Contextual stance classification using prompt engineering

Felipe Penhorate Carvalho de Fonseca[1], **Ivandré Paraboni**[1], **Luciano Antonio Digiampietri**[1]

[1]Escola de Artes, Ciências e Huamnidades – Universidade de São Paulo (USP)
03828-000 – São Paulo – SP – Brazil

felipe.penhorate@gmail.com, {ivandre,digiampietri}@usp.br

Abstract. *This paper introduces a prompt-based method for few-shot learning addressing, as an application example, contextual stance classification, that is, the task of determining the attitude expressed by a given statement within a conversation thread with multiple points of view towards another statement. More specifically, we envisaged a method that uses the existing conversation thread (i.e., messages that are part of the test data) to create natural language prompts for few-shot learning with minimal reliance on training samples, whose preliminary results suggest that prompt engineering may be a competitive alternative to supervised methods both in terms of accuracy and development costs for the task at hand.*

1. Introduction

The increasingly popular use of transformers [Vaswani et al. 2017] in NLP and related fields, and the availability of so-called Large Language Models (LLMs), machine learning tasks based on text data have undergone significant changes in how training and classification may be implemented. For a start, these advances allowed a fixed *pre-trained* language model to be reused across multiple tasks by means of fine-tuning, that is, by adjusting a general (and sometimes multilingual) model to a particular downstream task [Liu et al. 2023]. This approach, also known as "pre-train, fine-tune", is perhaps best illustrated by the wide-spread popularity of models such as BERT [Devlin et al. 2019] in the NLP field.

Nowadays the release of progressively larger language models is commonplace and, accordingly, further advances in the field have followed. In particular, since the introduction of GPT-2 [Radford et al. 2019] and GPT-3 [Brown et al. 2020] models, we may speak of a second paradigm shift known as "pre-train, prompt and predict". In this approach, instead of fine-tuning a pre-trained language model to address a specific task, the task itself is reformulated to serve as an input to the language model as text data [Liu et al. 2023]. For instance, in a "pre-train, prompt and predict" approach to, e.g., sentiment analysis of film reviews, we may craft a natural language instruction - or prompt - as in "$[X]$ the movie is $[Z]$", in which X is the input text to be classified and Z is the label (e.g., 'good', 'bad', etc.) that we would like the model to predict. By following this method, known as *prompt engineering* [Liu et al. 2023], a sufficiently robust LLM should be able to fill in the prompt slots with the most likely tokens, which in turn would provide a sentiment class label Z without the need for any fine-tuning of model parameters and, crucially, requiring little or no labeled training data, in what is known as few- or zero-shot learning.

A particular NLP task that may potentially benefit from these methods is the stance classification task, which consists of identifying the point of view or judgment of an individual (e.g,, in favor, against, neutral, etc.) upon a target object of interest [ALDayel and Magdy 2021]. For instance, given the target 'vaccination', a statement as in '*I believe that everyone should be vaccinated compulsory*' conveys a stance in favor of vaccination. Stance classification may in principle seem analogous to sentiment analysis [Zhang et al. 2018]) but, unlike sentiment (e.g., positive, negative, etc.) stance (e.g., for, against, etc.) is anchored on a specific target, and stance and sentiment do not actually correlate [ALDayel and Magdy 2021].

Central to our present work, we notice that standard (i.e., supervised) approaches to stance classification will require a usually large amount of training data for every target under consideration. Thus, for instance, we may need a labeled corpus of stances towards, e.g., vaccination, and in a second project we may need a new corpus labeled with stances towards climate change, and so forth. This unlimited dependency on labeled data arguably makes stance classification an ideal testbed for prompt engineering methods.

Based on these observations, in this work we introduce a prompt-based method for few-shot text learning using stance classification as an application example. More specifically, we focus on the issue of *contextual stance classification*, that is, the task of determining the stance expressed by a given statement within a conversation thread with multiple points of view [Derczynski et al. 2017]. Our method consists of using the existing messages (which are part of the test data) as prompts for few-shot learning with minimal reliance on training samples and, in doing so, we would like to show that prompt engineering is a competitive alternative to supervised methods both in terms of accuracy and development costs.

The rest of this article is organized as follows. Section 2 discusses the concept of prompt engineering and how it compares to standard supervised learning. Section 3 describes related work in the field of stance classification. Section 4 presents the materials and methods used in our own work. Section 5 presents the results of our experiments. Finally, Section 6 draws a number of conclusions from the present work.

2. Background

In this section we introduce the issue of prompt-based learning and discuss how this compares to standard supervised approach.

Let x be an input (text), and let y be an output (label or text, for instance) based on a model $P(Y|x, \theta)$. Learning the parameters θ require a labeled set of input $- >$ output pairs, from which we may train a model to predict the described conditional probability [Liu et al. 2023]. In a supervised approach, learning is dependent on a set of training samples to obtain the probability $P(Y|x, \theta)$. However, it is often the case that a sufficiently large train dataset is not available for the required class or domain. Moreover, as discussed in the previous sections, we notice that standard target-based stance classification will require a specific training dataset for every target topic under consideration [Mohammad et al. 2016].

As an alternative to supervised learning, the recent availability of large language models (LLMs) has enabled the use of prompt-based methods in text classification tasks.

Prompt-based learning circumvents the lack of training data by creating language models that output the probability $P(x, \theta)$ based on x by itself, that is, models that can describe an output without the need to specify what the expected output would be in a training fashion [Liu et al. 2023].

Prompt-based learning comprises of three main steps: prompt addition, answer search, and answer mapping. In what follows we briefly review each of these steps in turn. Further details are provided in [Liu et al. 2023].

Prompt addition makes use of a prompting function $f_{prompt}(x)$ to modify an input text x into a prompt $x' = f_{prompt}(x)$. This consists of applying a pre-defined template to the input x based on an input slot [X] and output slot [Z], and then filling in the slot [X] with the input text x [Liu et al. 2023], thus creating a natural language instruction (or prompt) to be submitted to the language model. There are at least two methods for modeling the input [X] and the slot [Z] as text. The first method, called *cloze prompt*, takes place when slot Z appears in the middle of the text. The second method, called *prefix prompt*, takes place when the input appears entirely before [Z].

Answer search computes the highest-score text z' that maximizes the score of the language model [Liu et al. 2023]. This involves defining a set of permissible values Z for z, and then using a function $f_{fill}(x_0, z)$ to fill in the location [Z] in prompt x_0 with the candidate answer z. In classification tasks, we may define, e.g., $Z = \{excellent, good, OK, bad, horrible\}$ to represent a set of possible classes $Y = \{++, +, \ , -, --\}$. A prompt is said to be an *answer prompt* when it fills in the output slot correctly. [Liu et al. 2023].

Finally, answer mapping establishes a mapping between the computed answer z' and the target output value y. This step may in some cases be trivial but, since multiple answers may result in the same output (e.g., 'bad', 'very bad', 'horrible' etc. may all be mapped onto a '0' class label in a particular application), it is often necessary to establish a mapping from z' to y [Liu et al. 2023].

3. Related Work

Stance classification has been established as a major research topic in the NLP field since the SemEval stance detection shared task series [Mohammad et al. 2016, Derczynski et al. 2017] in 2016-2017, followed by RumourEval 2019 [Gorrell et al. 2018]. In what follows we briefly review these initiatives and the best-performing participant systems in each task.

SemEval 2016 Task 6 [Mohammad et al. 2016] introduced two stance detection tasks by providing a stance corpus of tweets in the English language. Task A addressed stance classification in a standard supervised setting, and Task B addressed the task in an unsupervised fashion. The SemEval 2016 corpus consisted of 4,163 tweets conveying a stance (for, against, or neutral) towards five target topics (Atheism, Climate Change, Feminist Movement, Hillary Clinton, and Abortion Legalization). An additional, unlabeled topic (Donald Trump) was used in (unsupervised) task B.

The three best-performing participant systems in SemEval 2016 Task 6 [Mohammad et al. 2016] were Mitra [Zarrella and Marsh 2016], Pkudblab [Wei et al. 2016] and Takelab [Tutek et al. 2016]. Mitra's approach was based on a

recurrent neural network, whereas Pkudblab [Wei et al. 2016] used a convolutional neural network. Takelab, on the other hand, took a different approach by using an ensemble of models created with the aid of a genetic algorithm.

Of particular interest to the present work, SemEval 2017 Task 8 [Derczynski et al. 2017], also known as RumourEval, introduced a novel Twitter dataset for stance classification that included contextual information represented by rumors associated with the stance target, and which could be used as an aid to the classification task. The RumourEval corpus is divided into pre-defined training and a test subsets. The training portion contains 297 conversations about 8 rumors discussed across 297 *tweets* that initiated a conversation thread, and 4,222 answers, making 4,519 *tweets* in total. The test dataset has 28 conversations, being 20 about the same rumors introduced in the training dataset, and 8 are about different rumors. Table 1 presents the class distribution of the train and test datasets.

Table 1. RumourEval 2017 class label distribution.

	Support	**Deny**	**Query**	**Comment**
Training	910	344	358	2,907
Test	94	71	106	778

The three best-performing participant systems in SemEval 2017 Task 8 were Turing [Kochkina et al. 2017], UWaterloo [Bahuleyan and Vechtomova 2017] and ECNU [Wang et al. 2017]. Turing proposed an approach based on a recurrent neural network with LSTM neurons and additional features derived from the training data. UWaterloo [Bahuleyan and Vechtomova 2017] based their approach mainly on feature selection and engineering, some of which manually curated with external knowledge provided by annotators, and using a XGBoost classifier. ECNU [Wang et al. 2017] combined an *ensemble* approach with hierarchical training to take advantage of the contextual information provided.

SemEval 2019 Task 7 [Gorrell et al. 2018] (also known as RumourEval 2019) improved upon the original RumourEval task definition by adjusting a number of issues found in the original dataset, and by adding data from Reddit. The three best-performing participant systems were BLCU NLP [Yang et al. 2019], BUT-FIT [Fajcik et al. 2019], and eventAI [Li et al. 2019]. BLCU NLP fine-tuned a Generative Pre-Trained Transformer (GPT) for contextual stance classification taking as an input the entire conversation history, and not only the target tweet. To this end, the conversation history was submitted to the model as a natural language prompt with tweets divided by separators, and the model was subsequently fine-tuned using a fully connected layer that followed the GPT layers. BUT-FIT's used a fine-tuned BERT model prompted with a contextual representation comprising the text that generated the conversation thread (i.e., the first text in a conversation), and the texts that appeared before and after it in the conversation. The eventAI approach, by contrast, did not use any LLM, proposing instead a recurrent neural network approach based on LSTM neurons alongside a rule-based model.

After the initial SemEval and RumourEval shared tasks, multiple stance classification datasets and models have been publicly released. These include, for instance, studies devoted to Arabic [Alhindi et al. 2021, Jaziriyan et al. 2021], Portuguese

[Won and Fernandes 2022], German [Gohring et al. 2021], and multilingual scenarios [Chen et al. 2022]. Moreover, although most studies are purely text-based, the issue of multimodal stance classification (e.g., combining text and social media relations or other knowledge sources) has also been investigated [Sakketou et al. 2022]. We notice also that some of these resources are considerably larger than the original SemEval corpus. This is the case, for instance, of the P-Stance corpus in [Li et al. 2021], comprising over 21k labeled tweets.

Finally, we notice that all of the above studies, including those that used an LLM in their architecture, addressed the issue of contextual stance classification in a standard supervised fashion, that is, none of them addressed the task using zero- or few-shot prompt engineering. Examples of this kind are only beginning to emerge in the field, and include, for example, [Yin et al. 2019, Zhang et al. 2023].

4. Materials and Methods

We envisaged an experiment in prompt-based learning to address the task of contextual stance classification as described in the previous sections. In what follows, we outline the materials and methods employed in the present work.

Our experiment makes use of the contextual stance data provided by the SemEval 2017 Task 8 corpus [Derczynski et al. 2017]. The corpus consists of a series of conversation threads in which individual messages may either *Support*, *Deny*, *Query* or *Comment* the root statement. This structure is illustrated in Figure 1, in which each example u_i consists of an input text x_i, an output yi and a context Ci, where Ci is every u_j that occurs before the current example in the conversational tree.

SDQC support classification. Example 1:

u1: We understand there are two gunmen and up to a dozen hostages inside the cafe under siege at Sydney.. ISIS flags remain on display #7News **[support]**

u2: @u1 not ISIS flags **[deny]**

u3: @u1 sorry - how do you know it's an ISIS flag? Can you actually confirm that? **[query]**

u4: @u3 no she can't cos it's actually not **[deny]**

u5: @u1 More on situation at Martin Place in Sydney, AU –LINK– **[comment]**

u6: @u1 Have you actually confirmed its an ISIS flag or are you talking shit **[query]**

SDQC support classification. Example 2:

u1: These are not timid colours; soldiers back guarding Tomb of Unknown Soldier after today's shooting #StandforCanada –PICTURE– **[support]**

u2: @u1 Apparently a hoax. Best to take Tweet down. **[deny]**

u3: @u1 This photo was taken this morning, before the shooting. **[deny]**

u4: @u1 I don't believe there are soldiers guarding this area right now. **[deny]**

u5: @u4 wondered as well. I've reached out to someone who would know just to confirm that. Hopefully get response soon. **[comment]**

u4: @u5 ok, thanks. **[comment]**

Figure 1. Two 'support' classification instances from [Derczynski et al. 2017].

In this scenario, our present approach makes use of the existing conversation thread to implement prefix prompt addition. More specifically, given an user who authored a sample message u_i, the prompt is introduced by using a structure as follows.

> *This is a conversation between some friends about an article in Twitter. They decided that they can only support, deny, query or add a comment about the article.*

The introductory statement is followed by the context C of the current example, its corresponding text [X], and the answer [Y] provided by the model, in the form "[C][X]. A:[Y]". The context itself comprising a series of (few-shot) query-answer example pairs in the format "**Q:** User said [X]. **A:** User wants to [Y] the article" according to the structure of the conversation. In our approach, all available contextual messages are taken as learning prompts and, if necessary, additional prompts are created as discussed below.

An example of the complete prompt structure is illustrated in Figure 2, in which the current text is shown in green, the expected answer [Z] appears in red, and the context [C] appears in blue. The [Y] labels for each example ares shown as **[Y]**.

This is a conversation between some friends about an article in Twitter. They decided that they can only support, deny, query or add a comment about the article.

Q: John said "We understand there are two gunmen and up to a dozen hostages inside the cafe under siege at Sydney.. ISIS flags remain on display #7News"

A: John wants to **[support]** the article

Q: Benjamin answered John "*[mention]* not ISIS flags"

A: Benjamin wants to **[deny]** the article

Q: Charlotte answered John "*[mention]* sorry - how do you know it's an ISIS flag? Can you actually confirm that?"

A: Charlotte wants to **[query]** the article

Q: Amelia answered Charlotte "*[mention]* no she can't cos it's actually not"

A: Amelia wants to **[deny]** the article

Q: Mia answered John "*[mention]* More on situation at Martin Place in Sydney, AU –LINK–"

A: Mia wants to **[comment]** the article

Q: Paul answered John "*[mention]* Have you actually confirmed its an ISIS flag or are you talking shit"

A: Paul wants to **[query]** the article

Figure 2. A prompt structure example and its expected output (in red).

In this representation, whenever an input is a reply to another message, the text changes to "**Q:** UserA answered UserB [X]. **A:** UserA wants to [Y] the article", in which User is a placeholder for the user to whom a message is assigned. To this end, the original username of each individual is replaced with a name within a pre-defined set of possible names $N = \{$*John, Paul, Lily, Noah, Olivia, James, Lucas, Emma, Amelia, Henry, Liam, Charlotte, Elijah, Ava, William, Sophia, Benjamin, Isabella, Mia, Evelyn, Theodore, Harper*$\}$.

The goal of the classifier is to complete the last query-answer pair in the sequence of the conversation with the intended class prediction, and for that reason it is imperative that context [C] includes at least one example of each possible answer. Since not all corpus

conversations are complete in this way, if necessary the context will be expanded with additional query-answer pair taken from the training portion of the RumourEval corpus, which is otherwise discarded. These additional samples are selected from messages with a time prior to the time of the current message and according to cosine similarity.

By following this procedure, prompts were engineered for every test instance in the RumourEval corpus, and then submitted to the OpenAI GPT 3.5 *text-davinci-003* model with default temperature. This choice was partially motivated by its ability to handle up to 4,097 tokens as an input, which is sufficiently large to handle most of the prompts generated by the present method.

As a generative model, GPT 3.5 may naturally provide answers in multiple formats and, accordingly, some form of answer mapping is called for. In the present work, answers are mapped onto *Deny*, *Support*, or *Query* class labels according to the presence of certain keywords, or otherwise mapped onto *Comment* class labels as summarized in Table 2.

Table 2. Keyword-based answer mapping.

Class	Keywords
Deny	deny, denies, denying
Support	back up, reinforce, support
Query	query, querying, queries
Comment	none of the above

5. Results

Table 3 presents RumourEval test data F1 results obtained by our prompt engineering approach, and by the two top-performing systems at RumourEval. The best results for each class are highlighted.

Table 3. RumourEval F1-score test results.

Approach	Overall	Comment	Deny	Query	Support
Turing [Kochkina et al. 2017]	0.43	**0.87**	0.00	0.46	**0.40**
UWaterloo [Bahuleyan and Vechtomova 2017]	0.45	**0.87**	0.06	0.49	**0.40**
Our work	**0.47**	0.76	**0.37**	**0.54**	0.22

Results in Table 3 show that our current work, although only outperforming the baseline systems in two individual tasks (*Deny* and *Query*), obtained overall highest F1-scores among the systems under evaluation. Moreover, our work was the only system capable of handling - albeit still in a limited fashion - the more challenging *Deny* task. This outcome, and the observation that our work, unlike the two baseline systems, does not require training data, suggest that the use of prompt-based methods for contextual stance classification may represent a compelling alternative to standard approaches that rely on model supervision.

6. Conclusion

This article introduced a few-shot method to contextual stance classification using test messages available from the current conversation thread (i.e., within which the target message occurs) to prompt a large generative model, with results that show improvement over the two best-performing participant systems at RumourEval. In addition to that, results also show a considerably higher accuracy in handling so-called 'Deny' statements if compared to previous work, which is a likely benefit of using a large language model for the task.

More importantly, unlike previous work in the field, we notice that the current results were obtained in a few-shot fashion, that is, with no reliance on a large training dataset. Thus, if taking into account its underlying development costs, the present approach affords a significant advantage over existing methods. This is particularly the case of manual corpus annotation, a task that, in standard stance classification, would normally have to be performed for every single target topic of interest, with substantial costs that are presently negligible.

The present work leaves a number of opportunities open to investigation. First, we notice that the current model may be further assessed using the extended RumourEval 2019 dataset in [Gorrell et al. 2018], or other similar resources. Second, we may consider alternative prompt engineering methods including, for instance, enriching the prompt instructions with external knowledge about the conversation topic (e.g., from news articles, Wikipedia, etc.) Moreover, the present approach may in principle be applied to other text classification tasks based on contextual information including, for instance, sarcasm or sentiment detection.

References

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

Alhindi, T., Alabdulkarim, A., Alshehri, A., Abdul-Mageed, M., and Nakov, P. (2021). AraStance: A multi-country and multi-domain dataset of Arabic stance detection for fact checking. In *4th Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 57–65, Online. Assoc. for Computational Linguistics.

Bahuleyan, H. and Vechtomova, O. (2017). UWaterloo at SemEval-2017 task 8: Detecting stance towards rumours with topic independent features. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 461–464, Vancouver, Canada. Association for Computational Linguistics.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners.

Chen, N., Chen, X., and Pang, J. (2022). A multilingual dataset of covid-19 vaccination attitudes on twitter. *Data in Brief*, 44:108503.

Derczynski, L., Bontcheva, K., Liakata, M., Procter, R., Wong Sak Hoi, G., and Zubiaga, A. (2017). SemEval-2017 task 8: RumourEval: Determining rumour veracity and support for rumours. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 69–76, Vancouver, Canada. Association for Computational Linguistics.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Fajcik, M., Smrz, P., and Burget, L. (2019). BUT-FIT at SemEval-2019 task 7: Determining the rumour stance with pre-trained deep bidirectional transformers. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 1097–1104, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Gohring, A., Klenner, M., and Conrad, S. (2021). DeInStance: Creating and evaluating a german corpus for fine-grained inferred stance detection. In *17th Conference on Natural Language Processing (KONVENS 2021)*, pages 213–217, Düsseldorf, Germany. KONVENS 2021 Organizers.

Gorrell, G., Bontcheva, K., Derczynski, L., Kochkina, E., Liakata, M., and Zubiaga, A. (2018). Rumoureval 2019: Determining rumour veracity and support for rumours.

Jaziriyan, M. M., Akbari, A., and Karbasi, H. (2021). ExaASC: A General Target-Based Stance Detection Corpus in Arabic Language. In *11th International Conference on Computer Engineering and Knowledge (ICCKE)*, pages 424–429, Mashhad, Iran. IEEE.

Kochkina, E., Liakata, M., and Augenstein, I. (2017). Turing at SemEval-2017 task 8: Sequential approach to rumour stance classification with branch-LSTM. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 475–480, Vancouver, Canada. Association for Computational Linguistics.

Li, Q., Zhang, Q., and Si, L. (2019). eventAI at SemEval-2019 task 7: Rumor detection on social media by exploiting content, user credibility and propagation information. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 855–859, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Li, Y., Sosea, T., Sawant, A., Nair, A. J., Inkpen, D., and Caragea, C. (2021). P-stance: A large dataset for stance detection in political domain. In *Findings of ACL-IJCNLP-2021*, pages 2355–2365, Online. Assoc. for Computational Linguistics.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., and Neubig, G. (2023). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9).

Mohammad, S., Kiritchenko, S., Sobhani, P., Zhu, X., and Cherry, C. (2016). SemEval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 31–41, San Diego, California. Association for Computational Linguistics.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners.

Sakketou, F., Lahnala, A., Vogel, L., and Flek, L. (2022). Investigating user radicalization: A novel dataset for identifying fine-grained temporal shifts in opinion. In *LREC-2022 proceedings*, pages 3798–3808, Marseille, France. ELRA.

Tutek, M., Sekulić, I., Gombar, P., Paljak, I., Čulinović, F., Boltužić, F., Karan, M., Alagić, D., and Šnajder, J. (2016). TakeLab at SemEval-2016 task 6: Stance classification in tweets using a genetic algorithm based ensemble. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 464–468, San Diego, California. Association for Computational Linguistics.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Wang, F., Lan, M., and Wu, Y. (2017). ECNU at SemEval-2017 task 8: Rumour evaluation using effective features and supervised ensemble models. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 491–496, Vancouver, Canada. Association for Computational Linguistics.

Wei, W., Zhang, X., Liu, X., Chen, W., and Wang, T. (2016). pkudblab at SemEval-2016 task 6 : A specific convolutional neural network system for effective stance detection. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 384–388, San Diego, California. Association for Computational Linguistics.

Won, M. and Fernandes, J. (2022). SS-PT: A stance and sentiment data set from portuguese quoted tweets. In *PROPOR-2022 proceedings*, pages 110–121, Fortaleza, Brazil. Springer.

Yang, R., Xie, W., Liu, C., and Yu, D. (2019). BLCU_NLP at SemEval-2019 task 7: An inference chain-based GPT model for rumour evaluation. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 1090–1096, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Yin, W., Hay, J., and Roth, D. (2019). Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach.

Zarrella, G. and Marsh, A. (2016). MITRE at SemEval-2016 task 6: Transfer learning for stance detection. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 458–463, San Diego, California. Association for Computational Linguistics.

Zhang, B., Ding, D., and Jing, L. (2023). How would Stance Detection Techniques Evolve after the Launch of ChatGPT?

Zhang, L., Wang, S., and Liu, B. (2018). Deep learning for sentiment analysis: A survey. *WIREs Data Mining and Knowledge Discovery*, 8(4):e1253.