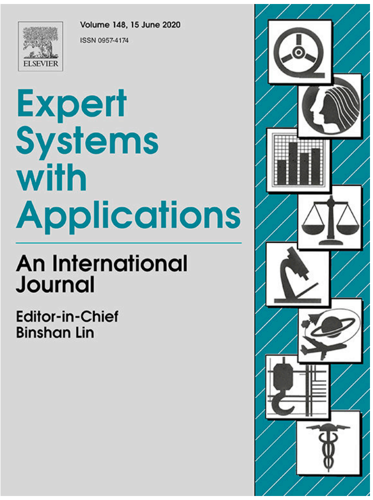


ELSEVIER
Volume 148, 15 June 2020
ISSN 0957-4174

Expert Systems with Applications

An International Journal

Editor-in-Chief
Binshan Lin

Stance detection in tweets: a topic modeling approach supporting explainability

Manuela Gómez-Suta, Julián Echeverry-Correa, José A. Soto-Mejia

PII:	S0957-4174(22)02064-4
DOI:	https://doi.org/10.1016/j.eswa.2022.119046
Reference:	ESWA 119046
To appear in:	*Expert Systems with Applications*
Received Date:	3 January 2022
Revised Date:	30 September 2022
Accepted Date:	12 October 2022

Stance detection in tweets: a topic modeling approach supporting explainability

Manuela Gómez-Suta[a] (madegomez@utp.edu.co), Julián Echeverry-Correa[b] (jde@utp.edu.co), José A. Soto-Mejia[a] (jomejia@utp.edu.co)

[a]*Data Envelopment Analysis Research Group. Faculty of Business. Universidad Tecnológica de Pereira. Colombia.*
[b]*Data Analysis and Computational Sociology Research Group. Faculty of Engineering. Universidad Tecnológica de Pereira. Colombia.*

Corresponding Author:
Manuela Gómez-Suta
Data Envelopment Analysis Research Group
Faculty of Business
Universidad Tecnológica de Pereira
Colombia
Tel: (+57) 312 285 6961
Email: madegomez@utp.edu.co

Stance detection in tweets: a topic modeling approach supporting explainability

Manuela Gómez-Suta[a,1], Julián Echeverry-Correa[b], José A. Soto-Mejia[a]

[a]Data Envelopment Analysis Research Group. Faculty of Business. Universidad Tecnológica de Pereira. Colombia.
[b]Data Analysis and Computational Sociology Research Group. Faculty of Engineering. Universidad Tecnológica de Pereira. Colombia.

Abstract

Stance detection improves fake information recognition in social media. This task encourages interpreting and explaining the misinformation identification, thus aligning with the importance of improving human trust in classification results. Nonetheless, most of the stance detection studies do not engage in understanding the reasons behind a solution. We propose a two-phase classification system for stance detection in tweets. We mainly exploit topic modeling features. Our proposal is remarkably different from previous ones since we provide an explanation of stance labels through the most relevant terms within topics over the tweets. Therefore, our approach is flexible because it adjusts to the vocabulary leveraging topic information. We additionally construct sets of features seeking tweet-specific content, sentiment and subjectivity markups, target attributes, term distribution, and word embeddings. Our classification system ranks second in the state-of-the-art regarding *SemEval-2016 task 6* dataset with a 74.63% overall F-measure. To the best of our knowledge, our results are superior to deep learning-based proposals and competitive with studies that do not provide an explanation for stance labels. Hence, we affirm that topic modeling features enhanced classification results and provided textual information about a plausible explanation of stance labels. We report the performance of our system considering variations in the feature set, besides discussing the explanation of the results.

Keywords: stance detection, topic modeling, explainability, opinion mining.

In brief, the highlights of our study include:

- We propose stance detection exploiting topic modeling features.
- Our proposal ranks second in the state-of-the-art.
- We present qualitative information to relate tweet content and stance prediction.
- We provide explanations of our classification models in contrast to other studies.
- Our main contribution is exploiting topic models to explain the classifier behavior.

1. Introduction

Social media platforms efficiently spread fake data, unverified claims, and fabricated attention-grabbing stories (Zhang & Ghorbani, 2020). The misinformation manipulates user behavior. For instance, inaccurate reviews can decrease sales opportunities and destroy a brand's reputation. Moreover, the misleading news has promoted great confusion on a viral scale to influence public opinion about socio-political determinations (Saquete et al., 2020). Consequently, the fake information has weakened the user's trust in governments, companies, and legitimate mass media. Therefore, it is relevant to study fake information detection to strengthen trust in social and digital processes.

Different approaches enable this task. The feedback-based approach applies stance detection to exploit secondary data as user viewpoints/reactions/comments to infer the information veracity (Zhou & Zafarani, 2020). Stance detection task consists of identifying the orientation (commonly *Favor*, *Against*, *None*) that each comment expresses behind a target (*i.e.* concept, idea, news, opinion, claim), even if the post explicitly or implicitly mentions the target (Küçük & Can, 2020). The reactions reflect the sentiments and attitudes of individuals toward an object's behavior. As a result, the information allows inferring the target truthfulness. This task occurs in the news and social networks (Saquete et al., 2020). The former consider classifying the stance of text relative to the respective headline. In social media, the object is to detect the post (such as a tweet) stance regarding a target.

Stance detection improves fake recognition in three situations. First, it is possible to obtain an accurate assessment of the user's trust in the data. This strategy responds to the trustworthiness subjective nature as it evaluates opinions that people build through different observations and system understanding. The second situation was mentioned previously and refers to the early detection of misleading information that has never been seen before. Specifically, stance detection helps to identify check-worthy topics regarding whether the content trigger large-scale discussions (Zhou & Zafarani, 2020). The last situation refers that this task encourages the interpretability of fake information detection. Stance information allows justifying and illustrating the behavior of the classification models. Hence, user comments provide new

[1] Corresponding author.
Email addresses: madegomez@utp.edu.co (Manuela Gómez-Suta), jde@utp.edu.co (Julián Echeverry-Correa), jomejia@utp.edu.co (José A. Soto-Mejia)

knowledge about text characteristics and extra-linguistic relations (Saquete et al., 2020).

Qian et al. (2018) report that mixing content-based techniques and user stance allows 88.05% accuracy during fake news detection. Figure 1 shows a real-world example of the stance of users' responses toward a fake tweet about the AC Milan footballer Michael Essien had contracted Ebola. The *disagreed* label means a stance against the tweet. Stance label benefits to suspect about the tweet veracity since all the responses are in opposition to the tweet. Hence, Figure 1 exposes a real example demonstrating the advantages of stance detection to recognize information credibility.

fake tweet: Unconfirmed reports claim that Michael Essien has contracted Ebola.

response 1: @FootballcomEN if it is unconfirmed, why do you tweet it? *[disagreed]*

response 2: @FootballcomEN Fuck off. You're an attention seeking cunt. *[disagreed]*

response 3: @FootballcomEN 😂😂😂 *[disagreed]*

response 4: @FootballcomEN how would someone start such a bizarre rumour about Essien?Human beings are heartless *[disagreed]*

response 5: @FootballcomEN This is not funny, people are dying from this. You should be ashamed of yourself. *[disagreed]*

response 6: @ohlozzie @FootballcomEN Who mentioned it being funny? 'UNCONFIRMED reports'.... to translate: 'it might not be true, but...' *[disagreed]*

Note. Created with information from PHEME dataset of rumors and non-rumors (Zubiaga et al., 2016).

Figure 1: Example of the stance of users' responses that discuss the veracity of a fake tweet. Stance label is in brackets and italic at the end of each response.

Stance detection overcomes other strategies as content-based that use external knowledge to check the information truthfulness regarding the content and style features (Saquete et al., 2020). Moreover, the content-based approach is limited to specific language variants "creating a cat-and-mouse game" (Zhou & Zafarani, 2020, p. 20), where malicious entities change their deceptive writing style to the models, and external resources do not identify fake information. Hence, stance detection is independent of fake content or style information. Consequently, this task is a starting point for the early identification of misleading information that feedback makes it easier to recognize the check-worthy content (Zhou & Zafarani, 2020).

Nonetheless, stance detection is a challenge considering that the definition of truth is subjective. Opinions and orientations arise from different expressions, statements, and linguistic compositions (Li et al., 2020). In news stance detection, the studies usually employ the pair headline/body news to establish features as common *n-grams* or overlapping words to train models (Wojatzki & Zesch, 2016; Al-Ghadir et al., 2021). The task is more demanding in social media, where the document commonly is a tweet that lacks contextual information since headlines are smaller than those used in news stance detection. Moreover, tweets are succinct, have many abbreviations, and the text is informal written and non-standardized because users usually do not follow grammar rules (Saquete et al., 2020).

SemEval-2016 task 6: Detecting Stance in Tweets - task A dataset (Mohammad et al., 2016) is the most widespread data for stance detection in tweets. The dataset consists of 4163 tweets divide into five targets *Atheism* (A), *Climate Change is a real Concern* (CCC), *Feminist Movement* (FM), *Hillary Clinton* (HC), *Legalization of Abortion* (LA) with *Favor*, *Against*, *None* stance labels. The authors labeled the dataset regarding the sentiments as *Positive*, *Negative*, and *Neutral*. Moreover, the dataset has a label to indicate if the tweet does not express a point of view (*No One*) or if the opinion regards the target (*Target*) or not (*Other*).

The participants in *SemEval-2016 task 6* presented feature-based machine learning and deep learning proposals. Nevertheless, the participants achieved a F-measure between 54-67% without overcoming the baseline results that followed a machine learning approach. Machine learning works achieve certain effectiveness and do not overcome this point. Meanwhile, the deep learning techniques show optimistic results but do not justify the computational cost and resources (Saquete et al., 2020).

Recent studies have reached scores higher than 70%; refining semantic features enhance subjectivity detection through external lexicons (Dey et al., 2017). Therefore, this proposal depends on external lexicons whose information is limited and may not respond to data variations or targets other than those present in *SemEval-2016 task 6* dataset. Alternatively, the state-of-the-art paper (Al-Ghadir et al., 2021) utilizes a retrieval strategy of taking the top-*k* terms following a ranking of a weighting scheme; also, the classifier is a weighted *K* nearest neighbor (WKNN) with *K=1* and experiments with principal component analysis (PCA). Although, the dimensionality reduction technique does not present a statistically significant improvement. Nonetheless, the authors do not make explicit what the strategy is to weight the feature vectors within WKNN; therefore, their results are not reproducible.

Unfortunately, even though such proposals achieve the highest overall F-measure in the *SemEval-2016 task 6* dataset, these works lack explicitness in the sense that it remains unclear how to use their output to understand the reasons behind the stance detection performance or how the feature engineering help to provide information on the disagreements generated by the target. Thus, previous research rarely indicates which features and textual content influence the task. It is necessary to explain why the system predicts that a tweet set expresses a *Favor* or *Against* stance given a target.

Addressing misinformation through feedback-based data such as stance detection involves not only the classification result but also an explanation and justified reasoning for the choice (Saquete et al., 2020). Consequently, the interpretability of computational models has become a significant aspect in fake information detection since people usually may not trust the system decision unless they understand the reasons for the judgment making (Liu et al., 2016). Moreover, humans accept or reject the veracity of information when they can contrast their prior knowledge and the explanation behind a system's performance (Ribeiro et al., 2016).

Devoting effort to seeking justification of the classification result increases user acceptance of the predictions of various systems. For instance, Przybyła & Soto (2021) implement a set of interactive visualizations to explain a credibility score regarding different features in the credibility assessment task. Thus, the authors report that human credibility assessment of text becomes more accurate as users understand the features of non-credible documents. Furthermore, Kirchner & Reuter (2020) analyze the perception of warnings of misinformation and find that 65% of users claim that their confidence in the predicted label of the tweet increases as they obtain reasons for this result. Additionally, Kirchner & Reuter (2020) state that seven out of ten participants wish that the platform explained why it labeled the tweet as fake.

Likewise, understanding the reasons behind an automatic classification leads to the interpretation of misinformation (Saquete et al., 2020). Therefore, through an automatic classifier, it is possible to recognize linguistic and extra-linguistic features to discriminate false information. This has three main advantages. First, the discovered knowledge enriches the credibility of the domain. Second, it promotes user confidence in automatic classification systems (Liu et al., 2016). Third, the model provides transparency to justify the predicted label of each instance.

Most of the current stance detection studies do not explain how the values of the feature vectors provide an explanation of the difference between tweet stances given a target. For this reason, there is a gap in the interpretability of model results, as it is unknown how the input attributes guide the classification of an instance set (Wojatzki & Zesch, 2016). Commonly, research in different tasks tends to report a low performance when they commit to providing interpretability of the classification results (García-Cuesta et al., 2020; Przybyła & Soto, 2021). This situation occurs in stance detection during *SemEval-2016 task 6* dataset.

Some efforts use external resources to categorize the topics discussed in the documents. For instance, Mohammad et al. (2017) explore the dataset using tags assigned by human evaluators. These external resources potentially help during stance detection by contributing to interpreting the label assigned to the tweet. The main drawback of the proposal is a dependence on pre-established knowledge on resources that may or may not respond to new data or domains. Hence, the transparency obtained leads to a decrease in system performance due to the static nature of the inference features (see table 15).

Li & Caragea (2020) suggest that using attention weights (learned with memory update mechanism from deep learning techniques) allow for differentiating the stance of the tweets. The authors observe that the proposal fails because the memory mechanism led to overfitting. Thus, the model generalization is poor in test data.

We propose stance detection exploiting topic modeling features. Our research aims to find semantic structures and discussed topics in each target, assigning multinomial distributions over words. We build the vocabulary with the most relevant terms within the topics, allowing us to characterize and explain tweet stance.

The term distribution in topics has been shown to improve stance recognition. For instance, Thonet et al. (2016) outperform the state-of-the-art in Bitterlemons dataset during viewpoint classification when distinguishing term-topic distribution. Specifically, the topic terms encode the information related to the stance implicit in texts. Furthermore, Lin et al. (2019) mine term distribution to create features in standpoint classification and affirm that exploiting the latent topic information helps to identify a more fine-grained stance in the documents.

Thus, term distribution from topic modeling allows extracting crucial information since not all words in the tweets make the same contribution to attitude expressions. Therefore, topic modeling helps elucidate the text content and explain the main features during stance detection. We follow a conceptualization similar to Ribeiro et al. (2016) since we exhibit explanation and interpretability of the model, presenting textual data to give qualitative understanding. Hence, we present qualitative information on the relationship between the structure and content of tweets with the stance prediction of the model. To the best of our knowledge, our proposal achieves an overall F-measure superior to many baselines. It ranks second in the state-of-the-art when the studies with similar scores do not provide explanations for the behavior of their classification models.

We built topics using Biterm Topic Model (BTM) appropriate for short text (Yan et al., 2013). The input of the topic model was the set of tweets in training partition. We experimented with various representations of tweets. To illustrate, we transformed the texts to exclusively contain the words from the tweets with *Favor* and *Against* labels and no terms of the tweets with no opinion (*None* label). Our intuition behind tweet codifications was to accentuate terms that facilitate recognizing a stance toward a specific target. We only analyzed the train set to determine which terms would be used to represent the tweets.

The most relevant terms within the retrieved topics compose the vocabulary. We use this vocabulary to build feature vectors following a *tf-idf* weighting scheme. Namely, our approach is flexible because it is data-based to adjust the vocabulary by leveraging topic information. In addition, the selected textual data can exemplify terminological characteristics to differentiate tweets with and without an opinion given a specific target. Consequently, the most relevant topic terms contribute to building an explanation regarding what the classification system infers is a no opinion tweet.

Section 2.2 shows that other investigations have worked on topic models to generate feature vectors. One of the novelties of our work is that we do not exploit the weights in the term-topic matrix or document-topic matrix. We employ the topic composition to establish the vocabulary. Zhang & Lan (2016) present a similar work because they collect the top

20 words in each topic to analyze the similarity between the top terms and the content of the tweets through an overlap measure. Also, Mourad et al. (2018) count the number of top words found in the tweet per topic and the ratio of the number of top words in the tweet with the total top words in all topics. Alternatively, our proposal uses the most relevant terms to build feature vectors and provide an explanation of the stance labels.

Further, we construct collections of features seeking tweet-specific content, sentiment and subjectivity markups, target attributes related to tweet words, term distribution over the tweets regarding *tf-idf* weighting scheme, and word embeddings using BERTweet (Nguyen et al., 2020). Some of these features have been used in similar stance detection research (Dey et al., 2017; Al-Ghadir et al., 2021; Zhang & Lan, 2016; Tutek et al., 2016), although we made modifications to the coding of the information. These modifications are explained in section 3.2.

We support the idea that stance detection is a target-dependent task. Accordingly, we established a different set of features for each target. In particular, the feature vectors always accounted for the topic features as well as some tweet-specific content, sentiment and subjectivity markups, target attributes, term distribution with weighting scheme, or word embeddings.

Our classification model has two phases. First, we train a system to differentiate tweets with the *None* label from the other stances. Thus, the classifier recognizes texts that do not or do express an opinion. In the second phase, we discriminated the *Favor* and *Against* stances apart from only the texts containing an opinion after the results of the first phase.

We experimented with SVM and Logistic Regression (LR) classifiers, additionally two ensemble methods, Extremely Randomized Trees (ET) and AdaBoost (AB). The best results indicate no unique classifier between the targets during the two classification phases. For example, *Climate Change is a real Concern* (CCC) has the highest values with ET in the first phase, while SVM provides more accuracy between *Favor* and *Against* tweets. Our proposal tends to generate high dimensionality feature matrices. Therefore, we experiment with PCA as a dimensionality reduction strategy. In addition, we perform a recursive search for the best feature subset using the Recursive Feature Elimination (RFE) algorithm (Guyon et al., 2002). We achieve 74.63% overall F-measure in the *SemEval-2016 task 6* dataset. Thus, the system ranks second in the current state-of-the-art to the best of our knowledge.

Our main contribution is exploiting topic models to explain the classifier behavior and indicate the textual content influencing the text stance. Consequently, our proposal is remarkably different from previous ones since we provide a qualitative understanding of the performance system using the most relevant terms within topics over the tweets. In addition, we construct a classification system that provides superior results to deep learning-based proposals. Moreover, our results are competitive with those approaches that do not provide an explanation for stance detection results. We report the performance of our system regarding if tweets express opinion toward the targets.

Furthermore, comparison scenarios (varying the feature set) are presented to evaluate the advantages and shortcomings of the proposed features. We expose the contribution of the features to the overall performance through ablation tests. In addition, we discuss the explanation of the results considering the composition of the topics.

The rest of the document is organized as follows: section 2 presents studies that address stance detection in the *SemEval-2016 task 6* dataset, as it is the objective data of this work. Section 3 describes the research proposal of this paper. Section 4 states and discusses the results. Finally, section 5 indicates the conclusions and future work.

2. Related work

SemEval-2016 task 6 is the dataset commonly studied during stance detection in tweets. Table 1 shows the dataset statistics (Mohammad et al., 2016). Essentially, the classes are imbalanced. The difference in the distribution between targets could be why previous works fail to detect stances across all targets. In other words, a classifier regarding diverse targets obtains a lower performance than an architecture that generates a model for each target (Küçük & Can, 2020).

Some studies examine stance detection from deep learning and machine learning techniques. We follow this division to expose how similar research have modeled *SemEval-2016 task 6* dataset.

	Training (%)			Count	Testing (%)			Count
Target	Favor	Against	None		Favor	Against	None	
A	17.93	59.26	22.81	513	14.55	72.73	12.73	220
CCC	53.67	3.80	42.53	395	72.78	6.51	20.71	169
FM	31.63	49.40	18.98	664	20.35	64.21	15.44	285
HC	17.13	57.04	25.83	689	15.25	58.31	26.44	295
LA	18.53	54.36	27.11	653	16.43	67.50	16.07	280
Total	25.84	47.87	26.29	2914	24.34	57.25	18.41	1249

Table 1: Statistics about *SemEval-2016 task 6* dataset (Mohammad et al., 2016).

2.1. Deep learning techniques

Word embeddings codify the tweet as a dense vector of lower dimension compared to representations constructed with all the terms of the train set (Wei et al., 2018). The second place in *SemEval-2016 task 6* challenge used word2vec as input to Convolutional Neural Network (CNN) models regarding a classifier for each target (Wei et al., 2016). Alternatively, Zarrella & Marsh (2016) ranks first in the competition through a RNN with word embeddings learned on their corpus of more than two million tweets.

Deep learning techniques do not beat machine learning approaches (Küçük & Can, 2020). To illustrate, Siddiqua et al. (2019) report 72.11% in the task through a neural ensemble model from two long short-term memory (LSTM). Meanwhile, the current state-of-the-art has 76.45% through feature-based machine learning (Al-Ghadir et al., 2021). Thus, deep learning techniques do not show outstanding improvement that compensates for the computational effort and required resources (Saquete et al., 2020).

Ahmed et al. (2020) attempt to combine the advantages of content features and deep learning techniques. This study exploits sentiment, emoticon, and intensifier lexicons to train a layer within a LSTM variation. Despite this, the system only attains 70.46% during stance detection (see table 15). Even if the proposal does not obtain a high result in the classification task, it is an exciting work since it is one of the few efforts from the deep learning approach that point out which features are relevant in stance detection.

In brief, most of the deep learning work does not provide information to understand the decisions of classification systems. Moreover, the proposals claiming to give this information are characterized by low performance. Therefore, the deep learning approaches continue to show shortcomings in classification results and the production of more explainable models during stance detection.

2.2. Machine learning techniques

As previously stated, our proposal follows a machine learning approach. Therefore, this section presents the information in a way that our methodological model can be easily compared with similar investigations. Table 2 summarizes the studies based on machine learning techniques and our proposal regarding the different features such as i) lexical information, ii) tweet-specific content, iii) sentiment and subjectivity markups, iv) target attributes, v) term distribution, vi) word embeddings and, vii) topic modeling.

The lexical information usually is *n-grams* and *n-chars* up to a length of four or five grams codified as binary, count-based or weighted vectors (Tutek et al., 2016). Some approaches record the number of punctuation and exclamation marks, capitalized or repeated letters, as well as negation and wh-question terms (Ahmed et al., 2020).

On the other hand, tweet-specific content concerns syntactic attributes helpful in defining characteristics of the stance toward a target (Mourad et al., 2018; Ahmed et al., 2020). For example, Tutek et al. (2016) obtained third place in *SemEval-2016 task 6* challenge measuring the number of retweet symbols, hashes, emoticons, and mentioned users.

No system defeated the baseline of *SemEval-2016 task 6* challenge, which was a classifier based on binary vectors of *n-grams*. It demonstrates the benefit of morphological marks as lexical attributes and tweet-specific content during stance detection. Nonetheless, recent research with higher scores do not use these attributes (see tables 2 and 15). Further, lexical and tweet-content information do not provide an explanation to discriminate the different stances semantically.

Sentiment and subjectivity marks enrich the tweet with linguistic information. Sentiment features provide a score after contrasting the composition of the tweet and information from a lexicon. The most common resources are SentiWordNet, and NRC Emotion and Sentiment Lexicons[2]. There are diverse proposals for sentiment codification. Some studies count the tweet's positive, negative, and neutral words (Tutek et al., 2016; Elfardy & Diab, 2016).

In contrast, Dey et al. (2017) exploit SentiWordNet information and designate a polarity score to tweet terms. Despite this, the sentiment and stance expressed in the tweet do not always coincide. For instance, 51.70% of tweets in *SemEval-2016 task 6* dataset had a favorable stance but a negative sentiment label (Sobhani et al., 2016). Alternatively, Dey et al. (2017) rank third in the current state-of-the-art scoring whether the tweet expresses strong or weak subjectivity given its terminology composition and the information from MPQA subjectivity-polarity lexicon[3].

These linguistic markups enrich the succinct content of tweets. Hence, they potentially give an explanation of why a tweet expresses a stance *Favor* or *Against* a target. Although, these features have two weaknesses. First, the systems rely on static resources such as lexicons that may not respond to new data or domains. Second, linguistic-based strategies examine words in isolation from their context (Elfardy & Diab, 2016). As a result, each term is given a category regardless of its syntactic functions or the semantic information its context may provide about its interpretation.

Besides, Sobhani et al. (2016) expose a classifier regarding linguistic marks with a lower performance when compared to a system that codifies the presence of opinion toward a target. Further, the label distribution of *SemEval-2016 task 6* dataset indicates marked differences in tweets where the subject of the text is other than the target of the stance detection task (Mohammad et al., 2017). Target attributes usually are boolean vectors to model the presence of a target label in the tweet or the construction of *n-grams* and *n-chars* vectors regarding the target name (Dey et al., 2017).

Another feature used during stance detection is term distribution to measure the contribution of the terms in the tweet. Specifically, the system creates vector representations with the top *k* words within the set of tweets per target. Zhang & Lan

[2] NRC Emotion and Sentiment Lexicons: `https://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm`

[3] MPQA subjectivity-polarity lexicon: `https://mpqa.cs.pitt.edu/lexicons/`

(2016) determines the top 20 words within tweets from the same target through *tf-idf* weighting scheme. The current state-of-the-art founds its features in term distribution (Al-Ghadir et al., 2021). The authors claim to generate two lists of the 10 words with the highest *tf-idf* weight within the dataset. One list contains the tweet words and the other the stems. Thus, the system is trained with a fused feature vector of the normalized *tf-idf* weights of the top terms. Nonetheless, this proposal is not reproducible since it is ambiguous. The authors do not specify how to merge the weights of the two lists. In addition, it is not clear whether the top terms are selected by discriminating between related tweets by target or for the entire database. Besides, the authors report using a WKNN model without explaining how to weigh the features. Even more, there is no mention that the proposal constructs a classifier for each target or a common system. Al-Ghadir et al. (2021) do not provide information to understand the results of stance detection.

System	A	CCC	FM	HC	LA
Our system	Sentiment and subjectivity markups Target attributes Term distribution Word embeddings Topic modeling	Tweet-specific content Sentiment and subjectivity markups Target attributes Term distribution Word embeddings Topic modeling	Tweet-specific content Sentiment and subjectivity markups Target attributes Term distribution Word embeddings Topic modeling	Tweet-specific content Sentiment and subjectivity markups Target attributes Term distribution Word embeddings Topic modeling	Sentiment and subjectivity markups Target attributes Term distribution Word embeddings Topic modeling
Al-Ghadir et al. (2021)	Term distribution				
Dey et al. (2017)	Lexical information Sentiment and subjectivity markups Target attributes				
Mohammad et al. (2017)	Lexical information Target attributes Word embeddings				
Mourad et al. (2018)	Lexical information Tweet-specific content Labeled-based features				
Baseline	Lexical information				
Tutek et al. (2016)	Lexical information Tweet-specific content	Lexical information Sentiment and subjectivity markups	Lexical information Sentiment and subjectivity markups	Lexical information Tweet-specific content Word embeddings	Lexical information Tweet-specific content
Zhang & Lan (2016)	Lexical information Tweet-specific content Target attributes Term distribution Word embeddings Topic modeling	Lexical information Tweet-specific content Target attributes Term distribution Word embeddings Topic modeling	Lexical information Tweet-specific content Target attributes Term distribution Word embeddings Topic modeling	Lexical information Target attributes Term distribution Word embeddings Topic modeling	Lexical information Tweet-specific content Target attributes Term distribution Word embeddings Topic modeling
Elfardy & Diab (2016)	Lexical information Topic modeling Tweet-specific content				

Note. If the system supports target-dependent stance detection, then there is a feature set per target.

Table 2: Systems based on machine learning techniques.

Word embeddings are advantageous because they give a common ground to compare tweets with different compositions (Wani et al., 2021; Mohammad et al., 2017). Particularly, word embeddings enhance stance detection. For instance, Mohammad et al. (2017) use word2vec as a feature generator and describe better the performance. Furthermore, word embeddings trained in a tweet corpus present better results. Qian et al. (2018) use word2vec and report a lower overall F-measure than Mohammad et al. (2017), which builds word embeddings in their tweet corpus with word2vec algorithm. Likewise, the codification can impact the results. Mohammad et al. (2017) represent each tweet as the average word embeddings for all words appearing in the tweet.

Topic modeling is one of the features less explored in stance detection. Only two studies consider these features in *SemEval-2016 task 6* dataset. First, Elfardy & Diab (2016) apply Weighted Textual Matrix Factorization to model out-of-vocabulary words. In particular, the authors train the model in an external corpus (WordNet, Wiktionary definitions, and the Brown Corpus), setting 100 topics. Consequently, the system learns a vector representation for the words that could appear in the train set and other terms (such as words that may be present in the test set), but this proposal does not clarify how to encode each tweet regarding the learned vectors. Besides, Elfardy & Diab (2016) do not offer an interpretation of the topics retrieved and how these could contribute to constructing an explanation of the results. Second, Zhang & Lan (2016) use Latent Dirichlet Allocation model to generate a document-topic matrix and term-topic matrix of tweets associated with a target. As in the previous study, the authors do not attempt to interpret the retrieved topics and their contribution to the explanation of stance labels. Therefore, current proposals based on topic modeling do not provide results that explain the stance detection results. Moreover, the mentioned research does not have remarkable performance on the task. Zhang & Lan (2016) rank 14th, and Elfardy & Diab (2016) rank 16th in the state-of-the-art.

Consequently, our proposal has two points of innovation. First, we exploit topic composition to generate an explanation of stance labels. We provide this information without relying on lexical resources. In addition, we benefit from the philosophy of topic models. In this way, we avoid analyzing terms in isolation. Instead, the approach encompasses the context of the words and the underlying semantic information. Second, we combine the interpretability results and the classification performance; thus, the system ranks second in the state-of-the-art.

Regarding the classification scheme, Küçük & Can (2020) report that two-phase classification is adequate to discriminate between *Favor*, *Against*, and *None* stances. In the first phase, the system differentiates between tweets without opinion (*None* label) and tweets with an opinion (*Favor* or *Against* labels). In the second phase, the classifier inputs tweets categorized as opinion in the previous phase, and the classification between labels *Favor* and *Against* occurs.

Some strategies to manipulate the high dimensionality of the feature vectors involve applying reduction techniques such as PCA. Nonetheless, Al-Ghadir et al. (2021) report a reduction in overall F-measure by applying PCA.

Classification schemes vary depending on whether stance detection is proposed as a target-dependent task. Some researchers implement a system considering the same feature vectors for all targets when they support stance detection as a target-independent task (Mohammad et al., 2016; Elfardy & Diab, 2016; Mohammad et al., 2017). This scheme has the advantage that information from other domains can enrich classes with few instances. Conversely, some studies declare that words and concepts from stance labels do not have the same meaning and intention across the targets (Tutek et al., 2016; Zhang & Lan, 2016). In this sense, Küçük & Can (2020) expose that "training separate classifiers for each stance target is a recommended practice" (p. 10) given the literature review on stance detection. Commonly, the classifier systems use models such as SVM, RF, LR, Gaussian Naïve Bayes, KNN variations, and ensemble models.

3. Methodology

Our proposal is target-dependent; hence we trained a scheme with a particular feature set per target. Moreover, our system has a two-phase classification scheme. As a result, there are two classifiers trained for each target (see the system architecture shown in figure 2). This section documents the tweets preprocessing, feature construction, and classification scheme. The evaluation metrics that we used are presented in section 4.

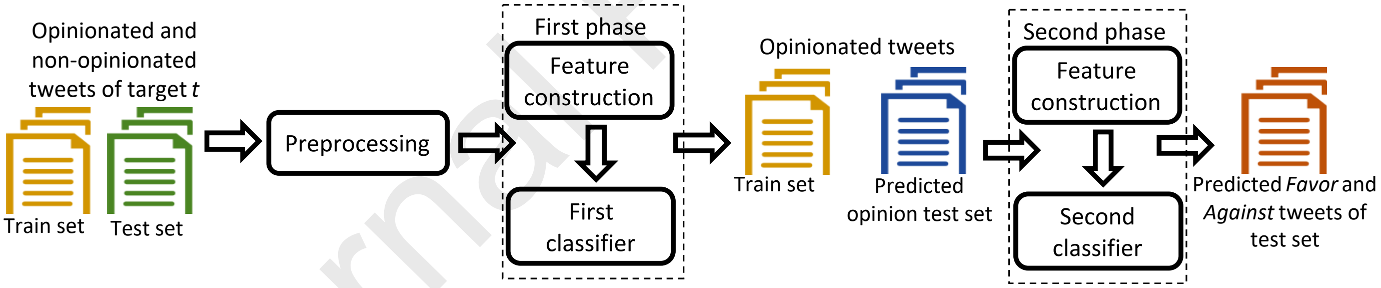


Figure 2: System architecture.

3.1. Preprocessing

Preprocessing consists of normalization and cleaning tweets regarding their impact on feature extraction. First, we normalized the tweets using a slang dictionary[4]. Thus, colloquial written expressions as *aiight* was replaced for *all right*. This strategy was also implemented by Dey et al. (2017). Second, we removed any non-alphabetic character in the English language. For instance, *@billclinton* became *billclinton*, and *#WhyImNotVotingForHillary* to *WhyImNotVotingForHillary*. Third, we split the concatenated expressions from hashtags and usernames. Particularly, we used Word ninja package[5]. In this way, *billclinton* turned into *bill clinton*, and *WhyImNotVotingForHillary* emerged as *Why Im Not Voting For Hillary*. Our fourth step was stopwords removal to eliminate none-informative content for the task. The stopwords used were a refined list of the available English inventory in Google code project[6]. Finally, we performed stemming through Porter stemmer and normalized the text to lowercase.

It is important to mention that some features were built with variations of preprocessed tweets. That is, omitting any preprocessing step to model the tweet-specific content or to have a common basis for comparing the tweet content with external lexicons.

[4] Slang dictionary: `https://www.noslang.com/dictionary/`

[5] Word ninja package: `https://github.com/keredson/wordninja`

[6] Stopwords list: `https://code.google.com/archive/p/stop-words/`

3.2. Feature construction

We propose features to model the linguistic content of tweets and encode opinion toward the target. We suggest as the main strategy exploiting topic composition retrieved with BTM topic modeling. Additionally, we used a set of attributes from previous research. For instance, our system uses features such as tweet-specific content, sentiment and subjectivity markups, target attributes, term distribution, and word embeddings.

Feature	First phase	Second phase
Topic modeling	Opinion codification All training tweets	Majority stance codification Minority stance codification Opinionated tweets
Tweet-specific content		Hashtags vector Usernames vector
Sentiment and subjectivity markups	Stanza vector Vader vector MPQA boolean feature Adjective boolean feature	
		MPQA weighted feature Adjective feature SentiWordNet vector
Target attributes	Target label vector *N-grams* and *n-chars* target label feature	
Term distribution	Weighted features	
Word embeddings	BERTweet vector	

Note. The input to the first phase was all training tweets. Alternatively, the opinionated tweets (*Favor* and *Against* labels) were modeled in the second phase. We **bold** the feature names in the rest of the document.

Table 3: Summary of constructed features.

Consequently, we did not use lexical information or label-based features. The former did not show an improvement in performance that would compensate for the increased dimensionality of the training matrix. As mentioned in section 2.2, the latter involves a high manual labeling cost and restricts the proposal's applicability to other domains. Therefore, we did not implement these features.

Table 3 summarizes the constructed features for each classification phase, as shown in figure 2. Below, we describe how we built each feature.

3.2.1. Topic modeling

Topic modeling considers the existence of latent topics through the corpus. The latent topics are represented by sets of words within the texts learned without supervision. In particular, topics are term abstractions to capture the corpus content beyond the words in each document. Hence, latent representation leads to better generalization and the possibility of making inferences from unseen data (Thonet et al., 2016).

BTM is an appropriate topic model to analyze documents with sparse content such as tweets (He et al., 2020). This characteristic arises because BTM identifies topics by directly coding the co-occurrence of words, *i.e.* BTM inspects the corpus as a set of *bigrams* or *biterms* (Cheng et al., 2014). As a result, two words are in the same topic when they co-occur frequently. In this sense, BTM assumes that *K* topics are expressed over *W* words of the vocabulary. Let $z \in [1,K]$ be a topic indicator variable. The probability of *z* topic in the corpus (*i.e.* $P(z)$) can be modeled by the multinomial distribution $\Theta = \{\theta_z\}_{k=1}^{K}$.The word distribution for topics (*i.e.* $P(w \,|\, z)$) is represented by Φ matrix with $K \times W$ dimensions where its *z*th row ϕ_z is a multinomial distribution with entries $\phi_{k,w} = P(w \,|\, z = k)$. Cheng et al. (2014) use symmetric Dirichlet priors for Θ and ϕ_k with single-valued hyperparameters α and β, respectively. Hence, BTM model hyperparameters are K, α, and β.

We used BTM to retrieve the topic structures of the training tweets from a specific target t. Particularly, we implemented the code available from the model authors[7]. The hyperparameters values were $\alpha = 0.05$, $\beta = 0.08$ since Cheng et al. (2014) state that these values ease building coherence topics in similar databases[8]. We varied the number of topics (n_k) between $\{3, 5, 7\}$.

Our proposal has two different aspects from previous works (Mourad et al., 2018; Zhang & Lan, 2016; Elfardy & Diab, 2016) that employ topic modeling during stance detection in *SemEval-2016 task 6* dataset. In the following, we detail each

[7] Source code of BTM: `https://github.com/xiaohuiyan/OnlineBTM`

[8] Question collection from a popular Chinese QA website and Tweets2011 dataset: `http://trec.nist.gov/data/tweets/`

of the differences.

Target:	Atheism	(example **a**)
Stance:	Against	
Tweet:	Prayer is a gift. Trusting and walking in it is a blessing. #Yeshua #GetToKnowHim #RelationshipIsKey #KeepWalkingByFaith	
Preprocessed tweet:	prayer gift trust walk bless yeshua get know relationship key keep walk by faith	
Preprocessed and encoded tweet:	prayer gift trust walk yeshua key walk faith	
Target:	Atheism	(example **b**)
Stance:	Favor	
Tweet:	@ChooseToBFree @xfranman @NBCNews Thomas Jefferson & George Washington did not believe in an afterlife, Thomas Paine was atheist	
Preprocessed tweet:	choos b free fran man nbc new thoma jefferson georg did not believ afterlif thoma pain atheist	
Preprocessed and encoded tweet:	choos b nbc thoma jefferson washington afterlif thoma atheist	

Figure 3: Examples of preprocessed and transformed tweets after **Opinion codification**. Terms in blue are in the vocabulary from tweets with *None* label, and then they are removed.

Input to topic model: tweet codification

Unlike previous works, we experimented with the representation of tweets to highlight the terminological differences between the stance labels. Our premise is to establish the terms that support describing the stance toward a target. Thus, tweet codifications aim to emphasize the information that the system considers during classification. Specifically, we generated the following codification considering that D_t is the set of training tweets from the target t and S is the set with the stance labels of interest.

3.2.1.1. Opinion codification

We fixed D_t as the set of all training tweets from the target t, and $S = [Favor,\ Against,\ None]$. Then, we built two vocabulary lists considering the preprocessing activities proposed in section 3.1. One list for the opinionated tweets (labeled as *Favor* or *Against*) and another for the tweets with *None* label. Then, we filtered the list of opinionated tweets to eliminate terms in both vocabularies. Consequently, the filtered list of opinionated tweets did not contain the words from tweets with the *None* label.

We used this list to modify D_t. In particular, we transformed the texts to contain only the words from the generated list. Thus, tweets without opinion were removed, and the content of tweets with opinion was reduced. Our proposal aims to recognize the terms that express an opinion. For instance, figure 3 presents two examples of **Opinion codification** regarding the *Atheism* (A) target.

The preprocessed and encoded tweet from example **a** consists of words that are directly related to *Against* stance. Alternatively, example **b** has *atheist* as term which could be a signal to clarify *Favor* stance, but the other terms require a context to discriminate the label (*i.e.* it is necessary to know that *thoma jefferson* was not a fervent Christian to set *Favor* label). Moreover, removing the expression *did not believ* makes it challenging to recognize the standpoint of the persons mentioned in the text and its impact on tweet stance. Hence, **Opinion codification** generated that the tweet in example **b** was made up of words related to having an opinion about the target, but it is not clear if it is a *Favor* or *Against* stance. Therefore, we presumed that our codification proposal is appropriate to build a feature focused on the first phase of classification (*i.e.* discriminating between tweets with and without opinion).

3.2.1.2. All training tweets

In this form of tweet codification, we used the original tweets as input to the BTM, thus employing the preprocessed tweets of D_t without any transformation or terminological filtering. Furthermore, we specified $S = [Favor,\ Against,\ None]$ then all the training tweets were used. For this reason, we exploited this representation during first phase classification. This tweet codification has been explored in previous works. Nonetheless, we propose a new approach to manipulate topic model results during feature construction.

3.2.1.3. Majority stance codification

We set D_t as the train set of opinionated tweets, then $S = [Favor,\ Against]$. Specifically, this coding was appropriate for classifying the second phase, as it allowed highlighting the terminology of the tweets with the opinion of the majority class in each target. We examined the label distribution in table 1 to recognize the majority class per target. For instance, *Favor* is the majority label of *Climate Change is a real Concern* (CCC) target.

Target:	Climate Change is a Real Concern	(example **a**)
Stance:	Favor	
Tweet:	As the world emitted CO2 water vapour built up in the atmosphere and caused the Lord to send the flood in retribution #bible	
Preprocessed tweet:	as world emit co water vapour built up atmospher lord send flood retribut bibl	
Preprocessed and encoded tweet:	as world emit water vapour built up atmospher lord send flood retribut bibl	
Target:	Legalization of Abortion	(example **b**)
Stance:	Against	
Tweet:	Rise & Shine its a new day & you're alive. Thank God 4 another day of precious life. #Christian #Catholic #TeamJesus	
Preprocessed tweet:	rise shine new day you aliv thank god anoth day preciou life christian cathol team jesu	
Preprocessed and encoded tweet:	rise shine aliv preciou christian cathol team jesu	

Figure 4: Examples of preprocessed and transformed tweets after **Majority stance codification**. Terms in blue are in the vocabulary from tweets with minority stance label, and then they are removed.

Like the **Opinion codification**, we constructed two vocabularies regarding *Favor* and *Against* labels. Thereby we refined the list related to the majority class to eliminate the terms included in the other list (*i.e.* minority class vocabulary). We exploited the adjusted vocabulary to alter D_t. Hence, the tweets from the minority class were discarded. We also reconstructed the tweets of the majority class using the adjusted vocabulary words. Figure 4 shows two examples of **Majority stance codification**.

These examples illustrate that few words were removed during majority class codification for some targets. This is consistent with the distribution of labels in *SemEval-2016 task 6* dataset. Our proposed coding enhances the elimination of redundant words even when there is a small size of the minority class vocabulary. In example **b**, words such as *new* and *anoth* that provide little information about the tweet stance were dropped.

3.2.1.4. Minority stance codification

Likewise, we propose a tweet representation to identify the terms in tweets labeled with a minority stance toward a specific target. Thus, we only consider the minority class terminology. We applied the same D_t and S list from the previous codification. Hence, we designed this proposal to highlight the words of the minority class during the second phase of classification. We followed steps pretty similar to those described in the **Majority stance codification**. The difference was that the adjusted list corresponded to the stance label with the lowest number of tweets.

Figure 5 exposes examples from this codification. We found two characteristics of our proposal. First, minority class tweets tend to have unique terminology, leading to low-frequency words within the dataset. For example, our proposed coding almost did not remove the terms in example **a**, as most of its words appear exclusively in this tweet. Hence, we hypothesized that our coding would produce topics that describe the particular content of tweets rather than generalize topics across them. In the second result, example **b** exhibits that our proposal alternatively generates a sparse representation of tweet when it is composed of words used in the majority stance label. Therefore, we suspect that the content of these tweets would not be represented in the retrieved topics. This could imply a failure to discriminate the tweet stance and a weakness in our proposal.

3.2.1.5. Opinionated tweets

We established D_t as the preprocessed tweets with *Favor* and *Against* labels ($S = [Favor,\ Against]$) to retrieve topics per target. In particular, we exploited this set without any special codification to create features that captured the content of opinionated tweets. We expect these features to be composed of common terms between the stance labels. To the best of our knowledge, this simple tweet representation has not been explored for topic construction in *SemEval-2016 task 6* dataset.

Information for features construction: top terms from topics

Section 2.2 reports that the previous studies (Zhang & Lan, 2016; Elfardy & Diab, 2016) exploited term-topic or document-topic matrix from topic models. On the contrary, we propose to use topic composition to recognize the most relevant words within the content of the tweets, considering the presented codifications. Our idea is similar to the implementation of Zhang & Lan (2016); Mourad et al. (2018) since the authors compute the similarity between top terms in topics and tweet words. Nonetheless, we used the top terms to build weighted vectors for tweet representation.

Target:	Hillary Clinton	(example **a**)
Stance:	Favor	
Tweet:	Let's bring prosperity to the line workers & to the kindergarten teachers!!!- @amyklobuchar #inspiredbyhillary	
Preprocessed tweet:	bring prosper line worker kindergarten teacher myk lob inspir by hillari	
Preprocessed and encoded tweet:	bring prosper kindergarten teacher myk nspir	
Target:	Feminist Movement	(example **b**)
Stance:	Favor	
Tweet:	Did you know? Gender stereotypes as we know then developed with beginning of the 18. century. #gender	
Preprocessed tweet:	did you know gender stereotyp as know then develop centuri gender	
Preprocessed and encoded tweet:	stereotyp centuri	

Figure 5: Examples of preprocessed and transformed tweets after **Minority stance codification**. Terms in blue are in the vocabulary from tweets with majority stance label, and then they are removed.

In particular, we consider $K_{t,c}$ as the set of topics retrieved from the training tweets of the t target following the c codification. Each $k = [w_1, w_2, \cdots, w_W]$ topic from $K_{t,c}$ is a list with the W terms that compose the vocabulary of training tweets. The W words are arranged regarding the probability that the terms belong to the k topic (*i.e.* $\phi_{k,w} = P(w \mid z = k)$) in such a way that the top terms (with the highest probability) are at the top of k. Subsequently, we selected the n top terms from each $k \in K_{t,c}$ and created R as the set with these words. Thus, duplicates are not allowed.

R size varies according to the number of topics retrieved (n_k), the n number of the top terms captured, and tweet codification (c). For instance, R had a higher number of words through **Minority stance codification** than **Majority stance codification** (see table 4). This is a consequence of the characteristics described for **Minority stance codification**, particularly, the presence of terms with low occurrence in the coded tweets. We experimented with $n = \{10,\ 15\}$ Table 4 shows that, considering the vocabulary of **Opinionated tweets**, the text with the *Against* stance has a higher number of terms with a value different from zero compared to the tweet with *Favor* label.

		Majority stance codification		**Minority stance codification**				**Opinionated tweets**			
Stance	**Preprocessed tweet**	*dnc*	*lb*	*captor*	*lon*	*longer*	*z*	*babi*	*have*	*kill*	*pro*
Against	agre that not ok kill lb babi uteru dw s tweet dnc clinton hillari for i pro compromis	$tf - idf_{d,w}$	$tf - idf_{d,w}$	0	0	0	0	$tf - idf_{d,w}$	0	$tf - idf_{d,w}$	$tf - idf_{d,w}$
Favor	m lon z life new hq captor have been born no longer part bodi pregnant woman that s differ	0	0	$tf - idf_{d,w}$	$tf - idf_{d,w}$	$tf - idf_{d,w}$	$tf - idf_{d,w}$	0	$tf - idf_{d,w}$	0	0

Note. The terms related to the tweet codifications are examples of top terms from topics. $tf - idf_{d,w}$ represents *tf-idf* weighting of the term w in the tweet d. The tweets are related to the target *Legalization of Abortion* (LA).

Table 4: Example representation of tweets regarding top terms from topics during the second phase.

Therefore, we built a different R list for each $K_{t,c}$. We presume R would contain words to explain the tweet's content regarding codification (c). To illustrate, we would identify the particular terms expressing the *Against* stance (toward *Atheism* (A) target) when we inspect R from K topics of atheism with **Majority stance codification** (c). Our hypothesis arises from the fact that we exploited the distribution probability of terms in topics since we recognize that not all words provide the same contribution to expressing a standpoint. Hence, we seek to provide an explanation of the information that the system considers discriminating the stance of tweets. Moreover, we present textual data from tweets to elucidate a qualitative understanding of classification system behavior. Thus, our proposal is a data-based approach that can easily adapt to the target information.

Furthermore, we use the terms from R to build feature vectors following a *tf-idf* weighting scheme. We set D_t (*i.e.* preprocessed d tweets from t target) as the corpus and w words from R as the vocabulary to compute weighted vectors.

Primarily, we implemented Eq. 1, where $f_{d,w}$ is the frequency of term w in tweet d, $|D_t|$ the number of tweets in the corpus, and n_d is the number of tweets that contain w. Eq. 1 corresponds to the implementation of *tf-idf* from Scikit-learn library.

$$tf - idf_{d,w} = f_{d,w} * \left(\log\left(\frac{1+|D_t|}{1+n_d}\right) + 1\right) \qquad (1)$$

It is essential to clarify that D_t represents all the training tweets (regarding the three stance labels) during the first phase. Meanwhile, in the second classification phase, D_t represents the opinionated tweets.

Table 5 shows an example of tweet representation regarding the codifications in the first phase. The terms of the **Opinion codification** have a zero score for the tweet with the *None* label. The vocabulary of **All training tweets** poorly illustrates the tweet without opinion (row 1), contrary to the tweet with a stance *Against* the target *Legalization of Abortion* (LA) (row 2). It occurs because *Against* label is the majority class of the target LA (see table 1). Therefore, the respective terms may have a larger presence in the topics.

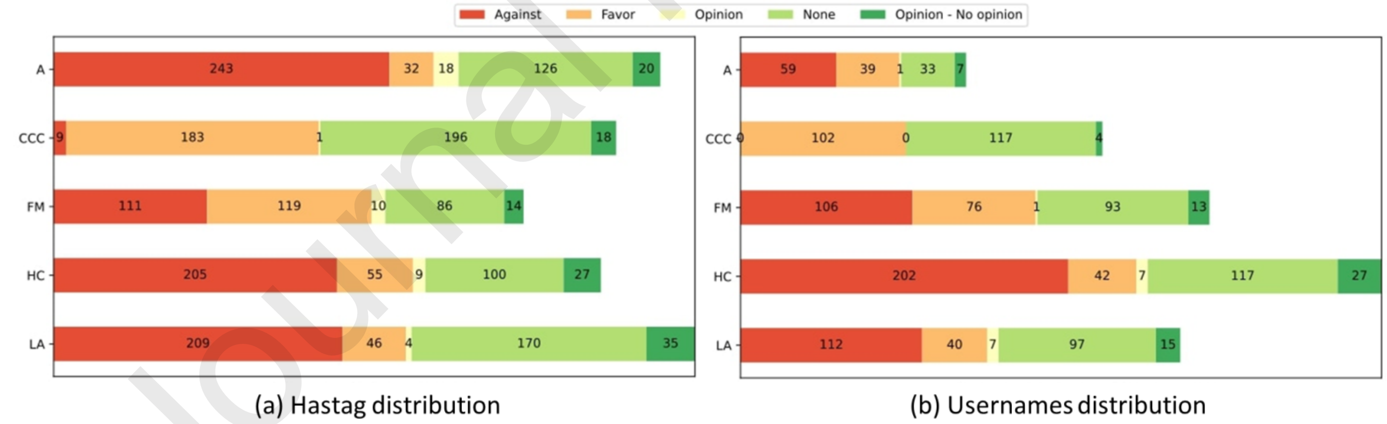
		Opinion codification		**All training tweets**		
Stance	**Preprocessed tweet**	*babi*	*concept*	*for*	*life*	*on*
None	on manila tv monday pac quia appeal indonesia presid dont execut mari jane velo so death row drug traffick christian	0	0	0	0	$tf - idf_{d,w}$
Against	prayer for babi urgent prayer detroit mi san diego one san antonio life concept t	$tf - idf_{d,w}$	$tf - idf_{d,w}$	$tf - idf_{d,w}$	$tf - idf_{d,w}$	0

Note. The terms related to tweet codifications are examples of top terms from topics. $tf - idf_{d,w}$ represents *tf-idf* weighting of the term w in the tweet d. The tweets are related to the target Legalization of Abortion (LA).

Table 5: Example representation of tweets regarding top terms from topics during the first phase.

3.2.2. Tweet-specific content

We analyzed the distribution of hashtags and usernames in training tweets regarding the stance labels. In particular, we examined the tweets normalized with slang dictionary. Thus, the subsequent preprocessing steps were not applied. Figure 6 shows hashtags and usernames appearing in each stance tag and the merged tags.



(a) Hastag distribution

(b) Usernames distribution

Figure 6: Information of tweet-specific content in the train set. *Opinion* label shows computes the number of attributes simultaneously present in tweets with *Favor* and *Against* labels. *Opinion - No opinion* is the number of characteristics that are in tweets of the three stance labels. *Favor* represents the number of attributes included in the tweets with a *favor* stance. Meanwhile, *Against* counts the number of characteristics in tweets with *against* label. (**a**) Hashtag distribution. (**b**) Usernames distribution.

Notably, hashtags and usernames are more frequent in tweets with a majority stance label and *None* label per target. Consequently, exploiting these attributes during the first phase of classification would not have allowed discrimination between opinion tweets and non-opinion tweets. Alternatively, the hashtags and usernames distributions are different between *Favor* and *Against* labels. This indicates an opportunity to use these features to distinguish tweets during the second phase. Therefore, we built tweet-specific content markups with hashtags and usernames in the second classification phase.

3.2.2.1. Hashtags vector and Usernames vector

We generated a feature vector considering hashtags (**Hashtags vector**) and another one for usernames (**Usernames vector**). We set D_t as the training tweets with *Favor* and *Against* labels. We executed the following process:

1. We defined the set $B = [b_1,b_2,\cdots,b_n]$ of the respective attribute (*i.e.* hashtags or usernames) regarding the textual content of D_t in both cases.
2. To each $b_i \in B$, we established whether b_i appeared in tweets with *Favor*, *Against*, or both labels (*i.e.* when the attribute occurred simultaneously in *Favor* and *Against* tweets).
3. We evaluated for $d \in D_t$ whether d contained any $b_i \in B$. In this way, we computed the number of attributes related to each stance label or both tags.

We assigned a value of 1 (one) when the majority of attributes in d appeared in *Favor* tweets or a value of 2 (two) if the tweet had a predominant presence of attributes associated with *Against* label. Conversely, we assigned a value of 0 (zero) when the tweet did not have any b_i. In addition, zero emerged when the attributes related to both labels were predominant or the tweet had an equal proportion of characteristics of stance labels.

Hashtags vector and **Usernames vector** had $|D_t| \times 1$ dimensions each one. Moreover, our tweet-specific content features differ from previous work where the authors' coded *n-gram* hashtags or usernames frequency (Mourad et al., 2018; Tutek et al., 2016; Zhang & Lan, 2016; Wojatzki & Zesch, 2016), as mentioned in section 2.2. Alternatively, we attempted to assign a value to each tweet based on its content, and the distribution of hashtags and usernames in the train set.

3.2.3. Sentiment and subjectivity markup

These linguistic markups facilitate stance detection when ideological position influences the sentiment and subjectivity toward the target (Elfardy & Diab, 2016; Dey et al., 2017). For instance, 66.74% of training tweets with *Against* stance have a *Negative* sentiment label in *SemEval-2016 task 6* dataset (Mohammad et al., 2017). Moreover, 95.44% of training tweets with an opinion (*Favor* or *Against* labels) present a *Negative* or *Positive* sentiment tag[9]. Therefore, we built features to provide a score to interpret the sentiment and subjectivity of tweets to improve stance detection during the first and second phases of classification. Table 6 shows examples of these linguistic markers.

3.2.3.1. Stanza vector

We sought to use an unsupervised technique for sentiment analysis. We explored the available model of Stanza library[10]. This model is a CNN trained in different data sources, including the Airline Twitter Sentiment[11]. Moreover, it supports negative, neutral, and positive labels represented in 0, 1 and 2 scores for each sentence within the text.

Stanza vector was constructed with D_t as the tweets normalized with slang dictionary and avoiding subsequent preprocessing steps. As mentioned above, we respected the tweet sets established for the first and second phases. Stanza's sentiment analysis sometimes provided more than one score for each tweet, as the model found multiple sentences in the tweet. We assigned the most frequent score within the tweet, and in case of a tie, we allocated the value of one.

3.2.3.2. Vader vector

We also exploited VADER, a rule-based sentiment analysis lexicon specifically adapted to expressions from social media. A **Vader vector** was built for the first and second phases considering D_t as the tweets set without any preprocessing (as VADER considers the traits from social media like slang, acronyms, etc.). We used this model from NLTK module and manipulated the polarity score that the model gives. Essentially, we encoded the sentiment labels provided by the model as $negative = 0$, $neutral = 1$, and $positive = 2$. We designated the value of 1 (one) when VADER rated more than one sentiment label with the same score.

3.2.3.3. MPQA boolean feature and MPQA weighted feature

Likewise, we explored how previous works examine sentiment and subjectivity during stance detection. In particular, we analyzed the work of Dey et al. (2017), which ranks third in the state-of-the-art with a two-phase classification scheme and a machine learning approach. For these reasons, this study is similar to our proposal. We modified Dey et al. (2017) codifications, although we sought to enrich our work with the lessons learned.

[9] SemEval-2016 dataset description: `http://www.saifmohammad.com/WebPages/StanceDataset.htm`

[10] Stanza package: `https://stanfordnlp.github.io/stanza/sentiment.html`

[11] Airline Twitter Sentiment dataset: `https://data.world/crowdflower/airline-twitter-sentiment`

We used the MPQA subjectivity-polarity lexicon to construct features to detect the subjectivity of tweets during the first and second classification phases. In this sense, we set up D_t as the tweets with all preprocessing steps except stemming. **MPQA boolean feature** was generated to discriminate between opinionated and non-opinionated tweets. We implemented a coding similar to Dey et al. (2017), where a token matched to the strong subjective set was assigned a value of 2 (two), while weak subjective words were scored with a value of 1 (one). In addition, if the token was related to the positive polarity, the indicated score was positive, while the term identified with the negative polarity received a negative score. We found a score for each tweet considering the sum of the values assigned to its terms. **MPQA boolean feature** facilitated the discernment between subjectively or non-subjectively inclined tweets, whereby tweets with values greater than +2 or less than -2 were associated with subjectively inclined tweets. For the second phase of classification, we followed the study of Dey et al. (2017) and generated **MPQA weighted feature** as the result of adding up the scores given the tweet content.

		Sentiment and subjectivity markup						
	Tweet content	Stanza vector	Vader vector	MPQA boolean feature	MPQA weighted feature	Adjective boolean feature	Adjective feature	SentiWordNet vector
Target: Atheism **Stance:** Favor **Sentiment:** Negative	@latikia @seangillies Yeah, right? Lol wink wink nudge nudge. The greatest part is how "God's" playing along! #religiondoesharm	Positive	Neutral	1	10	1	greatest	0
Target: Atheism **Stance:** Against **Sentiment:** Positive	heavens and the earth were one connected entity, then We separated them And We havemade from water every living thing quran 21:30	Neutral	Neutral	1	2	1	every	0
Target: Atheism **Stance:** None **Sentiment:** Positive	Just watched the #SCfeatured segment on the life of Manny Pacquiao. If I wasn't on #TeamPacquiao already, I would be now. #respect	Negative	Neutral	1	9	0	-	0
Target: Hillary Clinton **Stance:** Favor **Sentiment:** Positive	@Jeff_Nichols_82 @HillaryforIA #freedom_justice_equality_education for innovation and development to make #happy_life in Utopia	Positive	Positive	1	10	1	happy	1
Target: Hillary Clinton **Stance:** Against **Sentiment:** Negative	#Hillary is as transparent as a brick wall #LibertyNotHillary	Negative	Neutral	1	2	1	transparent	-1
Target: Hillary Clinton **Stance:** None **Sentiment:** Neutral	@Temp15544 @icedecay @BrooklynJuggler @classic_mouth In this context, to all practical extents and purposes? It means	Neutral	Neutral	1	-2	1	practical	-1

Note. The stance and sentiment labels correspond to the reference tags. The **MPQA weighted feature** and **SentiWordNet vector** columns show the sum of the weights of the valued terms. For instance, in the third row, the value 9 is the sum of the weights *[2,1,1,0,2,2,1]*, meaning that six out of seven terms from the respective tweet appear in the MPQA subjectivity-polarity lexicon. The column of the **Adjective feature** exhibits the tweet term (or terms) whose adjective category is indexed in WordNet. The mark (-) denotes that the tweet does not contain adjectives indexed in WordNet.

Table 6: Example of **Sentiment and subjectivity markups** on different tweets.

3.2.3.4. Adjective boolean feature and Adjective feature

Besides, we constructed an **Adjective boolean feature** to evaluate the presence of adjectives in the content of the tweets since sentences that offer an opinion tend to use qualifiers to judge the target of interest. We applied the proposal of Dey et al. (2017) to use WordNet to assess whether a term possess the category of adjective. Especially we made a list of all words classified as adjectives (even if they had other tags) in the lexicon. To this list, we added the stemmer of the previously selected terms. D_t represented the preprocessed tweets. Then, we assessed if any token of the tweet was in the generated list. Dey et al. (2017) employed the **Adjective boolean feature** during the first phase of classification.

Additionally, we propose a modification to Dey et al. (2017) work. We constructed an **Adjective feature** for the second phase of classification. Unlike the boolean vector, we generated a boolean matrix where each column is one of the elements of the list created with WordNet information. The matrix counts whether a particular adjective appears in a tweet. In this sense, we discriminate what kind of adjectives are used to express *Favor* or *Against* stance toward a target.

3.2.3.5. SentiWordNet vector

Finally, we built a **SentiWordNet vector** to provide a sentiment score like Dey et al. (2017) for the second phase of classification. We established D_t as the tweets set with *Favor* and *Against* labels. We applied slang normalization, removal of non-alphabetic characters, and splitting of concatenated expressions in the modeled tweets. SentiWordNet from NLTK was used. We took the information of the first synset of the terms within each tweet and then identified the scores regarding objective, positive and negative tags. If the token had a higher positive score, we assigned it a +1 value. On the contrary, we allocated -1 when the word had a dominant negative score. In case of a tie between positive and negative scores or when the objective score was higher, a value of zero was attributed. Subsequently, we rated the sentiment of each tweet, summing up the values associated with its terms.

Table 6 presents examples of the linguistic markers on different tweets. For these examples, **Stanza vector** tends to predict the correct sentiment label compared to **Vader vector**; it may occur due to the limitations of the lexicon-based approach. Nevertheless, both markups fail to recognize the tweet sentiment in target *Atheism* (A). We hypothesize that these features do not benefit stance detection at target A. The examples in table 6 show the pertinence of our approach where stance detection is a target-dependent task. For instance, **Adjective boolean feature** provides a score to discriminate the tweets with and without an opinion given the target A, but it fails to target *Hillary Clinton* (HC). Comparably, **SentiWordNet vector** brings information to differentiate *Favor* and *Against* stances for target HC, whereas it is unsuccessful for target A.

We suspect **MPQA boolean feature** is not suitable for stance detection in target A and HC, as table 6 illustrates. **MPQA weighted feature** and **Adjective feature** facilitate distinguishing *Favor* or *Against* stance of the tweets since they evaluate the degree of subjectivity and sentiments. Therefore, we presume these markups enhance the classification during the second phase.

3.2.4. Target attributes

It is relevant to model the relationship between the target and the tweet. Some tweets do not explicitly mention the target tag; instead, they express an opinion on a topic related to the target, which allows inferring the stance label (Mohammad et al., 2017).

3.2.4.1. Target label vector, N-grams and n-chars target label feature

As in previous studies (Dey et al., 2017; Mohammad et al., 2017), we generated a boolean feature to evaluate the presence of the target's name in the tweet (**Target label vector**). Additionally, we built a set of boolean features to assess the presence of *n-grams* and *n-chars* of the target name in each tweet (**N-grams and n-chars target label feature**). The length of the *n-grams* varied between two and three, while *n-chars* were sought in the range of two to four. We considered D_t as the set of preprocessed tweets. These vectors were implemented during both phases.

3.2.5. Term distribution

Term distribution brings the semantic representation of a tweet through the frequency of the words.

3.2.5.1. Weighted features

We proposed **Weighted features** for both classification phases with D_t as the set of preprocessed tweets. Primarily, we constructed a document-term matrix with *tf-idf* weighting scheme. The term columns emerged from the vocabulary of preprocessed training tweets.

3.2.6. Word embeddings

Some research (Wani et al., 2021; Ghosh et al., 2019) uses BERT or a variation of it to convert the tweet into a vector regarding the sentence structure.

3.2.6.1. BERTweet vector

We codified the tweets regarding BERTweet (Nguyen et al., 2020), that is, word embeddings pre-trained for English tweets. These vectors were trained with +850M tweets (16B word tokens) following the same architecture as BERT and perform better than word embeddings trained on general corpora during tasks such as Twitter Sentiment Analysis and irony detection in tweets. In particular, we set D_t as the preprocessed tweets before stemming. We converted every token to a vector of 1×768 dimensions for each tweet using BERTweet. Out-of-vocabulary words were discarded. Next, we created a new vector (**BERTweet vector**) by concatenating the mean and standard deviation of the word vectors related to the tweet.

3.3. Classification scheme

As mentioned above, we proposed a two-phase classification scheme in which two classification models were trained per target. In particular, we trained the classifier of the first phase with all the tweets set to predict tweets expressing opinions considering the test set. For the second phase, we trained the model with opinionated tweets of the train set and evaluated the classifier in the output of the first phase (see figure 2).

We experimented with SVM, LR, ET, and AB classification models in each phase. We calibrated the classifiers through an exhaustive search over specified parameters. For SVM, we modified the value of the regularization parameter and the kernel. LR utilized *liblinear* algorithm and varied the value of the regularization parameter. For ET, we searched the number of estimators and varied between entropy and gini to evaluate the quality of the splits. AB used decision trees as the base estimator, and we varied the number of estimators and learning rate. Moreover, we did the calibration using 5-fold cross-validation on the training data; consequently, we adjusted the system's parameters in a development set.

Some of the features used tended to have high dimensionality. For this reason, we applied PCA as a dimensionality reduction strategy during the first phase. We experimented with PCA and RFE algorithms for the second phase since the former reduced the classification performance for some targets (see table 14). We adjusted the number of components to retain in PCA regarding the amount of variance explained. Alternatively, we used the weights trained in a LR as an input to RFE. Furthermore, we exhaustively search for the number of components to retain and the number of features to be selected in RFE.

4. Results and analysis

Our proposal is for stance detection. We used the evaluation metric of *SemEval-2016 task 6* challenge. In this sense, we applied the macro-average of the F1-score for *Favor* and *Against* labels.

$$F_{avg} = \frac{F_{favor} + F_{against}}{2} \qquad (2)$$

Where F_{favor} and $F_{against}$ are calculated as:

$$F_{favor} = \frac{2P_{favor}R_{favor}}{P_{favor} + R_{favor}} \qquad (3)$$

$$F_{against} = \frac{2P_{against}R_{against}}{P_{against} + R_{against}} \qquad (4)$$

Where P and R stand for Precision and Recall, respectively, we computed the F_{avg} to each target during development. Overall F-measure is calculated with the averages of F_{favor} and $F_{against}$ between the targets. We use the overall F-measure to compare our proposal with other studies since it is the metric used in the competition. In the next section, we present the feature selection and the resulting classifier systems for each phase. The experimental results compare our performance with the studies mentioned in section 2. Finally, we show how our proposal aims to provide information regarding a plausible explanation of stance labels. It is worth mentioning that the standard error of these metrics in the development and testing set lies in the order of e^{-16}. Therefore, we do not report confidence intervals and claim that each change is statistically significant. All the schemes and approaches described in section 3 were implemented in Python 3.7. The experiments were run on Google Colab with the basic configuration. We implemented BTM with the code from the model authors (Cheng et al., 2014). We used Stanza and NLTK libraries to build the **sentiment and subjectivity markups**. Furthermore, we utilized the Scikit-learn library to implement *tf-idf* weighting scheme and the classification models. We exhaustively experimented with the parameter values of the classification models. As the number of experiments was quite large, we do not include these results and only indicate the details of the best-performing models. The details of the models are in Appendix A.

4.1. Feature space for the first phase

Initially, we determined the features for each phase by taking the targets independently. Table 7 shows five groups that help to get the highest classifier results during the first phase. Recall that we varied the number of topics as $n_k = \{3, 5, 7\}$ and the number of top terms from each topic between $n = \{10, 15\}$ for **Opinion codification** and **All training tweets** features.

Groups from table 7 contain **Topic modeling** features to guide the discrimination across topic compositions. Also, we employed the **Term distribution** feature to represent the content of tweets regardless of latent topics. Group 1 aims to rate adjectives and BERTweet. Likewise, group 2 allows judging the contribution from BERTweet and target markups. Alternatively, group 3 involves sentiment features and enriched knowledge of the MPQA lexicon to study the influence of not exploiting adjectives. Group 4 represents a scenario for handling sentiment information and target attributes together.

Instead, group 5 includes all the sentiment and subjectivity markups to assess whether the linguistic information enhanced the first phase.

	Topic modeling		Sentiment and subjectivity markups				Target attributes		Term distribution	Word embeddings
	Opinion codification	All training tweets	Stanza vector	Vader vector	MPQA boolean feature	Adjective boolean feature	Target label vector	*N-grams* and *n-chars* target label feature	Weighted features	BERTweet vector
Group 1	✓	✓				✓			✓	✓
Group 2	✓	✓					✓	✓	✓	✓
Group 3	✓	✓	✓	✓	✓				✓	
Group 4	✓	✓	✓	✓			✓	✓	✓	
Group 5	✓	✓	✓	✓	✓	✓			✓	

Note. Topic modeling features varied considering the number of topics $n_k = \{3, 5, 7\}$ and the number of top terms from each topic $n = \{10, 15\}$.

Table 7: Feature groups for the first phase.

We implemented these feature groups by experimenting with the four classifier systems (*i.e.* SVM, LR, ET, and AB previously mentioned in section 3.3). Table 8 shows the F_{avg} obtained in the first classification phase by applying the feature groups outlined in table 7. These results represent the system performance in the development set, given that feature selection is a parameter. In addition, the classifier that provided each result is indicated next to the score with an acronym in parentheses.

	A	CCC	FM	HC	LA
Group 1	**87.39 (SVM)**	82.69 (LR)	76.79 (LR)	80.31 (ET)	77.57 (ET)
Group 2	81.59 (ET)	**87.59 (ET)**	75.80 (LR)	79.51 (LR)	79.36 (AB)
Group 3	81.09 (SVM)	82.13 (ET)	**87.19 (LR)**	82.92 (ET)	87.35 (AB)
Group 4	81.34 (ET)	85.66 (LR)	86.41 (LR)	**83.81 (ET)**	74.65 (AB)
Group 5	85.09 (SVM)	83.40 (ET)	77.15 (LR)	82.18 (ET)	**91.69 (AB)**

Note. The classifier used is indicated next to each result with an acronym in parentheses. The highest results per target are marked in **bold**.

Table 8: Results of feature groups in development set during the first phase.

We found the best discrimination for target *Atheism* (A) with an SVM considering a polynomial kernel of the third degree and a regularization parameter equal to 3. In particular, **Topic modeling** features were constructed with the information of 7 topics ($n_k = 7$) and the 10 top terms in each topic ($n = 10$). Table 8 allows us to recognize that the **Target attributes** are not relevant to classify opinionated and non-opinionated tweets related to target A since the system performance decreased in feature groups 2 and 4. Moreover, we identified that the **Adjective boolean feature** is the only sentiment and subjectivity markup that improve the classification.

Target *Climate Change is a real Concern* (CCC) has the best score in an ET classifier with 155 trees using entropy to assess the splits. Additionally, the topic features were generated with the 15 top terms ($n = 15$) of 5 topics ($n_k = 5$). Table 8 shows that this target has the highest score with feature groups 2 and 4. Thus, target CCC did not leverage the linguistic information; instead, the presence of **Target attributes** strengthened its scores. This is coherent with the dataset description, where 98.81% of tweets with the *None* label do not express an opinion toward the target CCC, and 82.85% of opinionated tweets declare an idea explicitly about CCC[12].

Target *Feminist Movement* (FM) presents the best performance in a LR classifier with a regularization parameter of 2. **Topic modeling** features were built with $n_k = 7$ and $n = 10$. We detected that the **Adjective boolean feature** did not contribute to this classification. When examining the composition of this feature, we found that 57.83% of FM training tweets do not contain adjectives. Furthermore, this attribute did not discriminate between tweets with opinion and *None* label, as about 50% of tweets in each label were not composed of any adjective.

We also observed that unsupervised techniques for sentiment analysis as the **Stanza vector** and **Vader vector** provided linguistic information to categorize opinionated and non-opinionated tweets in target *Hillary Clinton* (HC). The **Adjective boolean feature** diminished the F_{avg} score as it assigned the same value to all tweets. In the first phase, we recognized that sentiment and **Target attributes** are distinctive marks to classify tweets related to target HC. Consequently, we elicited the best score with feature group 3 regarding an ET model with 95 trees using the gini measure. Additionally, we constructed topic features setting $n_k = 5$ and $n = 15$.

[12] SemEval-2016 dataset description: `http://www.saifmohammad.com/WebPages/StanceDataset.htm`

Discrimination of opinionated and non-opinionated tweets in target *Legalization of Abortion* (LA) took advantage of linguistic features when using an AB model with 20 estimators and a learning rate of 0.1; we exploited the information of 3 topics ($n_k = 3$) considering their 15 top terms ($n = 15$). **Adjective boolean feature** improved tweets differentiation since there are differences in the presence of adjectives in opinionated and non-opinionated tweets. Moreover, **Sentiment and subjectivity markups** are decisive during the first phase of classification for target LA.

Our findings suggest that the **Adjective boolean feature** improved the performance in targets A and LA. Conversely, it reduced the score for the other targets. We noticed that **MPQA boolean feature** supported discrimination of opinionated and non-opinionated tweets in two out of five targets. These results are different from the work of Dey et al. (2017) since the authors claim that adjective information and MPQA data play a decisive role in the first phase of classification without considering the composition of each target. Likewise, we affirm that **Target attributes** enhanced the classification of tweets related to target CCC. Our affirmation is similar to the feature composition proposal of Zhang & Lan (2016) (see table 2); furthermore, we agree with these authors on using word embeddings during the first phase of classification considering the targets A and CCC.

Table 9 shows the ablation test considering the feature groups that provided the highest score per target. It is essential to clarify that the classifier related to the best performance for each target was used with no modifications when adding the features. In particular, we place *NA* to point out the features not used in certain groups following the information in table 7, and the white spaces correspond to the last feature implemented during the ablation test. For instance, 83.69% is the score for the test set related to target A when we trained a SVM model using only **BERTweet vector** (**Word embeddings**).

	A		CCC		FM		HC		LA	
	Dev	Test	Dev	Test	Dev	Test	Dev	Test	Dev	Test
All features	88.01	87.39	87.59	86.43	89.15	87.19	85.53	83.81	92.51	91.69
(-) Opinion codification	86.76	86.71	87.01	85.46	85.19	71.48	83.07	83.04	90.76	86.72
(-) All training tweets	86.60	86.19	86.49	84.37	84.99	63.85	82.51	82.60	90.39	83.84
(-) Term distribution	86.81	84.26	84.76	84.48	61.99	52.55	82.00	76.34	84.23	78.32
(-) Target attributes	*NA*	*NA*	85.11	81.82	*NA*	*NA*	55.84	48.92	*NA*	*NA*
(-) Sentiment and subjectivity markups	86.21	83.69	*NA*	*NA*						
(-) Word embeddings					*NA*	*NA*	*NA*	*NA*	*NA*	*NA*

Note. Dev: Scores in development set. Test: Scores in test set.

Table 9: Ablation test in the first phase of classification regarding F_{avg} score.

We noticed that **Word embeddings** are the main feature to classify the tweets of targets A and CCC since this attribute by itself provided values higher than 80%. Besides, we identified that **Term distribution** features produced results of more than 80% in targets HC and LA. This can be explained by the size of these sets (see table 1), where the weight of a term is different if it appears in opinionated and non-opinionated tweets.

Table 9 exposes a decrease between four and two points in the development set by eliminating the **Opinion codification** markup. Moreover, this reduction is even more significant when removing all topic features. Hence, **Topic modeling** features are crucial to discriminate between opinionated and non-opinionated tweets of targets FM, HC, and LA. Nonetheless, the topic features are not decisive when classifying tweets related to targets A and CCC.

We applied PCA to improve our classification results during the first phase. In particular, we implemented PCA in the feature matrix built with the feature group that provides the highest score per target (see table 8). We experimented with all the classifier models (*i.e.* SVM, LR, ET, and AB regarding the variations mentioned in section 3.3). Table 10 shows the results with and without PCA per target.

	A		CCC		FM		HC		LA	
	Dev	Test	Dev	Test	Dev	Test	Dev	Test	Dev	Test
without PCA	88.01 (SVM)	87.39 (SVM)	87.59 (ET)	86.43 (ET)	89.15 (LR)	87.19 (LR)	85.53 (ET)	83.81 (ET)	92.51 (AB)	91.69 (AB)
with PCA	**90.91 (ET)**	**89.28 (ET)**	**89.70 (ET)**	**89.86 (ET)**	**92.35 (LR)**	**90.02 (LR)**	**95.00 (AB)**	**93.01 (AB)**	**94.92 (AB)**	**93.50 (AB)**

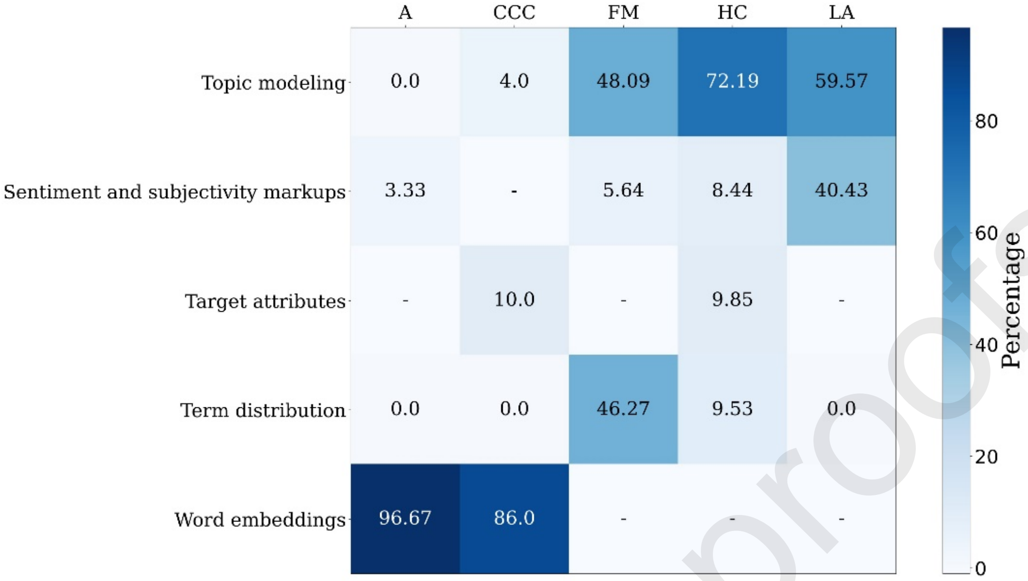
Note. Dev: Scores in development set. Test: Scores in test set. The classifier used is indicated next to each result with an acronym in parentheses. The highest results per target are marked in **bold**.

Table 10: F_{avg} scores with and without PCA in the first phase of classification.

It is clear that PCA improves classification results in all targets. This finding differs from Al-Ghadir et al. (2021), where authors report that PCA as a dimensionality reduction technique did not increase F_{avg} scores. Nonetheless, our proposal is divergent from the above-mentioned since we applied a two-phase classification scheme considering stance detection as a target-dependent task. Moreover, Al-Ghadir et al. (2021) tend to express that the feature matrix has low dimensionality, although in their work, it is not clear how the attribute vectors were constructed.

We analyzed which features had the highest weight in the components constructed with PCA. In particular, we selected 10% of the components that explained the greatest variance per target. Then, we identified the attribute type of the 20 features with the greatest weight in absolute value within each selected component. Figure 7 exhibits the percentage of feature participation per target.

In this sense, we identify that **Topic modeling** features have the highest absolute values inside the 10% of the components that explained the greatest variance in targets FM, HC, and LA during the first phase of classification. As we have already noted, **Word embeddings** are the main features in targets A and CCC.



Note. The mark (-) is associated to features not used in the group regarding table 7.

Figure 7: Percentage of feature participation in the 10% of the components that explained the greatest variance per target.

4.2. Feature space for the second phase

As in the first phase, we generated feature groups to determine which improved the classification between *Favor* and *Against* labels in each target. Table 11 exhibits the feature groups implemented. Moreover, we implemented **Topic modeling**, **Term distribution**, and **Target attributes** features in all groups since these improved classification results, and some of them represent information to examine the difference in stance labels.

	Topic modeling			Tweet-specific content		Sentiment and subjectivity markups					Target attributes		Term distribution	Word embeddings
	Majority stance codification	Minority stance codification	Opinionated tweets	Hashtags vector	Usernames vector	Stanza vector	Vader vector	MPQA weighted feature	Adjective feature	SentiWordNet vector	Target label vector	*N-grams* and *n-chars* target label feature	Weighted features	BERTweet vector
Group 6	✓	✓	✓					✓	✓		✓	✓	✓	
Group 7	✓	✓	✓	✓	✓	✓	✓		✓		✓	✓	✓	
Group 8	✓	✓	✓	✓	✓	✓	✓				✓	✓	✓	✓
Group 9	✓	✓	✓	✓		✓	✓		✓	✓	✓	✓	✓	✓
Group 10	✓	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓

Note. Topic modeling features varied considering the number of topics $n_k = \{3, 5, 7\}$ and the number of top terms from each topic $n = \{10, 15\}$.

Table 11: Feature groups for the second phase.

Table 12 shows the classification results in the development set by implementing the feature groups of table 11 for each target.

We recognized that the highest value to target A occurs through feature group 6 in an ET with 105 trees considering gini as a measure to evaluate the splits. In particular, we built 3 topics ($n_k = 3$) and took the 10 top terms per topic ($n = 10$). In this order, we established that **Sentiment and subjectivity markups** as **MPQA weighted feature** and **Adjective feature** are pivotal during the second classification phase in target A. We found that **MPQA weighted feature** assigned higher weights to tweets labeled as *Against*, indicating strong subjectivity and positive polarity. Contrarily, this feature gave more negative values to tweets labeled as *Favor* pointing to strong subjectivity and negative polarity. Further, the **Adjective feature** illustrated a terminology difference between the stances. Thus, we recognized that tweets with words such as *devotee*, *graceful*, *love*, and *spiritual* express an *Against* stance toward target A.

Table 12 exhibits that SVM model achieved the highest score for target CCC. We considered a sigmoid kernel with a regularization parameter of 8. Also, we built 7 topics ($n_k = 7$) and analyzed the 10 top words per topic ($n = 10$). Hence, we detected that **Tweet-specific content** improved classification results in target CCC. We examined the **Hashtags vector** and found that no tweets expressing *Favor* stance are related to hashtags associated with against stance; also, there are no tweets with *Against* label related to hashtags associated with *Favor* stance. This is a consequence of the proposed coding and evidences the discrimination capability provided by this feature. Nonetheless, **Tweet-specific content** has no contribution in 42.29% of training tweets. This suggests the shortcomings of these features by not providing any assessment to discriminate the stance of a substantial percentage of the tweets.

	A	CCC	FM	HC	LA
Group 6	**73.58 (ET)**	65.95 (SVM)	63.05 (ET)	78.24 (LR)	70.01 (AB)
Group 7	70.73 (ET)	**66.18 (SVM)**	63.80 (SVM)	79.23 (LR)	70.26 (AB)
Group 8	69.80 (ET)	60.75 (SVM)	**71.24 (LR)**	75.68 (LR)	71.14 (LR)
Group 9	70.96 (SVM)	60.69 (SVM)	68.30 (LR)	**82.47 (LR)**	71.50 (LR)
Group 10	71.30 (ET)	62.23 (SVM)	64.23 (AB)	81.17 (LR)	**75.47 (LR)**

Note. The classifier used is indicated next to each result with an acronym in parentheses. The highest results per target are marked in **bold**.

Table 12: Results of feature groups in development set during the second phase.

Sentiment and subjectivity markups are other features contributing to the score of the target CCC. This situation is consistent with the dataset description, in which all tweets expressing an *Against* stance toward target CCC possess a negative sentiment label[13]. Moreover, this finding is similar to the feature composition of Tutek et al. (2016), where the authors reported a F_{avg} of 72% in development set toward target CCC codifying sentiment information. Additionally, we report a higher value exploiting adjective information that indicated the words used in each stance to qualify the target. Therefore, we recognized that words such as *blissful*, *boiled*, *repentant*, and *prime* are presented in tweets with *Favor* label.

Adjective feature did not improve the classification results to target FM. We identified that LR model with 3 as the regularization parameter provided the best score using feature group 8. We generated 7 topics ($n_k = 7$) and examined the 15 top words per topic ($n = 15$). It is possible to affirm that **Tweet-specific content** contributed to the performance. We examined the composition of these features and found a situation similar to that described for the target CCC. These attributes allow to discriminate the stance but assign values to a few tweets. For example, only 20% of the training tweets received a non-zero value in the **Usernames vector**.

Target HC has the best result in a LR classifier with a regularization parameter of 2 considering feature group 9. We built topic features with $n_k = 3$ and $n = 15$. This result could be explained by the presence of the **Hashtags vector** and the lack of **MPQA weighted feature** since these are different between groups 9 and 10, where target HC achieved the highest ratings. We explored the composition of the **Hashtags vector** in the train set and found the same situation described above, where this feature designated zero value to 48.09% of tweets with *Against* label and 38.14% of tweets with *Favor* label.

Furthermore, we detected that **MPQA weighted feature** was not appropriate to classify the target HC during the second phase. It occurs since it assigns similar values to opinion tweets, especially if they contain words associated with negative polarities. Consequently, the lack of **MPQA weighted feature** was pertinent and facilitated the identification of the contribution of other features such as adjective information. For instance, we established that terms such as *ail*, *soil*, *vacant*, and *tension* appear in *Against* label. Likewise, the **SentiWordNet vector** allocated mean values below zero to tweets labeled *Against* and positive values to tweets with *Favor* label in target HC. Although, this feature gave a value of zero to 45.01% of the train set, as these tweets include terms that are not present in the SentiWordNet lexicon.

Target LA presents the best score with feature group 10 in a LR model, considering 7 as the regularization parameter. **Topic modeling** features were built with 7 topics ($n_k = 7$) and 10 top terms per topic ($n = 10$). We detected that **Sentiment and subjectivity markups** are essential during the second phase. **Adjective feature** provides discrimination on the terminology used to qualify each stance. To illustrate, we found that tweets with *Favor* label have adjectives such as *neither*, *dumb*, *ultimate*, and *pregnant* meanwhile, tweets with *Against* labels use words like *pride*, *anger*, *satanic*, and *witty*.

We noticed that **MPQA weighted feature** assigns higher values to tweets with *Favor* stance indicating a positive polarity and strong subjectivity. A similar situation occurs with **SentiWordNet vector**, nonetheless this feature assigned a value of zero about 40% of the tweets which implies that it did not provide a value to distinguish stance.

In brief, we affirm that **Adjective feature** strengthened the classification result in four of five targets. Recall that we proposed to use the adjective information by modifying the work of Dey et al. (2017). We did not implement the **Adjective boolean feature** since it is to be expected that tweets with *Favor* and *Against* labels use adjectives. Therefore, it was pointless to evaluate the presence or absence of adjectives in tweets from the second phase of classification. Alternatively, we discriminated which adjectives were presented in each tweet to identify the qualifications of the stance labels.

[13] SemEval-2016 dataset description: `http://www.saifmohammad.com/WebPages/StanceDataset.htm`

Consequently, we recognize the terminological differences between stance as exemplified in the preceding paragraphs, moreover we provide this information to the classifier to improve its performance.

Our findings suggest that **SentiWordNet vector** is not adequate to discriminate *Favor* and *Against* stances for most of the targets (see tables 11 and 12). Initially, we did not expect our result as we implemented preprocessing steps and a codification of SentiWordNet information similar to Dey et al. (2017), since they report that this feature facilitated the second phase classification independently of the target. We claim that this difference occurs because our preprocessed tweets do not contain words included in the SentiWordNet lexicon.

We highlight that the MPQA lexicon provides information to enrich classification when applied to specific targets. Hence, we do not support the proposal of Dey et al. (2017) to use the same set of features indiscriminately among the targets. Alternatively, we noticed that **MPQA weighted feature** contributes to obtaining the highest scores for targets FM, HC, and LA. Therefore, we supply evidence to incentive target-dependent approach in stance detection task by promoting the explanation of the classification results. Table 13 shows the results of the ablation test in the second phase of classification, regarding that we implemented the classifier that provided the highest result per target. Furthermore, we did not change the experimental conditions when adding the features.

	A		**CCC**		**FM**		**HC**		**LA**	
	Dev	Test	Dev	Test	Dev	Test	Dev	Test	Dev	Test
All features	73.58	72.50	66.18	68.76	71.24	64.81	82.47	78.03	75.47	72.97
(-) Majority stance codification **Minority stance codification**	73.43	67.28	65.74	64.30	70.04	64.62	82.17	74.06	75.45	69.87
(-) Opinionated tweets	72.90	66.20	65.56	62.88	69.96	64.49	81.92	73.94	75.44	68.31
(-) Term distribution	55.98	53.23	64.61	61.94	69.53	63.41	81.82	71.33	75.28	65.22
(-) Target attributes	51.02	52.86	64.62	61.94	65.78	63.31	81.82	72.33	74.88	62.50
(-) Tweet-specific content	*NA*	*NA*	63.68	60.20	60.52	61.23	74.39	71.45	*NA*	*NA*
(-) Sentiment and subjectivity markups					59.97	60.55	74.57	70.01	72.7	61.21
(-) Word embeddings	*NA*	*NA*	*NA*	*NA*						

Note. Dev: Scores in development set. Test: Scores in test set. *NA*: Associated to features not used in the group following table 11. White spaces: Corresponds to last feature implemented during the ablation test.

Table 13: Ablation test in the second phase of classification regarding F_{avg} score.

Our codification of **Tweet-specific content** increases the classification results, but only by an average of 1.23% in the test set. This occurs because our proposed coding does not to provide an evaluation value to a significant percentage of tweets. Despite this, **Tweet-specific content** gives adequate information to discriminate those tweets that receive a weight different from zero. Other studies (Tutek et al., 2016; Zhang & Lan, 2016; Mourad et al., 2018) assess the overlap of hashtags and usernames in the content of tweets, but their performance is not outstanding. Thus, we consider that further research is needed on how to encode tweet-specific information to increase its contribution to stance detection results.

Likewise, we found that **Word embeddings** contribute to the classification results of three targets (see tables 12 and 13). Our result differs from Tutek et al. (2016), where these features did not improve classification results even when they combined word embeddings with other features. This divergence could be due to the type of word embedding implemented since Tutek et al. (2016) experimented with word2vec while we use BERTweet. Although, we do not have experimental evidence to indicate that BERTweet is a more suitable representation than word2vec. The related works (see section 2) allow us to establish that embedding vectors trained on tweet corpus enhances classification results. This is evidenced by works such as (Mohammad et al., 2017; Sobhani et al., 2016; Zarrella & Marsh, 2016) obtain a high overall F-measure (see table 15) through word embeddings learned in tweet corpus (see table 2). Moreover, we achieved 70.01% in target HC using only the **BERTweet vector**, and this result is higher than reported by top teams in *SemEval-2016 task 6* challenge.

Table 13 exhibits that **Target attributes** increase the performance in all the targets. Nevertheless, their contribution rate is low. Our results are similar to those of Mohammad et al. (2017), where the authors use comparable coding. To illustrate, Mohammad et al. (2017) reported that target attributes enhanced in 0.3% the F_{avg} in the test set toward target LA. Meanwhile, our result raised 1.29% (*i.e.* the difference between 62.50 and 61.21 scores in table 13). On the other hand, **Term distribution** enlarged F_{avg} score for all targets, except target HC.

We detected that **Topic modeling** features increase the F_{avg} score in all targets. Mainly, **Majority stance codification** and **Minority stance codification** features improve the classification in the test set. These features helped the second phase of classifying of tweets related to targets CCC, HC, and LA. **Topic modeling** features have a high contribution for *Atheism* (A), where the performance grew by 6.3% in the test set. Nonetheless, we found that the score of target FM increased by 0.32% (*i.e.* the rate raised from 64.49% to 64.81%). Therefore, we assert that our proposal enhanced the classification of some targets. This provides further evidence of the target-dependent approach.

We experimented with PCA and RFE to obtain the best classification performance regarding the feature group that provides the highest score per target (see table 12). Table 14 shows the results with and without reduction strategies per target.

The increase in classification results is undoubtedly due to the application of reduction strategies. Nonetheless, our finding consistently differs from that reported in the related works (see section 2). First, almost no previous studies indicate using reduction techniques to increase the classification score. Second, the authors in the state-of-the-art (Al-Ghadir et al., 2021) claim that PCA decreases the overall F-measure. We hypothesize that these divergences occur given our feature composition regarding **Topic modeling** features. To the best of our knowledge, we are the first to assert that reduction strategies improve stance detection in *SemEval-2016 task 6* dataset.

	A		CCC		FM		HC		LA	
	Dev	Test	Dev	Test	Dev	Test	Dev	Test	Dev	Test
without PCA/RFE	73.58 (ET)	72.50 (ET)	66.18 (SVM)	68.76 (SVM)	71.24 (LR)	64.81 (LR)	82.47 (LR)	78.03 (LR)	75.47 (LR)	72.97 (LR)
with PCA	76.20 (SVM)	67.58 (SVM)	**78.20 (SVM)**	**73.43 (SVM)**	72.44 (SVM)	66.78 (SVM)	80.41 (LR)	76.53 (LR)	72.59 (LR)	72.83 (LR)
with RFE	**80.88 (ET)**	**76.20 (ET)**	70.84 (SVM)	70.27 (SVM)	**75.17 (SVM)**	**70.36 (SVM)**	**82.47 (LR)**	**78.55 (LR)**	**75.64 (LR)**	**74.61 (LR)**

Note. Dev: Scores in development set. Test: Scores in test set. The classifier used is indicated next to each result with an acronym in parentheses. The highest results per target are marked in **bold**.

Table 14: F_{avg} scores with and without reduction strategies in the second phase of classification.

We used PCA for the target CCC and analyzed the composition of the latent components constructed. In particular, we organized the latent components from highest to lowest, considering the explained variance. Then, we divided these components into ten groups. Subsequently, we determined the type of the 20 features with the highest absolute values in each group. Figure 8 exhibits the percentage of feature participation in each component group.

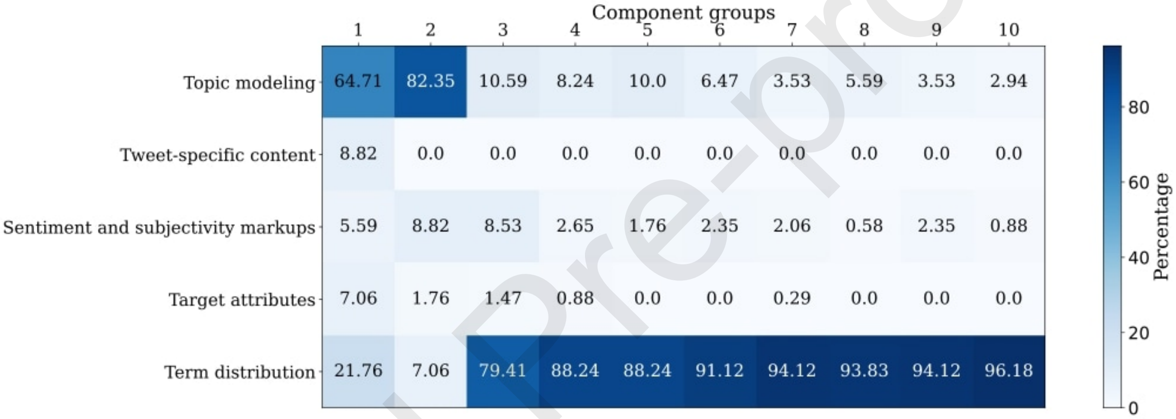
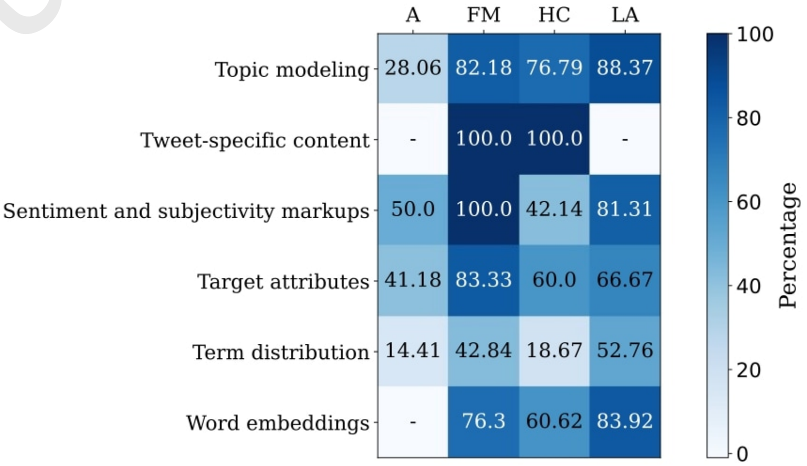


Figure 8: Percentage of feature participation in each component group for target CCC.

Topic modeling features had the leading role in the components that explained most variance. In contrast, **Term distribution** accounted for the highest values within the latent components with the lowest variance. This establishes that topic features have an essential contribution in explaining the variance of the data. The opposite situation occurs with **Term distribution** attributes.

Furthermore, RFE enhanced the classification in four of five targets during the second phase (see table 14). We suspect this is a consequence of selecting feature vectors where RFE seeks to reduce the noise some feature vectors impose during classification. Figure 9 exhibits the percentage of features retained after applying RFE per target.



Note. The mark (-) is associated to features not used in the group following table 11.

Figure 9: Percentage of retained features after applying RFE.

Tweet-specific content vectors were retained for targets FM and HC. Hence, these features are relevant during discrimination despite the shortcomings previously mentioned. Furthermore, we detected that most of the **Topic modeling** vectors were retained, implying their contribution to the performance of the classification system. In this order, we support evidence of the advantage of using **Topic modeling** feature during the second phase of classification. Section 4.4 describes how these features provide information to generate an explanation of stance labels.

Rank	**Technique**	**System**		**Overall**				F_{avg}		
			Favor	Against	Overall F-measure	**A**	**CCC**	**FM**	**HC**	**LA**
2	ML	Our system	70.37	78.89	74.63	**76.20**	73.44	70.36	78.55	74.61
Related work. Unreported values are marked "-"										
1	ML	Al-Ghadir et al. (2021)	**84.49**	68.36	**76.45**	73.52	**79.95**	72.99	75.02	74.74
3	ML	Dey et al. (2017)	69.53	**79.36**	74.44	72.5	53.59	**78.77**	79.7	**83.60**
4	DL	Siddiqua et al. (2019)	66.56	77.66	72.11	67.73	44.27	66.76	60.28	64.23
5	DL	Wei et al. (2018)	65.62	76.55	71.04	64.60	43.02	59.35	66.21	66.21
6	DL	Ahmed et al. (2020)	64.42	76.49	70.46	61.99	42.57	63.38	63.46	65.43
7	ML	Mohammad et al. (2017)	-	-	70.32	68.25	43.80	58.72	57.74	66.91
8	DL	Wani et al. (2021)	65.06	75.00	70.03	-	-	-	-	-
9	ML	Mourad et al. (2018)	-	-	70.04	72.11	46.80	42.56	**79.87**	62.72
15	DL	Li & Caragea (2020)	-	-	65.33	69.22	59.18	61.49	68.33	68.41
Top teams in *SemEval-2016 task 6*										
10	ML	Baseline (Mohammad et al., 2016)	62.98	74.98	68.98	65.19	42.35	57.46	58.63	66.42
11	DL	Zarrella & Marsh (2016)	59.32	76.33	67.82	61.47	41.63	62.09	57.67	57.28
12	DL	Wei et al. (2016)	61.98	72.67	67.33	63.34	52.69	51.33	64.41	61.09
13	ML	Tutek et al. (2016)	60.93	72.73	66.83	67.25	41.25	53.01	67.12	61.38
14	ML	Zhang & Lan (2016)	60.55	70.54	65.55	61.96	41.32	56.20	57.84	61.25
16	ML	Elfardy & Diab (2016)	54.99	72.21	63.60	55.68	39.41	53.88	51.19	59.38
17	ML	Wojatzki & Zesch (2016)	48.71	74.75	61.73	52.47	35.50	55.12	44.23	57.25

Note. ML: Machine learning technique. DL: Deep learning technique. F_{avg} is reported per target. It is essential to mention that the standard error of these metrics in the development and testing set lies in the order of e^{-16}; hence, the numerical change represent a statistically significant difference in the results.

Table 15: Systems performance with *SemEval-2016 task 6* dataset.

4.3. Experimental results

Table 15 shows a comparison between our proposal and related studies from section 2. We report F_{avg} score per target and overall F-measure to evaluate the system performance. All the scores correspond to the classification in the test set. Additionally, we indicate the ranking of the works considering the overall F-measure and express the main technique used in each study.

Our proposal outperforms all the top teams in *SemEval-2016 task 6* challenge. In particular, we improve the F_{avg} score to *Atheism* (A) and rank second in the state-of-the-art. To the best of our knowledge, we obtain classification results superior to the works with deep learning approach.

Namely, our results are remarkably near to the work of Dey et al. (2017) (third in the ranking). Although, the systems are divergent regarding feature composition and classification strategies. In addition, we have compared our proposal and the study of Dey et al. (2017) throughout the previous sections (see sections 4.1 and 4.2). Hence, we claim that our proposal obtains better results because it considers the target-dependent during stance detection and exploits the reduction strategies to create an appropriate representation to classify tweets. We have no evidence to indicate that **Topic modeling** features are the primary reason for the difference in results. This occurs since we could not implement all the features proposed by Dey et al. (2017), as there are gaps in their codification (*i.e.* frame semantics as discussed in section 2.2). We also identify some features that, the authors point out as influential, do not provide a numerical rating to discriminate most of the tweets (*i.e.* **SentiWordNet vector** as discussed in section 4.2).

Our system has an overall F-measure lower than the state-of-the-art (Al-Ghadir et al., 2021). Moreover, these works are remarkably different. We designed a two-phase classification scheme considering stance detection as a target-dependent task and exploited attributes of different types to build our feature space. Alternatively, Al-Ghadir et al. (2021) implemented a one-phase approach and used term distribution to create the features since they declared that sentiment information did not improve the classification results. At first, our proposal may seem more complex, given the number of features and parameters required. Despite this, and as we have stressed before, Al-Ghadir et al. (2021) are ambiguous about how they constructed their feature space and trained the classification system. Thus, it is not possible to replicate their proposal. In addition, our proposal is the only one that obtains classification results close to the state-of-the-art and provides information that allows explaining the discrimination of stance labels (see section 4.4).

SemEval-2016 task 6 challenge also evaluates whether the systems can detect stances when opinion is expressed toward some other target. Therefore, the database is split following the opinion labels provided by the SemEval workshop. Table 16 shows our system performance and related works that reported these measures. It is essential to clarify that we did this experiment implementing the feature groups and classifiers related to the best results.

Our proposal has the second position in the rank by evaluating its performance only on tweets that directly talk about the target. Notably, all studies obtain a lower performance when tweets do not express an opinion toward the target. Nonetheless, our system outperforms the best-performing system by 0.73% (*i.e.* the difference between 53.18 and 52.45). Thus, our proposal delivers major improvement when the tweet does not explicitly express an opinion toward the target.

System	Target	Other	All
Our system	79.37	**53.18**	**74.63**
Dey et al. (2017)	**79.89**	52.45	74.44
Baseline (Mohammad et al., 2016)	74.54	43.20	68.98
Zarrella & Marsh (2016)	72.49	44.48	67.82
Tutek et al. (2016)	73.66	37.47	66.83
Zhang & Lan (2016)	70.29	44.25	65.55
Elfardy & Diab (2016)	67.89	45.28	63.60
Wojatzki & Zesch (2016)	67.23	42.45	61.73

Note. It is essential to mention that the standard error of these metrics in the development and testing set lies in the order of e^{-16}; hence, the numerical change represent a statistically significant difference in the results.

Table 16: Systems performance with *SemEval-2016 task 6* dataset discriminating tweets toward the target or other target.

4.4. Explanation through topics features

In order to expose how the **Topic modeling** features provided an explanation of stance labels, we exemplify some of the terms used in the construction of some topic features. Table 17 shows example terms from the **Topic modeling** features of targets FM and A. It is important to clarify that the topic terms presented correspond to the topic features related to the best scores for each target.

	Target	Codification	Example terms									
First phase	FM	Opinion codification	valent	cook	power	nick	makeup	same	wage	gap	anita	respect
		All training tweets	man	feminist	equal	rape	women	male	if	femal	want	not
Second phase	A	Majority stance codification	lord	power	alway	kingdom	praise	done	never	enemi	hope	heart
		Minority stance codification	bullshit	canadian	dawkin	harm	read	cd	harper	scoc	richard	c
		Opinionated tweets	god	love	lord	that	you	as	religion	jesu	if	not

Note. The terms presented were subjected to the preprocessing described in section 3.1.

Table 17: Example terms from **Topic modeling** features used during the first and second phases of classification.

We recognize that some topic terms do not contribute to building an explanation of stance labels when they are analyzed in isolation. To illustrate, we indicate terms such as *if* and *not*. These words tend to provide general information and to be used regardless of the tweet's stance toward a target. We are not saying that these types of terms are not useful for conveying an opinion. On the contrary, these words provide knowledge about a topic when they are used together with other terms. In this sense, some words like negations modify other terms explaining the stance label.

Moreover, we recognize that contextualized interpretation of the terms retrieved through **Topic modeling** is appropriate. Thus, words such as *valent* or *anita* can be easily linked to *@JessicaValenti* and *#Anitasarkeesian*, respectively. Hence, these terms provide insight into tweet content since they refer to representatives of feminism.

The terms associated with **Opinion codification** correspond to controversial topics between genders, such as the wage gap, power, cook, respect and makeup. This finding enables us to establish that **Opinion codification** retrieves terms to describe the content of tweets that express a judgment about the target FM. Likewise, **All training tweets** codification involves general words of target FM since it points out that terms such as *man*, *femal*, *women* and *male* are related to the main actors in the feminism conversation.

A similar situation occurs to the term topics retrieved with **Opinionated tweets codification**. These terms are generic in the domain of religion. Besides, these words tend to generate the notion that the content of the tweets supports spiritual beliefs, leaving aside other stances. Alternatively, **Minority stance codification** retrieved terms such as *canadian*, *harper*, and *scoc* that are present in tweets like "Now that the SCOC has ruled Canadians have freedom from religion, can someone tell Harper to dummy his 'god bless Canada' #cdnpoli" which expresses a *Favor* stance to *Atheism* (A). Thus, we confirm our hypothesis that this coding captures the particular content of the tweets.

Nonetheless, we found evidence that contradicted our expectation that this coding would not represent information from tweets that contained terms of the majority class primarily. For example, *bullshit* is retrieved even when it occurs in "When it comes to scientific discoveries, the #religious call them bullshit until their texts already knew it" which is composed of many words used to express an *Against* stance toward target A. On the other hand, the **Majority stance codification** notably produced terminology related to religion and permitted identification of the underlying topics associated with Christianity.

In brief, we assert that each coding allows us to obtain different information about the target; in some cases, it is possible to establish the actors involved in the discussion or the issues causing controversy. Furthermore, the contextualized analysis of topic terms provides the opportunity to recognize the characteristics of the stance labels toward a target. Finally, we emphasize that our proposal provides textual information to construct an explanation of the content of the tweets. In contrast, previous works (Wojatzki & Zesch, 2016; Elfardy & Diab, 2016) exploit lexicons, thus depending on their content and how it has been encoded. Therefore, our work is flexible in the sense that it adapts to the set of tweets without requiring the data to contain any knowledge previously encoded in an external resource.

5. Conclusions and future work

Stance detection is a relevant approach in fake information detection, mainly because it aids the interpretability of the results. Nonetheless, most current stance detection studies do not explain how the feature space supports an explanation that allows for different stance labels toward a target. In this work, we proposed a two-phase classification scheme for stance detection that mainly exploits topic modeling features. Thus, a practical advantage is that topic modeling helps us to provide an explanation of classifier behavior and to present textual data to give a qualitative understanding of text stance. In particular, we proposed different tweet codifications (see section 3.2.1) to emphasize the terminological differences between stance labels. We modeled the tweets with BTM and retrieved the most relevant terms in the topics. These terms are used to construct an explanation of stance labels. Thereby, our proposal is remarkably different from previous ones since we provide an explanation of stance labels using the most relevant terms within topics over the tweets.

Consequently, a strength of our methodology is that our approach is flexible and data-based because it adjusts to the vocabulary, leveraging topic information. Moreover, our classification system ranks second in the state-of-the-art *SemEval-2016 task 6* dataset (see table 15) with a 74.63% overall F-measure. To the best of our knowledge, our results are superior to deep learning-based proposals and competitive with studies that do not provide an explanation for stance labels. Hence, we affirm that the leverage of our proposal is that topic modeling features enhanced classification results and provided textual information about a plausible explanation of stance labels. Additionally, we defined five different sets of attributes (i.*e.* **Tweet-specific content**, **Sentiment and subjectivity markups**, **Target attributes**, **Term distribution**, and **Word embeddings**) using proposals from previous research.

We modeled **Tweet-specific content** such as hashtags and usernames (see section 3.2.2). Nonetheless, we identified that this attribute provided a lower contribution to classification results (see table 13). A methodological limitation in our codification was the assignment of non-discriminative values to most of the tweets. The poor performance of this attribute has been reported in previous studies such as Tutek et al. (2016), Zhang & Lan (2016), and Mourad et al. (2018). Hence, further research is needed on encoding and using tweet-specific information.

Besides, we codified **Sentiment and subjectivity markups** (see section 3.2.3) using an unsupervised technique and manipulating information from lexicons. Specifically, we implemented various features proposed by Dey et al. (2017). We used the codification of the MPQA subjectivity-polarity lexicon to assess whether the tweet contains words related to subjectivity. Also, we evaluated the adjective information of tweets using WordNet and designated a sentiment score regarding SentiWordNet. Dey et al. (2017) claim that MPQA data and adjective information are decisive to stance detection indiscriminately among the targets.

Conversely, our findings suggest that **MPQA boolean feature** and adjective information did not improve the performance in all the targets. For instance, we identified that the **Adjective boolean feature** enhanced the results in only two targets and reduced the score for the others (see table 8). Furthermore, we reported that the **SentiWordNet vector** is inadequate in stance detection for most targets (see tables 11 and 12). We did not anticipate these results since Dey et al. (2017) declare that this feature facilitated the classification scores independently of the target. To increase our classification results, we proposed to modify Dey et al. (2017) work related to adjective codification. In particular, we implemented **Adjective feature** to discriminate the adjectives within the tweet in an effort to evaluate the qualifications of the stance labels. Therefore, our proposed attribute increased the discrimination in four of five targets (see table 12).

Target attributes were modeled leveraging the *n-grams* and *n-chars* of target names (see section 3.2.4). Likewise, we claim that **Target attributes** strengthened the classification of tweets, though their contribution rate is low (see tables 9 and 13). Our results are similar to those from previous studies (Mohammad et al., 2017; Zhang & Lan, 2016). Hence, a deeper understanding of how to use target information to improve stance detection is needed.

We employed **Term distribution** to represent the tweet content without considering the relevance of the terms given topic modeling information (see section 3.2.5). Our results allow us to state that **Term distribution** has a minor contribution to the results after applying reduction strategies (see figures 7, 8, and 9). This finding contradicts the state-of-the-art proposal (Al-Ghadir et al., 2021), where **Term distribution** is the main feature.

Furthermore, we used BERTweet as **Word embeddings** to provide a common ground to compare tweets with different compositions (see section 3.2.6). Unlike Tutek et al. (2016), we found that **Word embeddings** improved the classification results for three targets during stance detection (see tables 12 and 13). We hypothesize that this finding occurs because we employed BERTweet, trained on a tweet corpus; meanwhile, Tutek et al. (2016) used word2vec. Taking into account this result and the conclusions of similar research (Mohammad et al., 2017; Sobhani et al., 2016; Zarrella & Marsh, 2016), we affirm that embedding vectors trained on tweet corpus tend to enhance stance detection.

In this sense, we tackled stance detection as a target-dependent task through a two-phase classification scheme. Hence, we trained two classifiers with a particular feature set per target (see figure 2). In the first phase, we identified the attributes that contribute to classifying opinionated and non-opinionated tweets (see tables 7 and 8). As a result, we established that **Word embeddings** were the main feature to classify the tweets of targets *Atheism* (A) and *Climate Change is a real Concern* (CCC) since this attribute provided values greater than 80% in F_{avg}.**Topic modeling** features were central discriminating tweets of targets *Feminist Movement* (FM), *Hillary Clinton* (HC), and *Legalization of Abortion* (LA), given the results of the ablation test (see table 9). Moreover, we recognized that PCA improved the system performance by an average of 3.8% compared to a non-reduction approach (see table 10).

In the second phase of classification, we determined the features to classify tweets with *Favor* or *Against* labels toward a target (see tables 11 and 12). Specifically, **Topic modeling** features increased the F_{avg} score in all targets (see table 13). Furthermore, these features enhanced the performance to target A since it improved by 6.3% in the test set. Nevertheless, the score of target FM grew by just 0.32%. Therefore, we claim that our proposal further increased the classification of some targets, proving the effectiveness of the target-dependent approach. Likewise, we identified that the reduction strategies as PCA and RFE helped the system results in the second phase (see table 14).

Consequently, we supply evidence to the incentive target-dependent approach in stance detection and state that the features provided a distinctive contribution to the classification of each target. Hence, our methodology differs from previous studies (Dey et al., 2017; Al-Ghadir et al., 2021; Mohammad et al., 2017) that use markups without considering the composition of each target. Moreover, our proposal is the only one that obtains classification results close to the state-of-the-art and provides information that allows explaining stance labels. In this order, we exemplified some of the terms used in the construction of the topic features and how they can be employed to explain the contents of tweets with a particular stance (see section 4.4). Thus, the practical advantage of our proposal is that we illustrated how the proposed tweet representations give information about the target, such as the actors involved in the discussion or the topics causing controversy.

In future work, we will evaluate the performance of our proposal on databases that are not fully labeled, as this is a common scenario during stance detection in tweets. We suggest further experimental studies on the impact of feature reduction strategies to characterize their performance during stance detection. A limitation of our proposal is that some fundamental features during classification do not provide an explanation of stance labels. **Word embeddings** are essential in the first phase for targets A and CCC, and these features play a prominent role in the second phase for targets FM, HC, and LA. We will explore strategies to answer how **Word embeddings** support the explainability of classification results. For instance, Lovera et al. (2021) enrich a Deep Learning classifier with information from Linked Data. The methodological approach produces accurate and explainable results in tweet sentiment analysis. Nonetheless, the authors expose that the proposal could generate an underfitting classifier for tweets without terms present in the knowledge graph. Another possible strategy is to provide explainability through attention mechanisms in neural network classifiers. Tian et al. (2020) propose quantum-driven text encoding and an attention mechanism to recognize the importance and stance of a tweet to justify fake information detection. In that work, they reflect that text coding facilitates high F-measure scores without overfitting. This proposal is similar to the one presented by Li & Caragea (2020).

Besides, we recognize that a methodological limitation is the reliance on external resources for the construction of some **Sentiment and subjectivity markups**. Especially, we realize that these linguistics markers are relevant for classification performance. For example, **Sentiment and subjectivity markups** produced an overall F-measure of 60.20% in the second phase (see table 13) to target CCC. Most studies report a lower score for this target (see table 15). Therefore, we plan to search for unsupervised strategies that evaluate the linguistic features of tweets. We consider that addressing these points will improve the classification results and the retrieval of information to establish explanations about the stances toward a target.

Like many of the works addressed in section 2, our methodological proposal covers tweet stance detection. It is worth recognizing whether our proposal is helpful for analyzing longer texts than those modeled. A new approach tackles stance detection as an optimization problem. Can & Alatas (2021) propose a metaheuristic to model the tweets related to targets A, FM, HC and LA. Nonetheless, this proposal is not comparable to previous work, as the evaluation metrics differ from those established in *SemEval-2016 task 6* challenge. Future research might use this approach with experimental conditions that allow comparison of results.

Appendix A

The following tables present the details of the best-performing models for each target in the first and second phase of classification.

	A	**CCC**	**FM**	**HC**	**LA**
% variance explained	56.76	64.50	98.54	93.15	98.28
Classifier	ET No. estimators: 90 Measure of splits: entropy	ET No. estimators: 85 Measure of splits: entropy	LR Regularization parameter: 3	AB No. estimators: 60 Learning rate: 0.1	AB No. estimators: 5 Learning rate: 0.1

Table 18: Details of the models applied together with the PCA in the first phase of classification. This information is of the models that provide the best score.

	A	CCC	FM	HC	LA
Reduction strategy	RFE No. of selected features: 350	PCA % variance explained: 96.11	RFE No. of selected features: 2150	RFE No. of selected features: 1700	RFE No. of selected features: 3650
Classifier	ET No. estimators: 105 Measure of splits: gini	SVM Kernel: linear Regularization parameter: 3	SVM Kernel: polynomial Regularization parameter: 6	LR Regularization parameter: 8	LR Regularization parameter: 1

Table 19: Details of the models applied together with the reduction strategy in the second phase of classification. This information is of the models that provide the best score.

Acknowledgements

This work was supported by Universidad Tecnológica de Pereira [project code 6-21-3, 2021].

References

Ahmed, M., Chy, A., & Chowdhury, N. (2020). Incorporating Hand-crafted Features in a Neural Network Model for Stance Detection on Microblog. In *ACM International Conference Proceeding Series* (pp. 57-64). https://doi.org/10.1145/3442555.3442565.

Al-Ghadir, A., Azmi, A., & Hussain, A. (2021). A novel approach to stance detection in social media tweets by fusing ranked lists and sentiments. *Information Fusion, 67,* 29-40. https://doi.org/10.1016/j.inffus.2020.10.003.

Can, U., & Alatas, B. (2021). A novel approach for efficient stance detection in online social networks with metaheuristic optimization. *Technology in Society, 64*. https://doi.org/10.1016/j.techsoc.2020.101501.

Cheng, X., Yan, X., Lan, Y., & Guo, J. (2014). BTM: Topic modeling over short texts. *IEEE Transactions on Knowledge and Data Engineering, 26,* 2928-2941. https://doi.org/10.1109/TKDE.2014.2313872.

Dey, K., Shrivastava, R., & Kaushik, S. (2017). Twitter Stance Detection — A Subjectivity and Sentiment Polarity Inspired Two-Phase Approach. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)* (pp. 365-372). https://doi.org/10.1109/ICDMW.2017.53.

Elfardy, H., & Diab, M. (2016). CU-GWU perspective at SemEval-2016 task 6: Ideological stance detection in informal text. In *SemEval 2016 - 10th International Workshop on Semantic Evaluation, Proceedings* (pp. 434-439). https://doi.org/10.18653/v1/s16-1070.

García-Cuesta, E., Gómez-Vergel, D., Gracia-Expósito, L., López-López, J., & Vela-Pérez, M. (2020). Prediction of opinion keywords and their sentiment strength score using latent space learning methods. *Applied Sciences (Switzerland), 10*. https://doi.org/10.3390/APP10124196.

Ghosh, S., Singhania, P., Singh, S., Rudra, K., & Ghosh, S. (2019). Stance Detection in Web and Social Media: A Comparative Study. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 11696 LNCS* (pp. 75-87). https://doi.org/10.1007/978-3-030-28577-7_4.

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning, 46*, 389-422. https://doi.org/10.1023/A:1012487302797.

He, J., Li, L., Wang, Y., & Wu, X. (2020). Targeted aspects oriented topic modeling for short texts. *Applied Intelligence, 50*, 2384-2399. https://doi.org/10.1007/s10489-020-01672-w.

Kirchner, J., & Reuter, C. (2020). Countering Fake News: A Comparison of Possible Solutions Regarding User Acceptance and Effectiveness. In *Proceedings of the ACM on Human-Computer Interaction.* https://doi.org/10.1145/3415211.

Küçük, D., & Can, F. (2020). Stance Detection: A Survey. *ACM Computing Surveys, 53.* https://doi.org/10.1145/3369026.

Li, S., Cao, X., & Nan, Y. (2020). Multi-level feature-based ensemble model for target-related stance detection. *Computers, Materials and Continua, 65,* 777-788. https://doi.org/10.32604/cmc.2020.010870.

Li, Y., & Caragea, C. (2020). Multi-task stance detection with sentiment and stance lexicons. In *EMNLP-IJCNLP 2019*

- 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference (pp. 6299-6305).

Lin, J., Kong, Q., Mao, W., & Wang, L. (2019). A topic enhanced approach to detecting multiple standpoints in web texts. *Information Sciences, 501*, 483-494. https://doi.org/10.1016/j.ins.2019.05.068.

Liu, H., Cocea, M., & Gegov, A. (2016). Interpretability of computational models for sentiment analysis. *Studies in Computational Intelligence, 639,* 199-220. https://doi.org/10.1007/978-3-319-30319-2_9.

Lovera, F., Cardinale, Y., Buscaldi, D., Charnois, T., & Homsi, M. (2021). Deep learning enhanced with graph knowledge for sentiment analysis. In *CEUR Workshop Proceedings* (pp. 74-86).

Mohammad, S., Kiritchenko, S., Sobhani, P., Zhu, X., & Cherry, C. (2016). Semeval-2016 task 6: Detecting stance in tweets. In *SemEval 2016 - 10th International Workshop on Semantic Evaluation, Proceedings* (pp. 31-41). https://doi.org/10.18653/v1/S16-2021.

Mohammad, S., Sobhani, P., & Kiritchenko, S. (2017). Stance and sentiment in Tweets. *ACM Transactions on Internet Technology, 17*. https://doi.org/10.1145/3003433.

Mourad, S. S., Shawky, D. M., Fayed, H. A., & Badawi, A. H. (2018). Stance Detection in Tweets Using a Majority Vote Classifier. In *The International Conference on Advanced Machine Learning Technologies and Applications* (pp. 375-384).

Nguyen, D. Q., Vu, T., & Nguyen, A. T. (2020). BERTweet: A pre-trained language model for English Tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstration* (pp. 9-14).

Przybyła, P., & Soto, A. (2021). When classification accuracy is not enough: Explaining news credibility assessment. *Information Processing and Management, 58*. https://doi.org/10.1016/j.ipm.2021.102653.

Qian, F., Gong, C., Sharma, K., & Liu, Y. (2018). Neural user response generator: Fake news detection with collective user intelligence. In *IJCAI International Joint Conference on Artificial Intelligence* (pp. 3834-3840). https://doi.org/10.24963/ijcai.2018/533.

Ribeiro, M., Singh, S., & Guestrin, C. (2016). Why should I trust you? Explaining the predictions of any classifier. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1135-1144). https://doi.org/10.1145/2939672.2939778.

Saquete, E., Tomás, D., Moreda, P., Martínez-Barco, P., & Palomar, M. (2020). Fighting post-truth using natural language processing: A review and open challenges. *Expert Systems with Applications, 141*. https://doi.org/10.1016/j.eswa.2019.112943.

Siddiqua, U., Chy, A., & Aono, M. (2019). Tweet stance detection using an attention based neural ensemble model. In *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference* (pp. 1868-1873).

Sobhani, P., Mohammad, S., & Kiritchenko, S. (2016). Detecting Stance in Tweets And Analyzing its Interaction with Sentiment. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics* (pp. 159-169). https://doi.org/10.18653/v1/S16-2021.

Thonet, T., Cabanac, G., Boughanem, M., & Pinel-Sauvagnat, K. (2016). VODUM: A topic model unifying viewpoint, topic and opinion discovery. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (pp. 533-545). https://doi.org/10.1007/978-3-319-30671-1_39.

Tian, T., Liu, Y., Yang, X., Lyu, Y., Zhang, X., & Fang, B. (2020). QSAN: A Quantum-Probability Based Signed Attention Network for Explainable False Information Detection. In *Proceedings of the 29th ACM International Conference on Information amp; Knowledge Management CIKM '20* (pp. 1445-1454). https://doi.org/10.1145/3340531.3411890.

Tutek, M., Sekulić, I., Gombar, P., Paljak, I., Čulinović, F., Boltužić, F., Karan, M., Alagić, D., & Šnajder, J. (2016). TakeLab at SemEval-2016 Task 6: Stance classification in tweets using a genetic algorithm based ensemble. In *SemEval 2016 - 10th International Workshop on Semantic Evaluation, Proceedings* (pp. 464-468). https://doi.org/10.18653/v1/s16-1075.

Wani, M., Agarwal, N., & Bours, P. (2021). Impact of unreliable content on social media users during COVID-19 and stance detection system. *Electronics (Switzerland), 10,* 1-21. https://doi.org/10.3390/electronics10010005.

Wei, P., Mao, W., & Zeng, A. (2018). A Target-Guided Neural Memory Model for Stance Detection in Twitter. In *Proceedings of the International Joint Conference on Neural Networks.* https://doi.org/10.1109/IJCNN.2018.8489665.

Wei, W., Zhang, X., Liu, X., Chen, W., & Wang, T. (2016). Pkudblab at SemEval-2016 task 6: A specific convolutional neural network system for effective stance detection. In *SemEval 2016 - 10th International Workshop on Semantic Evaluation, Proceedings* (pp. 384-388). https://doi.org/10.18653/v1/s16-1062.

Wojatzki, M., & Zesch, T. (2016). Ltl.uni-due at SemEval-2016 task 6: Stance detection in social media using stacked classifiers. In *SemEval 2016 - 10th International Workshop on Semantic Evaluation, Proceedings* (pp. 428-433). https://doi.org/10.18653/v1/s16-1069.

Yan, X., Guo, J., Lan, Y., & Cheng, X. (2013). A biterm topic model for short texts. In *WWW 2013 - Proceedings of the 22nd International Conference on World Wide Web* (pp. 1445-1455). https://doi.org/10.1145/2488388.2488514.

Zarrella, G., & Marsh, A. (2016). MITRE at SemEval-2016 Task 6: Transfer learning for Stance detection. In *SemEval 2016- 10th International Workshop on Semantic Evaluation, Proceedings* (pp. 458-463). https://doi.org/10.18653/v1/s16-1074.

Zhang, X., & Ghorbani, A. (2020). An overview of online fake news: Characterization, detection, and discussion. *Information Processing & Management, 57*. https://doi.org/10.1016/j.ipm.2019.03.004.

Zhang, Z., & Lan, M. (2016). ECNU at SemEval-2016 Task 6: Relevant or not? Supportive or not? A two-step learning system for automatic Detecting Stance in Tweets. In *SemEval 2016 - 10th International Workshop on Semantic Evaluation, Proceedings* (pp. 451-457). https://doi.org/10.18653/v1/s16-1073.

Zhou, X., & Zafarani, R. (2020). A Survey of Fake News: Fundamental Theories, Detection Methods, and Opportunities. *ACM Computing Surveys, 53*. https://doi.org/10.1145/3395046.

Zubiaga, A., Liakata, M., Procter, R., Wong Sak Hoi, G., & Tolmie, P. (2016). Analysing How People Orient to and Spread Rumours in Social Media by Looking at Conversational Threads. *PLOS ONE, 11,* 1-29. https://doi.org/10.1371/journal.pone.0150989.

We propose stance detection exploiting topic modeling features.
Our proposal ranks second in the state-of-the-art.
We present qualitative information to relate tweet content and stance prediction.
We provide explanations of our classification models in contrast to other studies.
Our main contribution is exploiting topic models to explain the classifier behavior.

Credit author statement

Manuela Gómez-Suta: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing - Original Draft.
Julián Echeverry-Correa: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing - Original Draft, Resources, Writing - Review & Editing, Supervision, Project administration, Funding acquisition.
José A. Soto-Mejia: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing - Original Draft, Resources, Writing - Review & Editing, Supervision, Project administration, Funding acquisition.