# SSSD: Leveraging Pre-Trained Models and Semantic Search for Semi-Supervised Stance Detection

**André Mediote de Sousa**
Informatics Institute
Federal Univ. of Rio Grande do Sul
Porto Alegre - Brazil
andremediote@inf.ufrgs.br

**Karin Becker**
Informatics Institute
Federal Univ. of Rio Grande do Sul
Porto Alegre - Brazil
karin.becker@inf.ufrgs.br

## Abstract

Pre-trained models (PTMs) based on the Transformers architecture are trained on massive amounts of data and can capture nuances and complexities in linguistic expressions, making them a powerful tool for many natural language processing tasks. In this paper, we present SSSD (Semantic Similarity Stance Detection), a semi-supervised method for stance detection on Twitter that automatically labels a large, domain-related corpus for training a stance classification model. The method assumes as input a domain set of tweets about a given target and a labeled query set of tweets of representative arguments related to the stances. It scales the automatic labeling of a large number of tweets, and improves classification accuracy by leveraging the power of PTMs and semantic search to capture context and meaning. We largely outperformed all baselines in experiments using the Semeval benchmark.

## 1 Introduction

Stance Detection (SD) is the task that automatically determines whether the author of a text is in favor of, against or does not manifest about a given target. Targets can be companies, movements, people or ideas (Mohammad et al., 2016b). It was initially applied to the analysis of political debates in online forums and has become very attractive to measure public opinion on social networks (Aldayel and Magdy, 2019).

SD on social media can be categorized based on different criteria, including the type of target, the type of stance (i.e., in favor, against, or neutral), and the level of analysis (i.e., post level or network level). The features used for classification vary according to the analysis level: textual features only (post level) or user-related attributes and behaviors such as mentions and the number of followers (network level) to improve the model accuracy (ALDayel and Magdy, 2021).

The state-of-the-art methods for SD (Al-Ghadir et al., 2021; Lai et al., 2017) are based on Machine Learning (ML) and have shown to be effective in various scenarios (Aldayel and Magdy, 2019). However, they rely on manual and complex feature engineering, particularly when applied at the network level. On the other hand, Deep Learning (DL) based methods for SD (Siddiqua et al., 2019; Li and Caragea, 2019) do not require feature engineering, but they can easily overfit if not trained with enough labeled data, due to their high number of parameters (Han et al., 2021). Unfortunately, labeling data is an expensive and time-consuming task, leading to small labeled datasets for specific domains (Al-Ghadir et al., 2021).

Transfer learning (Zhang et al., 2020; Giorgioni et al., 2020) and unsupervised approaches (Darwish et al., 2020; Rashed et al., 2021; Wei et al., 2019) are promising directions for SD, but they still face challenges in achieving comparable results to supervised machine learning approaches, especially in highly polarized environments such as Twitter. This is due to the difficulty of detecting stances in a noisy and polarized platform such as Twitter, where people express their opinions in nuanced and complex ways. Despite these challenges, researchers continue to explore new approaches to improve the accuracy of SD in various contexts (Rashed et al., 2021).

Using pre-trained models (PTMs) based on the Transformers architecture (Vaswani et al., 2017), such as BERT (Devlin et al., 2018) and GPT (Radford et al., 2019), researchers can address the challenge of data scarcity and the variability and noise inherent in Twitter, while capturing the relevant semantic and contextual information needed to classify stances accurately. PTMs are trained on massive amounts of data and can capture nuances and complexities in linguistic expressions, making them a powerful tool for detecting stances. By fine-tuning these models on smaller labeled datasets,



*Proceedings of Recent Advances in Natural Language Processing*, pages 264–273
Varna, Sep 4–6, 2023
https://doi.org/10.26615/978-954-452-092-2_030

they can learn specific patterns of stances in different contexts, which can help overcome the challenges of variability and noise on Twitter. Additionally, PTMs can be used to search or compare tweets with similar stances through cosine similarity (Han et al., 2021), aiding the task of stance detection. PTMs represent a promising approach to improving the accuracy of DS on Twitter.

In this paper, we propose SSSD (Semantic Similarity SD), a semi-supervised method for stance detection on Twitter that leverages PTMs and semantic search to automatically label a large, domain-related corpus for training a stance classification model. The method assumes as input a domain set of tweets about a given target and a labeled query set of tweets of representative arguments related to the stances. The tweets of the domain and query sets are converted into a contextual representation using a PTM, such that a similarity function can identify the semantic proximity of the tweets of both sets. For each tweet of the query-set, the search function selects the k most similar tweets from the domain set, assigning them the respective stance label. This set of labeled tweets is then used to train an SD classification model using some ML classification algorithm. The remaining unlabeled tweets can be classified using this model. SSDS improves stance classification performance by leveraging the power of PTMs and semantic search to capture the context and meaning of tweets in a specific domain, addressing the complexity of stance labeling. It reduces the need for manual annotation, an expensive and time-consuming task, enabling the accurate automatic label of a large volume of tweets with minimal computational costs.

Our experimental setting involved three classification algorithms and the SD benchmark datasets and metrics (Mohammad et al., 2016b), which includes six targets. SSSD outperformed the baselines (Al-Ghadir et al., 2021) by 13.9 percentage points (pp) and (Lai et al., 2017) by 11.2 pp in the overall averaged f-measure metric. We also assessed the influence of the value of $k$ on the similarity of retrieved tweets, number of labeled tweets and stance classification performance.

The main contributions of our study can be summarized as follows:

- a semi-supervised SD method that leverages PTMs and semantic search to automatically label data and train an SD classifier. By leveraging PTM and semantic search, it achieves superior performance compared to unsupervised/semi-supervised solutions (Gómez-Suta et al., 2023; Aldayel and Magdy, 2019) and outperforms state-of-the-art supervised systems (Al-Ghadir et al., 2021; Lai et al., 2017). The method is not dependent on a specific PTM or ML classification algorithm, nor requires a large, domain set of labeled data.
- A complete experimental assessment using datasets and metrics of a benchmark for stance detection (Mohammad et al., 2016b), demonstrating its effectiveness and robustness. Our approach is reproducible, and all the code is available in a public repository.

The remaining of this work is structured as follows. Section 2 presents the related work. Section 3 details the proposed semi-supervised SD method. Section 4 describes the experiments. Section 5 outlines conclusions and future work.

## 2 Related Work

Stance detection is a complex form of subjectivity analysis that focuses on identifying the attitude or perspective that a speaker or writer has towards a particular topic or issue. Unlike sentiment analysis (i.e., positive, negative), SD attempts to identify more subtle variations in the speaker's position, such as whether they are in favor of or against a particular policy or support or oppose a particular political candidate (ALDayel and Magdy, 2021).

The task of SD gained significant popularity following the launch of a competition on Twitter during Semeval 2016. Two tasks were proposed: supervised approaches (Task A) and unsupervised/semi-supervised approaches (Task B). The competition provided labeled data encompassing different targets and a well-defined methodology to assess the solutions, with a common evaluation metric (Mohammad et al., 2016a). Most studies in SD for the English language rely on SemEval datasets and evaluation methodology as a benchmark, which are limited in scope and size. The SemEval datasets cover only a specific set of domains and targets, and their small size may not capture the full complexity of the task, leading to overfitting or generalization issues. Therefore, it is important to create new datasets that can expand the scope of research in stance detection to other domains, languages, and targets (ALDayel and Magdy, 2021).

As a reflection of the scarcity of labeled data, state-of-the-art SD methods heavily rely on complex feature engineering techniques, making their reproduction a challenging tasks. For example, the leading SD system (Al-Ghadir et al., 2021) utilizes sentiment lexical dictionaries and ranked lists of TF-IDF weighted words to train K-NN classi-

fiers, but its operational details are unclear, hindering its reproducibility (Gómez-Suta et al., 2023). Other studies (Aldayel and Magdy, 2019; Lynn et al., 2019; Darwish et al., 2018) leverage network information (e.g., hashtags, retweets) to enhance classifier performance. However, these approaches require additional user behavior data, which limits their applicability beyond social media platforms.

Recent studies have focused on developing unsupervised SD models using clustering techniques. The system in (Trabelsi and Zaiane, 2018) used clustering at the author and topic levels, leveraging both the content and interaction networks of the users. Clustering was leveraged in (Darwish et al., 2020) to create an initial set of stance partitions for annotation and showed that retweets as a feature provided the best performance score upon implementing the clustering algorithm. The work in (Rashed et al., 2021) introduced embedding representations of users' tweets to enhance the SD model using hierarchical clustering to analyze fine-grained polarization between groups of tweets related to the Turkish election. While unsupervised methods are useful for minimizing the need for manual labeling, they generally perform worse than supervised methods when labeled data is available. Some unsupervised approaches (Darwish et al., 2020; Wei et al., 2019) still require some level of human supervision or adjustment, but this can be done more quickly than the manual labeling of large datasets.

To address the limited availability of labeled data for SD tasks, some studies (Zhang et al., 2020; Kawintiranon and Singh, 2021) have incorporated transfer learning techniques. These works involve fine-tuning a pre-trained language model on the source target data to learn a target-specific semantic-emotion representation. The resulting representation is then used to train a classifier for stance detection on the target with limited labeled data. By leveraging the transferred representation, which encodes information about the semantic and emotional characteristics of the target, the classifier can be trained with a smaller number of labeled examples (Han et al., 2021). The transfer learning approaches CrossNet and TextCNN-E were proposed in (Zhang et al., 2020) for enhancing SD across multiple targets. However, this approach requires a large labeled dataset and falls short of surpassing current state-of-the-art systems in SD.

Works as (Giorgioni et al., 2020; Ferreira and Vlachos, 2019) have proposed Transformer-based architectures combined with data augmentation and fine-tuning. They trained specific sentence classifiers based on UmBERTo using auxiliary datasets from tasks like sentiment analysis, irony detection, and hate-speech detection. The resulting labels were then augmented as new sentences in the SardiStance dataset. This training dataset was expanded by labeling additional tweets using distant supervision based on specific hashtags. Similarly, (Hanawa et al., 2019) utilized Wikipedia articles to extract knowledge for each topic in a seven-themed dataset. These studies incorporated the concept of transfer learning by utilizing new datasets beyond the SemEval stance task.

In summary, complex feature engineering techniques and network information can improve the performance of SD classifiers, but they are difficult to reproduce and not practical for use in contexts other than social media. Unsupervised methods can minimize the need for manual labeling but generally perform worse than supervised methods when labeled data is available. Transfer learning techniques are useful for addressing the limited availability of labeled data and can be used with smaller labeled examples, but some approaches require a large labeled dataset.

We contribute to the field by proposing a novel semi-supervised method that leverages the PTMs and semantic search to automatically label a large domain-related corpus and train an accurate stance classification model. This approach reduces the need for manual and costly annotation efforts, enabling labeling a large volume of tweets with minimal computational costs.

## 3 SSSD Overview

SSSD is a novel approach to conducting SD on Twitter using PTMs and semantic search. It explores PTMs to capture the semantic and contextual meaning of tweets, taking advantage of the strengths of deep learning-based approaches. By leveraging the power of PTMs and semantic search, we aim to automatically label a domain corpus for training SD models. PTMs are pre-trained on extensive text data to acquire general language representations that can be further fine-tuned for specific tasks such as SD on Twitter.

SSSD is semi-supervised: it relies on a set of labeled queries as input to the semantic search algorithm that automatically labels a larger corpus of domain-related tweets, which is then used to train a stance classification model. This reduces the effort required to label a large volume of tweets, while still achieving good classification performance.

By using semantic search to identify the most relevant tweets for each query, SSSD can focus on the most important posts for the stance classification problem while ignoring irrelevant or noisy data. This is an advantage compared to unsupervised approaches, which may struggle to identify the most relevant data, especially in noisy and complex datasets like Twitter.

The remaining of this section describes the input data required by SSDS, and the semantic stance detection process.

### 3.1 Input Data

SSSD requires two inputs: a set of tweets representing the domain (*domain-set*) and a set of labeled tweets with representative arguments used to express a stance (*query-set*). The domain-sets are unlabeled tweets about the target, and we aim to label them. The query-sets are a sample of tweets manually annotated with stance labels, typically in favor, against, and none. They are used to automatically label tweets of the domain-set, to compose a *training set*, i.e. a set of labeled tweets used as input to some classification algorithm.

Domain-set tweets can be collected using the Twitter API. Typically, tweets are filtered within a period of interest, and keywords representative of the target. Hashtags can be a useful strategy as they tend to capture the homophily and social influence related to the target (Darwish et al., 2020). Relevant hashtags can be found in Twitter's top trends section. They also serve as seeds in a snowballing process that identify other related hashtags based on co-occurrence. It is crucial to define an appropriate search period to avoid bias. For instance, when detecting stances regarding the candidates of an election, the search period should be carefully chosen to represent the stances as the election campaign progresses.

The critical task in our approach is the definition of a proper set of seeds to compose the query-set. In case labeled data does not exist, and the knowledge about the data is limited, a possible approach is to use advanced topic modeling methods such as BERTopic (Grootendorst, 2022) to gain a global understanding of the corpus and identify tweets representing different stances. An advantage of this particular method is that it uses semantic similarity and density-based clustering, and hence topics are dense regions of similar tweets. It also provides visualization and interpretation features to explore and understand the topics and select representative documents from each topic. For instance, (Ebeling et al., 2022) identifies the representative arguments and political bias in anti/pro-vaccination stances using BERTopic.

Standard pre-processing techniques should be applied to improve the quality and effectiveness of semantic search in tweets. These include the removal of punctuation marks, case conversion, and elimination of irrelevant characters (e.g., hashtags, links, and numbers), among others.

A labeled *validation set* is necessary to evaluate the performance of the trained stance classification model, using traditional metrics such as accuracy or F-measure. This can be a separate input set, but our method assumes (part of) the query-set can also be used for this purpose. To avoid bias, we included a maximum similarity threshold in the semantic search, as explained in the next section.

### 3.2 Semantic Stance Detection

Capturing contextual information and nuances in language can be crucial for accurate stance detection. SSSD uses a chosen PTM to transform tweets into embedding to capture the semantic meaning of the text and enable effective comparison and retrieval of similar tweets. This process requires a search function f(q, k), which returns the $k$ tweets from the domain-set with the highest similarity scores concerning the argument $q$.

We performed two adaptations to this search function. First, we assume $q$ is a pair $<$*tweet,stance*$>$ belonging to the query-set, to enable the automatic labeling of the $k$ most similar tweets. We also introduced an additional parameter to filter the retrieved tweets based on a maximum similarity threshold. This threshold ensures that tweets from the query sets are not included in the labeled training tweets, thus avoiding potential biases in model evaluation.

We divided our method into two steps, *Semantic Labeling*, and *Stance Detection*, detailed below.

**(a) Semantic Labeling:** This step is responsible for automatically labeling tweets to compose a training set, given a *domain-set* and a *query-set*. The output is a set of labeled tweets (*training-set*), which is used in the next step to train a stance classification model using a supervised ML algorithm. Table 1 presents the pseudo algorithm.

First, both the query-sets and domain-sets are converted into embeddings using a chosen PTM (e.g. BERT, GPT) or similar models (Step 1). After obtaining the embeddings, a search function is used to compare each element $q$ of the query-set with the domain-set tweets. This comparison is

```
Function: perform_semantic_labeling(query_set, domain_set, k, similarity_threshold)
Input:
  query_set: Labeled tweets with stance labels
  domain_set: Unlabeled tweets
  k: Number of similar tweets to select
  similarity_threshold: Maximum similarity threshold
Output: training-set (Labeled tweets from domain-set)
Step 1: Convert query-sets and domain-sets into embeddings using a chosen PTM
  training set = []
  query_embeddings = convert_to_embeddings(query_set)
  domain_embeddings = convert_to_embeddings(domain_set)
Steps 2-5: Loop over each query in query_set
  for q in query_set do
    Step 2: Calculate similarity scores between query_embeddings[q] and domain_embeddings
      similarity_scores = get_scores(query_embeddings[q], domain_embeddings)
    Step 3: Select the top-k tweets with the highest similarity scores
      top_k_tweets = select_top_k_tweets(similarity_scores, k, similarity_threshold)
    Step 4: Assign the corresponding stance labels from query_set[q] to top_k_tweets
      labeled_tweets = assign_stance_labels(top_k_tweets, stance(q))
    Step 5: Add to training set, handle ties using similarity
      training_set = append_and_handle_ties (training_set, labeled_tweets)
  end for
Return: training_set
```

Table 1: Pseudo Code for the Semantic Labeling of SSSD

done by calculating similarity scores between the embeddings of query $q$ and the embeddings of the domain-set tweets (Step 2). The similarity score can be computed using various methods, such as cosine similarity. Then, using the input $k$, the top-k tweets with the highest scores are selected (Step 3). There are situations where the same tweet can be present in both the labeled data and the query-sets. To avoid any biases, particularly when using part of the query-sets for performance validation, it is recommended to set a maximum similarity threshold smaller than 1 (e.g., 0.95).

The selected top-k tweets are assigned the corresponding stance label for $q$ (Step 4). Finally, the labeled tweets are included in the training set (Step 5). It is possible that a given tweet of the domain-set is similar to different queries from the query-set. If ties occur, we select the stance associated with the highest similarity score. Notice that the higher the value of $k$, the higher the likelihood of ties. Therefore, it is advisable to choose an appropriate value for $k$ to minimize ties and ensure more consistent labeling results.

This process enables to scale the labeling of tweets in the domain-set that have a similar stance to the ones in query-set, facilitating effective stance detection on Twitter. The number of labeled tweets in the training set depends on both the value of $k$ and the size of the query-set. Increasing the value of $k$ results in more labeled tweets, but it is important to find a balance between the number of labeled tweets and maintaining high similarity scores. The size of the domain-sets also affects the maximum number of labeled tweets that can be obtained. If the domain-sets are smaller, there will be a limit on the number of tweets that can be labeled.

Experimentation is key to determine the optimal value of $k$ for effective stance detection on Twitter. The ideal value can be identified by varying the value of $k$ and assessing the results using metrics such as F1-score. This iterative process of adjusting $k$ and analyzing performance metrics leads to improved accuracy and effectiveness in the stance detection task.

**(b) Stance Detection:** The process described above is effective in SD, but it does have limitations. Increasing $k$ can expand the coverage of labeled data, but it also increases the risk of more incorrect classifications due to degraded similarity scores. Training classification models using labeled data generated in the previous step is recommended to enhance accuracy and generalization. Then, the remaining unlabeled tweets of the domain-set can be assigned a label using this model.

There are various supervised machine-learning models suitable for this task, including Logistic Regression, Decision Trees, Support Vector Machines, RNNs, CNNs, and LSTMs. The choice of model and feature extraction method depends on the specific task, dataset, and available computational resources. In some cases, using the embeddings generated in the previous step as input features can be a more efficient and effective approach. The performance of the SD model can be assessed using the validation set.

## 4 Experiments

Our experiments were designed to assess the performance of SSSD against baseline systems and the influence of the value of $k$ in our results. In this section we describe the data and chosen baselines, and detail the experiments. All our experiments are reproducible, and the code and tools used in their development are available in a public repository[1].

### 4.1 Data

We developed our experiments using the Semeval datasets (Mohammad et al., 2016b) for tasks A and B. Task A included five different targets: "Atheism (Ath)", "Climate Change is a real concern (Cls)", "Feminism (Fmn)", "Abortion (Abt)", and "Hillary Clinton (Hlr)". The training dataset for Task A consisted of 2,914 labeled tweets, while the testing dataset had 1,246 labeled tweets. Task B focused on an unsupervised approach with the target "Donald Trump (Trp)". The evaluation for Task B involved a dataset of 707 labeled tweets and 78,000 unlabeled tweets. The documentation provides further information on the period and the hashtags used for collecting this datasets[2].

We constructed the *domain-sets* for each target from scratch, using the Twitter API. We parameterized each search to use the same period as Semeval (January 1 to December 31, 2016), and the same keywords. For the creation of the *query-sets*, for each target of Task A we combined the training and testing sets. For the target of Task B, we used the validation set. Each instance in a query-set includes a tweet and a stance label, indicating support, opposition, or neutrality toward the target. A summary of the distribution of tweets across the data sets is shown in Table 2. These datasets were pre-processed as described in Section 3.1.

To evaluate the performance of the trained model for all targets, we used the respective Semeval test/validation tests. To avoid biases, we introduced a similarity threshold of 0.95. Consequently, any query result with a similarity score above 0.95 was deemed dissimilar to the original query, guaranteeing the integrity and fairness of the labeling process while mitigating potential biases in the similarity of training and test sets.

### 4.2 Evaluation Metrics and Baselines

The evaluation metric used for both tasks was the macro-average F1-score, which was computed for SemEval's "Favor" and "Against" classes for all five targets in Task A and for the single target "Donald Trump" in Task B. This metric regards the class "None" as of no interest, i.e. a negative class in terms of Information Retrieval (IR) (Mohammad et al., 2016b). As baselines, we chose (Al-Ghadir et al., 2021) for Task A, and (Lai et al., 2017) for Task B. To the best of our knowledge, these are the state-of-the-art systems for these tasks, with F1-avg of 76.4% and 79.7%, respectively.

### 4.3 Experimental Setup

SDDD can be configured according to several components, and our choices are detailed below:

1. PTMs: We selected the "all-MiniLM-L6-v2" model (Wang et al., 2020). It provides comparable quality to models like MPNET (Ahmed et al., 2020) but with significantly faster performance.
2. Classification Algorithms: To assess if the choice of algorithm influenced the results, and if any model exhibited overfitting for specific targets, we experimented with multiple classification algorithms. We report here the results of the ones that yielded the best performance, namely Logistic Regression (SSSD-RL), Support Vector Machines (SSSD-SVM) and Random Forest (SSSD-RF).
3. Feature Extraction: to extract features from labeled tweets, we employed TF-IDF and bigrams. These techniques capture important information from the text and serve as inputs to the classification models.
4. Parameter $k$: We conducted experiments with a range of $k$ values, experimenting 20 values for $k$, starting from 5 and incrementing by 5 in each iteration. This iterative process is akin to traditional K-NN models, allowing us to determine an optimal $k$ value that enhances classification performance.

For each target (6) and classification algorithm (3), we performed a total of 20 iterations (values of $k$), resulting in the creation of 60 models per target.

### 4.4 Experiment 1: Method Perfomance

The goal of this experiment is to compare the performance of SSSD against the chosen baselines. The best results for each Semeval task are presented in Tables 3 and 4, together with the respective $k$.

In Task A, our method significantly outperformed the baseline (Al-Ghadir et al., 2021), which achieved an F-score of 76.4% for overall stance detection (Favg). In contrast, SSSD-RL achieved

[1] https://github.com/mediote/stance-detection

[2] www.saifmohammad.com/WebPages/StanceDataset.htm

| | Ath | Abt | Clc | Fmn | Hlr | Trp | Total |
|---|---|---|---|---|---|---|---|
| **query-sets** | 804 | 882 | 564 | 959 | 929 | 707 | **4.845** |
| **domain-sets** | 688.854 | 225.889 | 249.656 | 121.049 | 1.481.868 | 598.991 | **3.366.307** |

Table 2: Summary of tweets the representing targets

| **Systems** | Ffavor | **Overall** Fagainst | Favg | **Ath** Favg | **Abt** Favg | **Clc** Favg | **Fmn** Favg | **Hlr** Favg |
|---|---|---|---|---|---|---|---|---|
| **Baseline** | | | | | | | | |
| Al-Ghadir | 84.4% | 68.3% | 76.4% | 73.5% | 74.7% | 73.4% | 72.9% | 75.0% |
| **Our systems** | | | | | | | | |
| SSSD-LR | **87.3%** | **93.5%** | **90.4%** | **89.1%**$^{75}$ | **82.0%**$^{75}$ | **89.3%**$^{55}$ | **78.5%**$^{40}$ | 80.1%$^{55}$ |
| SSSD-SVM | 86.3% | 92.7% | 89.5% | 88.5%$^{80}$ | 80.0%$^{20}$ | 88.2%$^{80}$ | 77.2%$^{20}$ | **81.2%**$^{85}$ |
| SSSD-RF | 80.0% | 87.8% | 84.3% | 80.0%$^{85}$ | 74.9%$^{80}$ | 79.6%$^{35}$ | 70.1%$^{80}$ | 71.5%$^{70}$ |

Table 3: Results on Task A datasets

| **Systems** | Ffavor | **Overall** Fagainst | Favg | **Trp** Favg |
|---|---|---|---|---|
| **Baseline** | | | | |
| Lai el al. | 79.7% | 62.9% | 79.4% | 75.0% |
| **Our systems** | | | | |
| SSSD-LR | 87.4% | 93.2% | 90.3% | 84.7%$^{85}$ |
| SSSD-SVM | **88.0%** | **93.2%** | **90.6%** | **85.2%**$^{40}$ |
| SSSD-RF | 80.6% | 86.3% | 83.4% | 75.1%$^{65}$ |

Table 4: Results on Task B datasets

an impressive Favg of 90.3%, representing a substantial increase of 13.9 pp (percentage points). Similarly, SSSD-SVM achieved an Fav) of 90.6%, outperforming the baseline by 14.2 pp. SSSD-RF presented a slightly inferior performance compared to SSSD-RL and SSSD-SVM, but it outperformed the baseline by 7 pp. When considering individual targets, the performance differences were also remarkable. For instance, the SSSD-LR model showed performance differences ranging from 5.1 pp in the Hlr dataset to 15.9 pp in the Clc dataset.

Table 4 shows that all our systems outperformed the baseline for Task B proposed by (Lai et al., 2017) in terms of overall Favg, Ffavor, Fagainst, and Favg Trp. The best results were yielded by SSSD-SVM, which outperformed the baseline Overall Favg in 11.2 pp, due to an improvement in both Ffavor (8.3 pp) and Fagainst (30.3 pp). The worst results were achieved by SSSD-RF, and despite that, it also outperformed the baseline. Our solutions outperformed all metrics, in improvements that range from 0.1 pp (SSSD-RF Favg Trp) to 30.3 pp (SSSD-SVM overall Favg).

Our approach has demonstrated remarkable performance in both Task A and Task B of SemEval, positioning us as the new state-of-the-art in Stance Detection. In Task A, we achieved a substantial increase of 18.5 pp compared to the baseline proposed by (Al-Ghadir et al., 2021). This significant improvement showcases the effectiveness of our method in accurately detecting stances across different datasets. Similarly, in Task B, our systems outperformed the baseline proposed by (Lai et al., 2017) by approximately 14.1 pp, highlighting our advancements in stance detection for this task. These impressive results not only demonstrate the superiority of our approach but also solidify our position as the leading solution in the field.

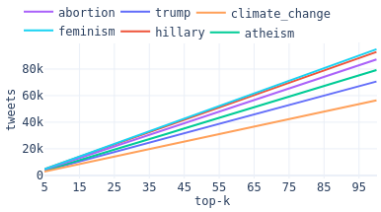
### 4.5 Experiment 2: Influence of K

The value for $k$ plays a crucial role in balancing the similarity scores and the number of labeled tweets, thereby influencing the performance of our method. We assessed its impact on three variables: the number of labeled tweets, similarity scores of retrieved tweets, and the classification performance.

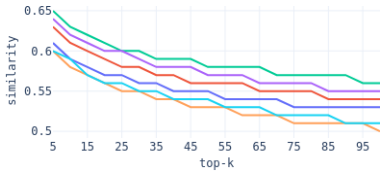
Figure 1 displays the results of the relationship between $k$ and the number of labeled tweets and the similarity. In Figure 1.(a) we can observe, as expected, a linear growth of the number of labeled tweets as the value of $k$ increases. It is interesting to note that, for all datasets, a significant number of tweets are labeled even with a low $k$ value (e.g., about 20k tweets for $k = 25$). Figure 1.(b) displays the mean similarity value according to the value of $k$. It is possible to observe the degradation of similarity scores as the value of $k$ increases.

Figure 2 illustrates a consistent pattern in the relationship between overall Favg metric (average F-score) and $k$ across all datasets and classification algorithms. As $k$ increases, Favg also increases until it reaches a point of stability, where there is a concentration of similar Favg values on the graph. However, as $k$ approaches 100, very often the Favg values start to decline, indicating a degradation in scores. This pattern is particularly evident in the Trump, Atheism, and Hillary datasets. This observation is further supported by the findings presented in Figure 1.(a).

Although most of our best results were achieved with $k = 60$, establishing a fixed value for all cases is not an adequate solution. Considering the results




(a) $k$ and number of labeled tweets




(b) $k$ and cosine similarity

Figure 1: Relationship between $k$, labeled tweets, and similarity scores.

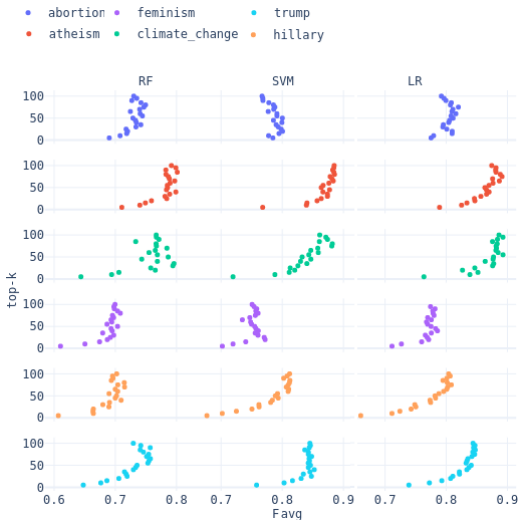



Figure 2: Relationship between K and Favg

in Tables 3 and 4, we see that for each dataset and classification algorithm, there is a specific $k$ that provides the best trade-off between $k$ and Favg.

The correlation matrix in Figure 3 summarizes all the points discussed so far. Higher $k$ values positively impact the number of labeled tweets, negatively impacts the similarity, with a minor impact on Favg. We also notice a negative impact caused by high similarities concerning Favg and number of labeled tweets, confirming the need for a balance between these variables for good results.

## 5 Conclusions

In this work, we proposed SSSD, a semi-supervised method for SD on Twitter based on semantic search. We leverage PTMs in combination with a top-k function to retrieve and label domain-specific tweets, which are then used the automatic label a

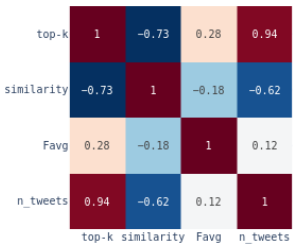



Figure 3: Correlation matrix

dataset to train a supervised classification model. It reduces the dependence on large annotated datasets while significant improving classification performance. We largely outperformed state-of-the-art supervised systems using the Semeval stance detection benchmark.

In our evaluation, we tested different $k$ values, assessing their impact on performance with various datasets and classifiers. The results showed that our method is robust and has a high degree of generalization. We also found that the optimal $k$ varied based on the specific scenario, with a trade-off between similarity scores and the number of labeled tweets to maximize ranking performance. Overall, our findings indicate that our method is effective for various SD scenarios, but the value of $k$ needs to be identified experimentally.

We have shown that by leveraging PTM and semantic search, our method handled the nuances and complexities of stance automatic labeling. Our approach is simple, computationally inexpensive, and the encouraging results motivates us to further investigate it in other text classification tasks, making it a valuable contribution to the field of NLP by addressing the challenge of labeled data scarcity.

As future work, we intend to qualitatively evaluate our method regarding some challenges faced when analyzing social phenomena on Twitter. One of them is the bias introduced in the interpretation of topics due to hashtags to represent the objects of study. A common example is false positives, where a tweet is falsely inserted in the context of a hashtag by refuting the idea represented by it, usually through replies. There is also the scenario where a hashtag is purposefully linked to events (e.g. games, famous artists) outside of its context to increase its relevance and impact artificially.

**Acknowledgments:** This study was partially financed by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, CNPq (131178/2020-2) and the PETWIN Project (FINEP financing and LIBRA Consortium).

## References


Mumtahina Ahmed, Abu Nowshed Chy, and Nihad Karim Chowdhury. 2020. Incorporating handcrafted features in a neural network model for stance detection on microblog. In *Proceedings of the 6th International Conference on Communication and Information Processing*, pages 57–64.

Abdulrahman I Al-Ghadir, Aqil M Azmi, and Amir Hussain. 2021. A novel approach to stance detection in social media tweets by fusing ranked lists and sentiments. *Information Fusion*, 67:29–40.

Abeer Aldayel and Walid Magdy. 2019. Your stance is exposed! analysing possible factors for stance detection on social media. *Proceedings of the ACM on human-computer interaction*, 3(CSCW):1–20.

Abeer ALDayel and Walid Magdy. 2021. Stance detection on social media: State of the art and trends. *Information Processing & Management*, 58(4):102597.

Kareem Darwish, Walid Magdy, Afshin Rahimi, Timothy Baldwin, Norah Abokhodair, et al. 2018. Predicting online islamophobic behavior after# parisattacks. *The Journal of Web Science*, 4(3):34–52.

Kareem Darwish, Peter Stefanov, Michaël Aupetit, and Preslav Nakov. 2020. Unsupervised user stance detection on twitter. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 14, pages 141–152.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Régis Ebeling, Carlos Abel Córdova Saenz, Jéferson Campos Nobre, and Karin Becker. 2022. Analysis of the influence of political polarization in the vaccination stance: The brazilian covid-19 scenario. *Proc. of the International AAAI Conference on Web and Social Media*, 16(1):159–170.

William Ferreira and Andreas Vlachos. 2019. Incorporating label dependencies in multilabel stance detection. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6350–6354.

Simone Giorgioni, Marcello Politi, Samir Salman, Roberto Basili, and Danilo Croce. 2020. Unitor@ sardistance2020: Combining transformer-based architectures and transfer learning for robust stance detection. In *EVALITA*.

Manuela Gómez-Suta, Julián Echeverry-Correa, and José A Soto-Mejía. 2023. Stance detection in tweets: A topic modeling approach supporting explainability. *Expert Systems with Applications*, 214:119046.

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.

Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Yuan Yao, Ao Zhang, Liang Zhang, et al. 2021. Pre-trained models: Past, present and future. *AI Open*, 2:225–250.

Kazuaki Hanawa, Akira Sasaki, Naoaki Okazaki, and Kentaro Inui. 2019. Stance detection attending external knowledge from wikipedia. *Journal of Information Processing*, 27:499–506.

Kornraphop Kawintiranon and Lisa Singh. 2021. Knowledge enhanced masked language model for stance detection. In *Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: human language technologies*, pages 4725–4735.

Mirko Lai, Delia Irazú Hernández Farías, Viviana Patti, and Paolo Rosso. 2017. Friends and enemies of clinton and trump: using context for detecting stance in political tweets. In *Advances in Computational Intelligence: 15th Mexican International Conference on Artificial Intelligence, MICAI 2016, Cancún, Mexico, October 23–28, 2016, Proceedings, Part I 15*, pages 155–168. Springer.

Yingjie Li and Cornelia Caragea. 2019. Multi-task stance detection with sentiment and stance lexicons. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 6299–6305.

Veronica Lynn, Salvatore Giorgi, Niranjan Balasubramanian, and H Andrew Schwartz. 2019. Tweet classification without the tweet: An empirical examination of user versus document attributes. In *Proceedings of the third workshop on natural language processing and computational social science*, pages 18–28.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016a. A dataset for detecting stance in tweets. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3945–3952.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016b. Semeval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 31–41.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Ammar Rashed, Mucahid Kutlu, Kareem Darwish, Tamer Elsayed, and Cansın Bayrak. 2021. Embeddings-based clustering for target specific

stances: The case of a polarized turkey. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 15, pages 537–548.

Umme Aymun Siddiqua, Abu Nowshed Chy, and Masaki Aono. 2019. Tweet stance detection using multi-kernel convolution and attentive lstm variants. *IEICE TRANSACTIONS on Information and Systems*, 102(12):2493–2503.

Amine Trabelsi and Osmar Zaiane. 2018. Unsupervised model for topic viewpoint discovery in online debates leveraging author interactions. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 12.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788.

Penghui Wei, Wenji Mao, and Guandan Chen. 2019. A topic-aware reinforced model for weakly supervised stance detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7249–7256.

Bowen Zhang, Min Yang, Xutao Li, Yunming Ye, Xiaofei Xu, and Kuai Dai. 2020. Enhancing cross-target stance detection with transferable semantic-emotion knowledge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3188–3197.