Finding the truth in a sea of misinformation

Misinformation detection pipeline

Hugo Albert Bonet
Iván Arcos Gabaldón
David Borregón Sacristán
Kexin Jiang Chen
José Fco. Olivert Iserte
Diana Yaser Haj

PROJECT III: DATA ANALYSIS

Bacherlor's Degree in Data Science
Universitat Politècnica de València
Valencia, Spain
May 26, 2023

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Abstract

UNICC posed the challenge to build an automatic system capable of effectively detecting false claims, hoaxes, and other forms of misinformation spread on Twitter in the context of Nigeria's 2023 Presidential Election. For this, a misinformation detection pipeline was developed, consisting of two main components: a fact checking algorithm and a machine learning model. State-of-the-art Natural Language Processing (NLP) techniques (such as BERT-style transformers) were used to train the model. We have achieved F1-Scores close to 0.9 by combining embeddings and classical classifiers. Incorporating information on toxicities and emotions further improved performance. The fine-tuning technique using a combination of Convolutional Neural Networks (CNN) and evidence sentences reached an F1-Score of 0.907, emphasizing the importance of providing supporting information. On top of that, this endeavor brings value in two additional ways: by having built an extensive fact-checking database from scratch (given the lack of available annotated data on this topic) and by having generated reusable Python classes and modules, and project documentation that can help build an analogous pipeline for future misinformation detection projects.

Contents

List of Figures

List of Tables

1 Introduction

Misinformation refers to false information, regardless of whether or not it is intended to mislead or deceive people. With the constant evolution of digital platforms and technologies like social media, misinformation spreads farther, faster, and deeper than truthful information where the most common issues are related to immigration, gender, politics, equality and vaccination that can cause real-world consequences like deterioration of the trust in journalism and science. It is crucial to address this problem in order to create safe digital spaces. Misinformation detection comes with several challenges, such as models becoming outdated due topic/vocabulary mismatch between the new social media post and the training data used to build models. Additionally, the availability of annotated data is limited as it takes time and effort to compile an up-to-date annotated dataset.

United Nations International Computing Center (UNICC) is an Information Technology (IT) Service Provider to the United Nations that provides many different services to different organizations and agencies [1]. One of the major challenges that UNICC aims to address is the widespread of misinformation on social media platforms. In collaboration with UNICC, our main focus will be tackling misinformation in Nigerian elections, where the spread of false information can have serious consequences. UNDP (United Nations Development Programme, another UN agency) [2] is also one of the end users of this project, tasked with monitoring elections and reporting issues. Therefore our main goal is to help UNICC effectively detect and prevent false claims, hoaxes, and other forms of misinformation from spreading on social media. To accomplish this, we will develop a misinformation detection pipeline (an automatic system capable of distinguishing fake news) consisting of two main components: a machine learning module and a fact checking module. This pipeline solves the issues of misinformation detection mentioned above by re-training the model and annotating data semi-automatically. Previous research on automated fact-checking models such as those by (Kotonya & Toni, 19 October 2020, #) [3], (Nakov et al, 22 May 2021, #) [4], provided valuable insights about the process followed to automate the fact-checking tasks, and the challenges faced in the process. We found that BERT-style transformers are the most commonly used models by researchers.

To further enhance our understanding of current and emerging methodologies on automated fact-checking, we conducted additional research and reviewed more papers. For instance, the study (Pérez-Rosas, 2017) [5] uses linguistic features such as n-grams, punctuation, psycholinguistic features, syntax as well as text readability properties to develop classification models. Seidman (2013) [6] compares the similarity between the given documents and a number of impostor documents, so that documents can be classified as having been written by the same author, if they are shown to be more similar to each other than to the impostors. Pan & Chen (2021) [7] proposed a framework Question Answering for Claim Generation (QACG) for training a robust fact verification model by using automatically generated claims. QACG framework works well in both the zero-shot and few-shot learning setting. Pan (2017) [8] used transformers, which had shown to be really effective in translation tasks and can be applied to other tasks such as English constituency parsing. Overall, the literature we found provided us great insights when building our own misinformation detection pipeline. However, the methods used are not mainly focused on automating the fact-checking process. Thus, our project could be a novelty in terms of the automatization of the misinformation detection process. The goals and values of this project align with the importance of combating misinformation and promoting the use of accurate and reliable information in society, which can

have a positive impact on individuals, organizations, and society as a whole. We hope that with this misinformation detection pipeline, we can help UNICC, UNDP and other individuals and organizations quickly and accurately identify and correct false information, thereby preventing it from spreading further on social media platforms, and strengthen trust between the users.

1. **Success criteria**
   To consider the project successful, we must implement the pipeline and establish a metric to evaluate the quality of the predictions made by the model. We ought to keep in mind that the threshold is based on how accurate we want to be with the classification and it depends on the data, the amount of evidence we have to contrast the claims. The starting goal was set to 70% of accuracy in the prediction of true and false claims.

2. **Alignment with Sustainable Development Goals**
   This project can be considered a society-oriented project due to its alignment with several Sustainable Development Goals (SGDs), specifically: 16.- Peace, Justice and strong Institutions: "Promote peaceful and inclusive societies for sustainable development, provide access to justice for all and build effective, accountable and inclusive institutions at all levels." [9] and 17.- Partnership for the Goals: "Strengthen the means of implementation and revitalize the Global Partnership for Sustainable Development." [10].

2 Technical implementation

One of the first steps we had to take was the data extraction process. We wanted to collect data related to the main topic, Nigerian elections, from two main sources; fact-checking pages and news outlets. To be able to accomplish this task, we used the Requests package of Python (Requests, n.d.), combined with Selenium [11].

The complete list of fact-checking pages was Africa Check [12], Dubawa [13], AFP [14], Lead Stories [15], The Dispatch [16] and News Verifier Africa [17]. The news outlets used were The Guardian [18], Punch Nigeria [19], Daily Post [20], Sahara Reporters [21], The Nation Online Nigeria [22], and Vanguard Nigeria [23].

Due to the disparity of the labels used in the different pages we decided to unify and keep only two of them; False and True. We obtain a total of 673 rows with the distribution of 546 False claims and 127 True claims. We can see that there is a clear problem of unbalanced classes, where the majoritarian one is the False clase, being approximately four times larger than the True class.

After detecting this problem, and knowing that our test data would be composed of tweets, we decided to label a subset of the data provided by UNICC. For this task, we created a new Comma-Separated values (CSV) file using the Evidence column of all the fact-checking pages separating sentences and adding a new row for each one. We also added general information extracted from Wikipedia about Nigeria (assuming and considering that Wikipedia is a whitelist) obtaining a total of 22567 observations. We call this file the Evidence Database. We initially labeled 2000 tweets using the API with GPT 3.5 Turbo [24]. After realizing that most of the tweets were labeled as False or containing misinformation (concretely around 200 were True and 1800 were False), we decided to generate synthetic tweets to be able to compensate both classes. We supplied five randomly selected sentences from the Evidence Database and generated a tweet related to the information provided. The final result was a total of 3004 observations, distributed in 1623 tweets with a True label and 1381 tweets with a False label. This data will help us to build the Fact-Checking Module.

2.1 Data description and preparation

The data preparation phase is a crucial step in any data analysis or machine learning project. The main steps involved in preparing the data for further analysis were the following.

1. The first step is to merge the individual CSV files from the different fact-checking pages into a single consolidated dataset. This was achieved by combining the rows from each CSV file into one large dataset, ensuring that the column structure remains consistent across all files.

2. Next, we extract the evidences from the combined dataset. Additionally, we created an index to facilitate easy reference and retrieval of information.

3. To ensure uniformity and consistency in the dataset, we converted all text data to lowercase, creating a standardized representation and simplifying subsequent analysis.

4. This step involves checking for any inconsistencies, errors, or missing labels within the dataset, unifying different classes and comparing and eliminating any duplicates, ensuring the dataset contains unique instances only.

5. In order to gain deeper insights from the data, it was relevant to extract additional features such as toxicity, sentiment, and subjectivity. We extracted these features using libraries like NLTK, Emo-RoBERTa [25], unbiased-toxic-roberta [26] and TextBlob. This can help uncover patterns, trends, and underlying sentiments within the data.

2.2 Model building

2.2.1 Machine learning module

The machine learning module consists of a machine learning model trained specifically to classify a claim according to the presence of misinformation in it. The process can be divided into different steps which we will explain in detail below:

1. **Embedding and Similarity Computation**
   The text of the claim and each retrieved document is converted into numerical vectors (embeddings) using the Sentence-Transformers library [27]. This facilitates the computation of similarity between the claim and the documents, which helps in ranking the documents based on their relevance to the claim. We then compute the cosine similarity between the claim embedding and each document embedding, selecting the top 5 most relevant documents (evidence) for further processing.

2. **Preprocessing, Feature Extraction and Dataset Creation**:
   We use a pre-trained SBERT model (Attention Is All You Need, 2017) [8], 'mitra-mir/setfit-model-Feb11-Misinformation-on-Media- Traditional-Social', designed for misinformation detection on traditional and social media, to classify the claim based on the retrieved documents. The data is preprocessed by concatenating the claim and the top 5 most relevant evidence sentences, and then encoded using the SBERT model to obtain embeddings. With that we create a structured dataset that includes the claim, the top 5 evidence sentences, and the label.

   To enrich the dataset and improve classification performance, we incorporate predictions from various models:

   - Zero-shot classification [28] using the 'facebook/bart-large-mnli' model for the initial prediction of whether the claim is True or False. Zero-shot classification allows the model to classify text into categories it has not seen during training.
   - Emotion detection using the 'arpanghoshal/EmoRoBERTa' model [25] to capture the emotional context of claims. This model detects 28 different types of emotions in the text. We have included the probability scores for each emotion in the dataset as this information can help in understanding the emotional context of the claims.
   - Toxicity detection using the 'unitary/unbiased-toxic-roberta' model [26] to identify and filter out toxic or biased claims such as general toxicity, insults, identity attacks, etc. By incorporating this information, we can potentially identify and filter out toxic or biased claims from the dataset.
   - Zero-shot classification for election-related categories to categorize claims into various election-related categories, such as voter registration, election administration, election

violence, political parties, campaign strategies, election monitoring, electoral transparency, and many more. This helps in organizing the claims and understanding their context better.

3. **Dimensionality Reduction**:

   To reduce the dimensionality of the embeddings and speed up the training process, we apply Principal Component Analysis (PCA). This step helps in reducing the computational complexity while retaining the most important information from the embeddings. However, we noticed that applying dimension reduction did not yield good results in our model training. Therefore, we didn't use it in the end.

4. **Model Training and Evaluation**:

   The dataset is split randomly into training and validation sets with a 80-20 split ratio. Various classification algorithms were trained on the training set using a 10-fold cross validation, and the model's hyperparameters were optimized using a parameters grid and repeating the process of training with all the combinations of these parameters. The output of this process will be the model fitted with the parameters that had better values in the metrics. The performance of these models is then evaluated on the validation set using accuracy as the performance metric. Since the output is balanced, accuracy is an appropriate metric to measure the model's performance. That being said, F1-Score was also used to give more importance to costly misclassifications. Different models have been compared, and they are described in the table of section 3.2 where results are discussed.

5. **Prediction and explanation**:

   The best-performing model is selected based on its accuracy on the validation set. This model is then used to classify the claim as true or false based on the retrieved evidence. We then built models to provide an explanation for the classification of the claim. We made use of Explainable AI and GPT3 to do so, which are explained in detail below:

   5.1. **Explainable AI**

   In the context of Nigerian elections, Explainable AI (XAI) is crucial to determine which features the model is using to determine whether a claim is true or false. This information can be useful in identifying patterns and sources of misinformation, which can then be addressed and corrected.

   5.2. **GPT3**

   Since we wanted to obtain a comprehensive explanation for the label assigned to a claim, we opted for GPT-3 in the end. We provided GPT-3 with the claim, the top 5 pieces of evidence and the predicted label obtained from the classification models, and then requested a detailed explanation as to why the claim was classified as True, False or Unproven.

In the Appendix we included three examples of implementation of the machine learning module using GPT3 as well as the surrogate model (decision tree).

2.2.2 Fact-checking module

We built a fact-checking module, which consists of a rule-based module that emulates the behavior of a journalist during the process of fact-checking [29]. To build the fact-checking module, we explored many different approaches which we discuss below:

- **Emotion Analysis**:
  This approach involves analyzing the sentiment of the text to determine if it contains misinformation. Misinformation often uses emotional language to manipulate the reader's perception of the facts. Sentiment analysis can be used to identify the emotional tone of the text and determine if it is consistent with the facts presented in the article.

- **Decision Tree Analysis**:
  We created a decision tree by analyzing our database of known true and false articles and identifying the features that distinguish them. The decision tree is built by selecting a feature that it believes is important in distinguishing true and false news articles. It then splits the data into two groups based on this feature and repeats the process for each group until we have a tree that accurately classifies the data. We then repeat the process for each group until we have a tree that accurately classifies the data. Once we have created the decision tree, we can explore it backwards to see the rule that it applies in each split. This can help us to understand the features that are most important in distinguishing true and false news articles and to identify patterns of misinformation.

With these ideas and using the data previously labeled, we built our no-machine-learning algorithm.

3 Results and discussion

3.1 Machine Learning Module

Over the past few months, our team has worked diligently to develop and fine-tune a model that can help us reach the goal mentioned in Section 1.1.

We conducted a comprehensive evaluation of different approaches. We compared the performance of the following models:

- **Baseline Model**: The 'roberta-fake-news' model in its original configuration serves as the baseline for measuring performance.

- **'roberta-fake-news' Embeddings + Classical Classifiers**. We applied supervised learning techniques using the embeddings generated by the 'roberta-fake-news' model. Classical classifiers such as SVM, MLP, and RF are used for the classification of misinformation. We compare their performance against the baseline model.

- **Fine-tuning 'roberta-fake-news' + CNN with Evidence, Sentiment, and Toxicity**, We fine-tuned the 'roberta-fake-news' model using a 1D Convolutional Neural Network (CNN) to incorporate evidence, sentiment, and toxicity information. We evaluate the impact of considering different numbers of evidence sentences (k = 0, 1, 2, 3, 4, 5) on the model's performance, as shown in Figure 1.

Furthermore, in each comparison, we consider the costs of false positives (FP) and false negatives (FN) and calculate precision, recall, and F1-score. This comprehensive evaluation enables us to assess the effectiveness of the models in detecting misinformation while considering the practical implications of classification errors.

The models have been trained on a dataset consisting of 2835 claims, of which 1577 are labeled as true claims and 1258 as false claims. These claims were used to train the models and teach them patterns and relationships between tokens to make predictions on new, unseen data. After training, we evaluated the models' performance on a separate test set containing 315 claims. Among these claims, 171 were labeled as true claims and 144 as false claims.

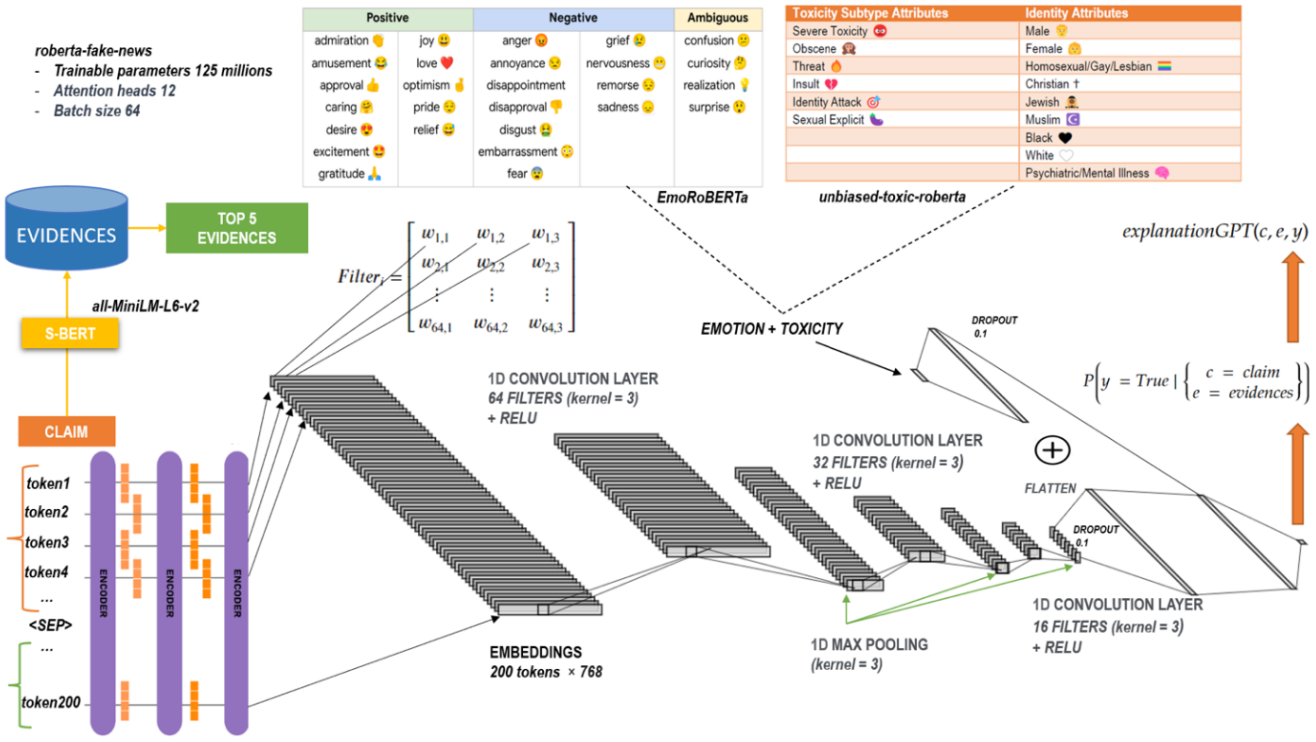

Figure 1: Fusion of Transformer, 1D Convolutional Networks, and Variable Concatenation for Misinformation Detection in Nigeria

3.1.1 Attention Between Claims and Evidences is All You Need: Tackling Misinformation in Nigeria with Transformers and 1D Convolutional Networks

We will perform fine-tuning on the "roberta-fake-news" model, which has 125 million trainable parameters and 12 attention heads. This model has been specifically trained on a Kaggle dataset for detecting fake information. However, our goal is to adapt it specifically to the task of misinformation detection in Nigeria.

The input to the transformer model will consist of tokens representing the claim we want to verify, followed by a special separation token "<SEP>", and tokens representing the top "k" (initially 5) evidence sentences that are closest to the claim. To determine these closest evidence sentences, we will calculate the cosine similarity between the embeddings using the "all-MiniLM-L6-v2" model.

Instead of using the "[CLS]" token for classification, we propose applying one-dimensional convolutional operations with a specific number of filters. Subsequently, we will apply one-dimensional max pooling layers successively until we obtain a flattened vector with the final outputs.

This approach aims to capture relationships between the claim tokens and the evidence tokens to predict whether it is misinformation or not. By leveraging the attention capabilities of the transformer and utilizing one-dimensional convolutional operations, we aim to capture relevant features that can help discern between accurate and misleading information.

After obtaining the representation from the convolution process, we concatenate it with a vec-

tor associated with the outputs of a dense layer applied to other types of variables that we have considered. These variables include emotions extracted from the claim using the EmoRoBERTa transformer. EmoRoBERTa is trained on the GoEmotions dataset, which consists of 58,000 Reddit comments labeled for 27 emotion categories, including Neutral.

Additionally, we have incorporated various toxicity variables using the unbiased-toxic-roberta transformer. This model is trained on The Civil Comments dataset, which comprises approximately 2 million public comments from the now-closed Civil Comments platform. The dataset is annotated for toxicity and other attributes, including identity labels, to facilitate research on improving online conversations. The toxicity labels in this dataset include: *Toxic*, *Severe*, *Toxic*, *Obscene*, *Threat*, *Insult* and *Identity Hate*.

Moreover, the model takes into account identity labels such as *male*, *female*, *homosexual_gay_or_lesbian*, *christian*, *jewish*, *muslim*, *black*, *white*, and *psychiatric_or_mental_illness*.

By incorporating these emotion and toxicity variables into the model architecture, we aim to capture additional contextual information and potential correlations with misinformation detection in Nigeria.

3.2 No Machine Learning Module

In order to build the no machine learning module we have tried different approaches besides the one explained in Section 2.2.2.

One of them was a XGBoost trained only with emotions and toxicity variables.This model was trained only with synthetic data created by GPT-3. In order to explain the predictions we used a surrogate model to get the splits, a simple decision tree.

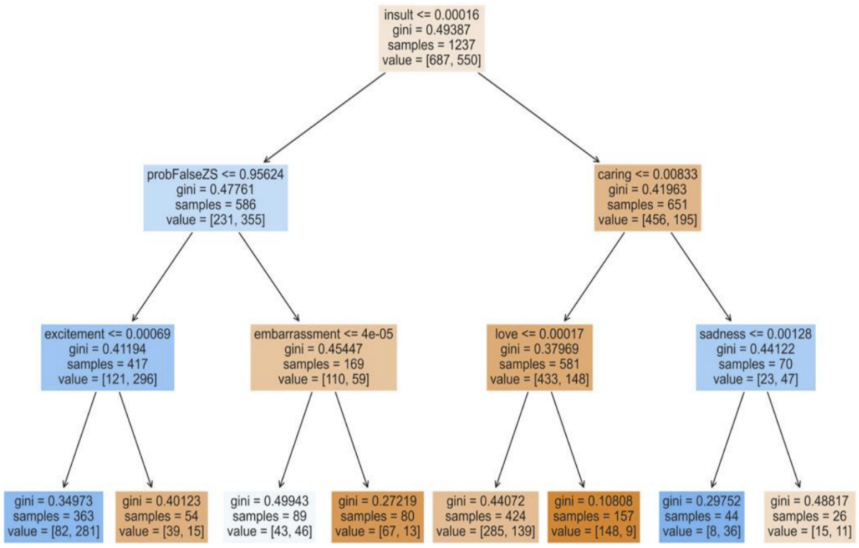

Figure 2: Decision tree surrogate model for Explainable Sentiment-Toxicity model
Blue means 'True': Orange means 'False'

This model achieved a F1 score of 0.65.

Then we tried different techniques and variables using similar models described in Section 2.2.2.

The first approximation was to clean the text removing hashtags and stopwords and vectorize it using a TDF-IDF representation and using a decision tree to classify whether the claim is True or False decided to try with a decision tree and classify. Surprisingly, we obtained a F1-score of around 0.85, which is a pretty good result keeping in mind that we only use text as the input with no other features. After, we thought of including sentiment analysis using the NLTK library in Python (sentiment intensity analyzer) and from the TextBlob library we used the sentiment.subjectivity module and the sentiment.polarity, to be able to classify the tweet as positive, negative or neutral. We combined both data frames and ran a Grid Search to find the best hyperparameters for the tree, hoping for better results. Unfortunately, results were a bit below the approach where we used just text (getting around 0.82 of F1-score), but we decided to represent the tree because it can give us more information.

We also created a model using just sentiments, and even though the predictive capacity of the model is a bit better than randomness, we obtained interesting conclusions from the polarity variable. We observed that positive tweets are classified as True and negative ones as False. Figure 8 shows the tree generated by the model including text and sentiment.

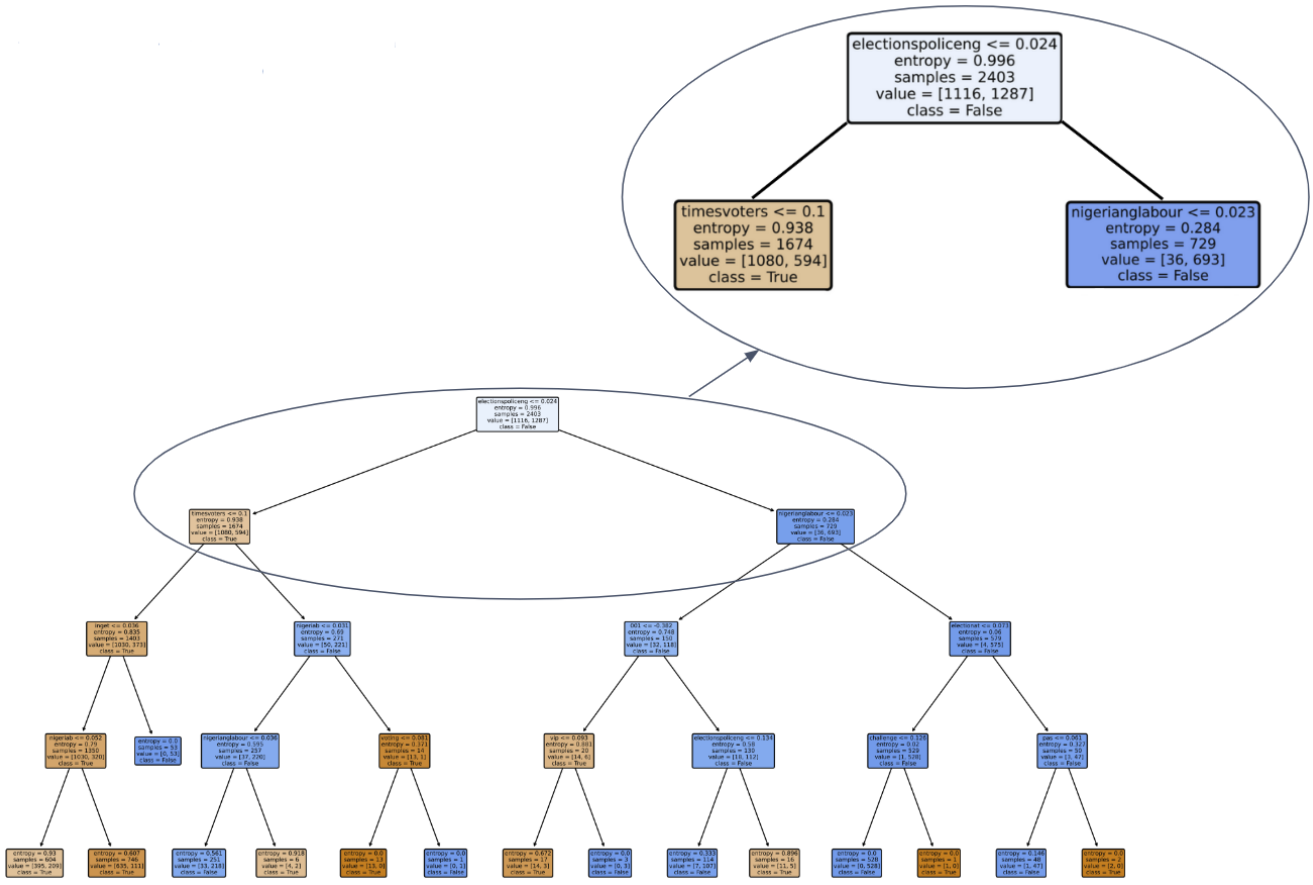

Figure 3: Decision tree combining sentiments
Blue means 'True': Orange means 'False'

Using a Decision Tree with a maximum depth of 4 we get a F1-score of 0.824. We can observe that the first and most important splits are word based using words like "elections" or "Nigeria elections". Sentiments come after in the splits of the tree and we can deduce that they are less important for the classification.

This improvement with respect to the first Decision Tree is due to different aspects.

Firstly, it came from the new features as the degree of subjectivity but mainly because of the data we made use of when training.

- GPT-3 has limited creativity so in the process of creating synthetic data tweets were really similar causing overfitting and making the predictions less reliable.
- GPT-3 was more effective at labeling the data, because it just got to compare the tweet with the evidences, rather than creating tweets on its own.

This reminded us of the importance of having a good evidence dataset in order to achieve good results.

The results of the baseline approach using the "roberta-fake-news" model were quite poor, achieving an F1-Score of 0.537. This indicates that the base model alone is not effective in detecting misinformation.

Approach	Model	F1-Score	Precision	Recall	Cost
Baseline	*roberta-fake-news*	0.537	0.460	0.643	1717
Embeddings + Classical Classifiers					
	SVM	0.907	0.907	0.912	265
	MLP	0.870	0.858	0.883	390
	XGBoost	0.884	0.879	0.889	343
	RandomForest	0.910	**0.938**	0.883	240
Embeddings + Classical Classifiers with emotions and toxicities					
	SVM	0.907	0.907	0.912	265
	MLP	0.897	0.900	0.895	296
	XGBoost	0.904	0.897	0.912	285
	RandomForest	**0.924**	0.924	**0.924**	**221**
Fine-tuning + CNN + k Evidences with emotions and toxicities					
	k=0	0.807	0.912	0.725	444
	k=1	0.898	0.920	0.877	308
	k=2	0.886	0.908	0.865	325
	k=3	0.897	0.900	0.895	297
	k=4	0.905	0.916	0.895	273
	k=5	0.907	0.927	0.889	276

Table 1: Performance Evaluation of Misinformation Detection Approaches

The combination of embeddings representations and classical classifiers, such as SVM, MLP, XGBoost, and RandomForest, proved to be much more effective. These approaches achieved F1-Scores close to 0.9, indicating good performance in detecting misinformation.

There was a beneficial effect when adding information on toxicities and emotions to the embeddings and classical classifiers models. This resulted in improvements of around 1-2 percentage points in the F1-Scores for most classifiers.

The fine-tuning technique using a combination of CNN and k evidences, where k represents the number of evidences used, also proved to be effective. The obtained F1-Scores were around 0.9. It was observed that adding evidence to the claims significantly improved the model's performance, increasing the F1-Score from 0.81 (without evidence) to almost 0.9 (with the closest evidence based on cosine similarity).

As the number of evidences (k) increased, the model's performance continued to improve, reaching an F1-Score of 0.907. This indicates that providing the model with more supporting information enhances its ability to detect misinformation more accurately.

It is important to note that these results were obtained with specific data and a limited evidence base. It is expected that a model with better data and a wide range of evidence will perform even better compared to other misinformation detection methods. Additionally, the inclusion of evidence and the use of attention matrices in the model provide explainability, as it allows understanding which aspects the transformer deems important in determining whether a claim is false

or true.

3.3 Deployment

In this section, we present the deployment of our misinformation detection pipeline through a prototype application, which aims to address the original objectives of the project. Our prototype application consists of a user-friendly interface that allows users to input a claim and receive a detailed analysis of the veracity. The interface is designed to be accessible for users with varying levels of expertise, including the employees at UNICC and United Nations Development Programme (UNDP) [2].

Our first python prototype which we called NigeriaFactCheck Bot has some defined directory structure that is essential in order to work.

Here we can see and example of how the chatbot works:

1. First the user has to run the main.py python file.

2. Secondly, the bot asks the user to enter a claim.

```
(environment) vant@agile-v2:~/Documentos/mySpace/fun/chatbot\$
 ↪  python main.py
Importing the modules...
Loading the models...
Models loaded.
Hello! I am NigeriaFactCheck Bot (but i am trained with Zambia
 ↪  elections data), nice to meet you. My duty is to provide a
 ↪  label to the claim you tell me. Hope I am accurate :D!
What is your question/claim?:
<Q>: (write bye to leave) Trump is the president of Zambia
```

3. Then, given that claim, the bot processes the claim through the misinformation detection pipeline and gives you a sentiment and toxicity analysis of it, altogether with the top nearest evidences it has found in our evidence database, as shown in Figure 17 in the appendix 3.

4. Finally, it gives you the label that the model predicted with the corresponding probability.

```
Checking: Trump is the president of Zambia
The evidences used to classify are:

- iVerify Zambia has determined as false the claim that police
 ↪  have arrested over 2,500 people for failing to pay back loans
 ↪  they got from Zamcash, an online lending platform that
 ↪  provides salary advance loans in Zambia.
- iVerify Zambia has established that the application form does
 ↪  not state that only people from a certain region should apply
 ↪  for the position.
```

```
        - Zambia has for the past three years seen its ranking position in
        ↪ this index worsen from 105 in 2018, 113 in 2019 and 117 in
        ↪ 2020.
        - She told iVerify Zambia in an interview that all those who took
        ↪ the aptitude tests had applied to ZAMSTATS and were invited by
        ↪ the organization and not herself or the mayor.
        - IVerify Zambia has established as false assertions that the
        ↪ Mayor and DC had invited people who did not receive invitation
        ↪ SMs from ZAMSTATS to attend census aptitude tests and that
        ↪ most of these are the ones who were recruited to train as
        ↪ enumerators.

        Analyzing emotions:
        Analyzing toxicity:
        <A>: This text is FALSE with a probability of 0.9261 by a XGBOOST
        ↪ model
```

This prototype demonstrates its potential to effectively identify and analyze false claims in the context of Nigerian elections. It is created to help human fact checkers to reduce their work time in searching and finding evidences to put a label on each claim, giving them some reliable model to lean on. Our objective is to create something similar to a chatbot that gives you a label given a claim about Nigeria. We have incorporated the sentiment and toxicity analysis to explain why our model is predicting that label. The goal of this is not to replace human labor in this kind of task. A human team is needed to supervise the model.

By automating the detection process and providing detailed explanations for the assigned labels, our prototype addresses the challenges of outdated models and the availability of annotated data, which is the main value that we wanted our project to have.

4 Conclusions

4.1 Main findings

Misinformation detection is not an easy task on any ground. Therefore, the hard work and the struggle has brought us some interesting findings.

First of all we got to mention the easy, fast and efficient way GPT-3 has given us to label the data that was not labeled. When data is scarce or the labeling task is so tedious, it is a pretty good option. It could work even better with the supervision of human experts.

We observed that state-of-the-art transformers and vector representations like Tf-idf with simple models perform really well. Sentiment analysis can be very useful to improve the explicability of our models and help the final user understand the decision taken for each instance by certain models.

Definitely, misinformation detection is a very challenging issue, but with a good understanding of the problem, hard work, fresh ideas, initiative and a proactive team, everything is possible.

4.2 Impact assessment

The spread of misinformation can manipulate public opinion, influence voter behavior, and undermine the integrity of the electoral process. Implementing a misinformation detection pipeline project specifically tailored for the Nigerian elections holds significant potential to mitigate the negative impacts of misinformation, enhance trust and credibility, strengthen democracy, and empower citizens with accurate and reliable information. By leveraging technology, collaboration, and public awareness initiatives, Nigeria can take a crucial step towards promoting fair, transparent, and credible elections, safeguarding the integrity of its democratic process.

4.3 Limitations

- Data bias: misinformation detection pipelines rely on large amounts of data, which can be biased and skewed towards certain perspectives or sources. This bias can affect the accuracy and effectiveness of the pipeline.
- Ethical concerns: misinformation detection pipelines can potentially infringe on individuals' privacy and free speech rights. It is crucial to ensure that the pipeline's development and implementation are ethically sound and do not violate individuals' rights.
- Adversarial attacks: misinformation actors can attempt to circumvent or manipulate the pipeline by introducing new forms of misinformation or finding ways to evade detection. This can require continuous updates and improvements to the pipeline's algorithms and methodologies.

4.4 Future work

Future work for this project includes several areas of improvement. Firstly, we could try to expand the scope of our model to cover other languages and regions beyond Nigeria. This will require the

collection of more annotated data and the development of language-specific models. Additionally, we could explore the use of transfer learning techniques to adapt our model to new languages and regions more efficiently.

We could also incorporate multimodal analysis, which involves analyzing different types of data such as images and videos instead of just text, as tweets containing misinformation also come in images and videos, which in this project we could not detect. This would enable a more accurate detection pipeline and improve the overall effectiveness of the project.

There are also several ways to extend the project and improve its effectiveness in detecting and preventing the spread of misinformation on social media. One way could be integrating the misinformation detection pipeline with social media platforms such as Twitter or Facebook. This would enable the automatic detection and removal of false information in real-time, thereby reducing the spread of misinformation.

5 Others

5.1 Acknowledgment

We would like to express our sincere gratitude to the United Nations International Computing Center (UNICC) for their invaluable collaboration and support throughout the development of our misinformation detection pipeline project. Their guidance and expertise have played a crucial role in shaping our project and enhancing its impact.

Finally, we would also like to acknowledge the efforts of our project advisors and professors. Their expertise, feedback, and continuous support have been invaluable in shaping our approach and refining our methodologies. Their constructive criticism and thoughtful suggestions have helped us navigate the challenges inherent in developing a misinformation detection pipeline.

Without the collaboration and support of UNICC, our project would not have been possible. We are immensely grateful for the opportunity to work with such a distinguished organization and for the knowledge and experience gained during this collaboration.

5.2 Software/data availability

In a Data Science project, the availability of software and data plays a critical role in its success. However, there are instances where accessing labeled data can be challenging to find, especially in certain areas. This was our case, leading us to employ web scraping techniques to obtain labeled data and overcome dara scarcity. Regarding software tools, we relied on free and open-source tools like Python and R to overcome the limitations of proprietary software. Python, with libraries like *BeautifulSoup*, *Scikit*, *NLTK*, *Pandas*, *Torch*, *Numpy*, *Transformers* and *Pretrained Bert* among others to be able to scrape, process and organize data, represent textual information and train models. R provided great data visualization tools with libraries like *Ggplot2*. Leveraging these free tools allowed the project team to overcome financial barriers and access a wide range of features and functionalities required for data acquisition and analysis. All code written throughout the project will be hosted on GitHub, and can be accessed through our repository [30].

5.3 Conflict of interest

The development of a misinformation detection pipeline, while aimed at addressing the pervasive issue of misinformation, inherently presents a complex landscape of potential conflicts of interest. Recognizing and managing these conflicts is crucial to ensuring the integrity, fairness, and effectiveness of such a system. Therefore, one should be on the lookout for algorithmic bias, political and ideological influence (given that the definition of what constitutes misinformation can be subjective, and the interpretation of data and content can vary across different cultural, social, and political contexts) and potential unintended consequences (like inadvertently contributing to the spread of false positives or false negatives, which would impact the credibility of legitimate sources).

To mitigate these conflicts of interest, we should have incorporated ethical guidelines and regulatory frameworks into the development process.

References

[1] "United Nations International Computing Center (UNICC)." [Online]. Available: https://www.unicc.org/

[2] "United Nations Development Programme (UNDP)." [Online]. Available: https://www.undp.org/es

[3] N. Kotonya and F. Toni, "Explainable Automated Fact-Checking for Public Health Claims," Oct. 2020, arXiv:2010.09926 [cs]. [Online]. Available: http://arxiv.org/abs/2010.09926

[4] P. Nakov, D. Corney, M. Hasanain, F. Alam, T. Elsayed, A. Barrón-Cedeño, P. Papotti, S. Shaar, and G. D. S. Martino, "Automated Fact-Checking for Assisting Human Fact-Checkers," Mar. 2021, arXiv:2103.07769 [cs] version: 1. [Online]. Available: http://arxiv.org/abs/2103.07769

[5] V. Pérez-Rosas, B. Kleinberg, A. Lefevre, and R. Mihalcea, "Automatic Detection of Fake News," Aug. 2017, arXiv:1708.07104 [cs] version: 1. [Online]. Available: http://arxiv.org/abs/1708.07104

[6] S. Seidman, "Authorship Verification Using the Impostors Method."

[7] L. Pan, W. Chen, W. Xiong, M.-Y. Kan, and W. Y. Wang, "Zero-shot Fact Verification by Claim Generation," May 2021, arXiv:2105.14682 [cs] version: 1. [Online]. Available: http://arxiv.org/abs/2105.14682

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," Dec. 2017, arXiv:1706.03762 [cs] version: 5. [Online]. Available: http://arxiv.org/abs/1706.03762

[9] "Goal 16 | Department of Economic and Social Affairs." [Online]. Available: https://sdgs.un.org/goals/goal16

[10] "Goal 17 | Department of Economic and Social Affairs." [Online]. Available: https://sdgs.un.org/goals/goal17

[11] "Selenium with Python — Selenium Python Bindings 2 documentation." [Online]. Available: https://selenium-python.readthedocs.io/

[12] "Fact-checks | Africa Check." [Online]. Available: https://africacheck.org/fact-checks

[13] "Home." [Online]. Available: https://dubawa.org/

[14] "Fact Check." [Online]. Available: https://factcheck.afp.com/

[15] "Lead Stories," May 2023. [Online]. Available: https://leadstories.com/

[16] "Fact Check Archives," May 2023. [Online]. Available: https://thedispatch.com/category/fact-check/

[17] "Home - News Verifier Africa." [Online]. Available: https://newsverifierafrica.com/

[18] "The Guardian Nigeria News - Nigeria and World News | The Latest news in Nigeria and world news. The Guardian Nigeria Newspaper brings you the latest headlines, opinions, political news, business reports and international news." [Online]. Available: https://guardian.ng/

[19] "Punch Newspapers - Breaking News, Nigerian News, Entertainment, Sport, Business and Politics." [Online]. Available: https://punchng.com/

[20] "Daily Post Nigeria - Nigeria News, Nigerian Newspapers." [Online]. Available: https://dailypost.ng/

[21] "Sahara Reporters — Breaking news, latest stories, photos, videos citizen journalism in Africa." [Online]. Available: https://saharareporters.com/

[22] "The Nation Newspaper - Latest Nigeria news update." [Online]. Available: https://thenationonlineng.net/

[23] "Vanguard News." [Online]. Available: https://www.vanguardngr.com/

[24] "Product." [Online]. Available: https://openai.com/product

[25] "arpanghoshal/EmoRoBERTa · Hugging Face," Jan. 2023. [Online]. Available: https://huggingface.co/arpanghoshal/EmoRoBERTa

[26] "unitary/unbiased-toxic-roberta · Hugging Face." [Online]. Available: https://huggingface.co/unitary/unbiased-toxic-roberta

[27] "SentenceTransformers Documentation — Sentence-Transformers documentation." [Online]. Available: https://www.sbert.net/

[28] [2105.14682], "Zero-shot Fact Verification by Claim Generation," May 2021. [Online]. Available: https://arxiv.org/abs/2105.14682

[29] "About Reuters Fact Check." [Online]. Available: https://www.reuters.com/fact-check/about

[30] J. F. Olivert, "pepe-olivert/unicc_project," Mar. 2023, original-date: 2023-02-13T16:55:28Z. [Online]. Available: https://github.com/pepe-olivert/unicc_project

6 Appendices

6.1 Appendix 1: Preliminar machine learning models

Here we provide a more detailed explanation of the models we tried for the machine learning module as well as the hyperparameters used.

- **Baseline Model**: The pretrained "roberta-fake-news" transformer has been used as the baseline model to evaluate the rest of the results, as it is explained in Section (Model Comparison). A transformer is a deep learning architecture capable of understanding the context of the language, the state of the art when it comes to natural language processing. This particular one is trained to determine if a piece of news is true or not based on 5 evidences, so we checked its performance in the context of tweets concerning the Nigerian elections.

- **Extreme Gradient Boosting (XGB)**: XGB is also an ensemble learning method that combines different decision trees. It builds the trees sequentially, with each tree trying to correct the errors of the previous tree. This method is particularly effective in handling imbalanced datasets and reducing bias. We trained the XGB model on the processed data by setting the number of trees to 200, the same number as the previous model.

- **Support Vector Machine (SVM)**: SVM is a linear model that tries to find the best hyperplane that separates the data into different classes with the largest margin. It can also use non-linear kernels to transform the data into a higher-dimensional space where the classes are separable. We trained the model on the preprocessed data setting the regularization parameter to 1. This parameter controls the trade-off between maximizing the margin (which helps to improve generalization) and minimizing the classification error. And we used the radial basis function (RBF) kernel. We chose this kernel function because it is a popular choice for non-linear classification models as it can capture complex patterns in the data and is less prone to overfitting compared to other kernel functions.

- **Multi-Layer Perceptron (MLP)**: MLP is a neural network model that consists of multiple layers of nodes, where each node applies a nonlinear activation function to its inputs. As for the parameters of the model we used a hidden layer with 50 nodes and a learning rate of 0.001. Adding more layers and nodes can increase the model's accuracy but in our case it takes a lot of time to train, so we didn't use many layers and nodes. The activation function used is ReLU for the hidden layers and a softmax function for the output layer.

- **Random Forests**: Random Forests is an ensemble method which combines the techniques of bagging and randomization. It takes advantage of the instability of decision trees to create diversity by creating lots of them trained on different subsamples from the train data. Moreover, adding randomization creates more variance among the trees, using the "knowledge of the crowd" in its favor. Then, the predictions of each tree are combined.

6.2 Appendix 2: Transformer model example analysis

Example Analysis: Detecting Misinformation in Nigeria with Transformer Model and Attention Mechanism.

In this section, we present an example that demonstrates the process of detecting misinformation in Nigeria using a transformer model. We provide a specific claim related to Nigeria, along with the top 5 closest evidence sentences. Furthermore, we generate an explanation using GPT to determine whether the claim qualifies as misinformation. Additionally, we visualize the attention matrix, which illustrates how the transformer model attends to the token relationships between the claim and evidence sentences, enabling accurate inference regarding the claim's veracity.

Claim: *Nigeria is facing some insecurity problems.*

Top 5 evidences:

1. "Nigeria has struggled with insecurity in recent years, but the resilient spirit of Nigerians continues to shine through.“

2. "Over the years, Nigerians have had to contend with security issues, yet their determination to overcome challenges remains unwavering.“

3. "Before the 2015 general elections, insecurity was a major problem in Nigerian society. However, the country has made significant strides in addressing this issue since then.“

4. "Therefore, talk of insecurity has dominated the Nigerian information space, but Nigerians are resolute in their commitment to build a safer and more secure nation.“

5. "“From this, you can see how serious the Federal Government of Nigeria is in making sure that the Nigerian Army is not only equipped but capable of effectively addressing the problem of insecurity across the country,” emphasized Lieutenant General Adeosun."

The claim is TRUE with a probability of 0.8328

GPT explanation:
The claim that "Nigeria is facing some insecurity problems" is true based on the five pieces of evidence provided. These evidences confirm that Nigeria has been struggling with security issues in recent years. The statements highlight how insecurity has been a major concern in Nigerian society, dominating the information space and instilling fear among Nigerians. The evidence also mentions that before the 2015 general elections, insecurity was a significant problem in Nigeria. Additionally, it acknowledges the seriousness of the Nigerian government in equipping and addressing the issue of insecurity across the country. In conclusion, the evidence presented supports the claim that Nigeria is currently facing insecurity problems.

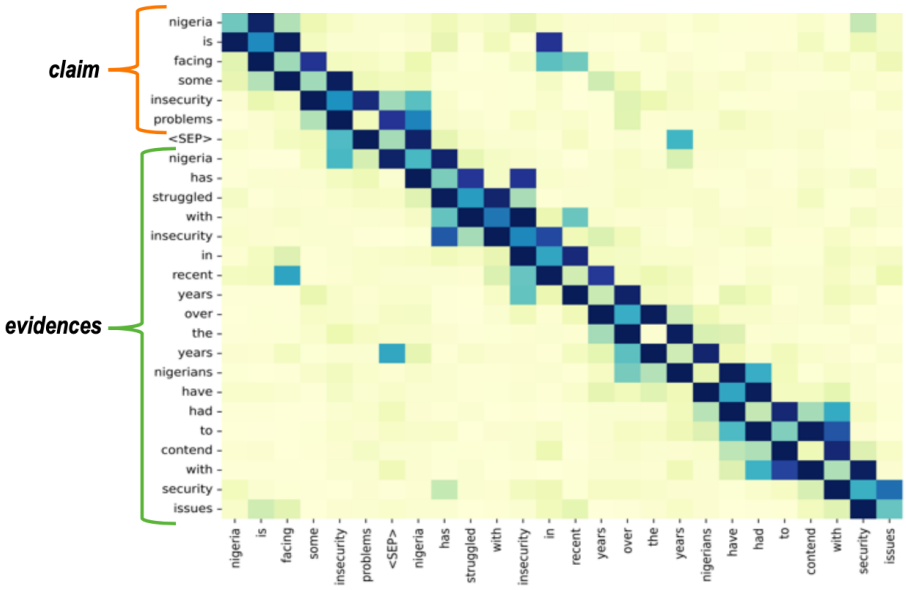

Figure 4: Attention Matrix: Relationships between the Claim and Evidences

To analyze the relationships between the tokens of the claim and the evidence sentences, we can examine the attention matrix generated by the transformer model. The attention matrix illustrates how each token attends to other tokens in the input sequence, indicating the importance or relevance of each token in the context of the others.

In the case of the claim "Nigeria is facing some insecurity problems" and the provided evidence sentences, we can observe the following relationships between tokens:

- The token Nigeria in the claim attends to tokens such as insecurity and security in the evidence sentences. This suggests that the model recognizes the association between Nigeria and the issue of insecurity.
- The token facing in the claim attends to tokens like in and recent in the evidence sentences. This is because the word facing implies a current or ongoing situation, and the tokens in and recent provide contextual information about the timeframe or period in which Nigeria is experiencing the insecurity problems. By attending to these tokens, the model recognizes the connection between the concept of facing problems and the specific time frame associated with the insecurity issues in Nigeria.

By analyzing the attention matrix, we can gain insights into how the transformer model processes and weighs the relationships between tokens, allowing for accurate inference regarding the veracity of the claim.

6.3 Appendix 3: Chatbot sentiment and toxicity bar plots

Here we included the sentiment and toxicity bar plots that the chatbot generates given a claim.

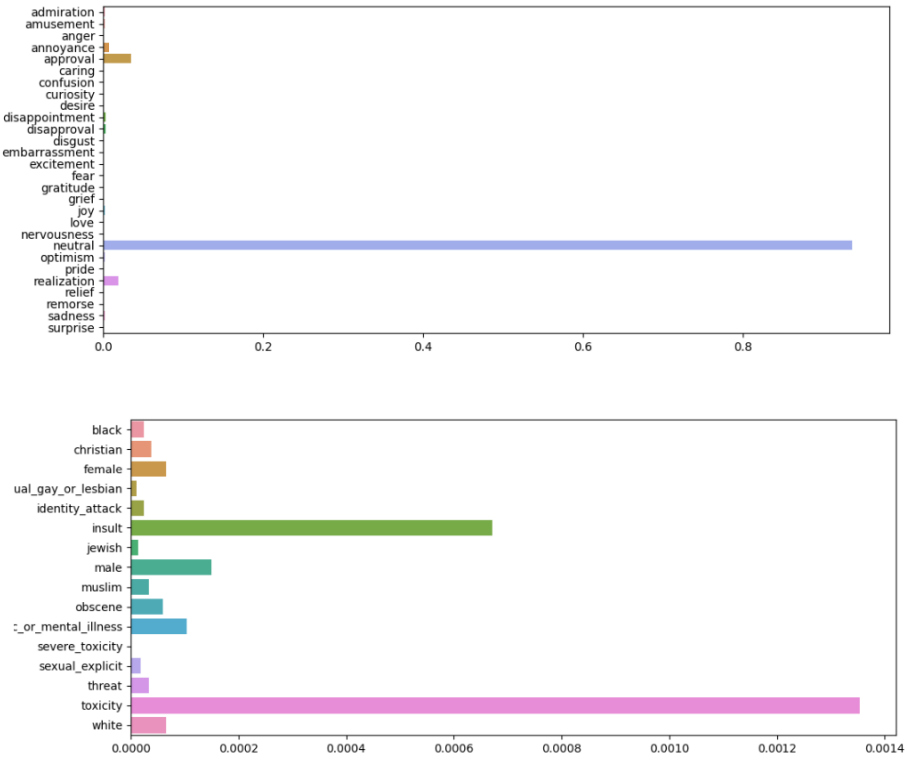

Figure 5: Bar plots of sentiment and toxicity analysis

6.4 Appendix 4: Code repository

You are free to check out our GitHub repository and all the code in it at the moment. This repository contains:

- **Main.py**: The script that makes the bot work.
- **Requirements.txt**: Text file with all the modules we have used.
- **Inference_fake_def3.py**: Python script that contains a function that preprocess the claim when it arrives.
- **Hfmodels.py**: Script that has the code of exporting the "Huggin Face" models.