NLP Project Report
Claim Detection in Social Media via Fusion of Transformer and Syntactic Features

Problem Statement:

Our problem statement involves classifying tweets into two categories: those deemed worthy of fact-checking by professional fact-checkers and those not worthy.

The task focuses on identifying fact-check-worthy claims from the vast pool of assertions made on social media platforms like Twitter. Not every statement qualifies as a claim; only assertions with a potential truth value are considered. Moreover, the focus lies on claims that have the potential to negatively impact a significant number of individuals. These are the assertions that warrant fact-checking by professional fact-checkers. Thus, the objective is to develop a model that can effectively classify such claims, separating those deserving of fact-checking from those that do not meet the criteria.

Example of Problem Statement:

Table 1: Sample tweets for Task-1 Check-Worthiness Prediction

Tweet	Claim	Check-Worthy
Dear @VP Pence: What are you hiding from the American people? Why won't you let the people see and hear what experts are saying about the #CoronaOutbreak?	0	0
Greeting my good friends from the #US the #Taiwan way. Remember: to better prevent the spread of #COVID19, say no to a handshake & yes to this friendly gesture. Check it out:	0	0
Corona got these flights cheap as hell I gotta job interview in Greece Monday	1	0
My mum has a PhD on Corona Virus from WhatsApp University	1	0
This is why the beaches haven't closed in Florida, and why they've had minimal COVID-19 prevention. Absolute dysfunction. <link>	1	1
COVID-19 cases in the Philippines jumped from 24 to 35 in less than 12 hours. This is seriously ALARMING. Stay safe everyone! <link>	1	1

Related Work:

Early studies employed classifiers such as Support Vector Machines, Decision Trees, and Naive Bayes, as cited in [2]. Initially, feature extraction focused on count-based techniques like TF-IDF, POS tags, and sentiment scores. Furthermore, researchers, as referenced in [6], introduced additional features including average embedding vectors of sentences, linguistic attributes, and sentence position awareness. These studies also utilized classifiers such as Deep Feed-Forward Neural Networks. [4] implemented an RNN-centric architecture, employing a blend of embeddings, POS tags, and one-hot encoding of syntactic dependencies to represent tokens. Meanwhile, [1] opted for n-gram features and employed a KNN classifier for their approach.

Lately, there's been a surge in the adoption of transformer models, as referenced in [3][5][7]. Ever since their introduction, these transformer models have emerged as the preferred approach for various Natural Language tasks including summarization, sequence classification, named entity recognition, text generation, language modeling, and extractive question answering, among others.

Methodology:

1. **Imports and Setup:**
   - Importing necessary libraries like pandas, numpy, nltk, etc.
   - Installing required packages using pip install commands.
   - Loading language models like 'en_core_web_sm' from spaCy.

2. **Data Preprocessing:**
   - Reading the training dataset (F2_Claim_Check_Worthiness_train.csv) into a pandas DataFrame.
   - Preprocessing the text data, including:
   - Removing white spaces, lowercasing, and tokenization.
   - Removing stopwords, punctuation, URLs, and HTML tags.
   - Spell checking using the autocorrect library.
   - Saving the preprocessed data into a new CSV file (preprocessed_train.csv).

3. **Feature Extraction:**
   - Extracting features such as parts of speech (POS) tags and dependency relations.
   - Calculating the occurrence count of unique dependencies in the dataset.
   - Identifying the most important dependencies based on their occurrence count.

Feature extraction process image:

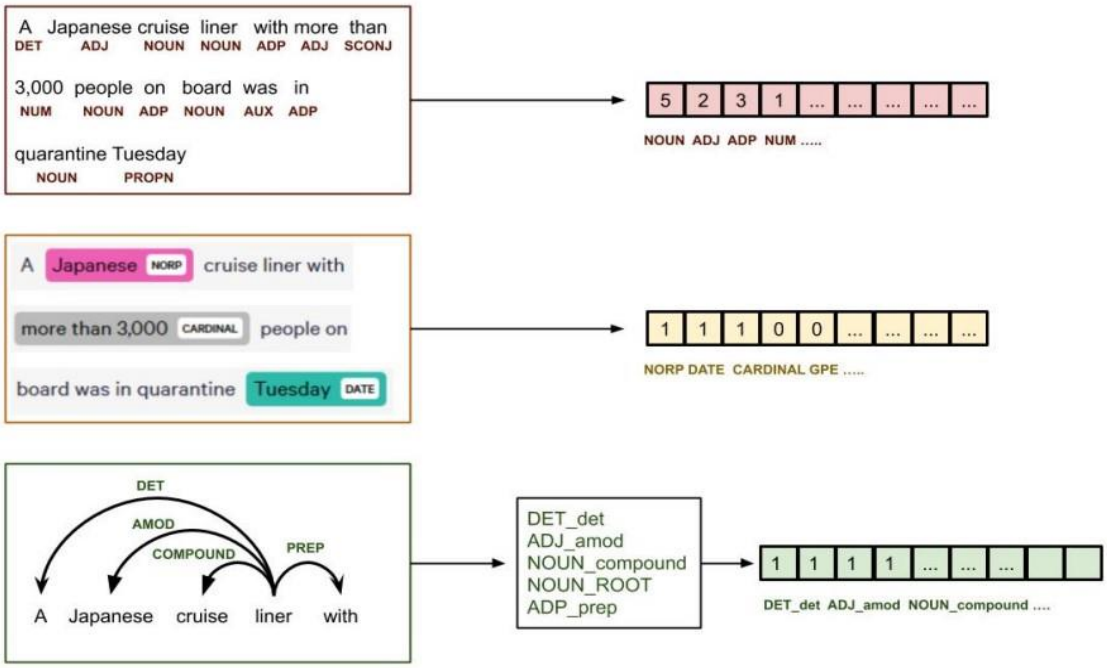


Fig. 1: Syntactic feature extraction and encoding process. Feature vectors are based on the number of times it is seen in the given sentence.

4. Data Transformation:

- Scaling the extracted features using MinMaxScaler.
- Applying Principal Component Analysis (PCA) for dimensionality reduction.

5. Modeling:

- Using the SentenceTransformer library to obtain sentence embeddings.
- Building a pipeline for data transformation and model training.
- Splitting the preprocessed data into training and validation sets.
- Training a Support Vector Machine (SVM) model on the training data.
- Evaluating the model's performance using macro F1-score on both train and test sets.

6. Predictions:

- Making predictions on the test dataset (F2_Claim_Check_Worthiness_test.csv) using the trained SVM model.

Image of Approach for Check-worthiness prediction:

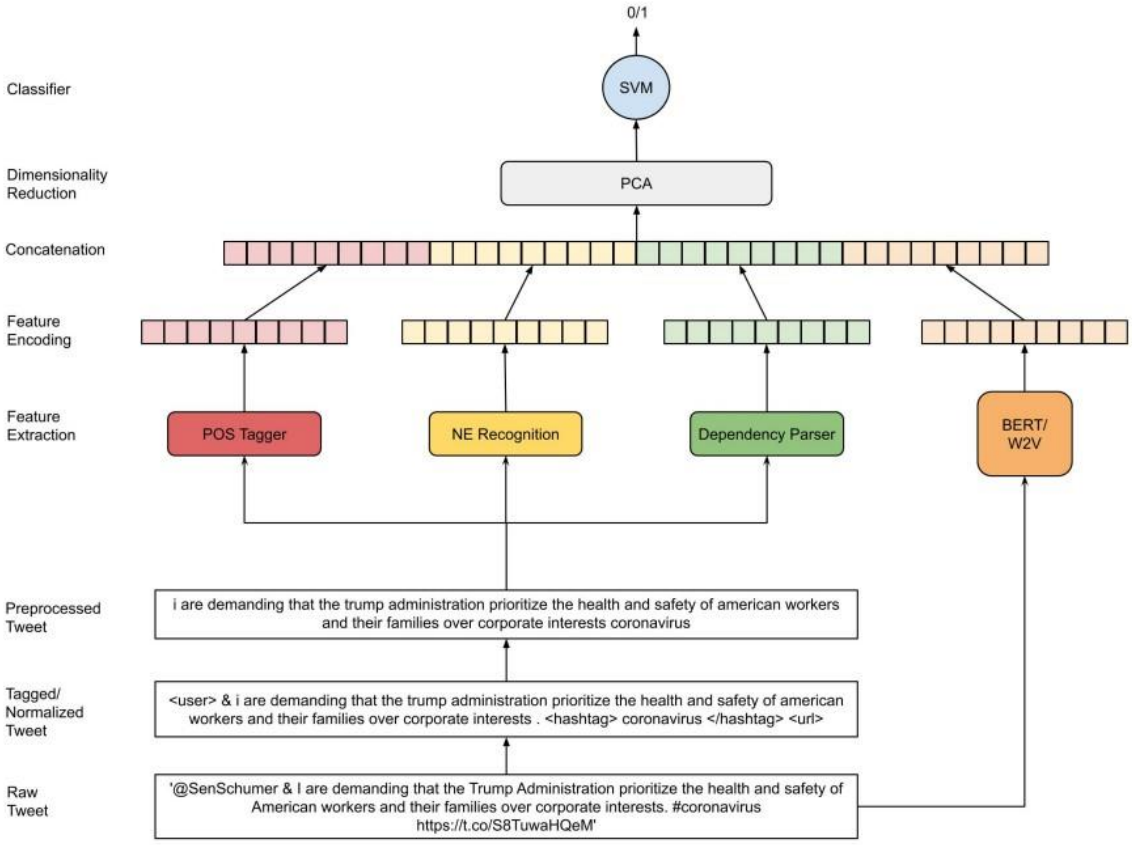


Fig. 2: Proposed Approach for Check-Worthiness Prediction

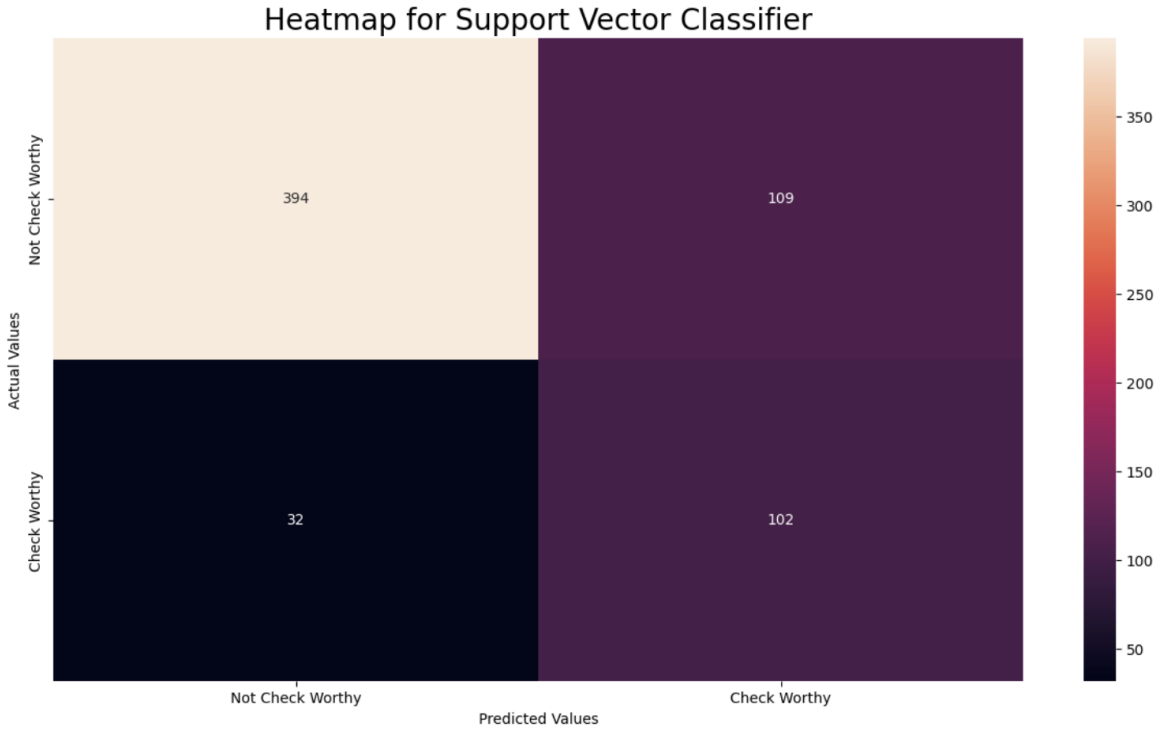
- Saving the predictions along with their corresponding IDs into a CSV file (output5.csv).

7. Evaluation:

- Computing and displaying a confusion matrix and a classification report for the model's performance on the validation set.

Results:

- **Confusion matrix of SVM:**



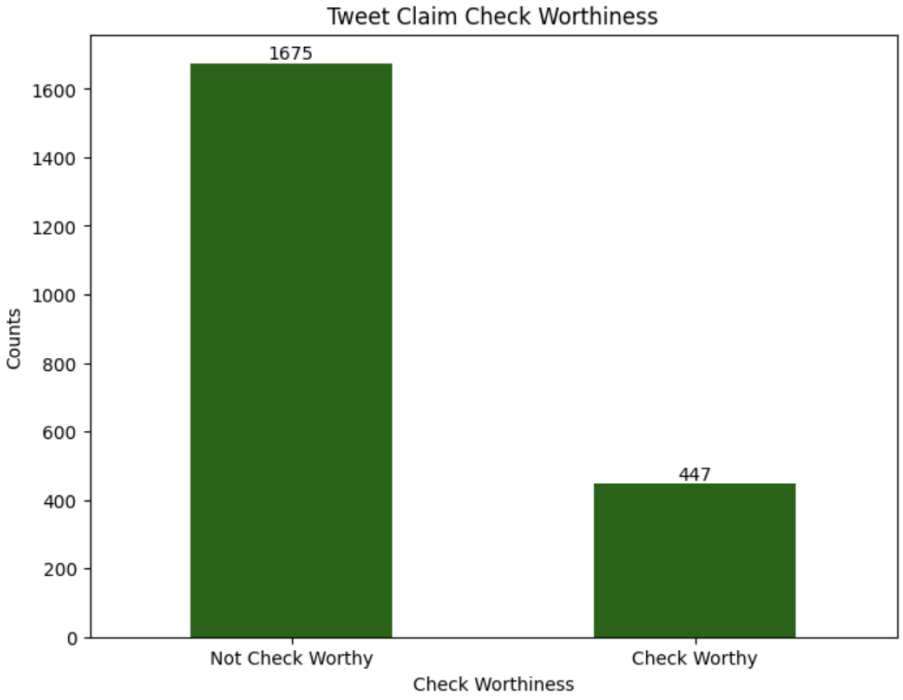
- **Accuracy Scores:**

```
Traning Accuracy:  0.8922558922558923
Validation Accuracy:  0.7786499215070644
```

- **Macro-F1 Scores**

```
Traning Macro-f1:  0.8599763327000052
Validation Macro-f1:  0.7197641222445827
```

- **Bar Plot for tweet claim check worthiness:**



- **Classification Report:**

Classification Report for Validation Set:

	precision	recall	f1-score	support
0	0.92	0.78	0.85	503
1	0.48	0.76	0.59	134
accuracy			0.78	637
macro avg	0.70	0.77	0.72	637
weighted avg	0.83	0.78	0.79	637

Conclusion:

In conclusion, the machine learning approach presented in this study offers a promising solution to the challenge of misinformation and fake news on Twitter. By leveraging advanced NLP techniques and robust classification algorithms, the model achieves high accuracy in identifying dubious content within tweet streams. Future research endeavors could explore the integration of additional features, such as named entity recognition, to further enhance the model's performance. Moreover, extending the scope of analysis to encompass other social media platforms would enable a comprehensive approach to combating misinformation in the digital sphere.

References:

[1] P. Gencheva, P. Nakov, L. Màrquez, A. Barrón-Cedeño, I. Koychev, A context-aware approach for detecting worth-checking claims in political debates, in: Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017, 2017, pp. 267–276.

[2] A. Nikolov, G. D. S. Martino, I. Koychev, P. Nakov, Team Alex at CLEF CheckThat! 2020: Identifying Check-Worthy Tweets With Transformer Models, arXiv:2009.02931 [cs] (2020). URL: http://arxiv.org/abs/2009.02931, arXiv: 2009.02931.

[3] B. Ghanem, M. Montes-y Gomez, F. Rangel, P. Rosso, UPV-INAOE-Autoritas - Check That: Preliminary Approach for Checking Worthiness of Claims (2018) 6.

[4 N. Hassan, C. Li, M. Tremayne, Detecting check-worthy factual claims in presidential debates, in: Proceedings of the 24th acm international on conference on information and knowledge management, 2015, pp. 1835–1838.

[5] E. Williams, P. Rodrigues, V. Novak, Accenture at CheckThat! 2020: If you say so: Post-hoc fact-checking of claims using transformer-based models, arXiv:2009.02431 [cs] (2020). URL: http://arxiv.org/abs/2009.02431, arXiv: 2009.02431.

[6] G. S. Cheema, S. Hakimov, R. Ewerth, Check_square at CheckThat! 2020: Claim Detection in Social Media via Fusion of Transformer and Syntactic Features, arXiv:2007.10534 [cs] (2020). URL: http://arxiv.org/abs/2007.10534, arXiv: 2007.10534.

[7] C. Hansen, C. Hansen, J. G. Simonsen, C. Lioma, The Copenhagen Team Participation in the Check-Worthiness Task of the Competition of Automatic Identification and Verification of Claims in Political Debates of the CLEF-2018 CheckThat! Lab (2018) 8.