Detection of Propaganda Techniques in News Articles

Vishwas Nalka
University at Buffalo
50419408
vishwasn@buffalo.edu

Vishal Raman
University at Buffalo
50376944
vraman2@buffalo.edu

Abstract

This paper presents the solution to detect and classify propaganda techniques. The main idea of our solution is to use RoBERTa model to train and predict the span of text containing propaganda, and predicting which class of propaganda this span belongs to. We deep dive into the preprocessing steps done, the way we handled the RoBERTa Tokeniser, and how we dynamically split sentences to retain context. Apart from this we provide the comparison of metrics between various models and various approaches we took, as well as some thoughts on how this work can be improved to achieve better results potentially.

1 Introduction

Propaganda is used to force a specific agenda on people and influence their stance on that topic. The commonly used ways to achieve this are logical fallacies and emotional language. In the former case, misrepresented points and data are used to deter the sense of logic. And in the latter case, the audience is persuaded to side with the speaker based on emotional manipulation and wording.

The objective of this project is to detect such propaganda in articles. This is done in two sub-tasks: Propaganda Technique Classification (TC) and Span Identification (SI). The goal of the TC task is to classify a given span and the context into one of the 14 defined propaganda techniques. There could be multiple techniques in a span also. The objective of the SI task is to identify the span using character indexes that contain a propaganda technique in the given text document. A single document can have multiple such spans containing propaganda techniques, and all such spans have to be identified.

We approach this problem in two phases. An additional task of Sentence Level Classification was introduced by us to classify a sentence as Propaganda or not. If the sentence is classified as Propaganda, we then proceed to the subsequent tasks of SI and TC. The Task TC is solved as a Multi-Label Multi-Class classification task and the Task SI is solved as a Sequence Token Classification problem (Rogge,). Both the tasks were solved using a fine-tuned RoBERTa (HuggingFace,) model. We use a RoBERTa based Transformer model for both the Tasks SI and TC, by fine-tuning the model to each of the tasks using the respective datasets. For the SI task, we split the sentences dynamically into equal length, and we preserve the context before tokenising and eventually feeding these sentences into the model. We also explore the ignore index present in RoBERTa to let the model ignore extra tokens generated by its tokeniser. For the Task TC, we utilise the Sigmoid functions to squash the logits returned by the RoBERTa model. This helped us in achieving Multi-Label Multi-Class classification.

The rest of the paper is organised as follows: Section 2 presents the literature survey and the approaches already presented before for this task. Section 3 talks about our approach, and elaborates on the Architecture, the various methods we tried, and the implementation details. Section 4 compares the results we achieved using various approaches. Section 5 briefs about the future directions that could potentially improve the results.

2 Related Work

The advancement in complex neural network based Transformer architectures has enabled the building of state-of-the-art Language Models such as BERT, GPT-3, etc. These models are utilized in implementing cutting-edge solutions for NLP tasks such as Machine Translation, Question Answering, Text

Classification, Summarization. The neural networks based models are widely used to solve the propaganda detection task. Multiple high-scoring solutions used modified versions of transformer models (mostly BERT) to identify the propagandistic article. The baseline algorithms devised by (Da San Martino et al., 2019). consisted of 3 models based on BERT (BERT, BERT-Joint, BERT-Granularity). These models included additional linear layers on top of the BERT architecture, which was fine-tuned, for the propaganda classification and span-identification task. Their proposed solution for the task was a multi-granularity network, also based on BERT, that used the low-granularity information to solve the higher granularity task. The model output from paragraph-level or sentence-level is the main task. And the output from higher-granular levels such as word-level, subword-level, character-level is used to improve the performance.

Another top-scoring model (2020 TC task winner and 2nd position in SI task) developed by Team ApplicaAI (Jurkiewicz et al., 2020) was based on RoBERTa model. They implemented an incremental semi-supervised learning algorithm, also referred to as self-training, using a RoBERTa-CRF architecture. The initial model was trained on the given dataset (called the gold data) and this model was used to annotate a different dataset which was called the silver data. Another model was trained using both the gold data and silver data. The final model was an ensemble of both the initial model and the retrained model. One of the solutions was an ensemble model based on Neural Architectures presented by (Gupta et al., 2019). They used three different models, Logistic regression, Convolutional Neural Network, and a fine-tuned BERT model. The prediction from each model was collected for each sentence and used two ensemble strategies were considered. One strategy was majority-voting in which the prediction is propaganda if the majority predict it as propaganda. In case of conflict, consider the output of the model with the highest F1 score. And the other strategy was relax-voting where the prediction was propaganda if with certain confidence models in the ensemble predict as propaganda. Finally, for Span Identification and Technique Classification, based on the classification from the previous model, a LSTM-CRF and BERT based ensemble is utilized.

3 Model architecture

3.1 Overview

The solution is presented in two phases, like the approach used by (Gupta et al., 2019). In the first phase, sentence level propaganda classification is done. In the second phase, only the sentence that are classified are propaganda are considered. Here with the help of RoBERTa model predictions are found. The exact models are mentioned below.

Task TC: Fine tuned model RobertaForSequenceClassification (base model loaded from pertained ‘roberta-base’)

Task SI: Fine tuned model RobertaForTokenClassification (base model loaded from pertained ‘roberta-base’)

The overall architecture of the solution is as given below:

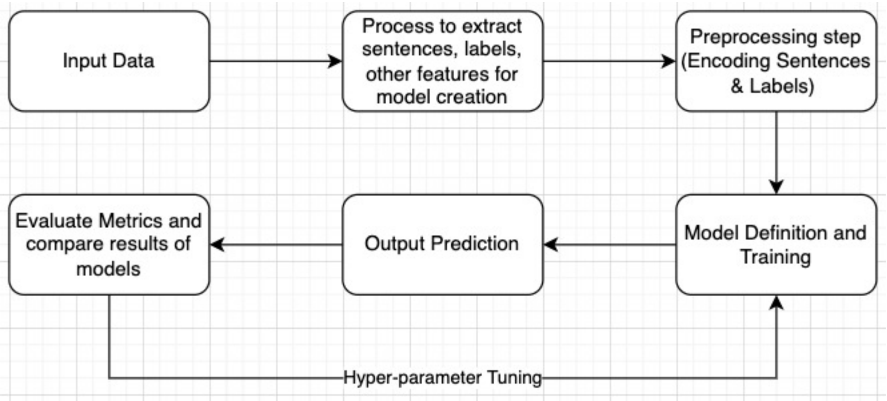

Input Data
Process to extract sentences, labels, other features for model creation
Preprocessing step (Encoding Sentences & Labels)
Model Definition and Training
Output Prediction
Evaluate Metrics and compare results of models
Hyper-parameter Tuning

Figure 1: Sentence Level Classification

3.2 Preprocessing

The new articles and the spans of the propaganda were provided in separate files. The span information was used to extract the actual texts/sentences. The given new article is split into multiple sentences based on period character or a newline character.

The tokenization and encoding of the sentences is done using BERT/RoBERTa Tokenizer. We tried converting the text to lowercase, removed stop-words, and lemmatised the text, to check if these improved the scores. But, these steps did not yield any significant changes. (In fact, retaining the case and using the 'bert-base-cased' model yielded slightly higher score).

The span information is just start-end index. This is converted to character level 1/0 labels, which is further converted to word-level 1/0 labels and finally this is converted to sentence level 1/0 labels. Depending on the task, the labels at the required granularity is used.

3.3 Implementation

Phase 1 : Sentence Level Classification This step involves going over each sentence and classifying it as a propaganda or not propaganda. If it is classified as propaganda, the sentence is considered in the second phase for Fragment Level Classification for Span Identification and Technique Classification. The training data for this model was created by us using the data already available for the other tasks. This task is treated as a "Binary Classification Task" and is implemented using RoBERTa model, BERT Sequence Classifier models with num-labels set to 2. The workflow of building the model is as follows:

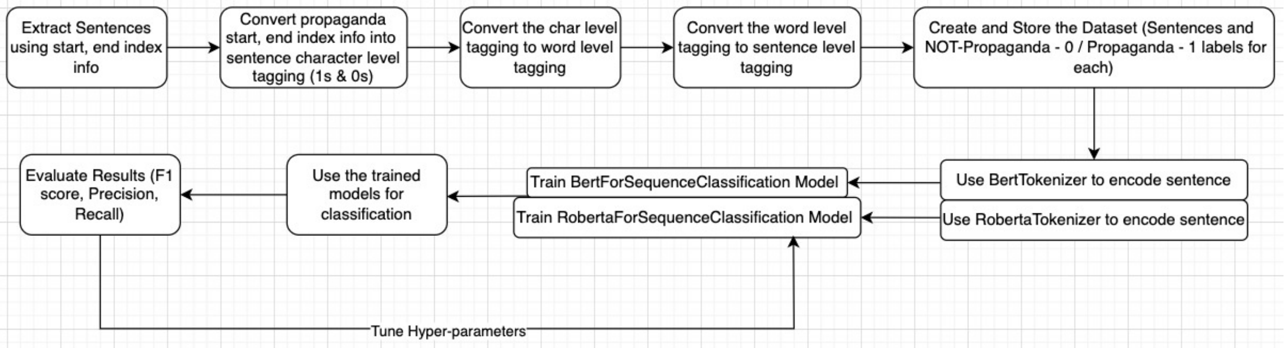

Extract Sentences using start, end index info
Convert propaganda start, end index info into sentence character level tagging (1s & 0s)
Convert the char level tagging to word level tagging
Convert the word level tagging to sentence level tagging
Create and Store the Dataset (Sentences and NOT-Propaganda - 0 / Propaganda - 1 labels for each)
Use BertTokenizer to encode sentence
Use RobertaTokenizer to encode sentence
Train BertForSequenceClassification Model
Train RobertaForSequenceClassification Model
Use the trained models for classification
Evaluate Results (F1 score, Precision, Recall)
Tune Hyper-parameters

Figure 2: Sentence Level Classification

Phase 2 : Fragment Level Technique Classification:

a) Span Identification: The dataset for this sub-task contains files representing the actual new articles and for each news article, Article ID, starting character index, ending character index are provided. The architecture for Span Identification involves using a RoBERTa Transformer model. The start and end index of training and dev data are given at character level, and these are transformed to word level labels before feeding into the model. Since the RoBERTa model accepts sentences that have at most 512 words, each sentence is split into multiple sentences of 256 length each dynamically. The data is tokenized using RoBERTa Tokeniser. They are then encoded and five fields are created, namely attention mask, input ids, labels, offset mapping and token type ids. Since RoBERTa Tokenizer creates extra words, 256 word limit was chosen. If a higher number had been chosen there is still a chance that after tokenising the word length would be more than 512, and hence the last few words would get trimmed. Also to retain context, last 50 words from the previous sentence are used for the subsequent sentence while splitting.

While tokenising using BertTokenizer, there are a lot of unwanted labels. For example, the word "investigating" is split into ('in', '##vs', '##ti', '##gating'). This creates extra tokens, and hence extra unwanted labels. Similarly, in case of RobertaTokenizer, additional word pieces are generated which increase the number of overall tokens. So to cater to these labels, during training, a label of -100 is assigned to such tokens based the "offset_mapping" provided by these tokenizers . This -100 is ignored by the model while training, and we can eliminate tokens with -100 label during testing as well.

The training data is fed into the RoBerta model, and the model is fine-tuned for this task using the train data. The hyper-parameters used were as follows, TRAIN_BATCH_SIZE = 2, DEV_BATCH_SIZE = 1, EPOCHS = 7, LEARNING_RATE = 1e-05, MAX_GRAD_NORM = 10, and the optimizer used was Adam. Then this fine-tuned RoBERTa is used to classify each token as propaganda/not. After this, using offset mapping, the exact span of each token is obtained. For predictions with label as 1, the corresponding offset mapping is written to the output file. The workflow of building the model is as follows:

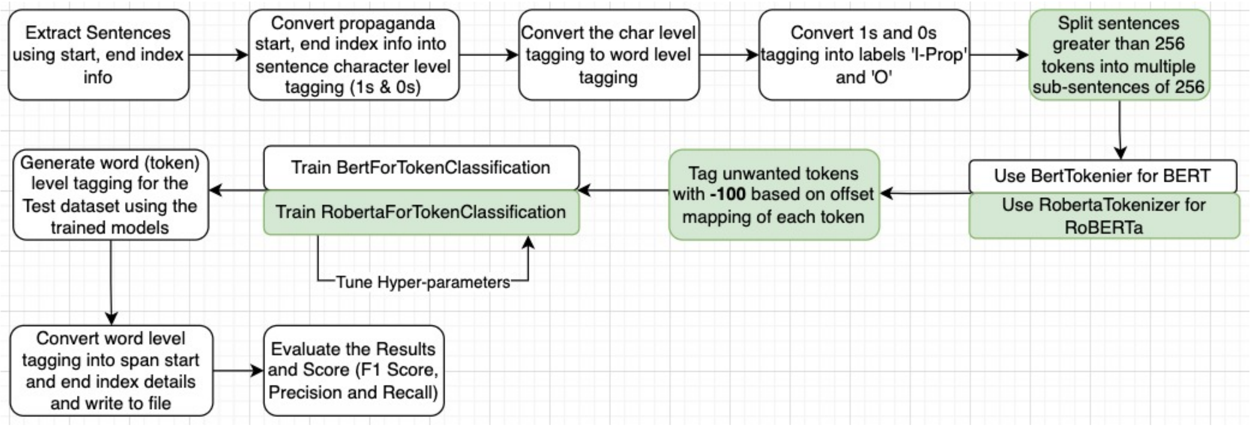

Extract Sentences using start, end index info
Convert propaganda start, end index info into sentence character level tagging (1s & 0s)
Convert the char level tagging to word level tagging
Convert 1s and 0s tagging into labels 'I-Prop' and 'O'
Split sentences greater than 256 tokens into multiple sub-sentences of 256
Use BertTokenier for BERT
Use RobertaTokenizer for RoBERTa
Tag unwanted tokens with -100 based on offset mapping of each token
Train BertForTokenClassification
Train RobertaForTokenClassification
Tune Hyper-parameters
Generate word (token) level tagging for the Test dataset using the trained models
Convert word level tagging into span start and end index details and write to file
Evaluate the Results and Score (F1 Score, Precision and Recall)

Figure 3: Span Identification

b) Technique Classification: The dataset for this sub-task contains files representing the actual new articles and for each news article, Article ID, the propaganda technique used, starting offset, ending offset are provided. The span information is used to extract the sentences that represent propaganda. The corresponding propaganda techniques are available which forms the training data. This task is performed as Multi-label Multi-Class Classification using the RoBERTa architecture. The training was done using RoBERTa model for the Multi-Class Classification. The hyper-parameters used were as follows, TRAIN_BATCH_SIZE = 16, DEV_BATCH_SIZE = 16, EPOCHS = 5, weight_decay=0.01, warmup_steps=500, and the optimizer used was AdamW. The model was designed as a true Multi-Label Multi-Class Classification. This was achieved by squashing the output class logits (raw outputs) from the predictions of the model using a Sigmoid function. This gives the probabilities of each class and the values are between 0 to 1 for each class. Finally, a cut-off is used (0.95) to get the classifications. Instead of choosing the class label with the highest probability, multiple labels are chosen, provided the probabilities of these classes is over the set threshold. The workflow of building the model is as follows:

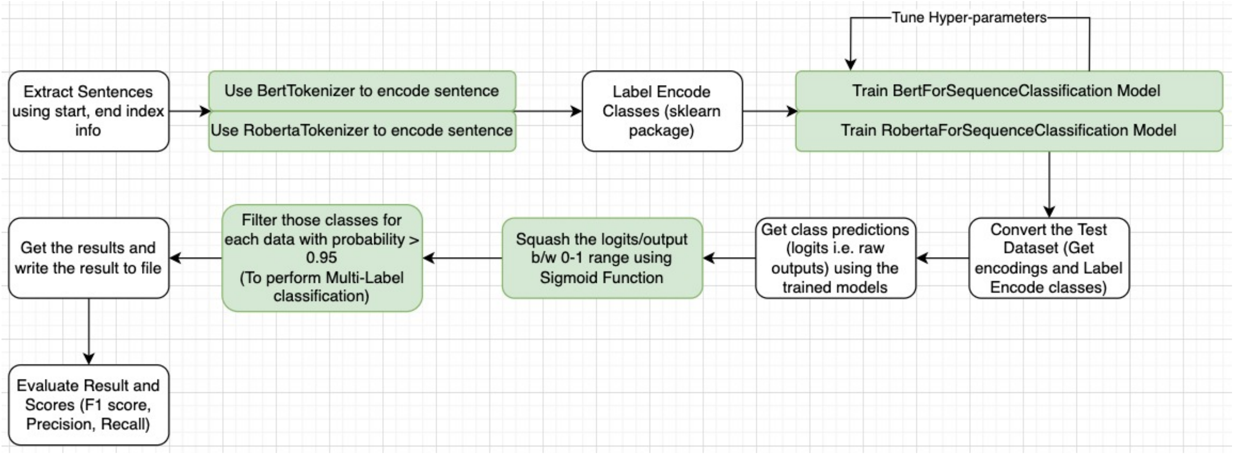
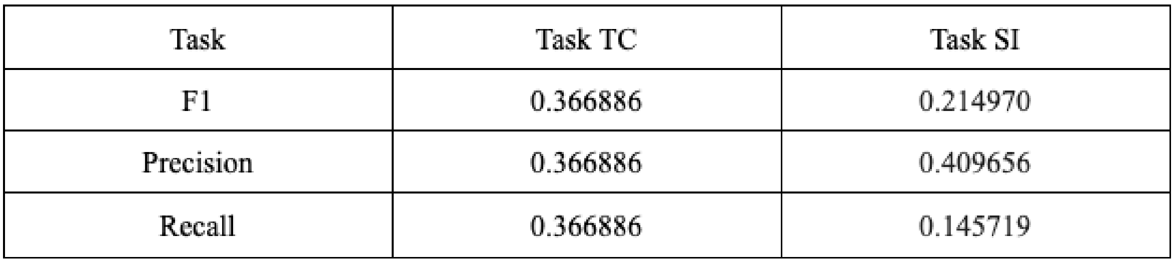

Tune Hyper-parameters
Extract Sentences using start, end index info
Use BertTokenizer to encode sentence
Use RobertaTokenizer to encode sentence
Label Encode Classes (sklearn package)
Train BertForSequenceClassification Model
Train RobertaForSequenceClassification Model
Convert the Test Dataset (Get encodings and Label Encode classes)
Get class predictions (logits i.e. raw outputs) using the trained models
Squash the logits/output b/w 0-1 range using Sigmoid Function
Filter those classes for each data with probability > 0.95 (To perform Multi-Label classification)
Get the results and write the result to file
Evaluate Result and Scores (F1 score, Precision, Recall)

Figure 4: Technique Classification

4 Results

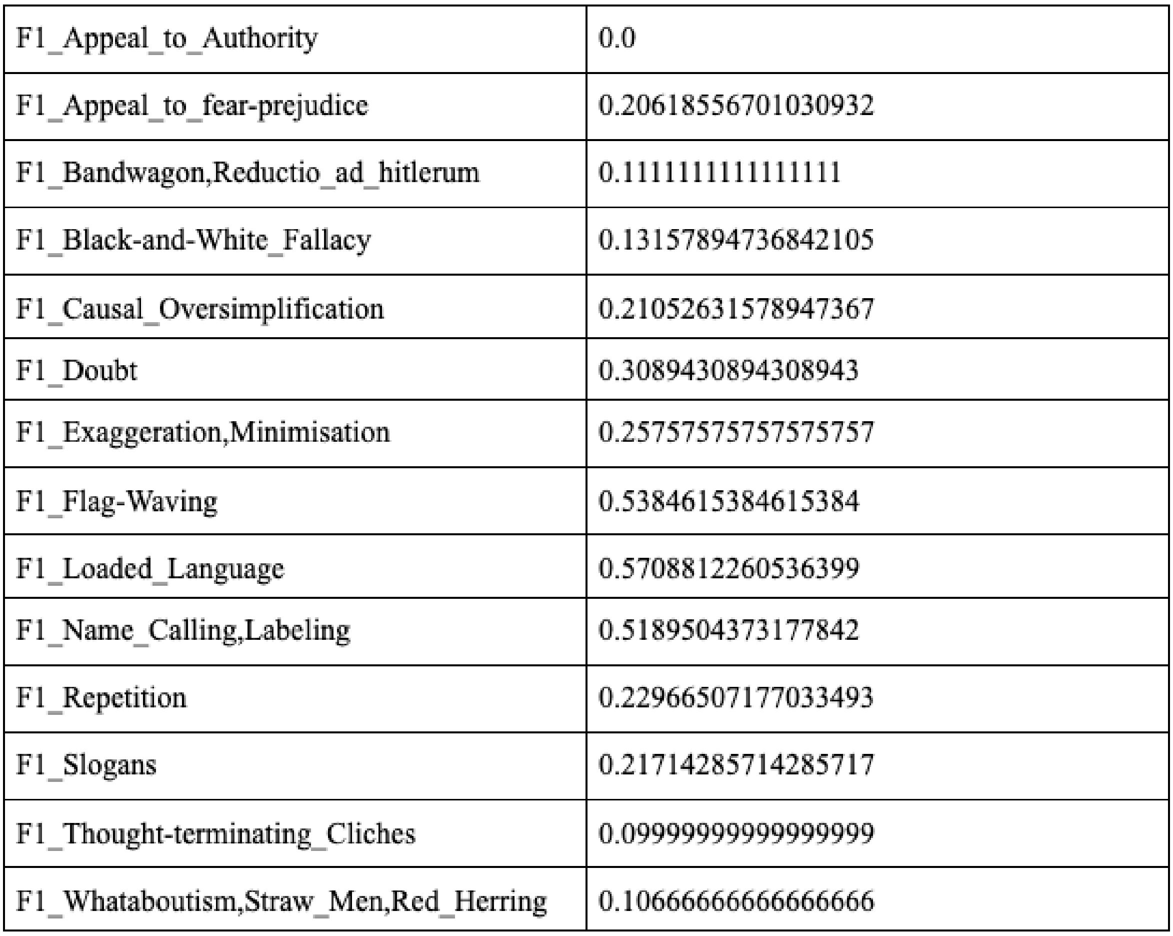
The results and the scores obtained are as follows:

4.1 Baseline Results



Task	Task TC	Task SI
F1	0.366886	0.214970
Precision	0.366886	0.409656
Recall	0.366886	0.145719

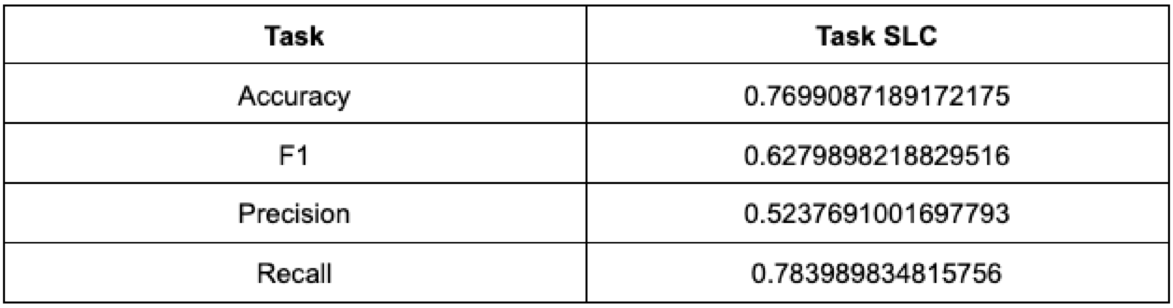
Figure 5: Baseline results



F1_Appeal_to_Authority	0.0
F1_Appeal_to_fear-prejudice	0.20618556701030932
F1_Bandwagon,Reductio_ad_hitlerum	0.1111111111111111
F1_Black-and-White_Fallacy	0.13157894736842105
F1_Causal_Oversimplification	0.21052631578947367
F1_Doubt	0.3089430894308943
F1_Exaggeration,Minimisation	0.25757575757575757
F1_Flag-Waving	0.5384615384615384
F1_Loaded_Language	0.5708812260536399
F1_Name_Calling,Labeling	0.5189504373177842
F1_Repetition	0.22966507177033493
F1_Slogans	0.21714285714285717
F1_Thought-terminating_Cliches	0.09999999999999999
F1_Whataboutism,Straw_Men,Red_Herring	0.10666666666666666

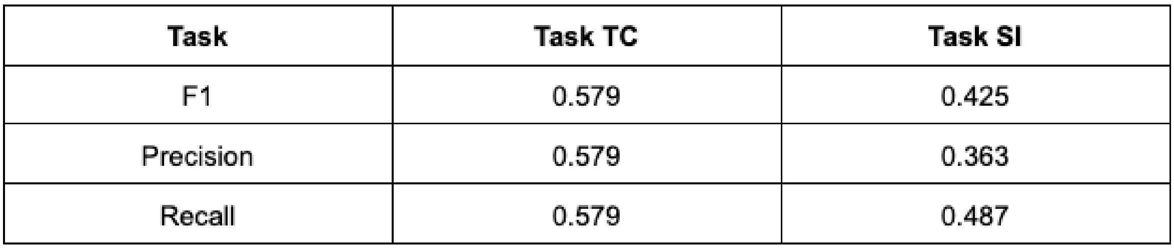
Figure 6: Baseline - F1 scores for individual classes

4.2 Final Results



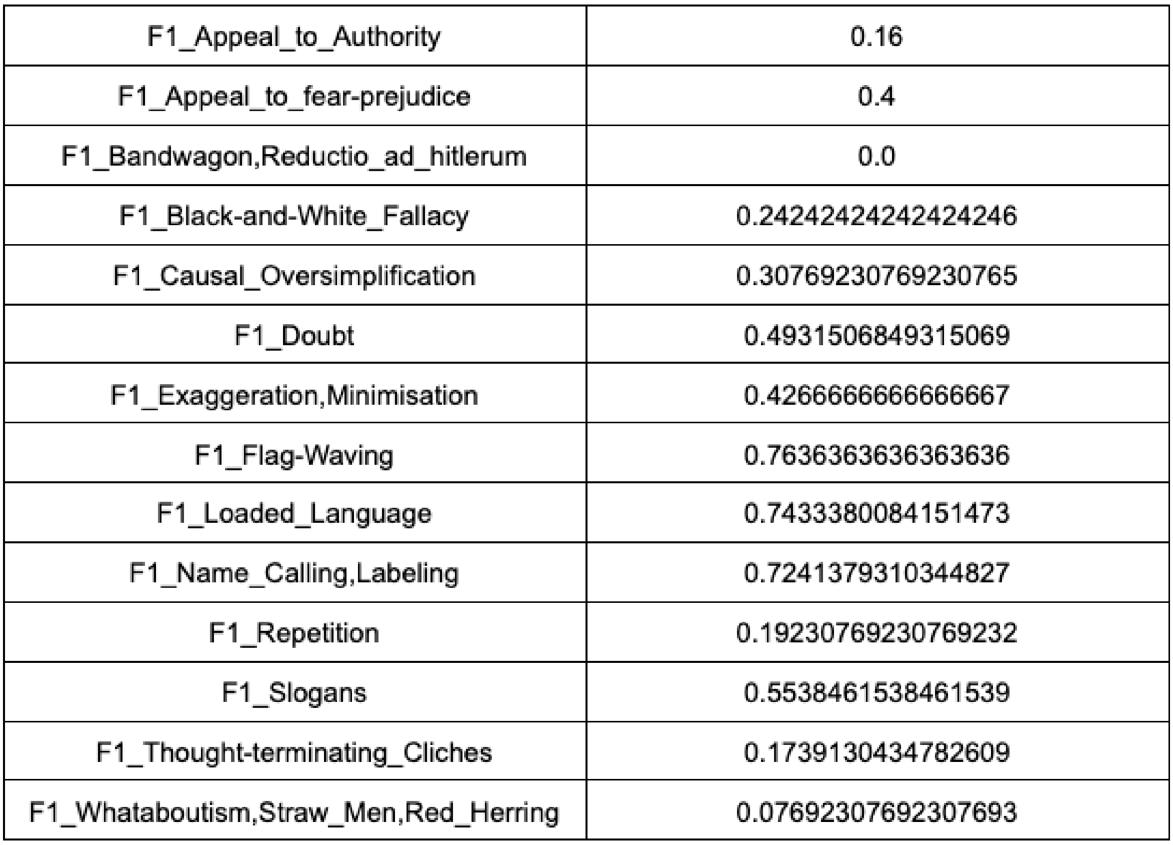
Task	Task SLC
Accuracy	0.7699087189172175
F1	0.6279898218829516
Precision	0.5237691001697793
Recall	0.783989834815756

Figure 7: Final Result - Task SLC using RoBERTa



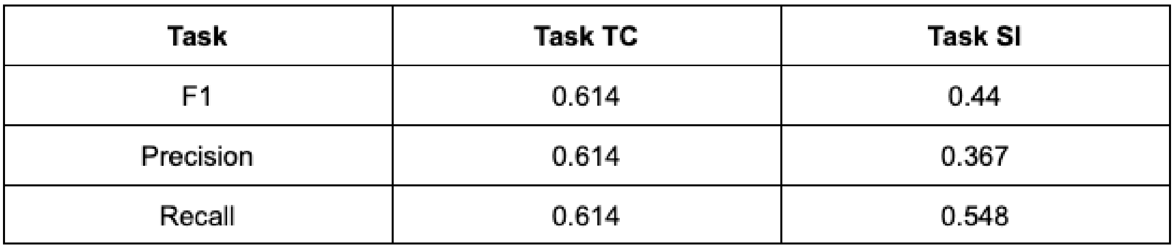
Task	Task TC	Task SI
F1	0.579	0.425
Precision	0.579	0.363
Recall	0.579	0.487

Figure 8: Final Result - BERT



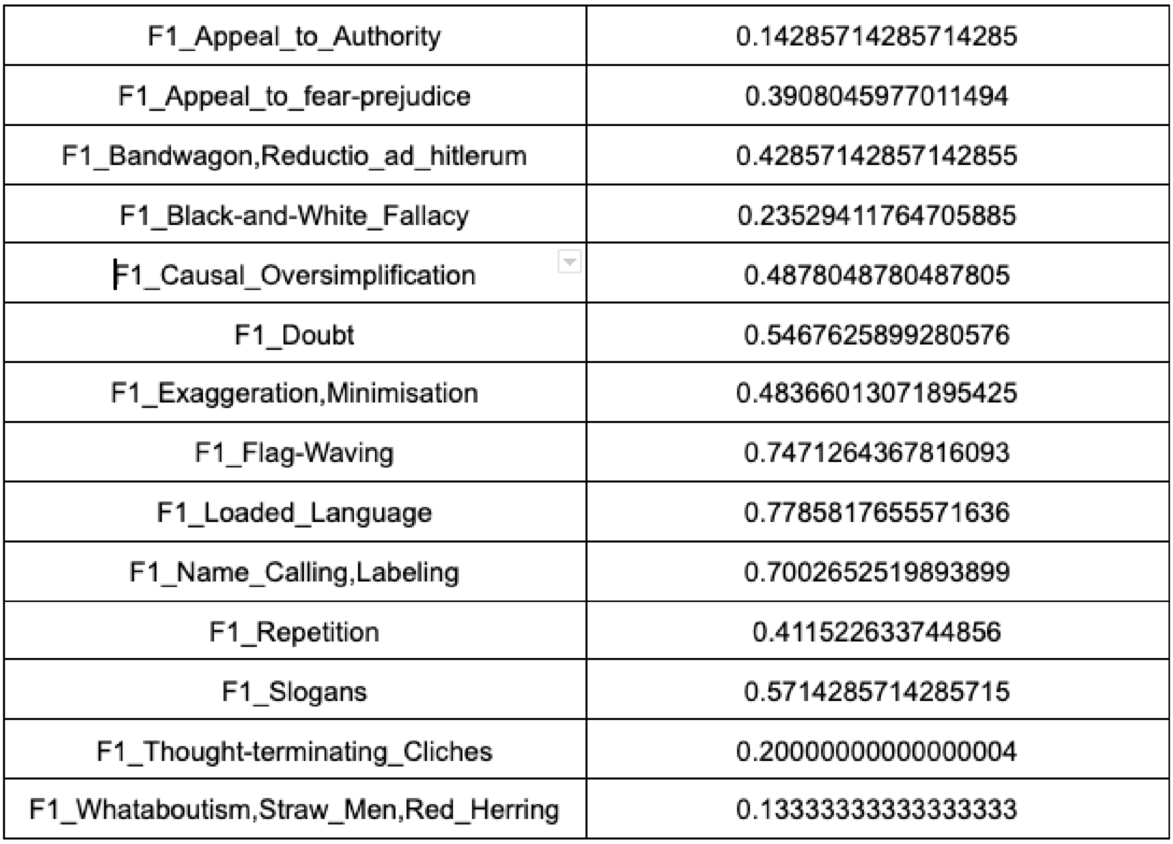
F1_Appeal_to_Authority	0.16
F1_Appeal_to_fear-prejudice	0.4
F1_Bandwagon,Reductio_ad_hitlerum	0.0
F1_Black-and-White_Fallacy	0.24242424242424246
F1_Causal_Oversimplification	0.30769230769230765
F1_Doubt	0.4931506849315069
F1_Exaggeration,Minimisation	0.4266666666666667
F1_Flag-Waving	0.7636363636363636
F1_Loaded_Language	0.7433380084151473
F1_Name_Calling,Labeling	0.7241379310344827
F1_Repetition	0.19230769230769232
F1_Slogans	0.5538461538461539
F1_Thought-terminating_Cliches	0.1739130434782609
F1_Whataboutism,Straw_Men,Red_Herring	0.07692307692307693

Figure 9: Final- Individual Class level F1 Scores for the task TC using BERT



Task	Task TC	Task SI
F1	0.614	0.44
Precision	0.614	0.367
Recall	0.614	0.548

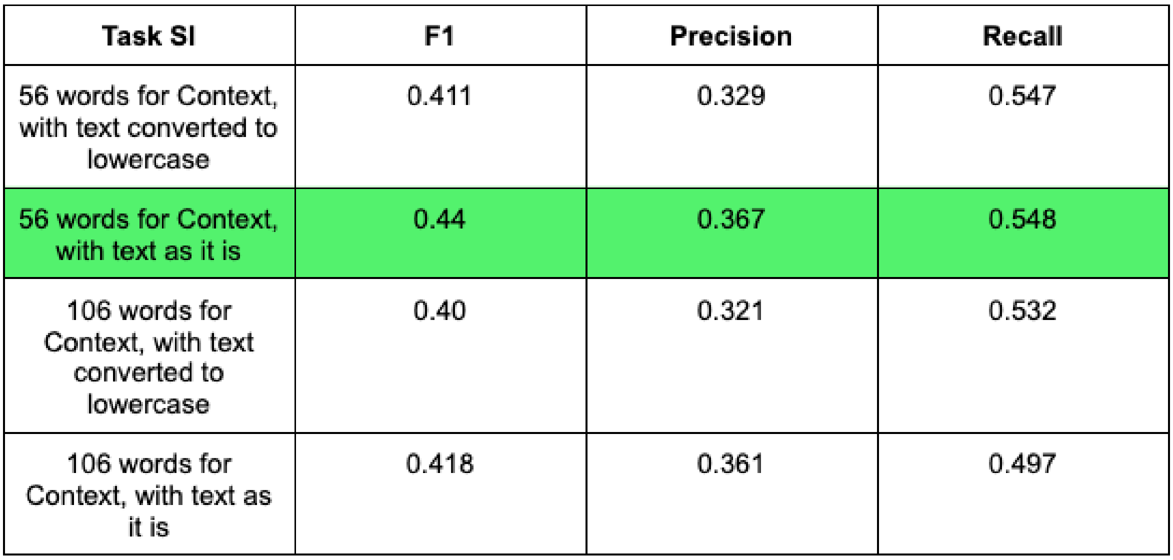
Figure 10: Final Result - RoBERTa



F1_Appeal_to_Authority	0.14285714285714285
F1_Appeal_to_fear-prejudice	0.3908045977011494
F1_Bandwagon,Reductio_ad_hitlerum	0.42857142857142855
F1_Black-and-White_Fallacy	0.23529411764705885
F1_Causal_Oversimplification	0.4878048780487805
F1_Doubt	0.5467625899280576
F1_Exaggeration,Minimisation	0.48366013071895425
F1_Flag-Waving	0.7471264367816093
F1_Loaded_Language	0.7785817655571636
F1_Name_Calling,Labeling	0.7002652519893899
F1_Repetition	0.411522633744856
F1_Slogans	0.5714285714285715
F1_Thought-terminating_Cliches	0.20000000000000004
F1_Whataboutism,Straw_Men,Red_Herring	0.13333333333333333

Figure 11: Final- Individual Class level F1 Scores for the task TC using RoBERTa

Comparison of Task SI scores using RoBERTa model with different context length, and different preprocessing.



Task SI	F1	Precision	Recall
56 words for Context, with text converted to lowercase	0.411	0.329	0.547
56 words for Context, with text as it is	0.44	0.367	0.548
106 words for Context, with text converted to lowercase	0.40	0.321	0.532
106 words for Context, with text as it is	0.418	0.361	0.497

Figure 12: Final Result - Comparison of various approaches using RoBERTa

As we can observe from the above table, the best F1 score for Task SI was obtained using the RoBERTa model with 56 words for Context, and using the text as it is. Also, the F1 scores for Task TC was the highest using the RoBERTa model as well.

5 Discussion and Error Analysis

1. In Task TC, the F1 scores for individual classes show that few of the classes have low scores, viz. ‘Whataboutism,Straw_Men,Red_Herring’, ’Appeal_to_Authority’. On analysis, these were the classes that had very low data examples. Data Augmentation techniques and added more manually curated samples for such classes would improve the overall model scores.
2. CRF can be used with RoBERTa to efficiently take into account nonlocal information. This might add extra context to the model, and this model might perform better in the Span Identification Task.
3. Ensemble of Logistic Regression, BERT and LSTM could be done to achieve better results for the Technique Classification Task.
4. The initial RoBERTa model will be trained using the given dataset. This model is then used to label additional data. Also negative training examples will be augmented i.e. sentences that do not have a propaganda. The model is again retrained with this new data. The final model will be a ensemble of the initial model and new model. This is then compared to just the old model without augmented data.
5. An Ensemble of Multi-task LSTM-CRF model with BERT could be used as mentioned by (Gupta et al., 2019), to predict the Spans and classify the spans as one of the 14 classes of Propaganda.

6 Conclusion

This paper describes the solution to identifying span and classifying the span into Propaganda techniques. The solution is based on the RoBERTa model fined-tuned to the use cases using the respective datasets. Several approaches were explored, and the best results comparing all the approaches are tabulated. We were able to improve the scores from our baseline models by almost two times, and this indicates that the language models are doing wonders in the field of NLP, and with the continuous advent of new larger models, the metrics can be improved further more.

7 Bibliography

References

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeno, Rostislav Petrov, and Preslav Nakov. 2019. Fine-grained analysis of propaganda in news article. 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 5636–5646.

Pankaj Gupta, Khushbu Saxena, Usama Yaseen, Thomas Runkler, and Hinrich Schütze. 2019. Neural architectures for fine-grained propaganda detection in news. *arXiv preprint arXiv:1909.06162*.

HuggingFace. https://huggingface.co/transformers/v4.4.2/custom$_{d}atasets.htmlseq - imdb.$

Dawid Jurkiewicz, Łukasz Borchmann, Izabela Kosmala, and Filip Graliński. 2020. Applicaai at semeval-2020 task 11: On roberta-crf, span cls and whether self-training helps them. *arXiv preprint arXiv:2005.07934*.

Niels Rogge. “https://colab.research.google.com/github/nielsrogge/transformers-tutorials/blob/master/bert/custom$_{n}amed_{e}ntity_{r}ecognition_{w}ith_{b}ert_{o}nly_{f}irst_{w}ordpiece.ipynbscrollto = k7rqmtdkdqqc''.$