# GraphNLI

ELAD LAKS, OR KATZ

Submitted as final project report
IDC, 2022

## 1 Introduction

Talk-backs and debates of users on public discussion on forums can escalate into hate or misinformative posts.(1) To be able to tackle or understand those posts the "GraphNLI" paper tries to classify whether a reply is supporting or attacking the post it is replying to. The problem is that when trying to identify the polarity between post and reply, the reply to a post may be based on an external context beyond the post. to explain the problem the paper defines the "local context" as the context between a post and its reply, and "global context" as the context to all the other replies before and after the local context, including the local context. The paper suggests a method to overcome the problem using the other replies and the replies to the reply from the 'root post'. They suppose it will give a lot more information to decide whether the reply is supporting or attacking the post it is replying to. They named this method "GraphNLI". They compared several NLP models and methods on a data-set from Kialo - a debate web, where every time user uploads a reply to a post he first labels his reply with an attack/support flag. All the methods based on GraphNLI were better than the others, and the best one was "GraphNLI: Root-seeking Graph Walk + Weighted Avg". It was inspired based on S-BERT using RoBERTa, Pooling, and aggregation between the results making the final embedding vector which is fed into Softmax-classifier in the end. The specialty in this method is the way the graph walks considering all the "tree" of replies and not only the reply and its post(considering the global context).

### 1.1 The final project mission

First, we will classify whether a reply is supporting or attacking the post it is replying to, based on the Kialo data set and the method of the anchor paper.
Then, we will use the method on different task - to classify whether a post is hate speech or not.

### 1.2 The data sources

Kialo data-set link to the data
Hate data-set link to the data

### 1.3 The evaluation method

Because we have labeled data, we split the data into train-test sets and evaluated our job with accuracy measure.

# 2 Related Works

## 2.1 A natural language bipolar argumentation approach to support users in online debate interactions (2013)

One of the first papers that tried to predict support or attack context between post and reply was published in 2013.(2) Obviously they were using old algorithms and tools. The paper proposes and evaluates the use of NLP techniques to identify the post and reply and their relations. In the paper, they used only the "local context" of the post and the reply and didn't use complex language models such as bert or deep learning tools. In particular, they adopted the textual entailment (TE) approach, a generic framework for applied semantics, where linguistic objects are mapped by means of semantic inferences at a textual level. TE is then coupled together with an abstract bipolar argumentation system which allows identifying the arguments that are accepted in the considered online debate. The paper archived an accuracy of 69%.

## 2.2 Identifying attack and support argumentative relations using deep learning (2017)

In this paper, they propose a deep learning architecture to capture argumentative relations of attack and support from one piece of text to another, of the kind that naturally occurs in a debate.(3) The architecture uses two (unidirectional or bidirectional) Long ShortTerm Memory networks and trained word embeddings and allowed to considerably improve upon existing techniques that use syntactic features and supervised classifiers for the same form of (relation-based) argument mining. The method was a deep learning architecture for Relation-based AM based on Long-Short Term Memory, within the architecture, each input text is fed, as a trained 100-dimensional GloVe embedding, into a (unidirectional or bidirectional) LSTM which produces a vector representation of the text independently of the other text being analyzed. The two vectors are then merged (using element-wise sum or concatenation) and the resulting vector is fed to a softmax classifier which predicts whether the pair of input texts belongs to the attack, support, or neither relations. the architecture achieved 89.53% (better than our paper). As our paper mentioned, the data and the code of this paper is not available so we couldn't compare between the models. Also in this paper, they used only the "local context" for the classification mission.

## 2.3 Modeling online debates with argumentation theory (2022)

This article overviews three recent pieces of work from argumentation theory to a better and scale understanding of online debates.(4)It presents 3 tools of automatic analysis and aggregating online debate information from a structured source ('Kialo"). first, they present the GraphNLI work which automatically identifies reply polarity (agreement or disagreement) between arguments submitted in online debates, then, locates where the justified arguments are likely to be and how that depends on the "degree of antagonism" of the debate using probabilistic model s.a. complex networks and simulations non-homogeneous trees, and lastly, present summarization of the arguments made in debates such that a reader would get as many of the justified arguments as possible without having to read the entire debate since the average reader will not be able to read the entire debate, and it is possible to read the "wrong" part of the debate and be misled in one's beliefs, found that present to the reader topological order of oldest to newest posted had statistically significantly higher evaluation result.

# 3 Anchor paper - GraphNLI

A Graph-based Natural Language Inference Model for Polarity Prediction in Online Debates
code git repositories

## 3.1 The data-set

The data set contains 324373 posts and replies, each one of them is a part of the "conversation tree". we split the data into train(80%) and test (20%). each conversation tree represents the global context, of the local context between 2 sentences in the tree.


Figure 1: convesation

## 3.2 Graphnli method

### 3.2.1 Distilroberta-base

The GraphNLI paper is mainly based on pre-trained model called distilroberta based on bert.(5) this is a distilled version of the RoBERTa-base model. On average DistilRoBERTa is twice as fast as Roberta-base. Roberta-base model was created by changing the training of the original BERT, including a new and larger data set, far more iterations, and much larger batches. It was pretrained with the Masked language modeling (MLM) objective and without labeled data. This way, the model learns an inner representation of the English language that can then be used to extract features useful for a downstream task which makes it modular and robust. It trained with 160GB of text, 500K iterations, 8K batch size, and 50K sub-words units.

### 3.2.2 Garph walk architecture

To catch the global context without losing the most important context(the local context) they were using the graph walk method called - Weighted Root-seeking GraphWalk. This means to walk starting from a given node up towards the root of the discussion tree. They limited the walk to 5 steps and weighted the nodes as follows constant(k = 0.75) power L = the number of steps from the given node $wi = k^{Li}$. At the end of a graph

walk for each node, we obtain at most L=5 arguments which are an ancestor of a given node. These sets of arguments are the input of the GraphNLI model.

#### 3.2.3 Model overview

Each of the L=5 arguments is input into the distilroberta-base model to get their corresponding embeddings, then, a mean-pooling operation is applied to derive a fixed-sized sentence embedding for each argument. They separated the given node(the point of interest) embeddings and the aggregated embeddings of its maximum L ancestors. The aggregation is with a weighted average, according to the weight calculation explained in the previous paragraph. Then the embedding vectors concatenate with $|u - v|$ to get the final embedded vector which is fed into the Softmax classifier for the prediction task.

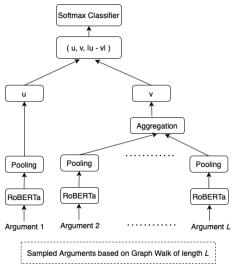


Figure 2: garph walk architecture

### 3.3 Our Coding effort

#### 3.3.1 The implementation of the method

We constructed the code in Colab notebook. First, achieved the data the anchor paper author used for the research, to end with similar results to their paper. We stored it in google drive, to be able to load them to the Colab notebook.
Then we cloned the git attached to the anchor paper. We found a few code mistakes and fixed them, as described in the next subsection. we opened our own git for the project and store the fixed code files in it.
The Colab notebook is built with the imports first, including uploading the data and cloning our GitHub to the notebook. We added to this section set_seed function to avoid randomness in the results.
Then, we let 2 options for training the model - to run a new model or to load and run a fine-tuned model. when loading fine-tuned model we need to load 3 files:
1: the data split to train and test sets used to train the model.
2: the folder of the trained model.
3: the trained Softmax model.
The training stage is identical in the 2 options - the data from the CSV files are used to build data loaders, and we choose the batch size, the number of epochs, and the graph-walk length and construct the train-loss from the Softmaxloss. Then the fit stage starts, we added a checkpoint every 10000 iterations that were saved in the drive in case of collapse, and choose the evaluation step of 1000. after the fit is ended, we evaluated our accuracy using labelAccuracyevaluator(sentence_transformers.evaluation).

### 3.3.2 Code mistakes in the original code

There were a few code mistakes in the original code, as follows:

1. In gen_dataset_graph_walks - the construction of the data-frame from the graph walk used a maximum of 4 sentences from each conversation. to walk with walklen= 5 they needed 6 sentences.

2. In SoftMaxLoss - they used mean aggregate instead of weighted_aggregate.

There was a great effort to find code mistakes in the published version of the code – it was not using the right function and arguments, the graph walk was to short and the code was not working.
After fixing all the errors, start to train distill-roberta model, as the paper asked. 4 epochs took 20 hours on GPU of Colab pro machine (Tesla P100-PCIE-16GB).

## 3.4 link to our notebook/git

github
colab

## 3.5 Results

### 3.5.1 Results of the paper

The original paper used learning rate $2 * 10^{-5}$, warm-up over 10% of the data. Batch size of 16, and Adam optimizer. The paper's best result came from the combination of root seeking graph walk, with weighted avg. they have got 82.87% accuracy.

**Table 1: Accuracy scores of different models trained on Kialo dataset for polarity prediction, discussed in Section 5.4.**

| Model | Accuracy (%) |
|---|---|
| Bag-of-Words + Logistic Regression | 67.00 |
| Prompt Embeddings + Logistic Regression | 61.20 |
| Sentence-BERT with classifier layer | 79.86 |
| BERT Embeddings: Root-seeking Graph Walk + MLP | 70.27 |
| GraphNLI: Root-seeking Graph Walk + Sum | 80.70 |
| GraphNLI: Root-seeking Graph Walk + Avg. | 81.96 |
| GraphNLI: Root-seeking Graph Walk + Weighted Avg. | **82.87** |
| GraphNLI: Biased Root-seeking Random Walk + Sum | 79.95 |
| GraphNLI: Biased Root-seeking Random Walk + Avg. | 80.44 |

Figure 3: paper results

### 3.5.2 Our results

After 20 hours of training with the same hyper-parameters of the anchor paper except batch = 12 (because of memory limitations) we achieved 82.41% accuracy. This is a reasonable gap, which we assume occurred due to the fact we used a smaller batch size, causing noisier training and the model has a smaller chance to converge to the global optima which might be the one reported in the original paper. one more reason might be the different random initialization of some of the model parameters and weights.

# 4 Innovative part

The first mission we wanted to accomplish was to improve the model. We achieved a better accuracy score than the anchor paper - 83%.
The second mission we wanted to accomplish is to use the method of GraphNLI on another mission - to classify hate speech.

## 4.1 First mission - Improve the model

To improve the performance of the original paper architecture, we tested a few possible adjustments which seem reasonable to us. All experiments share the same settings as reported above. we decided to focus on 3 options for changing the model. 1: to change the pre-trained distill-roberta model to another model. 2: to change the aggregation method of the softmax. 3: to change the classifier of the model. The changes of the aggregation and classifier were in the Softmax class, in which we upload the final .py file to our git.

### 4.1.1 First experiment - Change the model

We used few other models in order to get better results. we used "sentencetransformers/ all-mpnet-base-v2", "cross-encoder/nli-deberta-v3-base","sentencetransformers/ all-MiniLM-L6-v2". All of them was either really slow and we couldn't afford enough training time or got lower accuracy. ","sentencetransformers/all-MiniLM-L6-v2" was the best of them and achieved 82.23% accuracy.

### 4.1.2 Second experiment - Change the classifier

The second experiment focuses on the head of the model architecture, the 'classifier', which receives the features vector and uses it to classify between two labels. The original paper uses a single linear layer shape 3*768X2. In order to allow the model to learn more complex global patterns, we change the classifier into 2 linear layers shape 3*768X256 and 256X2 with Relu function as non-linearity between them. The motivation was to help in extracting the global relationship between the features. We add more fully connected layers with non-linearity to make sure these nodes interact well and account for all possible dependencies at the feature level. Indeed, we received better results than the original architecture, with 82.85 % Accuracy compared to 82.41 % Accuracy.

### 4.1.3 Third experiment - Change the aggregate method

The third experiment focuses on the aggregate methods of U and V, the embedded of the target post/sentence, and its surroundings. We followed the original paper reports, choosing as a baseline the features vector built from a concatenation of U, V, —U-V—. In order to improve the original paper results, it seems logical for us to add one more metric score to the concatenation features vector. As we know, in NLP the word or sentence embedded space build such that the distance and the angle or direction between two entities in the space have meaning, hence, we chose a metric that represents the direction of the two entities U and V using cosine similarity. Now the new features vector is composed of the concatenation of U, V, —U-V—, cosine(U, V). Here again, we received better results than the original architecture, with 82.54% Accuracy compared to 82.41% Accuracy.

#### 4.1.4 combined experiment - combination of the Third and second experiments

Since the two previous experiments were successful, we decided to implement and integrate the two upgrades in the original model. We assume that the combination of both changes in one model will result in the best results among the experiments reported so far. The results of this experiment, building combined model is Accuracy of 82.97%, which is better result than the original paper report 82.87% (we received it with less resources, forced us to use smaller batch size).

| Model | Accuracy |
|---|---|
| Anchor paper, batch size 16 | 82.87 |
| Our Model, batch size 12 | 82.41 |
| all-MiniLM-L6-v2, batch size 12 | 82.31 |
| Our Model + fully connected layers , batch size 12 | 82.85 |
| Our Model + cosine similarity, batch size 12 | 82.54 |
| Our Model + fully connected layers + cosine similarity, batch size 12 | **82.97** |

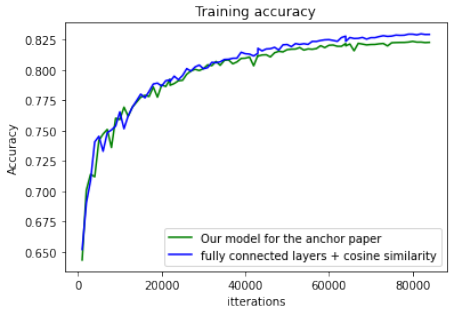


Figure 4: accuracy

### 4.2 second mission - Use the model to classify hate-speech

#### 4.2.1 The problem

In the last part of the anchor paper, they talked about future possible usage of the method, one of which was hate speech recognition.

#### 4.2.2 The data-set

The main problem was finding a data set that fits this mission. After a great effort of searching the web for the right dataset - with tree structure of post and reply, labeled to hate speech, We found this paper- "A Benchmark Dataset for Learning to Intervene in Online Hate Speech" (6) which made auto-reply to offensive language. The data was unordered, and we needed to pre-process it to fit our model method.

#### 4.2.3 Experiments

We assumed the ideal walk length to consider all the information of the global context is lower for the mission of classifying hate speech than to find whether a reply to a post is attacking or supporting it. So we search for the best length between 1 to 5.

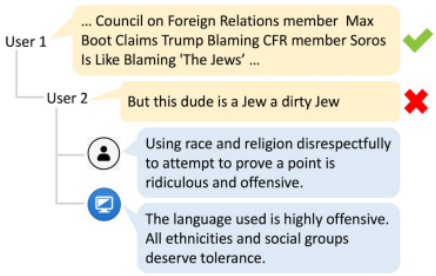


Figure 5: offensive language

#### 4.2.4 Results

In the experiments we large jump between walk length = 1 to walk length = 2, and we can see that higher the walk length higher the accuracy.

| Model | Accuracy |
|---|---|
| walk length = 1 | 0.858 |
| walk length = 2 | 0.920 |
| walk length = 3 | 0.920 |
| walk length = 4 | 0.926 |
| walk length = 5 | **0.926** |

#### 4.2.5 comparison to other methods

We found The paper "A Literature Review of Textual Hate Speech Detection Methods and Datasets" (7) from 2022 which gives a review of other methods used to detect hate speech. We found there is a high variance between papers and data sets, and it is very hard to determine how good our results without checking other models and methods on the same data. We can see that most of the best methods for each data set didn't achieve higher accuracy than 85 %. So, we can state that our method achieved high accuracy, much higher than the methods reviewed in the paper.

| Paper | Dataset | Best Method | Results | Limitation |
|---|---|---|---|---|
| [46] | Semeval-2019 task 5: multilingual detection of hate speech against immigrants and women on Twitter(19,600 tweets—13,000 in English and 6600 in Spanish) | LIBSVM with RBF | TASK 1: hateful or not: 0.58 accuracy TASK 2: individual or generic: 0.81 accuracy TASK 3: aggressive or not: 0.80 accuracy | Focused on detection of hate speech against immigrants and women on Twitter (HatEval). |
| [116] | 2228 sarcastic tweets | RF | 0.83 accuracy | Most of the sarcastic tweets do not fall in the category of sarcasm where a positive sentiment contrasts with a negative situation. Some authors did not recognize sarcasm as hate speech. |

Figure 6: other model results

# 5 Summary

## 5.1 What was achieved

Classification of supporting or attacking relationship between a post and a reply was found to be a hard mission. The main reason is there is a global context between the post and reply that does not take into consideration, even context that we can't find in the post tree such facts and knowledge. Our method tried to overcome part of this problem and succeeded partly.

## 5.2 comparison to related work

We succeeded to improve the anchor paper, and far as we know this is the best accuracy achieved for this mission. Also, using the method for hate speech detection, achieved outstanding accuracy as well, and proved the connection between the global context to hate speech detection.

## 5.3 (scientific) Insights gained

We discovered in our work that the anchor paper model was not enough complex, and that adding parameter such as cosine similarity and fully connected layer improved it's performances. Also, we proved that we will get higher accuracy for hate speech detection when using the previous post to a reply.

## 5.4 Open questions and future direction for further research

In the future, we think it's interesting to investigate the connection between the global context and to few more conversation missions as fake news. Also, we think it is possible to predict the popularity of a reply to a post before its sent to everyone. We have asked ourselves if there are better pretrained language models than distil-roberta for hate speech recognition because the hate language is different, but we didn't find a better model.

# 6 Code

colab notebook

# References


[1] Agarwal, Vibhor, Sagar Joglekar, Anthony P Young, and Nishanth Sastry: *Graphnli: A graph-based natural language inference model for polarity prediction in online debates*. In *Proceedings of the ACM Web Conference 2022*, pages 2729–2737, 2022.

[2] Cabrio, Elena and Serena Villata: *A natural language bipolar argumentation approach to support users in online debate interactions†*. Argument & Computation, 4(3):209–230, 2013. https://doi.org/10.1080/19462166.2013.862303.

[3] Cocarascu, Oana and Francesca Toni: *Identifying attack and support argumentative relations using deep learning*. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1374–1379, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. https://aclanthology.org/D17-1144.

[4] Young, Anthony P., Sagar Joglekar, Vibhor Agarwal, and Nishanth Sastry: *Modelling online debates with argumentation theory*. SIGWEB Newsl., (Spring), may 2022, ISSN 1931-1745. https://doi.org/10.1145/3533274.3533278.

[5] Sanh, Victor, Lysandre Debut, Julien Chaumond, and Thomas Wolf: *Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter*. ArXiv, abs/1910.01108, 2019.

[6] Qian, Jing, Anna Bethke, Yinyin Liu, Elizabeth M. Belding-Royer, and William Yang Wang: *A benchmark dataset for learning to intervene in online hate speech*. In *EMNLP*, 2019. https://arxiv.org/pdf/1909.04251.pdf.

[7] Alkomah, Fatimah and Xiaogang Ma: *A literature review of textual hate speech detection methods and datasets*. Information, 13(6), 2022, ISSN 2078-2489. https://www.mdpi.com/2078-2489/13/6/273.

[8] Chi Sun, Xipeng Qiu, Yige Xu Xuanjing Huang: *How to fine-tune bert for text classification?* 2020. https://arxiv.org/pdf/1905.05583.pdf.

[9] Reimers, Nils and Iryna Gurevych: *Sentence-bert: Sentence embeddings using siamese bert-networks*. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, November 2019. http://arxiv.org/abs/1908.10084.

[10] He, Pengcheng, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen: *Deberta: Decoding-enhanced bert with disentangled attention*. In *International Conference on Learning Representations*, 2021. https://openreview.net/forum?id=XPZIaotutsD.

[11] Qian, Jing, Anna Bethke, Yinyin Liu, Elizabeth Belding, and William Yang Wang: *A benchmark dataset for learning to intervene in online hate speech*, 2019. https://arxiv.org/abs/1909.04251.