Project Report:

Mis/Disinformation Analysis of the Israel/Palestine Conflict

Team Members:

Zaid Fada, ycq2zz | Tripp Mims, cef7dn | Ahyush Kaul, sdx6cq | Henry Newton, han5jn

1. Introduction and Motivation

The Israel/Palestine conflict has been a source of significant international tension and humanitarian concerns for decades. Journalism, particularly citizen journalism, concerning this conflict influences public opinion worldwide, yet it varies widely in its quality, often leading to the perpetuation of false accounts. Identifying and understanding these inaccuracies is crucial for promoting a balanced view of the conflict, which is essential for fostering informed discussions and, ultimately, peace. This project aims to use Artificial Intelligence for social good by applying misinformation and disinformation detection to the online discussion of and reporting of the Israel/Palestine conflict. We aim to detect potential indicators of falsehood in these reports, contributing to the broader effort of ensuring balanced media coverage.

2. Methods

We utilized a few different models for mis/disinformation analysis to see which best optimizes for our specific dataset. These include a deep/artificial neural network, support vector machine, and naive bayes classifier. Specifically, we plan to develop models that can classify articles as likely accurate or not based on the correlated features expressed within the text. This will involve preprocessing the text data, extracting relevant features, and training each model to classify articles accordingly.

Importantly, we will ultimately judge our models not only according to their effectiveness but also according to their apparent bias. We will ensure that whichever model we decide upon as the best is not significantly in favor of either side of the conflict.

3. Data

To conduct our analysis, we compiled a dataset of 55 data points, each consisting of text extracted from tweets or posts that were fact-checked by Snopes. These posts predominantly cover on-the-ground events in Israel and Palestine, as well as international responses from both governmental and grassroots movements. This collection aims to provide a comprehensive view of the discourse surrounding the conflict. Despite Snopes having a number of the following kind of article, we avoided including fact-checks of “Celebrity X did/did not say Y” as it seemed too extraneous from our intended subject.

While we could have included other fact-checking sources as well to not exclusively invite Snopes’ bias into our models, it likely would have lowered the accuracy without significant work to properly standardize data from all the sources. Even then, it may have greatly hampered our models ability to predict mis/disinformation.

Data Preprocessing

To prepare the data for analysis, we implemented several preprocessing steps designed to minimize potential biases in model training:

- To counteract potential biases, for a randomly selected half of our posts, we swapped all identifying information.
    - For instance, a tweet stating:
        > "USA President Joe Biden arrived in Israel. Where are the leaders of 57 Muslim countries who stands [sic] with Palestine?"
    - would be altered to:
        > "RUSSIA President Vladimir Putin arrived in Palestine. Where are the leaders of 57 Jewish countries who stands [sic] with Israel?"
    - This method is intended to obfuscate the parties involved, helping to prevent the model from developing skewed perceptions based on nationality or ethnicity.
- We also remove Hebrew and Arabic characters from the text to further reduce the risk of the model using these as indicators of whether the author of the post is Israeli or Arab, information which it could then use to bias in favor of one side against another.
- Snopes provides a range of labels to describe the accuracy of information. We standardized these into a binary true/false system for simplicity and excluded any "mixture" labels due to insufficient examples which could lead to unreliable training outcomes.

Libraries Used

For preprocessing and model development, we utilized popular data science libraries, including Keras for building and training neural network models, Pandas for data manipulation and analysis, and TensorFlow as our primary machine learning framework. These tools enhanced our model's development process, from data cleaning to complex algorithm implementations.

4. Experiments

Naive Bayes Classifier

The Naive Bayes classifier implementation primarily utilized the sklearn library's MultinomialNB. Initially, text data was transformed into a format suitable for model input using the TfidfVectorizer, which converts text into a matrix of TF-IDF features, highlighting important words in the dataset. To optimize the model's performance, parameter tuning was used on the smoothing parameter, alpha, which helps in managing unseen words in future data. Alpha values were tested from 0.1 to 1.0 in increments of 0.1 to determine the best setting for maximizing accuracy. This approach allowed for a systematic exploration of how each alpha value influenced the model's ability to generalize from the training data to unseen test data. The best alpha was found to be 0.2, achieving an accuracy of approximately 90.91%.

Support Vector Machine

The support vector machine implementation utilized the sklearn python library. It utilizes an additional library GridSearchCV, which allows for the parameterization of variables. This was important for testing the SVM with various different hyperparameters for the different kernels. For example, GridSearch enables us to pass in various gammas that are used for rbf kernels. Three different C and gamma values were used in the gridsearch to cover a range of values. Additionally, we use SMOTE to balance out the training class in order to prevent the majority class of False labels from being imbalanced. This is important in preventing bias when training the model.

Artificial Neural Network

The artificial neural network used was from the Sci Kit Learn library MLP. It was tuned for optimal parameters using grid-search values for alpha, hidden layers, and learning rate. 10 different values were attempted for all of these. This gave an alpha value of 0.001, a single hidden layer of 100 nodes, and an initial learning rate of 0.01. In addition to this 0.9 momentum was used along with Adam as an optimization algorithm and early stopping if accuracy did not improve after 10 iterations. This was done to try to reduce overfitting of the model. It was allowed to run for a maximum of 1,000 iterations to converge.

5. Results

Naive Bayes Classifier

The performance of the Naive Bayes classifier was evaluated based on its accuracy, precision, and recall metrics at various alpha levels. The optimal alpha value was identified as 0.2, yielding the highest overall accuracy of 90.91%. This alpha value indicates an effective balance in smoothing, which helps the model to manage unseen words in future data without excessively diluting the influence of known words. In terms of precision, the Naive Bayes model demonstrated a strong ability to predict both classes accurately. It achieved a precision of 90% for false claims and 100% for true claims. However, the recall metric revealed some limitations in the model's performance. For true claims, the recall was 67%, indicating that the model correctly identified only 67% of the true claims while missing the rest. This lower recall rate for true claims underscores a conservative prediction approach, where the model tends to be more cautious in predicting a claim as true, likely due to the imbalance in class distribution within the training data.

This conservative nature could be beneficial in contexts where the cost of false positives is high, but it also suggests the potential need for additional strategies, such as a larger sample size, to improve recall without compromising precision. The findings highlight the strengths and weaknesses of the Naive Bayes approach in text classification, particularly in scenarios where class imbalance is prevalent.

Support Vector Machine

The SVM results measured precision, recall, f1-score, and accuracy for predicting if a statement was true or false. Overall the scores for predicting false statements were higher than predicting true statements. The precision for false was 75% while only 50% for true predictions. Alone, these values indicate that the model wasn't performing well for predicting true statements. This is further supported by looking at the f1-scores which were 82% for false and 33% for true. The f1-scores indicate that when balancing precision and recall scores, the False label was predicted far more accurately than the True label. This was a result of overfitting the training data, which was imbalanced with False labels. Implementing a balancing factor for the training class, along with a larger sample size, ensures the model can learn the difference between false and true statements. The overall accuracy of the SVM was 71% with an optimal linear kernel and C value of 1.

Artificial Neural Network

Overall, the neural network overfit the training data frequently. It was able to perfectly fit the training data with a 73% accuracy on unseen data. It had a tendency to call more examples true than were actually the case. The false precision and especially the true recall. True recall was only 25% as it was very likely to label data as true which led to these discrepancies in values. However, it did perform well in not calling true values false. This could be beneficial in regards to journalists' reputations as labeling their insights false when they are true can be extremely harmful to their credibility.

It is very likely that this model was too complex for the data given. Other models could likely label the data with higher accuracy as it was more likely to label data as true even when this data was processed. Increasing the scope of data could improve the test accuracy through use of more labeled training data. This could be done through outsourcing to find true and false news stories that cover this conflict.

6. Conclusion

Overall, the naive bayes classifier worked the best on the data with the other models overfitting the data. A larger dataset could shed light onto which model would be ideal for classification as the conflict continues to expand. It is possible that naive bayes could continue to be the best model in spite of the fact that it is the most simplistic of the three. In addition to this, further models could be tested to see if they perform even better than naive bayes in attaining an even higher accuracy above 91 percent.

For future research in the same vein as our own, we would suggest trying to expand the data set used by the models. This could either be achieved by using multiple fact-checking sources or by including subjects other than just one, Israel/Palestine in our case. It would also greatly improve our models' flexibility to train on video and image data as well as text data.

The downside of the last two of these three options is that with them it becomes harder if not altogether impossible to suppress bias through de-identification. For instance, while it is easy to remove Hebrew and Arabic characters from text, it is much more difficult to identify and remove Hebrew or Arabic speech from audio. Similarly, clothing, flags, and other visual indicators of identity are much more difficult to swap than words in text.

7. References

“Snopes.com.” Snopes.com, 2024, snopes.com.

8. Code and Data

https://colab.research.google.com/drive/1kHnS1oW8QGNbR3XTXwAE1Jup3l1nVDUw?usp=sharing

https://docs.google.com/spreadsheets/d/10gKGxnbHhRqv0PgDucE36ks93yjSHLgN/edit?usp=sharing&ouid=117213969525265141226&rtpof=true&sd=true