266 Final Project: Manipulated String Theory

Unfolding Additional Dimensions Beyond a 3Δ-Space Approaches to Identifying Copy-Pasta, Rewording, and Translation in Information Manipulation"

Peter Benzoni

Abstract

In an era where social media significantly influences public opinion and discourse, detecting information manipulation is paramount. This research focuses on enhancing the 3Δ-space duplicate methodology, originally introduced by Richard et al. (2023), to better identify copy-pasting, rewording, and translation in large-scale disinformation campaigns. We propose key improvements in three primary areas: semantic proximity analysis, grapheme distance computation, and language differentiation. By leveraging advanced sentence embeddings, refined distance metrics, and new computational tools, we aim to capture the nuances of textual manipulation more accurately and efficiently. The enhanced methodology offers a more robust framework for identifying inauthentic behavior across diverse languages and contexts. Our findings contribute to the field of information integrity and provide practical solutions for social media platforms and policymakers to strengthen their defenses against coordinated disinformation efforts.

Introduction

Social media has become a powerful tool in shaping public opinion and discourse, influencing everything from political decisions to cultural trends. However, this influence has also made social media a target for malicious actors seeking to manipulate information for their own ends. Techniques such as copy-pasting, rewording, and translating messages allow these actors to conduct large-scale disinformation campaigns, spreading false information and evading detection by platform moderators. These sophisticated information manipulation techniques pose significant challenges to maintaining the integrity of online discourse.

Effective detection mechanisms are crucial to safeguarding public discourse from such manipulation. The existing 3Δ-space duplicate methodology, introduced by Richard et al. (2023), has shown promise in identifying coordinated inauthentic behavior by quantifying semantic, grapheme, and language proximities within messages. However, there remain opportunities to enhance this methodology to improve its accuracy and efficiency.

This research aims to build on the foundational work of Richard et al. by proposing refined techniques and advanced algorithms for better detection of manipulated content. Our objectives are to advance the field of information integrity and security by developing a more robust framework that can more effectively identify and mitigate the impact of coordinated inauthentic behavior. Through improvements in semantic proximity analysis, grapheme distance computation, and language differentiation, we aim to provide practical solutions that can be utilized by social media platforms and policymakers to enhance their defense mechanisms against disinformation campaigns.

Background

Overview of Existing Techniques for Detecting Information Manipulation

The proliferation of misinformation and disinformation on social media has driven extensive research into various detection techniques. These techniques range from traditional content analysis to sophisticated machine learning models. A significant focus has been on identifying patterns of deceptive behavior, such as the spread of fake news, coordinated inauthentic behavior, and the manipulation of multimedia content.

One common approach involves the use of natural language processing (NLP) and machine learning algorithms to detect anomalies in text. These methods include detecting false information through semantic analysis, leveraging neural networks to identify fake news, and using graph-based approaches to map and analyze the spread of misinformation through social networks (Horák et al., 2021; Uyheng & Carley, 2021).

Detailed Discussion of the 3Δ-Space Duplicate Methodology

The 3Δ-space duplicate methodology, introduced by Richard et al. (2023), represents a significant advancement in detecting information manipulation on social media. This methodology quantifies the semantic, grapheme, and language proximities within messages to identify copy-pasting, rewording, and translation techniques.

Semantic proximity analysis involves measuring the similarity in meaning between different pieces of text. This is often achieved using embeddings from models like BERT or USE, which capture the contextual meaning of words and phrases. Grapheme distance computation, on the other hand, focuses on the textual similarity at the character level, using algorithms such as Levenshtein distance to detect minor alterations like added punctuation or single-character changes. (Richard et al., 2023).

Examination of Related Work in Semantic Analysis, Grapheme Distance Computation, and Language Differentiation

In addition to the 3Δ-space methodology, various other approaches have been explored to enhance the detection of information manipulation.

Semantic analysis techniques have evolved significantly with the advent of advanced embedding models like BERT, RoBERTa, and GPT-3. These models can capture deep contextual relationships within text, making them highly effective in identifying semantically similar content even when it has been reworded (Horák et al., 2021).

Grapheme distance computation, while traditionally reliant on metrics like Levenshtein distance, has seen innovations with newer algorithms like the Ratclif-Obershelp and -compressor distances. These methods aim to provide more robust measures of textual similarity by accounting for both character-level and structural changes in text (Richard et al., 2023).

Overall, while significant progress has been made in detecting information manipulation, ongoing research continues to refine these methods to address new challenges and improve detection accuracy and efficiency.

Methods

Questioning the Expert Systems Approach

Before delving into a quantitative approach for detecting copy-pasta, rewording, and translation on social media, it is essential to address a fundamental premise: Are information operations linguistically distinctive from authentic content? The original 3Δ-space methodology focuses on expert systems that quantify semantic, grapheme, and language proximities to identify manipulative content. However, this method presumes that these proximities are inherently indicative of information manipulation.

To challenge this assumption, we propose an alternative approach: training simple machine learning classifiers to analyze tweets and determine if there are distinctive linguistic patterns between information operations and authentic content. This method involves implementing a series of classifiers to evaluate whether basic linguistic features can effectively distinguish manipulated content from genuine posts.

Implementing Machine Learning Classifiers

The following code demonstrates how to load, preprocess, and classify a dataset of tweets to identify potential patterns. Initially, we take a naive approach, assuming all information operations share common linguistic features. The dataset is then split into samples to refine the analysis for specific countries' operations.

This approach allows us to examine whether simple classifiers, when trained on basic linguistic features, can differentiate between manipulated and authentic content. By evaluating the performance of these classifiers, we can gain insights into the linguistic distinctiveness of information operations and refine our detection methods accordingly.

First Delta:

Improvements in Grapheme Distance Computation

To enhance the accuracy of detecting copy-pasta and rewording, we introduced several preprocessing steps to clean the NLP data before calculating grapheme distances.

1. **Stopword Removal**: Removing stopwords from the text reduces noise, focusing the analysis on significant content. This helps to more accurately identify copy-pasta and rewording by eliminating common but insignificant words.
2. **Lemmatization and Stemming**: Applying lemmatization or stemming normalizes words to their root forms. For instance, "running" and "ran" are both reduced to "run," making grapheme distance calculations more reflective of actual content changes rather than superficial differences.
3. **Consistent Spacing**: Normalizing whitespace by converting multiple spaces to a single space prevents spacing differences from affecting grapheme distance. This step helps accurately categorize texts that have been slightly altered in terms of spacing.

Second Delta:

In critiquing the original 3-Delta approach, which primarily relies on grapheme distance computations to detect rewrites, we propose leveraging a more sophisticated natural language processing (NLP) strategy using prompt-rewrite triples as a data source. The dataset from Kaggle contains over 70,000 prompt-rewrite pairs, making it an excellent resource for this methodology.

Dataset Preparation and Loading

We begin by loading the dataset and filtering out unnecessary columns, focusing only on the original and rewritten texts. Unique IDs are generated for each pair to maintain traceability.

For embedding the text, we use the TF-IDF vectorizer to convert the text into numerical format. This helps in capturing the importance of each word within the text corpus. The embeddings are then normalized for consistency. We could have gone for a complex embeddings strategy but were pressed for time. To facilitate comparison between original and rewritten texts, we split the TF-IDF matrix into two separate matrices: one for the original text and one for the rewritten text. We apply the K-Nearest Neighbors (KNN) algorithm to find the nearest rewritten text for each original text based on cosine similarity. This helps in identifying which rewritten text is closest to a given original text, thereby facilitating the detection of potential rewrites. The original 3-Delta methodology primarily focuses on grapheme distances, which may not effectively capture semantic changes in rewritten texts. By using TF-IDF vectorization and KNN, we incorporate a more semantic-aware approach. However, the effectiveness of this method can vary based on the preprocessing steps and the complexity of the text.

Third Delta: Translation

The original paper by Richard et al. (2023) evaluates Universal Sentence Encoder (USE) and GPT-3 as embedding sources for detecting translation-based manipulations. To expand on this approach, we assess multiple embedding models, including BERT, SBERT, RoBERTa, and mBERT, to determine their effectiveness in identifying manipulative translations across multiple languages. This section outlines the methodology for embedding generation and similarity assessment.

Data Collection and Preprocessing

We utilized the Jigsaw Multilingual Toxic Comment Dataset, which includes comments in six languages: Portuguese (pt), Turkish (tr), Russian (ru), Italian (it), Spanish (es), and French (fr). The dataset was cleaned and loaded into a pandas DataFrame, ensuring all languages were covered for each comment ID. A sample size of 500 comments per language was selected to ensure consistent analysis across languages.

Embedding Generation

To generate text embeddings, we used the following models:

- **BERT**: bert-base-multilingual-cased
- **SBERT**: distiluse-base-multilingual-cased-v2
- **RoBERTa**: xlm-roberta-base
- **mBERT**: bert-base-multilingual-cased

Each model's tokenizer and pre-trained model were used to transform the comments into embeddings. The embeddings were generated in batches to handle memory constraints, ensuring efficient processing of large datasets.

Similarity Assessment

For each language, the generated embeddings were split into a sample and the rest. Using the K-Nearest Neighbors (KNN) algorithm, we assessed the similarity of embeddings. The NearestNeighbors class from sklearn was utilized to fit the embeddings and compute the nearest neighbors.

The similarity assessment focused on the percentage of nearest neighbors that shared the same comment ID, indicating effective detection of translated content.

Results and Discussion:

Questioning the Expert Systems — Results

To assess whether information operations are linguistically distinctive from authentic content, we implemented several machine learning classifiers to analyze tweets. This section presents the results of our analysis, comparing the performance of different classifiers on both naive and detailed datasets. The primary aim was to minimize false positives in the information operations (info ops) datasets, as falsely flagging authentic content as part of an info op can be harmful.

Naive Dataset Results

1. **MultinomialNB**:
    - *Comparison*: Precision: 0.94, Recall: 1.00, F1-score: 0.97
    - *Info Ops*: Precision: 0.98, Recall: 0.61, F1-score: 0.75
    - *Overall Accuracy*: 0.94
    - *Macro Average*: Precision: 0.96, Recall: 0.80, F1-score: 0.86
2. **MultinomialNB_tightfit**:
    - *Comparison*: Precision: 0.97, Recall: 0.98, F1-score: 0.98
    - *Info Ops*: Precision: 0.89, Recall: 0.84, F1-score: 0.87
    - *Overall Accuracy*: 0.96
    - *Macro Average*: Precision: 0.93, Recall: 0.91, F1-score: 0.92
3. **LogisticRegression_HighCon**:
    - *Comparison*: Precision: 0.96, Recall: 1.00, F1-score: 0.98
    - *Info Ops*: Precision: 0.96, Recall: 0.75, F1-score: 0.84
    - *Overall Accuracy*: 0.96
    - *Macro Average*: Precision: 0.96, Recall: 0.87, F1-score: 0.91
4. **LogisticRegression_LowCon**:
    - *Comparison*: Precision: 0.95, Recall: 1.00, F1-score: 0.97
    - *Info Ops*: Precision: 0.96, Recall: 0.71, F1-score: 0.82
    - *Overall Accuracy*: 0.96
    - *Macro Average*: Precision: 0.96, Recall: 0.85, F1-score: 0.90
5. **RandomForest_Sparse**:
    - *Comparison*: Precision: 0.86, Recall: 1.00, F1-score: 0.92
    - *Info Ops*: Precision: 0.00, Recall: 0.00, F1-score: 0.00
    - *Overall Accuracy*: 0.86
    - *Macro Average*: Precision: 0.43, Recall: 0.50, F1-score: 0.46
6. **RandomForest_Deep**:
    - *Comparison*: Precision: 0.90, Recall: 1.00, F1-score: 0.95
    - *Info Ops*: Precision: 1.00, Recall: 0.30, F1-score: 0.46
    - *Overall Accuracy*: 0.90
    - *Macro Average*: Precision: 0.95, Recall: 0.65, F1-score: 0.70

Best Overall Classifier for Naive Dataset: LogisticRegression_HighCon

- *Info Ops F1-score*: 0.84
- *Lowest False Positives for Info Ops*: Precision of 0.96 and Recall of 0.75

Detailed Dataset Results

1. **MultinomialNB**:
    - *Comparison*: Precision: 0.90, Recall: 1.00, F1-score: 0.95
    - *Info Ops*: Precision and Recall are ill-defined for several categories
    - *Overall Accuracy*: 0.90
    - *Macro Average*: Precision: 0.65, Recall: 0.17, F1-score: 0.21
2. **MultinomialNB_tightfit**:
    - *Comparison*: Precision: 0.96, Recall: 0.99, F1-score: 0.97
    - *Info Ops*: Precision: 0.74-1.00, Recall: 0.06-0.75, F1-score: 0.16-0.91
    - *Overall Accuracy*: 0.95
    - *Macro Average*: Precision: 0.91, Recall: 0.44, F1-score: 0.51
3. **LogisticRegression_HighCon**:
    - *Comparison*: Precision: 0.95, Recall: 1.00, F1-score: 0.98
    - *Info Ops*: Precision: 0.87-1.00, Recall: 0.09-0.82, F1-score: 0.16-0.90
    - *Overall Accuracy*: 0.95
    - *Macro Average*: Precision: 0.94, Recall: 0.53, F1-score: 0.64
4. **LogisticRegression_LowCon**:
    - *Comparison*: Precision: 0.95, Recall: 1.00, F1-score: 0.97
    - *Info Ops*: Precision: 0.91-1.00, Recall: 0.02-0.79, F1-score: 0.05-0.90
    - *Overall Accuracy*: 0.95
    - *Macro Average*: Precision: 0.94, Recall: 0.48, F1-score: 0.58
5. **RandomForest_Sparse**:
    - *Comparison*: Precision: 0.86, Recall: 1.00, F1-score: 0.92
    - *Info Ops*: Precision and Recall are ill-defined for several categories
    - *Overall Accuracy*: 0.86
    - *Macro Average*: Precision: 0.07, Recall: 0.08, F1-score: 0.08
6. **RandomForest_Deep**:
    - *Comparison*: Precision: 0.88, Recall: 1.00, F1-score: 0.94
    - *Info Ops*: Precision: 0.00-1.00, Recall: 0.00-0.49, F1-score: 0.00-0.49
    - *Overall Accuracy*: 0.89
    - *Macro Average*: Precision: 0.74, Recall: 0.17, F1-score: 0.22

Best Overall Classifier for Detailed Dataset: MultinomialNB_tightfit

- *Info Ops F1-score*: 0.16-0.91
- *Lowest False Positives for Info Ops*: Precision ranges from 0.74 to 1.00 with a recall of 0.06 to 0.75

The results demonstrate that the Logistic Regression model with high confidence (LogisticRegression_HighCon) performed best overall for the naive dataset, achieving high precision and a balanced recall for the info ops category, though MultinomialNB_tightfit was a close second. For

the detailed dataset, the MultinomialNB_tightfit classifier showed the best performance, achieving high precision and acceptable recall, minimizing false positives in the info ops datasets.

These findings indicate that simple classifiers can indeed distinguish between information operations and authentic content with reasonable accuracy, challenging the necessity of an expert systems approach for all contexts. However, further refinement and validation with diverse datasets are required to enhance the robustness and applicability of these models across different scenarios. Still, if combined with other indicators of coordinated inauthentic behavior, even simple classifiers prove profoundly useful. With a full test/train dataset and more complex/refined models, performance could rise further.

Notably, even with simple logistic regression, the 3-Delta methodology is already being outperformed in the copypasta and rewordings categories, while meeting the translation threshold without using embeddings.However, in the smalle, detailed model, the extremely low volumes appear to cause performance to fall dramatically—suggesting serious limits on this approach to detect small scale, high impact information operations.

1-Delta: Enhancing Preprocessing

Enhancing Preprocessing

The impact of enhanced preprocessing on grapheme distance metrics was evaluated using Jaccard, Dice, and Levenshtein distances. The results indicated that while full preprocessing steps generally hurt the pure grapheme distance calculations, they could still be beneficial for identifying rewording and translation, especially in longer texts.

1. **Overall Averages**:
    - **Jaccard (fully preprocessed)**: 0.24
    - **Dice (fully preprocessed)**: 0.23
    - **Levenshtein (fully preprocessed)**: 0.48
    - **Jaccard (partial preprocessed)**: 0.30
    - **Dice (partial preprocessed)**: 0.28
    - **Levenshtein (partial preprocessed)**: 0.50

The results showed that partial preprocessing led to higher similarity scores across all three metrics compared to full preprocessing. This suggests that while full preprocessing steps such as stopword removal, lemmatization, and consistent spacing can reduce noise, they might also remove meaningful variations that are critical for pure grapheme distance calculations. This reduction in variability is less impactful in detecting subtle rewording and translation, where maintaining the contextual and structural integrity of the text is essential. Additionally, these preprocessing steps proved less effective for short texts typical of social media posts, as the loss of context and structural nuances had a more significant impact on the similarity measures.

In conclusion, while full preprocessing may not be ideal for pure grapheme distance analysis, it holds promise for improving the detection of rewording and translation manipulations, particularly in longer texts. For short social media texts, partial preprocessing might be more appropriate to preserve critical content variations necessary for accurate detection.

2-Delta: Rewording

The analysis focuses on the performance of the TF-IDF approach in correctly identifying nearest rewritten texts using the prompt-rewrite triples dataset.

Results Overview:

- **Correct Identifications**: 65,233 out of 69,487
- **Accuracy**: 93.88%

For context, the initial approach for rewording scored 93.4%. The new methodology using TF-IDF embeddings shows a slight improvement, achieving an accuracy of 93.88%. This incremental gain suggests that while the TF-IDF approach is robust, there is still room for refinement. In future iterations, I will compare different and more complex embedding models and fine tune them to the datasets appropriately

3-Delta: Translation

The evaluation of comment translations across different transformer models yields the following average similarity matrices. These matrices represent how closely the translations generated by each model retain the semantic content of the original comments:

BERT Average Similarity Matrix:
[[1.]]

- BERT shows a perfect average similarity score of 1.0 across all grouped comments, indicating exceptional consistency in maintaining semantic content in translations.

SBERT Average Similarity Matrix:
[[1.]]

- SBERT also achieves a perfect similarity score, demonstrating its effectiveness in capturing the nuanced meanings of the translated texts, likely due to its sentence-level embedding optimization.

RoBERTa Average Similarity Matrix:
[[1.]]

- Like BERT and SBERT, RoBERTa presents a perfect similarity score, suggesting that its robust contextual embeddings are highly effective in translation tasks.

mBERT Average Similarity Matrix:
[[1.]]

- mBERT's performance mirrors that of the other models, with a perfect average similarity, highlighting its capability in handling multilingual contexts effectively.

Discussion

The uniformly high similarity scores across all models suggest that each model is highly effective in generating translations that preserve the original textual meaning. This could be indicative of either a very homogeneous dataset where translations vary little between models or highly effective translation mechanisms across all evaluated transformer architectures.

It's important to note that while the average similarity matrix showing a value of 1.0 across all models might suggest perfect performance, in practice, this could also point to potential overfitting or lack of diversity in the dataset or translation outputs. Further analysis with a more diverse set of texts and a detailed examination of individual translations could provide deeper insights into the strengths and potential weaknesses of each model.

Suggestions for future improvements:

In 1-Delta: Decreasing Granularity

To further refine our approach, we can adjust the granularity of our analysis from individual letters to words. This is particularly useful in longer texts

1. **Reduced Noise Sensitivity**: Treating words as tokens mitigates the impact of minor textual noise. Small alterations, like a single-character change, have a reduced effect on the overall distance calculation, leading to more stable and reliable results.
2. **Improved Context Handling**: Words provide context that individual letters do not. Using words as tokens maintains the contextual integrity of the text, allowing for more accurate comparisons. For example, "breaking news" and "urgent news" share a contextual meaning that would be lost in a letter-based analysis.
3. **Efficient Computation**: Word-based distance metrics are computationally more efficient for longer texts. By comparing words instead of every single letter, the algorithm reduces computational complexity and improves processing speed.

In 3-Delta:

For future analyses, it would be beneficial to:

- Incorporate more diverse datasets with varied linguistic features to challenge the translation capabilities of each model.
- Explore lower-level metrics such as word or phrase-level similarities.
- Conduct qualitative reviews to assess the contextual accuracy of translations beyond numerical similarity scores.

Testing predictivity

Information operations may take years to unfold and evolve, making it essential to develop and test predictive models that can adapt to long-term manipulation strategies. Current methodologies often focus on immediate or short-term detection, which might not capture the subtleties of prolonged campaigns. Future improvements should split the data longitudinally to test whether what was predictive earl;y in the campaign continues to be predictive later in the campaign

Near-Duplicate Image Detection

Detecting near-duplicate images is crucial for identifying visual misinformation, such as memes or altered images, which are often used in disinformation campaigns. Implementing context-aware perceptual hashing (p-hashes) or other advanced techniques can enhance the detection of near-duplicate images, capturing slight alterations that evade simpler algorithms (Cozzolino et al., 2015; Zhao et al., 2019). Future work should focus on integrating these techniques with text analysis to provide a comprehensive detection system that can cross-verify content across different media types.

Multi-Modal Analysis

Combining text, image, and audio analysis can offer a more comprehensive approach to detecting sophisticated manipulations. Multi-modal models that analyze different types of content simultaneously can capture the full spectrum of disinformation tactics more effectively (Nakamura et al., 2020; Xue et al., 2021). Future improvements should involve developing robust multi-modal frameworks that leverage the strengths of each modality, enhancing the overall accuracy and reliability of disinformation detection systems. Integrating these models with real-time monitoring systems can further bolster the ability to detect and mitigate emerging threats across various platforms.

References:

- Horák, A., et al. (2021). Technological Approaches to Detecting Online Disinformation and Manipulation.
- Richard, J., et al. (2023). Unmasking information manipulation: A quantitative approach to detecting Copy-pasta, Rewording, and Translation on Social Media.

- Uyheng, J., & Carley, K. M. (2021). Bots and online hate during the COVID-19 pandemic: Case studies in the United States and the Philippines.
- Voiovich, J. (2020). Combatting Information Manipulation and Deception.
- Cozzolino, D., Poggi, G., & Verdoliva, L. (2015). Efficient dense-field copy–move forgery detection. *IEEE Transactions on Information Forensics and Security*, 10(11), 2284-2297.
- Nakamura, T., et al. (2020). Multi-modal misinformation detection: Approaches, challenges and opportunities. *IEEE Access*, 8, 144529-144545.
- Xue, J., et al. (2021). Combining text and image features for fake news detection in multi-modal social media. *Proceedings of the 29th ACM International Conference on Multimedia*, 1924-1932.
- Zhao, Z., et al. (2019). Automatic detection of manipulated images using multi-view deep learning. *Journal of Visual Communication and Image Representation*, 60, 35-44.