number of classifiers. Figure 1 presents the schematic representation of the methodological pipeline[9]. First, given the scarce availability of resources in Spanish, we labelled statements collected from news media outlets and presidential speeches,

[7]A public preliminary and outdated version of the model can be found at: https://github.com/chequeado/chequeabot/tree/master/claims_prediction

[8]Based on the data in the repository, precision and recall could have been around 0.85 and 0.72, respectively

[9]Data and implementation are available at: https://github.com/joacosaralegui

extracted several sets of features to represent each of the statements, trained different models, and, finally, evaluated their performance.

A. Data collection

The first step to build a data collection is to define what checkable statements are. According to Chequeado [14], a statement must meet several requirements to be checkable:

- It must be factual. It has to express a fact or data whose accuracy or veracity can be verified.
- It should be checked within a reasonable time. If checking a statement involves a prohibitive amount of time and resources, then, in practice, it cannot be checked.
- It must be verifiable based on existing open data.

Considering the type of fact-checking that Chequeado performs, three data sources were considered: presidential speeches, previous fact-checked statements and news shared in media outlets. Presidential speeches are representative of political speeches, which usually contain a high number of facts and references to statistics, promises and predictions. As a result, they provide a high number of checkable phrases, and also enough examples of the main cases for which it seeks to predict that something is non-checkable. Previous fact-checks refer to statements that have already been fact-checked. For example, "Milei: Global warming is a lie"[10]. This source allows obtaining a very high number of positive examples that also reflect the nature of the statements that are usually chosen to be verified. If the model can learn the main characteristics of these types of statements, it would be able to provide suggestions in the future that align with the usually checked statements. Finally, news from media outlets often cover diverse contexts and contain many examples of statements that cannot be checked. These would provide the negative instances for training the classifier to learn to filter the irrelevant information. In total, we retrieved 2300 statements from the presidential speeches at the opening of Congress sessions, approximately 1300 statements from the previous fact-checked statements collected between 2010 and 2021, and 1400 news media statements related to politics and the economy. Labelling was performed following Chequeado guidelines, involving the work of a professional fact-checker. After the labelling process, we obtained 4958 statements, with approximately 39% and 61% of fact-checkable or non-checkable statements.

B. Feature extraction

Once the data collection is built, it is necessary to define the features that will be used to describe the statements and then train the classification models. Previous works [8, 9, 11] provide a guideline regarding which features could be helpful. However, such features were proposed for classifying English texts. Then, it is necessary to evaluate their relevance to the Spanish language. The extracted features can be divided into two groups: traditional and semantic representations, which are summarised in Table I.

Traditional representation	Semantic representation
• Bag-of-Words representation. • Lemmas. • Named Entities. • Part-of-Speech (POS).	• Universal Sentence Encoder (USE).

TABLE I: Summary of extracted features

Traditional representations: This representation is mainly based on the lexical analysis of texts, i.e., the individual words. Texts are usually represented by the Bag-of-Words (BoW) model [19], in which each word or term is represented as an independent feature (similarly to a one-hot encoding). This representation disregards all grammar considerations, context, and word order but keeps the information about the frequency of each term. For the purpose of this study, a binary weighting scheme was considered, in which 1 and 0 indicated whether the term appeared in the statement. A simple tokenization process based on spaces and punctuation was applied for extracting words.

Although the experimental evaluation was performed on a closed set of statements, the goal of the analysis is to be used in a quasi-real-time setting. In such a setting, more complex weighted schemes, such as TF-IDF [19], might not be adequate, as statements would constantly arrive, which has two implications. Firstly, there is no fixed set of statements over which to compute the IDF. Second, if new statements are added to the collection, the statistics for computing the score for a feature would need to be periodically updated, which might be inefficient. Then, although some information related to the overall relevance of terms or features might be lost, in dynamic environments, it might be preferable to use more efficient weighted schemes, such as binary or term frequency. Note that, with each new statement, the statistics of the terms in such statement should be updated.

The traditional BoW representation was enriched by including the following features, also following a binary weighting scheme. In addition to including the unmodified words, we performed lemmatization to include the words lemmas. *Lemmatization* is one of the most common text pre-processing techniques. It leverages language's structure to reduce words to their base or canonical form by removing or replacing their suffix. For example, the lemma of `breaks` and `breaking` is `break`. The goal of lemmatization is to decrease the morphological variations of words, as multiple inflected word variations would be reduced to the same lemma, allowing to find more coincidences between the statements.

Named entity recognition is the task of identifying and categorizing critical information (i.e., the entities) in unstructured text. An entity is represented by a word or a sequence of words that consistently refers to the same thing, belonging to a specific category. For example, common entities are quantities, time expressions, names, and locations. For example, in the sentence "Joe Biden participated in an event in Washington DC" we can extract the entities "Joe Biden" (person) and

[10]In Spanish: "Milei: El calentamiento global es una mentira." https://chequeado.com/ultimas-noticias/milei-el-calentamiento-global-es-una-mentira/ More examples can be found in the Chequeado website

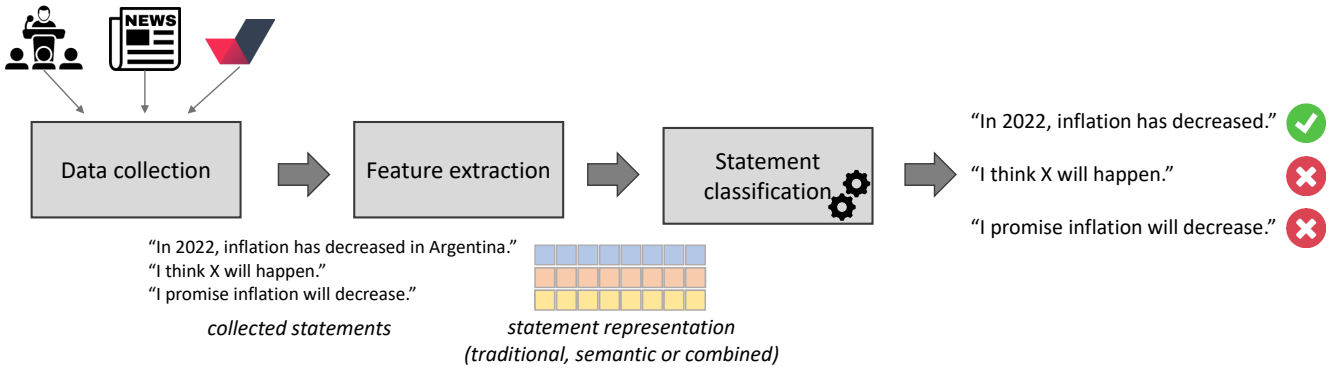

Fig. 1: Schematic representation of the methodological pipeline

"Washington DC" (location). Statements that mention people and institutions are likely to be checkable because they can negatively affect the mentioned entities [2]. In addition, entities provide unambiguous information that should be easy to verify. Therefore, including information regarding entities might provide good indicators of checkable claims [2]. Nonetheless, the information regarding the mentioned entity might not be as informative as knowing its category [28]. For the purpose of the study, we kept the information related to whether statements referred to locations, organizations and people, but not the particularities of which entity was mentioned. In addition, we included extra features associated with "miscellaneous" entities, i.e., the statement included any other type of entity and quantities. For recognizing quantities, we translated digits into a uniform representation, which acted as categories of quantities. For example, numbers like 2010 were replaced by $dddd$, and numbers like 15.66% were replaced by $dd.dd\%$.

Part-of-Speech (POS) tagging (also referred to as grammatical tagging) is the task of categorizing words in a text into particular parts of speech according to how words function in meaning and the grammatical context of a sentence. As POS tags describe the characteristics of a term within a sentence, they can be used as a proxy for shallow semantic features [30]. In this sense, these features have successfully distinguished between fake and authentic content and objective and subjective content [30]. We considered the following tags: adjectives, adposition, adverb, auxiliary verb, conjunction, determiner, interjection, noun, numeral, pronoun, proper noun, punctuation, symbol, and verb. POS information is used in the form of n-grams of size three, representing contiguous POS sequences (e.g., determiner-noun-adjective and verb-noun-adjective). These sequences are considered under the assumption that they are more informative than their individual counterpart, as shown in previous works [10, 18].

For this traditional representation, we obtained approximately 3500 features. In general, with a fixed number of training examples, the performance of any trained model first increases as the number of features increases, as more dimensions could imply a more accurate representation of instances. However, after a determined number of features, performance starts to decrease, while at the same time, the computational complexity increases. It would be possible that the feature space grew at the expense of having a larger number of features yielding low frequency and thus might not carry relevant information. Feature selection [24] techniques can be applied to reduce the feature space by removing redundant and irrelevant features. Based on preliminary evaluations, reducing the feature space by 50% allows reducing the noise caused by irrelevant features while increasing the model's performance. Features were selected based on their χ^2 score.

Semantic representations.: Despite its simplicity, the BoW model has a few drawbacks. First, it can create large feature spaces, which could become sparser as more instances are added, thus affecting the performance of the trained model. Second, as it is context-independent (with only a few exceptions, such as considering n-grams instead of individual words), it assumes that words are independent of each other, ignoring potential semantic relationships between them (e.g., synonyms) or that words might have different meanings depending on the context (e.g., polysemy). Third, it requires a large number of instances to extract relevant information, which could also lead to overfitting.

In the last decade, deep learning models have been developed for text representation in low dimensional spaces [15], i.e, "word embeddings". In general, these models can convert texts into numeric vectors aiming at capturing words' semantics. In this sense, words that usually appear in similar contexts would usually appear closer to each other in the embeddings space, which allows capturing syntactic and semantic regularities in language [15]. In turn, these representations allow improving the generalization of the trained model.

Word embedding techniques can either represent isolated words (e.g., Word2vec), contextualized words (e.g., ELMo), or complete sentences (e.g., Universal Sentence Encoder). Sentence embeddings differ from word embeddings in that they generate a unique representation for each sentence instead of one for each word that later needs to be summarized. As a result, sentence embeddings better capture polysemy, word order, out-of-vocabulary words, and sentence forms [1]. Then, for this study, we used the multilingual Universal Sentence Encoder (USE) [4] to generate a 512 fixed dimension and

dense representation of each statement. USE has already been used to classify Spanish text in a similar task [13].

C. Statement classification

As previously mentioned, we define the detection of checkable statements as a binary classification task in which statements are classified as "checkable" or "non-checkable". For the performed evaluation, we chose the most commonly used classification techniques: Multinomial Naïve Bayes (NMB), Random Forest (RF), Support Vector Machines (SVM) and Logistic Regression (LR). The first three models are the ones used by ClaimBuster and FullFact. The collected data was divided into train and test sets following a k-fold stratified cross-validation model, setting k to 5. To make results comparable, the same data partitions were used for all evaluations. To avoid data leakage, in the case of the traditional representation, the features used for model training were extracted only from the training data in each fold. The selected partition strategy follows the fact that there is no need to account for temporal relation in the data, which would have required a temporal train-test division. In addition, cross-validation splits have already been used in the literature for the same task [8, 9, 11].

Performance was evaluated considering the macro and per class Precision, Recall and F-Measure. The macro versions compute the metrics for each class independently and then average them, thus mixing and treating all classes equally. On the other hand, the per-class metrics allow us to assess how well the model performs for each class. Given the nature of the task, we are interested in models with higher recall (i.e., most of the actual checkable statements are identified) at the expense of lower precision.

In addition, we report a statistical analysis of results. A paired test was applied to the results obtained for each fold and each combination of features and classifier. We defined a null and an alternative hypothesis. The null hypothesis stated that no difference existed among the results of the different combinations, i.e., the combination of features and classifiers performed similarly. On the contrary, the alternative hypothesis stated that the observed differences were significant and non-incidental. The alpha value was set to 0.01.

D. Implementation details

All processing was implemented in Python. Three libraries were evaluated for the Natural Language Processing tasks: NLTK[11], SpaCy [12] and Stanza[13]. NLTK was discarded as it only provided limited Spanish support. SpaCy and Stanza provide different pre-trained models for tokenization, POS tagging, named entity recognition and lemmatization. SpaCy was selected as, despite achieving similar performance to Stanza, it provides a more optimized pipeline, allowing for faster processing. Although SpaCy provides extensive support for the Spanish language, it still has some limitations. For example, it can only detect a limited number of entities compared to those detected for the English language. The semantic representation of statements was implemented using the SpaCy integration [14] of the pre-trained Google USE models [15].

Classifiers were implemented using Sklearn [16]. Hyperparameter optimization was performed over a subset of the collected data. Reported results correspond to those obtained for the best parameters[17]. Experimental evaluation was run on a Ryzen 7 Serie 4000. Nonetheless, no special hardware resources are required for replicating the study.

IV. Evaluation

Table II presents the evaluation results for each combination of statement representation and classifier. For each metric, we report the average score across the 5 folds and the corresponding standard deviation. The best results are shown in **bold**, and the second-best are underlined. As a baseline for the evaluation, the Table includes the results of a random classifier. As it can be observed, all evaluated alternatives improved the results of the random classifier. Despite not being evaluated over the same data, it is worth considering that according to [14], human precision (over a balanced data collection) could be close to 70%. Similarly, approaches designed for the English language, on average, have reported precision and recall close to 0.72 and 0.79, respectively, as reported in Section II.

Traditional representation.: As the Table shows, LR and MNB achieved similar results for the three metrics, while RF and SVM achieved a slightly higher precision than recall. Note that the MNB alternative is the closest one to the original Chequeado implementation. Macro metrics were similar for all models, while some differences were observed for the checkable class metrics. In general, macro metrics were higher than those for the checkable counterpart. This could imply that one of the classes is easier to predict for the models, which is expected given the unbalanced nature of data. These differences were more noticeable for RF and SVM, which achieved both the highest precision and lowest recall. This means that classifiers are more confident that the identified checkable statements are actually checkable, at the expense of identifying fewer of them. For the purpose of the task, it is important to identify most of the statements, as missing a potentially relevant statement can have negative consequences[18]. Then, recall takes precedence as long as precision falls in an acceptable range (e.g., the human rate).

Semantic representation.: As for the traditional representation, standard deviations were lower than 0.03, implying that results are consistent across folds. There is also a difference between the prediction for both classes for this representation. In terms of the macro metrics, LR, RF and SVM achieved

[11]https://www.nltk.org/

[12]https://spacy.io/

[13]https://stanfordnlp.github.io/stanza/

[14]https://github.com/MartinoMensio/spacy-universal-sentence-encoder

[15]https://tfhub.dev/google/collections/universal-sentence-encoder/1

[16]https://scikit-learn.org/stable/

[17]More details can be found in the companion repository.

[18]For example, during the peak of the COVID-19 pandemic, there appeared many false statements regarding the consumption of toxic substances as a way to prevent getting sick, which needed to be debunked as fast as possible.

		Precision		Recall		F-Measure	
		Macro	Check. class	Macro	Check. class	Macro	Check. class
Random		0.49±0.002	0.39±0.002	0.50±0.002	0.50±0.004	0.49±0.002	0.43±0.003
Traditional representation	LR	0.83±0.01	0.79 ±0.02	0.83±0.01	0.78±0.02	0.83±0.01	0.79±0.02
	MNB	0.83±0.02	0.79±0.03	0.82±0.01	0.78±0.01	0.83±0.01	0.78±0.02
	RF	0.84±0.01	0.84±0.02	0.81±0.01	0.69±0.02	0.82±0.01	0.76±0.01
	SVM	0.85±0.01	0.85±0.02	0.82±0.01	0.72±0.01	0.83±0.01	0.78±0.01
Semantic representation	LR	0.83±0.01	0.78±0.01	0.84±0.02	**0.83**±0.03	0.84±0.01	**0.8**±0.01
	MNB	0.84±0.01	**0.90**±0.03	0.77±0.01	0.59±0.02	0.79±0.01	0.71±0.02
	RF	0.84±0.01	0.87±0.02	0.80±0.01	0.66±0.02	0.81±0.01	0.75±0.01
	SVM	0.86±0.01	0.86±0.02	**0.85**±0.01	0.77±0.02	**0.85**±0.01	0.82±0.01
Combined representation	LR	0.84±0.01	0.8±0.01	0.84±0.01	0.8±0.01	0.84±0.01	**0.8**±0.01
	MNB	0.84±0.01	0.82±0.02	0.83±0.01	0.79±0.02	0.84±0.01	0.80±0.01
	RF	0.86±0.02	**0.90**±0.03	0.80±0.01	0.65±0.01	0.82±0.02	0.75±0.03
	SVM	**0.87**±0.01	0.88±0.03	0.84±0.01	0.74±0.02	**0.85**±0.01	**0.80**±0.02

TABLE II: Results of checkable classification for a 5-fold evaluation

similar results than for the traditional representation, while MNB decreased its recall by 7%. Regarding the checkable class, recall decreased 18% on average for MNB and RF, while slightly increased 6% on average for LR and SVM. In the case of the MNB, the decrease in recall was accompanied by an increment of precision, i.e., the number of false positives decreased at the expense of increasing the false negatives. On the other hand, LR and SVM increased recall while maintaining precision, i.e., the number of both false negatives and false positives decreased.

Combined representation.: The goal of combining both representations was to leverage the different data perspectives that they provide and whether their combination could help boost performance. Regarding the macro metrics, results were similar to the other representations. For LR, differences in the checkable metrics with the semantic representations were not statistically significant, implying that adding more features did not allow improving performance. In the case of MNB, precision differences for the checkable class regarding the best previous results (i.e., traditional representation) were statistically significant, while recall differences were not statistically significant. Despite its improvements, MNB still had slightly worse performance than LR and SVM in terms of F-measure, precision (SVM) and recall (LR). RF improved its recall compared with the semantic representation, but results were still lower than those of the traditional representation. It kept the tendency for high precision but low recall, making the model unsuitable for the task. Finally, for SVM, precision differences were statistically significant, while recall differences were not. In this sense, it could be stated that combining the two representations, while it did not significantly increase the number of identified checkable statements, significantly decreased the number of false positives. Considering that the statements that the models identify will likely be passed down to a human fact-checker, reducing the number of false positives would reduce their workload, allowing them to concentrate on the following steps of the fact-checking process.

In summary, results showed that the evaluated representations and models obtained similar results to both the human study and the state-of-the-art models for the English language. LR and SVM were the best performing models for the three evaluated representations. The precision differences favouring SVM and the recall differences favouring LR were statistically significant. In this sense, LR would be able to identify more relevant statements (improving the coverage of statements), while SVM would be able to reduce the number of incorrectly identified statements (reducing the load of human checkers) while missing some relevant statements.

Time comparison.: Another factor to consider when choosing a model is its time complexity. Figure 2 presents the average time each trained model spent on classifying instances in one fold. As results showed, combining the features doubled the required time. Also, while NMB, RF and LR showed similar execution times, SVM was approximately 33% slower than the other models. On average, models can process approximately 500 statements per second. As a reference, the inaugural presidential speech of 2021 included 648 statements. Then, even when considering the combined representation, models would allow the real-time processing of statements.

Considering the time analysis, SVM was up to 49% lower than LR, which might hinder the model's scalability. As a result, LR trained on either a semantic or a combined representation seems suitable for identifying checkable statements.

V. Conclusions

This study tackled the identification of checkable statements in Spanish by evaluating different combinations of features and classifiers, aiming to achieve comparable results to state-of-the-art techniques for English text. The best performing model is publicly available and, at the time of writing, used by Chequeado and other fact-checking organizations in Latin America.

Several aspects could be tackled in future works. First, the model was trained and evaluated for a particular Spanish variant. In this context, additional evaluations could be performed

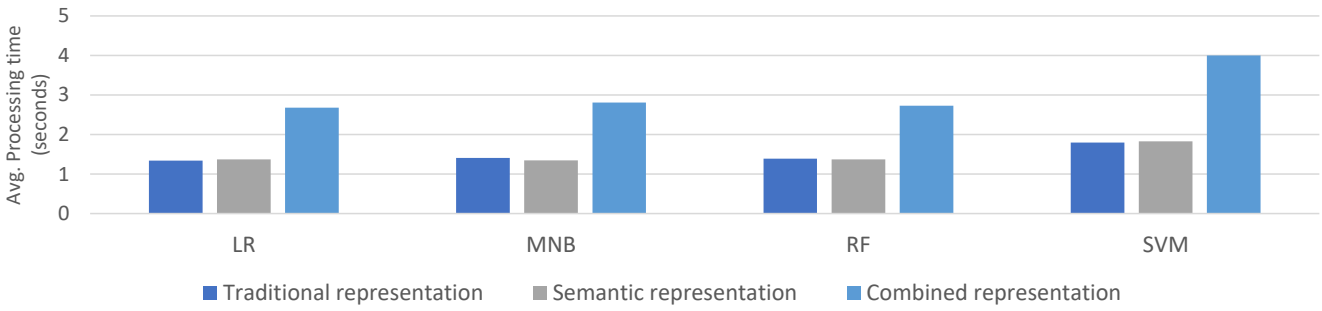

Fig. 2: Average processing times in seconds for checkable classification

over different data collections from different variants to assess the generalizability of results. Second, we used pre-trained models for the semantic representation of statements. These pre-trained models could be fine-tuned using the collected data to allow embeddings to adapt to the particularities of the data. For example, to adapt to specific topics that could not have been available in the full training data or to particularities of the Spanish variants. In addition, considering the differences observed for the traditional and semantic representation, a more in-depth study of the contribution of each feature type to model performance is needed. Third, the built data collection is based on semi-formal text, including complete sentences and correct grammatical structures. In the future, it could be possible to extend the search of checkable statements to other domains, such as messaging platforms or even video transcripts. These new sources might present different language uses and expressions which might be new to the model. Then, new data collections including such examples should be built to expand the model's capabilities. Finally, the built model could be used as a first step for identifying statements referring to the same event, or to detect previously fact-checked statements.

References

[1] Y. Adi, E. Kermany, Y. Belinkov, O. Lavi, and Y. Goldberg. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *arXiv preprint arXiv:1608.04207*, 2016.

[2] B. Altun and M. Kutlu. Tobb-etu at clef 2019: Prioritizing claims based on check-worthiness. In *CEUR Workshop Proceedings*. CEUR-WS, 2019.

[3] F. Arslan, N. Hassan, C. Li, and M. Tremayne. A benchmark dataset of check-worthy factual claims. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 14, pages 821–829, 2020.

[4] D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, et al. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*, 2018.

[5] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*, 2017.

[6] E. Ferrara. Measuring social spam and the effect of bots on information diffusion in social media. In *Complex spreading phenomena in social systems*, pages 229–255. Springer, 2018.

[7] P. Gencheva, P. Nakov, L. Màrquez, A. Barrón-Cedeño, and I. Koychev. A context-aware approach for detecting worth-checking claims in political debates. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 267–276, 2017.

[8] N. Hassan, F. Arslan, C. Li, and M. Tremayne. Toward automated fact-checking: Detecting check-worthy factual claims by claimbuster. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1803–1812, 2017.

[9] I. Jaradat, P. Gencheva, A. Barrón-Cedeño, L. Màrquez, and P. Nakov. Claimrank: Detecting check-worthy claims in arabic and english. *arXiv preprint arXiv:1804.07587*, 2018.

[10] J. Kapusta, M. Drlik, and M. Munk. Using of n-grams from morphological tags for fake news classification. *PeerJ Computer Science*, 7:e624, 2021.

[11] L. Konstantinovskiy, O. Price, M. Babakar, and A. Zubiaga. Toward automated factchecking: Developing an annotation schema and benchmark for consistent automated claim detection. *Digital Threats: Research and Practice*, 2(2):1–16, 2021.

[12] N. Kotonya and F. Toni. Explainable automated fact-checking: A survey. *arXiv preprint arXiv:2011.03870*, 2020.

[13] S. B. Majumder and D. Das. Detecting fake news spreaders on twitter using universal sentence encoder. In *CLEF (Working Notes)*, 2020.

[14] A. Merpert, M. Furman, M. V. Anauati, L. Zommer, and I. Taylor. Is that even checkable? an experimental study in identifying checkable statements in political discourse. *Communication Research Reports*, 35(1):48–57, 2018.

[15] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[16] M. J. Moon. Can it help government to restore public trust? declining public trust and potential prospects of it in the public sector. In *36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the*, pages 8–pp. IEEE, 2003.

[17] A. Patwari, D. Goldwasser, and S. Bagchi. Tathya: A multi-classifier system for detecting check-worthy statements in political debates. In *Proceedings of the ACM on Conference on Information and Knowledge Management*, pages 2259–2262, 2017.

[18] P. Przybyla. Capturing the style of fake news. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 490–497, 2020.

[19] G. Salton and M. J. McGill. *Introduction to modern information retrieval*. mcgraw-hill, 1983.

[20] C. Segovia Arancibia. *Political Trust in Latin America.* PhD thesis, 2008.

[21] S. Singla. Checking fact worthiness using sentence embeddings.

arXiv preprint arXiv:2012.09263, 2020.

[22] C. R. Sunstein and A. Vermeule. Conspiracy theories. *Harvard Public Law Working Paper*, 2008.

[23] R. M. Sutton and K. M. Douglas. Conspiracy theories and the conspiracy mindset: Implications for political ideology. *Curr Opin Behav Sci*, 34:118–122, 2020.

[24] J. Tang, S. Alelyani, and H. Liu. Feature selection for classification: A review. *Data classification: Algorithms and applications*, page 37, 2014.

[25] A. Vlachos and S. Riedel. Fact checking: Task definition and dataset construction. In *Proceedings of the ACL 2014 workshop on language technologies and computational social science*, pages 18–22, 2014.

[26] N. Vo and K. Lee. Where are the facts? searching for fact-checked information to alleviate the spread of fake news. *arXiv preprint arXiv:2010.03159*, 2020.

[27] S. Vosoughi, D. Roy, and S. Aral. The spread of true and false news online. *Science*, 359(6380):1146–1151, 2018.

[28] K. Yasser, M. Kutlu, and T. Elsayed. bigir at clef 2018: Detection and verification of check-worthy political claims. In *CLEF (Working Notes)*, 2018.

[29] V. S. Zúñiga and M. P. Torres. Confianza en instituciones políticas: factores que explican la percepción de confianza en chile. *Temas sociológicos*, (25):231–258, 2019.

[30] C. Zuo, A. Karakas, and R. Banerjee. A hybrid recognition system for check-worthy claims using heuristics and supervised learning. In *CEUR workshop proceedings*, volume 2125, 2018.