Report on Fake News Detection

using N-gram Analysis (Arabic Dataset)

Yaseen Ziqlam (2130397)

Belal Khaled (2136873)

Abdulrahman Abuhani (2132462)

Report on Fake News Detection using N-gram Analysis (Arabic Dataset)

1. Introduction

This report analyses a Jupyter Notebook project that implements fake news detection using N-gram analysis on an Arabic dataset called **AraFacts**. The project was completed by **Belal Khaled, Yaseen Naser, and Abdulrahman Abuhani** as part of an NLP and Text Mining assignment.

The project aims to detect fake news in Arabic text by analysing the linguistic patterns and features through N-gram analysis. N-grams are continuous sequences of n items (words or characters) that can reveal linguistic patterns indicative of deceptive content.

2. Dataset Description

The project uses the AraFacts dataset, which contains Arabic news articles labelled as either real or fake. The dataset is sourced from GitLab and includes the following columns:

- ClaimID: Unique identifier for each claim.
- claim: The text content of the claim.
- description: Detailed description of the claim.
- source: Origin of the claim.
- date: When the claim was published.
- source_label: Original label from the source.
- normalized_label: Standardized label (True, False, Partly-false).
- source_category: Category assigned by the source.
- normalized_category: Standardized category.
- source_url: URL of the source.
- claim_urls: URLs related to the claim.
- evidence_urls: URLs of evidence related to the claim.
- claim_type: Type of claim (encoded as numbers).

The dataset has 6,222 entries with 13 columns, making it a substantial corpus for fake news analysis in Arabic.

3. Methodology

The project follows a structured methodology for fake news detection:

3.1 Data Preprocessing

The preprocessing phase includes several steps:

1. **Loading libraries:** Essential libraries such as pandas, nltk, matplotlib, seaborn, and others are imported.
2. **Loading the dataset:** The AraFacts dataset is loaded from a CSV file.
3. **Cleaning the text:**
    - Removing diacritics (Arabic accent marks).
    - Removing punctuation.
    - Removing numbers.
    - Removing extra whitespaces.
    - Removing Arabic stop words.
4. **Tokenization**: The cleaned text is tokenized into individual words.
5. **Normalization**: Arabic text is normalized by:
    - Removing tatweel (elongation character ـ)-
    - Normalizing alif variants (آ, إ, أ to ا).

3.2 Feature Extraction

The core feature extraction technique used is N-gram analysis:

1. **Unigrams**: Single word tokens.
2. **Bigrams**: Two consecutive words.
3. **Trigrams**: Three consecutive words.

These N-grams are generated for each claim in the dataset and will serve as features for analysis.

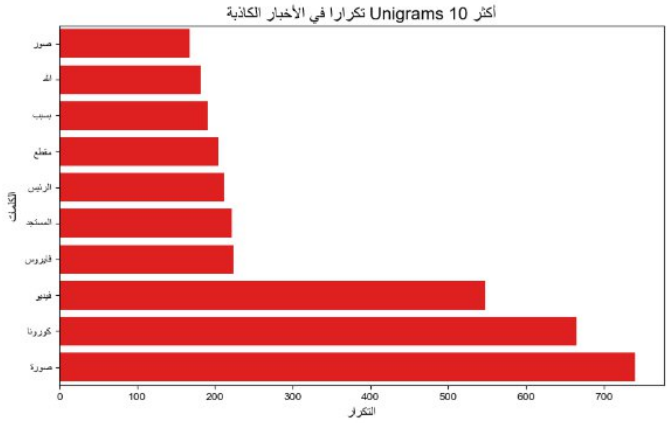
3.3 Data Analysis

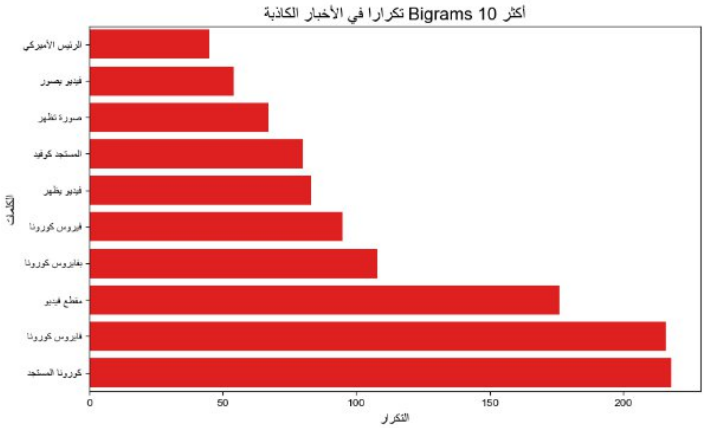
The project performs several analyses on the pre-processed data:

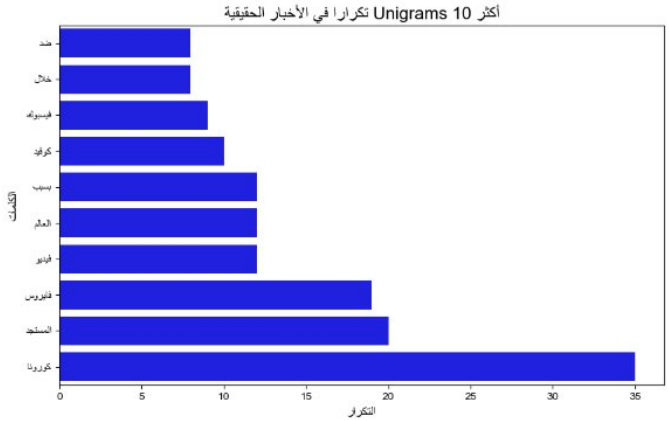
1. **Distribution analysis:** Comparing the frequency distribution of N-grams between fake and real news.

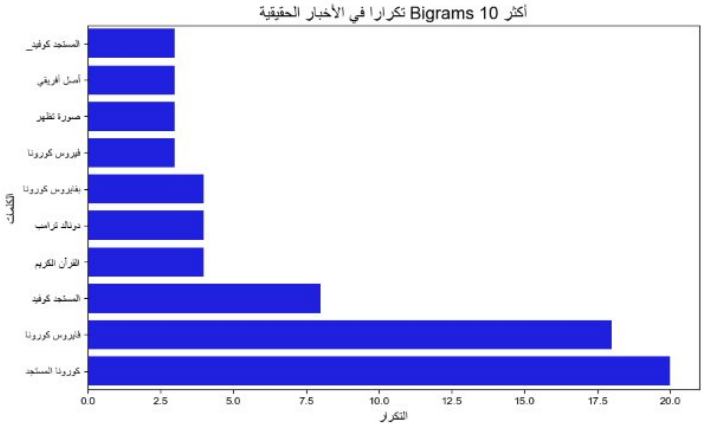
```
Top 10 Unigrams in Fake News: [(('صورة',), 740), (('كورونا',), 665), (('فيديو',), 548), (('فايروس',), 224), (('المستجد',), 222), (('الرئيس',), 212), (('مقطع',), 20
5), (('يسبب',), 191), (('الله',), 182), (('صور',), 168)]
Top 10 Unigrams in Real News: [(('كورونا',), 35), (('المستجد',), 20), (('فايروس',), 19), (('فيديو',), 12), (('العالم',), 12), (('يسبب',), 12), (('كوفيد',), 10), (('فيس
بوك',), 9), (('خلال',), 8), (('ضد',), 8)]
Top 10 Bigrams in Fake News: [(('المستجد ', 'كورونا'), 218), (('كورونا ', 'فايروس'), 216), (('فيديو ', 'مقطع'), 176), (('كورونا ', 'بفايروس'), 108), (('فيروس ', 'كورونا'), 9
5), (('يظهر ', 'فيديو'), 83), (('كوفيد ', 'المستجد'), 80), (('تظهر ', 'صورة'), 67), (('يصور ', 'فيديو'), 54), (('الرئيس ', 'الأميركي'), 45)]
Top 10 Bigrams in Real News: [(('المستجد ', 'كورونا'), 20), (('كورونا ', 'فايروس'), 18), (('كوفيد ', 'المستجد'), 8), (('القرآن ', 'الكريم'), 4), (('دونالد ', 'ترامب'), 4), (('بفايرو
س ', 'كورونا'), 4), (('كورونا ', 'فيروس'), 3), (('صورة ', 'تظهر'), 3), (('أصل ', 'أفريقي'), 3), (('كوفيد ', 'المستجد_'), 3)]
Top 10 Trigrams in Fake News: [(('المستجد ', 'كورونا ', 'فايروس'), 132), (('كوفيد ', 'المستجد ', 'كورونا'), 79), (('المستجد ', 'كورونا ', 'بفايروس'), 65), (('مقطع ', 'فيديو ', 'يظهر'), 37), (('العالمية ', 'الصحة ', 'منظمة'), 34), (('كوفيد ', 'المستجد ', 'كورونا_'), 27), (('أردوغان ', 'طيب ', 'رجب'), 19), (('كورونا ', 'فايروس ', 'انتشار'), 18), (('رجب ', 'التركي ', 'ر
ب'), 17), (('كورونا ', 'بفايروس ', 'إصابته'), 17)]
Top 10 Trigrams in Real News: [(('المستجد ', 'كورونا ', 'فايروس'), 15), (('كوفيد ', 'المستجد ', 'كورونا'), 8), (('المستجد ', 'كورونا ', 'بفايروس'), 3), (('المستجد ', 'كوفيد ', 'كورونا
_'), 3), (('كورونا ', 'فايروس ', 'انتشار'), 3), (('رئيس ', 'تركيا ', 'فايزر'), 2), (('الإيطالي ', 'يوفنتوس ', 'فريق'), 2), (('الرئيس ', 'التركي ', 'رجب'), 2), (('رجب ', 'طيب ', 'التركي
ب'), 2), (('أردوغان ', 'طيب ', 'رجب'), 2)]
```

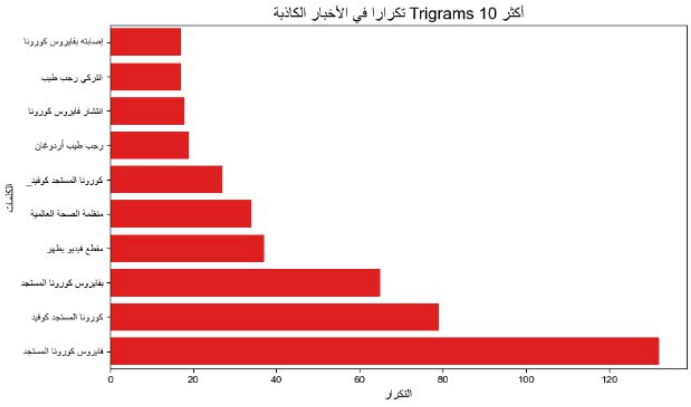
2. **Visual representations:** Creating bar plots and word clouds to visualize the most common N-grams in fake versus real news.

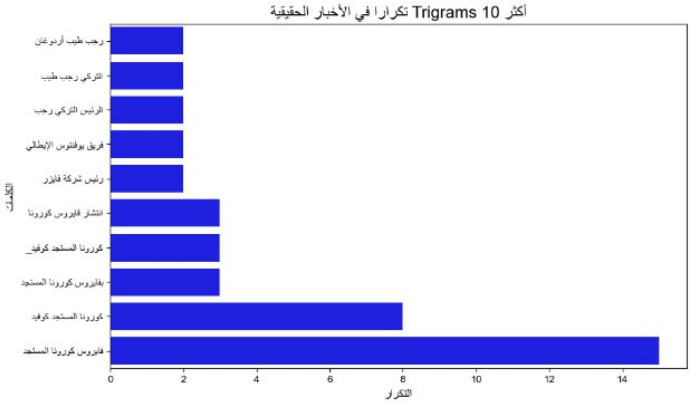












4. Key Findings

The analysis revealed several interesting patterns:

4.1 Unigram Analysis

The top unigrams in fake news include:

- صورة : 740 occurrences.
- كورونا : 665 occurrences.
- فيديو : 548 occurrences.
- فايروس : 224 occurrences.
- المستجد : 222 occurrences.

The top unigrams in real news include:

- كورونا : 35 occurrences.
- المستجد : 20 occurrences.
- فايروس : 19 occurrences.
- فيديو : 12 occurrences.
- العالم : 12 occurrences.

4.2 Bigram Analysis

The top bigrams in fake news include:

- كورونا المستجد : 218 occurrences.
- فايروس كورونا : 216 occurrences.
- مقطع فيديو : 176 occurrences.

The top bigrams in real news include:

- كورونا المستجد : 20 occurrences.
- فايروس كورونا : 18 occurrences.
- المستجد كوفيد : 8 occurrences.

4.3 Trigram Analysis

The top trigrams in fake news include:

- فايروس كورونا المستجد : 132 occurrences.
- كورونا المستجد كوفيد : 79 occurrences.
- بفايروس كورونا المستجد : 65 occurrences.

The top trigrams in real news include:

- فايروس كورونا المستجد : 15 occurrences.
- كورونا المستجد كوفيد : 8 occurrences.
- بفايروس كورونا المستجد : 3 occurrences.

4.4 Pattern Recognition

Several patterns emerge from the N-gram analysis:

1. Both fake and real news frequently mention COVID-19 related terms.
2. Fake news tends to use terms related to visual content (image, video) more frequently.
3. The same N-grams appear in both categories but with significantly different frequencies.
4. Fake news exhibits a higher frequency of certain N-grams compared to real news, suggesting more repetitive language patterns.

5. Limitations

The project has several limitations:

1. Reliance on frequency-based analysis without more sophisticated machine learning models.
2. Limited to N-gram features without considering semantic relationships.
3. The dataset might have an imbalance between fake and real news examples.
4. The analysis is primarily descriptive without predictive modelling.

6. Conclusion

This project demonstrates an effective approach to fake news detection in Arabic using N-gram analysis. The analysis reveals distinct linguistic patterns between fake and real news, particularly in the frequency of specific word combinations. While the current implementation focuses on exploratory analysis rather than predictive modelling, it provides valuable insights into the characteristics of deceptive content in Arabic news.

The findings suggest that N-gram analysis can be a useful tool for identifying potentially fake news in Arabic. The prevalence of terms related to visual content (images, videos) in fake news highlights a common strategy used to enhance the credibility of false claims. Additionally, the higher frequency of certain N-grams in fake news indicates more repetitive language patterns compared to genuine news articles. Overall, this project provides a solid foundation for more advanced approaches to fake news detection in Arabic, with potential applications in media literacy, fact-checking, and combating misinformation.

7. References

- https://gitlab.com/bigirqu/AraFacts/-/blob/master/Dataset/AraFacts/AraFacts.csv?ref_type=heads
- https://www.nltk.org/
- https://www.nltk.org/data.html
- https://www.nltk.org/api/nltk.tokenize.punkt.html
- https://stackoverflow.com/questions/77131746/how-to-download-punkt-tokenizer-in-nltk
- https://www.analyticsvidhya.com/blog/2021/09/what-are-n-grams-and-how-to-implement-them-in-python/
- https://www.geeksforgeeks.org/generating-word-cloud-python/