Narrative Detection Across Nations in Online Social Media Discourse

Stanford CS224N Custom Project

Sungbin Kim
Department of Symbolic Systems
Stanford University
sungbink@stanford.edu

Khaled Messai
Department of Computer Science
Stanford University
kmessai@stanford.edu

Vikram Srinivasan
Department of Symbolic Systems
Stanford University
vikr4m@stanford.edu

Abstract

In this project, we train a multi-task XLM-RoBERTa (XLM-R) model to concurrently handle Named Entity Recognition (NER) and sentiment analysis, focusing on what we define as "Narrative Detection." We define a narrative as an entity-sentiment tuple to capture both the subject and emotions associated with it. This is a version of Aspect-Based Sentiment Analysis (ABSA). While traditional ABSA Hu and Liu (2004) has predominantly focused on consumer reviews for commercial applications, we seek to better understand political discourse in a small subset of the internet. The multilingual nature of our model further distinguishes us from common ABSA applications, leveraging the XLM-R model due to its high performance in multilingual situations Conneau et al. (2019). We experiment with hard and soft parameter sharing techniques, a simple single task approach, and different modular architectures to evaluate their effectiveness in this specialized multitask setting compared to single-task fine-tuned models (XLM-R-NER Chaumond et al. (2021), XLM-R-Sentiment Barbieri et al. (2021)). This study seeks to reveal whether a multitask approach can achieve comparable or superior performance while optimizing computational resources, thus providing valuable insights for deploying efficient and powerful NLP systems in multilingual applications where resources may be limited.

1 Key Information

- Our TA mentor is Zhoujie Ding, we are not sharing this project with another class, and we have no external collaborators or mentors.

The work was divided evenly amongst the three of us, and even though some of the tasks were split up, we collaborated on everything. However we each focused on:

- Sungbin focused on the fine-tuning process and implemented the hyper-parameter optimization techniques. Additionally, he gathered the baseline results and scores from various models.
- Khaled gathered and pre-processed the Reddit comment data. He also led the Machine Learning engineering aspects, focusing on implementing and testing the different multi-task learning approaches.
- Vikram served as the project manager, ensuring all components were completed on schedule. He also developed the dataset labels, completed the data pre-processing step, developed our custom evaluation method, provided assistance in testing, and authored the majority of the report.

2 Introduction

Social media is the largest and most open battlefield of informational warfare, where state actors, corporations, and other organizations seek to influence public perception in order to further their objectives. State actors across the globe increasingly participate in online platforms to support their interests in what are known as “influence operations" Collins (2018). One contemporary example is the ongoing conflict in Gaza. This conflict has multiple narratives involving the same entities — Palestine and Israel — but with completely different sentiments associated with each depending on the author’s beliefs. These differing narratives are widely spread through social media, making it crucial to analyze the platforms for understanding their prevalence and spread. For the purposes of this project, we propose that a narrative is defined as a series of entities with a sentiment attached to each entity in order

to better understand the mobility and frequency of these ideas over time. Thus, our focus is on tracking sentiments and entities across international social media comment communities with the hope of aiding in detecting these influence operations. If you can track where a narrative started and its movements across different mediums, you may gain insight into whether or not it was planned or organic. In this project, we develop a multitask XLM-R model capable of extracting entities and sentiments from a diverse dataset of political social media comments from Reddit in a broad range of languages.

Our project resembles that of Aspect Based Sentiment Analysis (ABSA) or opinion mining, defined as the task of determining a sentiment with respect to a specific aspect in text Chen and Tang (2021). In their pioneering work on ABSA, Hu and Liu (2004) argued that sentiment analysis is possible at three levels: document, sentence, and entity or aspect. Focusing on sentiment analysis at the document or sentence level assumes that only one major topic is expressed in that document or sentence, which is often not the case. A more thorough analysis requires investigation at the entity and aspect level to identify specific entities and detect specific sentiments associated with these entities. Hu and Liu (2004)

We propose a multitask XLM-RoBERTa (XLM-R) model that can detect narratives, by handling NER and sentiment analysis simultaneously. The XLM-R model is a transformer-based masked language model focused on unsupervised learning that significantly outperforms other LLMs, like multilingual BERT (mBERT) on a variety of multilingual tasks Conneau et al. (2019). Through extensive experimentation, we settled on a strategy that employed hard parameter sharing, combined with a fine-tuning strategy that included freezing and unfreezing layers, as well as hyperparameter fine-tuning, to enhance the model's performance. Through this approach, we aimed to improve the model's ability to detect nuanced narratives in a multilingual context. Although the task proved challenging, our approach yielded valuable insights into the complexities of political discourse across different languages. Notably, our results indicated that while the model struggled with the diverse linguistic nuances, it showed promise in more homogeneous language settings, pointing to the potential for further refinement and targeted application.

3 Related Work

Previous work in Aspect-Based Sentiment Analysis (ABSA) has explored various techniques to improve sentiment classification at the entity level. The field has seen significant advancements, particularly in incorporating syntactic and semantic information to enhance sentiment prediction. In this section, we discuss several contributions in the field, the unique solutions they offer, and the challenges they leave unaddressed.

Zhou and Tang (2020) made a notable contribution with their Syntax and Knowledge-based Graph Convolutional Network (SK-GCN) model. This model leverages syntactic dependency trees and commonsense knowledge through Graph Convolutional Networks (GCNs). They introduced two strategies to model these structures: Syntax-based GCN (S-GCN) and Knowledge-based GCN (K-GCN), which model the dependency tree and knowledge graph independently, while SK-GCN combines them. By applying pre-trained BERT, their approach significantly enhanced the representation of sentences towards given aspects. This model addresses the problem of modeling syntactic information with complex tree structures and effectively incorporates commonsense knowledge into aspect-level sentiment analysis.

Building on the limitations identified in earlier works, Wang et al. (2021) proposed a span-based anti-bias aspect representation learning framework. This framework eliminates sentiment bias in aspect embeddings through adversarial learning and aligns distilled opinion candidates with aspects via span-based dependency modeling. Their method improved the interpretability of sentiment polarity predictions, highlighting the importance of addressing bias in sentiment models.

Further advancing the field, Zhang et al. (2022) introduced the Syntactic and Semantic Enhanced Graph Convolutional Network (SSEGCN) model for ABSA. This model combines aspect-aware attention with self-attention to capture both aspect-related semantic correlations and global sentence semantics. By integrating syntactic structure and semantic information through syntactic mask matrices, their approach demonstrated significant improvements on several benchmark datasets, notably SemEval-2014 Task-4 Pontiki et al. (2014). This highlights the ongoing efforts to fuse syntactic and semantic information more effectively for better sentiment analysis.

While syntactic and semantic integration has been a focal point, implicit sentiment remains a challenging aspect. Li et al. (2022) tackled this by focusing on implicit sentiment in aspect-based sentiment analysis. They adopted supervised contrastive pre-training on large-scale sentiment-annotated corpora to better capture implicit sentiment expressions. Their method showed effectiveness in learning both implicit and explicit sentiment orientations towards aspects, addressing a crucial gap in current ABSA models.

Despite these advancements, there remains a gap in the applicability of these models to current political discourse. Most of the discussed models were trained and evaluated on datasets from the mid-2010s, limiting their relevance to contemporary issues. Additionally, most approaches have focused on products or services in customer reviewsHercig et al. (2016). However, these consumer-centric approaches are not directly applicable to the complex nature of

political discourse on social media. Thus, we see a need for models that can effectively monitor political discourse and narratives on a global scale, addressing the unique challenges and nuances of this domain.

4 Approach

4.1 XLM-RoBERTa Transformer Architecture

XLM-RoBERTa (XLM-R) is a multilingual model designed to handle a wide range of languages, based on the Transformer model introduced by Vaswani et al. (2017). This model features a self-attention mechanism, which enables the model to weigh the importance of different words in a sentence. Self-attention calculates attention scores by computing the dot product of query (Q) and key (K) vectors, scaling these scores, and applying a softmax function to get the attention weights. These weights are then used to compute a weighted sum of value (V) vectors, capturing dependencies across the input sequence Vaswani et al. (2017). XLM-R uses multi-head attention, running several attention mechanisms in parallel, each with its own set of Q, K, and V matrices, which allows the model to attend to different parts of the sequence simultaneously. At the end, the outputs from each head are concatenated and linearly transformed Vaswani et al. (2017).

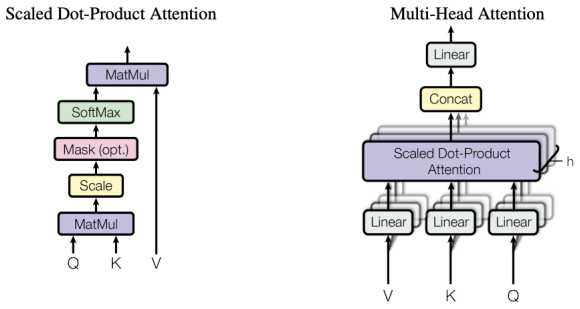

Figure 1: Scaled Dot-Product Attention (left), and Multi-Head Attention (right) consists of several attention layers running in parallel. Vaswani et al. (2017).

Each layer of the Transformer includes a position-wise feed-forward network (FFN) applied to each position independently. The FFN consists of two linear transformations with a ReLU activation in between Vaswani et al. (2017). Positional encodings are added to the input embeddings to provide information about the relative positions of tokens. These encodings, generated using sine and cosine functions of different frequencies, allow the model to distinguish between tokens based on their positions Vaswani et al. (2017).

The XLM-R Base model includes 12 layers, 12 attention heads per layer, and a hidden size of 768, totaling approximately 270 millio parameters. Conneau et al. (2019). XLM-R is pretrained using the masked language modeling (MLM) objective Devlin et al. (2019), which involves randomly masking a certain percentage of tokens in the input sequence, which helps the model generalize better. The final hidden states corresponding to the masked positions are fed into a softmax layer to predict the original tokens, and the loss is calculated based on the cross-entropy between the predicted and actual tokens. Devlin et al. (2019).

To manage diversity across different languages, XLM-R employs subword tokenization using SentencePiece Conneau et al. (2019). SentencePiece tokenizes text into subword units, which helps address the issue of out-of-vocabulary (OOV) words by breaking down rare or complex words into more frequent subword components. For example, "unhappiness" might be tokenized into "un", "happi", and "ness". SentencePiece uses Byte-Pair Encoding (BPE) to construct the subword vocabulary, which starts with a base vocabulary of individual characters and iteratively merges the most frequent pairs of subword units to form longer subwords Sennrich et al. (2016). By processing all languages with the same shared vocabulary, SentencePiece improves the alignment of embedding spaces across languages, making XLM-R versatile for multilingual tasks Conneau et al. (2019).

4.2 Baselines

For our baselines, we developed a Python script that combined the output layers from XLM-R NER model Chaumond et al. (2021) and XLM-R Sentiment Barbieri et al. (2021). It's important to note that each of these models are trained for a single-task use case. Thus, we mimicked a multi-task approach by combining the output layers. Using the default hyper-parameters for each model, we received a score on our custom metric (N-Score) of 0.432. This baseline N-Score score serves as our benchmark to compare as we make improvements.

4.3 Multitask Learning Approach

In our project, we experimented with three different strategies for multitask learning: soft parameter sharing, task-specific layers, and hard parameter sharing. Soft parameter sharing involves having separate models or layers for each task, but with regularization that encourages the parameters to be similar Long Duong and Cook (2015). This approach helps to mitigate the negative transfer and allows for task-specific adaptations. Task-specific layers involve adding layers unique to each task on top of a shared backbone, enabling the model to learn both shared and task-specific representations Roberts and Di (2024). Hard parameter sharing, on the other hand, involves using a single shared model for all tasks without any task-specific adaptations. This method is efficient and prevents overfitting but may suffer from negative transfer if tasks are too dissimilar Ruder (2017).

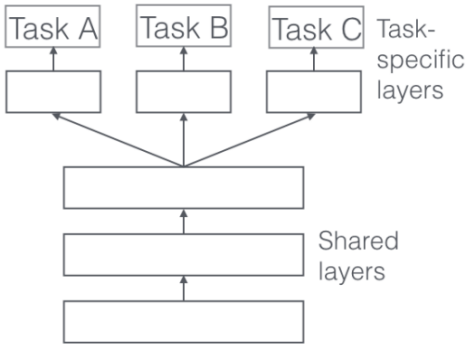

Figure 2: Hard Parameter Sharing Ruder (2017).

We began with a single-task linear model that combined both tasks into one output. The data consisted of IOB tags, e.g. 'B-LOC-neutral', 'I-TIME-positive,' and 'O.' Given that we divided our data into 4 subdivisions (PER, ORG, LOC, TIME), with two prefixes (B, I), and 3 sentiments, (positive, neutral, negative), we get (4 * 3 * 2) + 1 ('O') for 25 labels to classify the data. While the approach was simple and provided useful context, it faced significant challenges due to the expanded label space and lack of specialization, and so it performed rather suboptimally.

With that established, we transitioned to a multi-task model with hard parameter sharing. XLM-R served as the shared encoder for the model and there existed two separate classifiers for our tasks: A linear classifier for entity recognition and a linear classifier for sentiment analysis. We attempted a model employing a Conditional Random Field (CRF) for the NER task, but it struggled. The relative best performance was found with a linear classifier for both. Hard parameter sharing was chosen over its competitors, such as soft parameter sharing with task specific encoders and decoders, because we wanted the model to learn sentiments most commonly associated with the entities. E.g., if you take some historically abhorred figure who is often spoken of poorly, and the model grows to predict that better, it would end up a bit more robust which we thought would help given the broad span of languages. This strategy provided a balanced framework between leveraging shared representations and maintaining task specialization, ultimately improving the models performance.

Due to the nature of the IOB tags, there existed a rather large class imbalance in favor of 'O' tags. We mitigated this by inversely weighing rarer classes by frequency of appearances, so that rarer classes would have a higher weights in the CrossEntropyLoss.

4.4 Fine-tuning Approach

We experimented with multiple fine-tuning strategies, and ultimately adopted a two-phase strategy to leverage the pre-trained capabilities of the XLM-R model while ensuring it adapts effectively to our specific use case and dataset. Initially, we trained the model with a larger learning rate while keeping the backbone layers frozen. This phase allowed us to train the classifier layers without altering the pre-trained weights significantly, ensuring that the model retains its multilingual understanding and general language representations. Specifically, we started with the first six layers frozen and used a learning rate of 1e-4.

After achieving convergence in this initial phase, we continues with the second phase, where we unfroze the entire model and reduced the learning rate to a smaller value, 5e-5. This phase allowed for adjustments across all layers, enabling the model to learn task-specific nuances without forgetting its pre-trained knowledge. This hierarchical fine-tuning approach balances stability and adaptability, preventing catastrophic forgetting and ensuring that the model remains robust across complex contexts, like our political dataset. Goutam et al. (2020) He et al. (2019).

This approach was particularly helpful in our situation because XLM-R was already well-suited for multilingual tasks Conneau et al. (2019). Given our relatively small dataset, it was crucial to maintain the integrity of the original

model's learned features while allowing it to adapt to the specifics of our data, thus we employed this hierarchical fine-tuning approach.

Additionally, we experimented with hyper-parameter fine-tuning. The most popular methods of hyper-parameter tuning are manual grid search and random search. Manual grid search exhaustively searches through a predefined set of hyper-parameter values. Therefore, it can be time-consuming and computationally expensive Bergstra et al. (2012). Random search randomly samples hyper-parameter values within specified ranges and has been shown to be more efficient than grid search, but it does not guarantee the identification of the most optimal hyper-parameters Wu et al. (2019). Research has shown that Bayesian optimization algorithms based on Gaussian processes are able to achieve great accuracy in a few samples when fine-tuning hyper-parameters with a runtime significantly less than manual search Wu et al. (2019). Thus, in our experiment, we employed a Bayesian optimization approach. Similar to random search, this approach searches within a range of hyper-parameters, but it is not stochastic. Bayesian optimization is an advanced method that iteratively updates a probabilistic model based on prior observations to efficiently explore the hyper-parameter space. Bayesian optimization combines a prior distribution of a function with sample information to obtain a posterior distribution of the function. The posterior information is then used to identify where the function is maximized according to a specific criterionWu et al. (2019).

The 'optuna' Akiba et al. (2019) library was created to perdorm bayesian optimization on hyper-parameters. Thus, we used this library and evaluated using the Hugging Face Trainer class Face (2023). The objective function for the optimization process evaluated model performance based on accuracy. Using the 'optuna' library we explored learning rates between 1e-6 and 1e-4, epochs from 2 to 4, and batch sizes of 8, 16, and 32. The optimization process evaluated model performance based on accuracy over ten trials. The best hyper-parameters identified were then used to retrain the model.

5 Experiments

5.1 Data

Given that our goal was to analyze narratives in live, minimally processed social media contexts to capture differences in opinion across multiple nations, we collected a diverse dataset from reddit spanning 13 languages across 18 subreddits, primarily dedicated to politics or news. The languages these texposts come from are as follows: English, Simplified Chinese, Traditional Chinese, Russian, Hindi, Finnish, English, Norwegian, Icelandic, Swedish, German, Brazilian Portuguese, Japanese, and Korean, and the subreddits that we scraped data from are as follows: *"China-irl"*, *"Doubangousegroup"*, *"real-china-irl"*, *"liberalgoosegroup"*, *"mohu"*, *"hanren"*, *"Taiwanese"*, *"liberta"*, *"bakchodi"*, *"indianews"*, *"Suomi"*, *"norge"*, *"de"*, *"iceland"*, *"sweden"*, *"brasil"*, *"newsokur"*, *"hanguk"*.

To collect this data, we used Reddit's PRAW api to traverse the top 150 posts on the subreddit per year. Then, we retrieved every single comment and added a preliminary language tag to the data. To obtain the language tag, we used FastText's Joulin et al. (2016b) Joulin et al. (2016a) language identification model. We took the top three predictions and applied a 1.5 weight on the probability of the expected language. The expected language was calculated by looking at the most common language on the subreddit the message was found on. For example, the expected language for any post on *"indianews"* was Hindi.

Emoji's were another concern as they carry quite a bit of sentiment attached, but historically many models have been trained on corpuses stripped of emoji's. We employed a simple library, python's "emoji" Kim and Wurster (2021), to both convert emoji's into text delimited with colons (":thumbsup:") and translate it into the language detected by FastText in the passage.

We then took the comment body and included all comments with a character count greater than 12 excluding whitespaces. We used character count rather than word count to be inclusive of languages with different scripts which may not easily split into words, such as Japanese or Chinese. Overall, we ended up with a total of 500,000 Reddit comments over the past year.

Labeling this data proved to be quite challenging. As a collective we could speak 2 of the 13 languages we were attempting to analyze. Looking for solutions, we discovered that hand-labeled data has a reasonable degree of error. This is usually mitigated by having multiple labelers review the same data, and so we decided to employ a similar approach. Much work has been done highlighting that Prompt Guided Data Annotation (PGDA) is comparable to hand-labeled data Törnberg (2023) but generally in the context of an individual model. We mimicked techniques used to review hand-labelers and ran our dataset through three separate LLM's (GPT-3.5, Claude Opus, and Google Gemini 1.5) and took the aggregate.

For the NER labels, we used an Inside-Outside-Beginning (IOB) Ramshaw and Marcus (1995) tagging format. This is a commonly used format for NER labels, where B- (Beginning) indicates that the token is at the beginning of a named entity, I- (Inside) indicates that the token is inside a named entity but not at the beginning, and O (Outside)

indicates that the token is outside of any named entity. We attached the sentiment to the end of the IOB tag, so the sentence "I love Palo Alto" would have a corresponding label of ['O', 'O', 'B-LOC-positive', 'I-LOC-positive'].

The final dataset has two columns: text (input) and label (output), with the text column including each textpost, and the label column including the labeled entities and sentiment associated with each entity. An example row from our dataset is: "Text: 'Says Pranit who spread hatred against the United States in the name of supporting Palestine.' Label: ['O', 'B-PER-negative', 'O', 'O', 'O', 'O', 'O', 'B-LOC-negative', 'I-LOC-negative', 'O', 'O', 'O', 'O', 'O', 'B-LOC-neutral']"

5.2 Evaluation method

In evaluating our model, we initially planned to use the F1-score as our primary metric. However, we encountered issues where slight differences between the true sentiment and the predicted sentiment would significantly impact the F1 score, even when the overall label of the data was quite accurate. To address this, we decided to employ a hierarchical evaluation approach that separately assesses entity recognition and sentiment classification, then combines these evaluations into a single metric. We call this metric Narrative Score (N-Score):

5.2.1 Entity-Level Evaluation

For the entity-level evaluation, we focus on the recognition of entities without considering their associated sentiments. We use standard metrics: precision, recall, and F1 score, where:

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)+False Positives (FP)}}$$

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)+False Negatives (FN)}}$$

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision+Recall}}$$

5.2.2 Sentiment-Level Evaluation

For the sentiment-level evaluation, we use a distance-based metric that accounts for how close the predicted sentiment is to the true sentiment.

$$\text{Sentiment Distance} = 1 - \frac{1}{N} \sum_{i=1}^{N} \frac{|\text{True Sentiment}_i - \text{Predicted Sentiment}_i|}{2}$$

Where: N is the number of entities and sentiments are mapped to numerical values: negative = -1, neutral = 0, positive = 1.

5.2.3 Combined Metric

Finally, we combine the entity-level F1 score and the sentiment distance metric into a single metric.

$$\text{N-Score} = \frac{\text{Entity F1 Score} + \text{Sentiment Distance}}{2}$$

To see how our combined metric works, we provided an example calculation in the appendix.

5.3 Experimental details

We used XLM-R AI (2019) as our base model, and the XLMRobertaTokenizerFast from the Hugging Face Transformers library as our tokenizer. In the model fine-tuning phase, we implemented a two-phase strategy. Initially, we froze the first six layers of the XLM-R encoder and trained the model with a larger learning rate of 1e-4. After achieving convergence, we unfroze the entire model and continued fine-tuning with a reduced learning rate of 5e-6.

The final hyper-parameters we landed on were a learning rate of 5e-6 and a dropout rate of 0.1 to prevent over-fitting. The training was conducted for 20 epochs with a batch size of 32, leveraging a GPU to accelerate the process. The AdamW optimizer PyTorch (2023) was used to minimize the loss functions. For NER, we used a weighted cross-entropy loss function, with weights determined by the inverse frequency of class appearances to give higher weight to less frequent data. For sentiment classification, we used standard cross-entropy loss.

We divided our dataset into a train/test/validate split, allocating 60% for training, 20% for validation, and 20% for testing. This ensured that we could effectively train our model, tune hyper-parameters, and evaluate performance on unseen data to prevent over-fitting and ensure generalizability.

5.4 Results

In this section, we present the performance of our models evaluated using the custom N-Score metric. The models evaluated include the Baseline model, Single Task Linear Model, and our final Multi Task model. The N-Scores for each model are shown in Table 1 and Figure 3.

Model	N-Score
Baseline (XLM-R-NER & XLM-R-Sentiment)	0.432
Single Task Linear Model	0.213
Multi Task (Final) Model	0.362

Table 1: N-Scores for Different Models

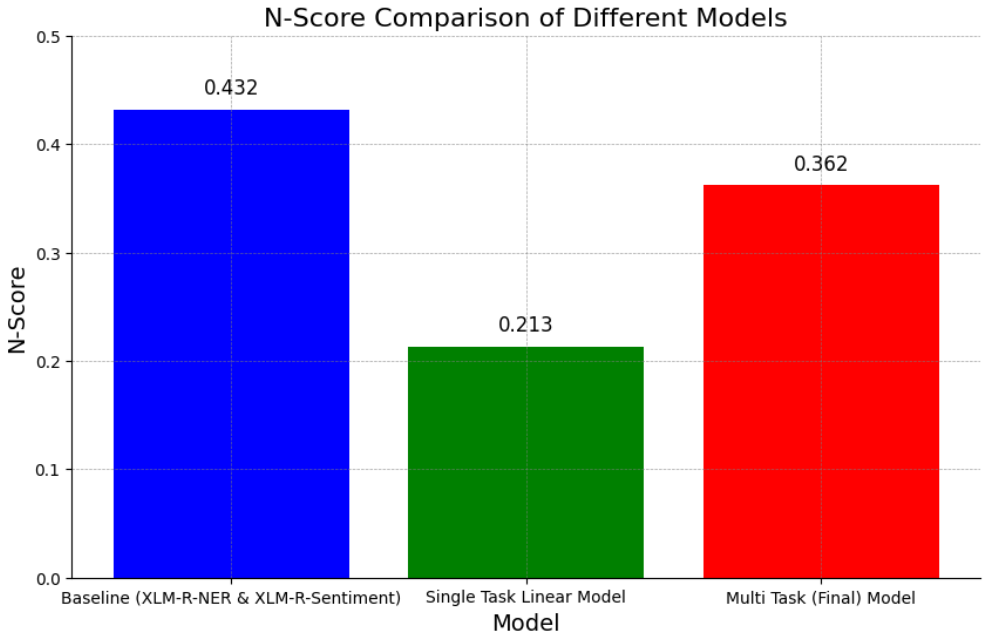

Figure 3: Comparison of N-Scores for Different Models

Table 1 compares the performance of three models using the custom N-Score metric. The N-Scores for these models were 0.432, 0.213, and 0.362, respectively. Although we aimed to surpass the baseline performance, the baseline model's N-Score of 0.432 proved to be a significant obstacle. Our final Multi-task model achieved a notable improvement over the Single-task linear model, indicating the potential benefits of multi-task learning. However, surpassing the baseline proved to be difficult due to the complexity of multilingual contexts.

6 Analysis

Our model's performance, as reflected by the N-Score, was very low, with a final score of 0.362. This outcome indicates that our model faced significant challenges in achieving high accuracy in narrative detection. We believe this is due to the inherent complexity of the task we set out to accomplish, which involves recognizing named entities and their associated sentiments across multiple languages. The task of NER in multilingual contexts is particularly demanding. Despite the robustness of the XLM-R model in handling multilingual tasks, the added complexity of detecting sentiments tied to specific entities proved difficult. Our experiments with various fine-tuning and optimization techniques, including the freezing/unfreezing strategy and different multitask learning approaches, were unable to significantly enhance the model's performance.

This difficulty is also evident when looking at the baseline scores from state-of-the-art LLMs like XLM-R-NER and XLM-R-Sentiment. While these models excel in their respective single-task environments, their performance dropped substantially when tasked with simultaneously identifying entities and determining sentiments associated with each entity.

One thing to note is that when we tested our model on an extremely small dataset of English text posts (100 text posts), the N-Score was significantly higher at 0.603. This result indicates that the multilingual aspect of this task added a considerable level of difficulty. We hypothesize that learning specific sentiments associated with Chinese

entities does not help in learning about specific sentiments associated with Norwegian entities. In fact, it is likely actively detrimental. This suggests that cross-linguistic transfer of sentiment information may be limited, making multilingual sentiment analysis a challenging problem, especially with different language families. Although XLM-R is pretrained on a vast array of languages, it's clear that the nuances of specific political discourse and the subtleties of sentiment in those contexts were not fully captured.

Despite these challenges, our approach did yield some insights. The process of developing and testing various fine-tuning strategies provided valuable lessons in limitations of current NER and sentiment analysis techniques in multilingual contexts. Additionally, the relatively low baseline scores of XLM-R-NER and XLM-R-Sentiment underscore the need for more sophisticated models capable of handling multitask learning in complex environments.

7 Limitations

We brought up the biggest limitation in the Analysis section, as narrative detection in a multilingual context is incredibly difficult. In addition to that, our dataset was labeled via prompt-guided data annotation, covering a variety of languages. Due to the multiple languages involved, it was impossible for us to verify the correctness of annotations, as we weren't familiar with many of the languages. This raises concerns about the consistency and accuracy of the dataset, which could impact the model's performance and reliability.

Similarly, the model trained on Reddit data may not generalize well to other domains, such as news articles, blogs, or other social media platforms. The specific language and style prevalent on Reddit might not be representative of broader online discourse. Consequently, the model's generalizability could be limited, potentially requiring further adaptation to maintain performance across diverse sources of text.

8 Conclusion

Our project set out to tackle the challenging task of narrative detection in a multilingual context, aiming to recognize named entities and their associated sentiments across various languages. Despite leveraging the robust XLM-R model and implementing a variety of fine-tuning and optimization strategies, our final N-Score of 0.362 highlights the significant difficulties inherent in this task. The complexity of NER in multilingual environments, combined with the nuanced task of sentiment analysis, proved challenging for our model.

The task of accurately detecting narratives across multiple languages is inherently difficult due to the unique linguistic and cultural nuances present in each language. Even though XLM-R's pretraining on a wide array of languages provided a strong foundation, the model struggled with the specific subtleties of political discourse. This challenge was compounded by our dataset being predominantly sourced from Reddit, a platform known for its liberal leaning, which may have introduced bias. We observed that our model performed significantly better on a smaller dataset of English text posts, achieving an N-Score of 0.603. This highlights the difficulty of cross-linguistic transfer of sentiment information, suggesting that the current multilingual models may not fully capture the intricacies of political discourse across different languages.

In light of these challenges, our work underscores the need for more sophisticated models and datasets to handle the intricacies of multilingual sentiment analysis. Looking ahead, we aim to expand and diversify our dataset by collecting data from a broader range of platforms, such as social media, news outlets, blogs, and forums. This approach will help mitigate the bias introduced by relying solely on Reddit, which is known for its more liberal leaning. By incorporating data from platforms with different ideological perspectives, we can develop a more balanced dataset that captures a wider range of viewpoints and discussions. This will significantly improve the model's ability to generalize and perform unbiased narrative detection and sentiment analysis.

Another key direction for future work is to refine our approach to narrative detection in multilingual contexts. Given the complexity of handling multiple languages, one potential strategy is to focus on individual languages or families of languages. For instance, we could experiment with narrative detection within the Romance languages (French, Italian, Spanish, Portuguese, and Romanian) to see if focusing on linguistically similar languages yields better results. Additionally, we could explore machine translation techniques to translate all texts to English before performing narrative detection, then translating the results back to the original languages. However, this approach may introduce its own challenges, especially for underrepresented languages. On the model architecture front, investigating more sophisticated multitask learning frameworks, such as multi-gate mixture-of-experts or dynamic task allocation, could better manage the complexities of different tasks and enhance our model's performance. Integrating external knowledge bases or contextual information to improve the model's understanding of entities and sentiments in ambiguous cases is another promising avenue for future research.

9 Ethics Statement

One significant ethical concern stems from the fact that our dataset is exclusively composed of Reddit comments. Reddit comments tend to lean towards a liberal perspective. We saw evidence of this by the prevalent negative sentiments towards entities such as the Chinese Communist Party observed in our dataset. This ideological bias could cause our model to generalize poorly and predominantly recognize one side of political narratives. This issue is particularly dangerous given the global scope of our model's application in detecting political narratives. If the model systematically fails to identify or accurately interpret narratives from different ideological perspectives, it could perpetuate misinformation or exacerbate political polarization. To address this, we could diversify our dataset by incorporating comments and posts from a wider range of social media platforms that cater to different ideological spectrums, such as conservative-leaning forums or international platforms like Weibo and VKontakte.

Another major ethical issue involves privacy concerns related to scraping sensitive information such as usernames, timestamps, and subreddits associated with each text post. Although the Reddit API grants permission to collect this data, we did not seek explicit consent from users for their private information. While none of this personally identifiable information was included in the model, and our dataset was not published publicly, the act of collecting it raises concerns. If we were to generalize to try and identify influence operations using this user metadata, there would be large concerns about improper classification and the harm that it could pose to users. To mitigate privacy these risks, we recommend implementing a script to strip all personally identifiable information (PII) from the dataset after the scraping process, but before we are able to analyze it. Although Reddit users grant permission for their data to be used, stripping PII before we implement our model would increase trust and transparency.

References

Facebook AI. 2019. Xlm-roberta base model. `https://huggingface.co/FacebookAI/xlm-roberta-base`. Accessed: 2024-06-06.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Francesco Barbieri et al. 2021. twitter-xlm-roberta-base-sentiment. *Hugging Face*.

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305.

Julien Chaumond et al. 2021. tf-xlm-r-ner-40-lang. *Hugging Face*.

Lihui Chen and Mingwei Tang. 2021. Aspect-based sentiment analysis. `https://www.sciencedirect.com/topics/computer-science/aspect-based-sentiment-analysis`. Accessed: 2024-06-05.

Ben Collins. 2018. Volunteers found iran's propaganda effort on reddit — but their warnings were ignored. *nbcnews*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Hugging Face. 2023. Trainer — transformers 4.10.0 documentation. Accessed: 2023-06-06.

Kelam Goutam, S. Balasubramanian, Darshan Gera, and R. Raghunatha Sarma. 2020. Layerout: Freezing layers in deep neural networks. *SN Computer Science*, 1(3):295.

He He, Xiaochang Tan, Sheng Chen, Junjie Liu, Wenqing Zhou, and Yanzhao Wu. 2019. Rethinking pre-training and self-training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 769–779.

Tomás Hercig, Tomás Brychcín, Lukáš Svoboda, and Michal Konkol. 2016. Uwb at semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of SemEval-2016*, pages 342–349.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 168–177.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016a. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016b. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Taehoon Kim and Kevin Wurster. 2021. emoji: Emoji for python. `https://pypi.org/project/emoji/`. Version 2.0.0.

Ruifan Li, Hao Chen, Fangxiang Feng, et al. 2022. Supervised contrastive pre-training for implicit sentiment analysis. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4916–4925.

Steven Bird Long Duong, Trevor Cohn and Paul Cook. 2015. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*.

Maria Pontiki, Haris Papageorgiou, Dimitrios Galanis, Ion Androutsopoulos, John Pavlopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland.

PyTorch. 2023. Adamw optimizer. `https://pytorch.org/docs/stable/generated/torch.optim.AdamW.html`.

Lance A. Ramshaw and Mitchell P. Marcus. 1995. Text chunking using transformation-based learning. *arXiv preprint cmp-lg/9505040*.

Josselin Somerville Roberts and Julia Di. 2024. Projected task-specific layers for multi-task reinforcement learning. *arXiv preprint arXiv:2309.08776*.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725.

Petter Törnberg. 2023. Chatgpt-4 outperforms experts and crowd workers in annotating political twitter messages with zero-shot learning.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Jingjing Wang, Jie Li, Shoushan Li, et al. 2021. A span-based anti-bias aspect representation learning framework for aspect-level sentiment classification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 246–256.

Jia Wu, Xiu-Yun Chen, Hao Zhang, Li-Dong Xiong, Hang Lei, and Si-Hao Deng. 2019. Hyperparameter optimization for machine learning models based on bayesian optimization. *Journal of Electronic Science and Technology*, 17(1):26–40.

Zheng Zhang, Zili Zhou, and Yanna Wang. 2022. Syntactic and semantic enhanced graph convolutional network for aspect-based sentiment analysis. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4916–4925.

Lihui Zhou and Mingwei Tang. 2020. Syntax- and knowledge-based graph convolutional network for aspect-level sentiment classification. *Knowledge-Based Systems*.

A Appendix

A.1 Evaluation Method Example

To see our N-Score evaluation method in action, consider this example from our dataset:

Text: "Says Pranit who spread hatred against the United States in the name of supporting Palestine."

True Labels: ['O', 'B-PER-negative', 'O', 'O', 'O', 'O', 'O', 'B-LOC-negative', 'I-LOC-negative', 'O', 'O', 'O', 'O', 'O', 'B-LOC-neutral']

Predicted Labels: ['O', 'B-PER-negative', 'O', 'O', 'O', 'O', 'O', 'B-LOC-negative', 'I-LOC-negative', 'O', 'O', 'O', 'O', 'O', 'B-LOC-positive']

Step-by-Step Calculation

1. **Entity-Level Evaluation**:

- Extract entities (ignoring sentiments):
  - True Entities: ['B-PER', 'B-LOC', 'I-LOC', 'B-LOC']
  - Predicted Entities: ['B-PER', 'B-LOC', 'I-LOC', 'B-LOC']
- Calculate TP, FP, FN:
  - TP = 4 ['B-PER', 'B-LOC', 'I-LOC', 'B-LOC']
  - FP = 0
  - FN = 0
- Precision = TP / (TP + FP) = 4 / (4 + 0) = 1.0
- Recall = TP / (TP + FN) = 4 / (4 + 0) = 1.0
- F1 Score = 2 * (Precision * Recall) / (Precision + Recall) = 2 * (1.0 * 1.0) / (1.0 + 1.0) = 1.0

2. **Sentiment-Level Evaluation**:

- Sentiment mapping: negative = -1, neutral = 0, positive = 1
- True Sentiments: [-1, -1, -1, 0]
- Predicted Sentiments: [-1, -1, -1, 1]
- Sentiment Distances:

$$\text{Distance for 'B-LOC-neutral' vs. 'B-LOC-positive'} = 1 - \frac{|0-1|}{2} = 1 - 0.5 = 0.5$$

- Average Sentiment Distance = $\frac{3 \cdot 1 + 0.5}{4} = \frac{3.5}{4} = 0.875$

3. **N-Score**:

$$\text{N-Score} = \frac{\text{Entity F1 Score} + \text{Sentiment Distance}}{2} = \frac{1.0 + 0.875}{2} = 0.9375$$