

**THE HASHEMITE UNIVERSITY**

Faculty of Prince Al-Hussein Bin Abdallah II For Information Technology

Information Technology Department

**Natural Language Processing**

**Assignment no.3**

**Arabic fake news detection**

**By**

JENEEN NABEEL MOHAMMED ISLEEM (2230934)

FIRAS BASSIL ZEIDAN AZIZ (2232786)

MOHAMMAD JEHAD KHALED ABUFARHAH (2232469)

KARAM MAHMOUD ABD ELHALEEM AL BALASMEH (2231827)

**Supervised by**

Dr. Esra'a Ahmad Helael alshdaifat

**Introduction**

This project aims to classify Arabic news into 2 label’s : fake , real

Through large dataset contains 22 columns which ['ClaimID', 'claim', 'description', 'source', 'date', 'source\_label','normalized\_label', 'source\_category', 'normalized\_category','source\_url', 'claim\_urls', 'evidence\_urls', 'claim\_type', 'label','combined\_text', 'cleaned\_text', 'tokens', 'unigrams', 'bigrams','trigrams', 'text\_length', 'word\_count']

With 6222 data records, and they are object datatype (string) but

We have one numeric column (claim\_type).

We are going to N-gram method :

Unigrams (1-word phrases)

Bigrams (2-word phrases)

Trigrams (3-word phrases)

by the helping of NLTK python Library which is a specified

library to handle the Natural Languages and process them and

other Arabic NLP tools we will talk about them later.

**Methodology**

First step was to loading the dataset which named arafacts.csv

Into pandas dataFrame then we start lookup and discover the date

by some pandas function to know the unique values and null

records and another data discovering techniques.

Then we start the preprocessing stage by doing these steps:

1. stripping Arabic diacritics (tashkeel)
2. Normalize characters:
3. Remove URLs
4. Remove punctuation and special chars
5. Remove English and numbers
6. Remove extra spaces

By the helping of the araby and regx libraries in python

Then we jumped to the tokenization process by taking every

token in each courps and removing the stopping words

using NLTK library .

Next, we generated n-grams with frequency filtering—keeping

only those unigrams, bigrams, or trigrams that occurred at least

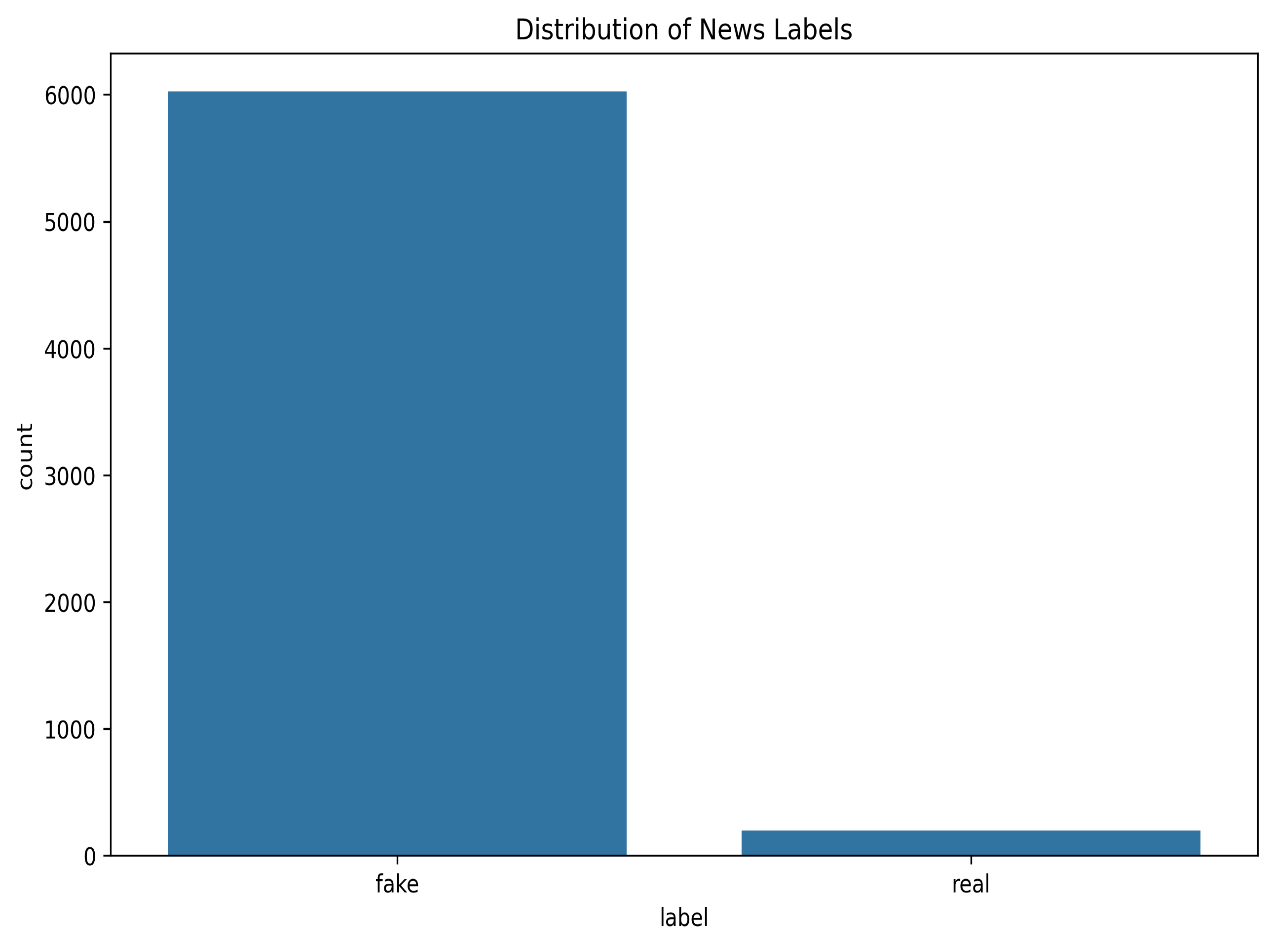
twice.

This balanced removing trivial noise and retaining meaningful

collocations. Finally, we computed basic text‐length and word‐

count statistics before visualizing the most frequent n-grams per

class.

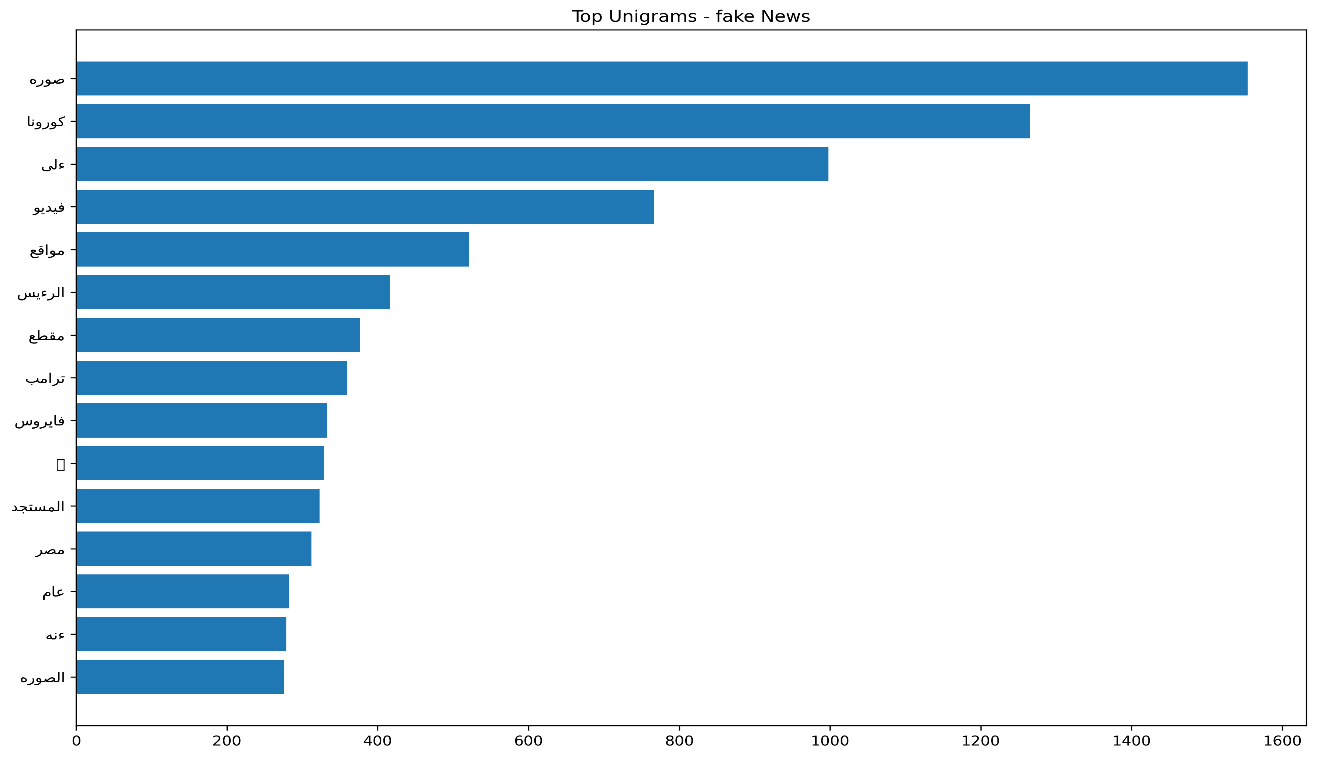
 **Visual representation**

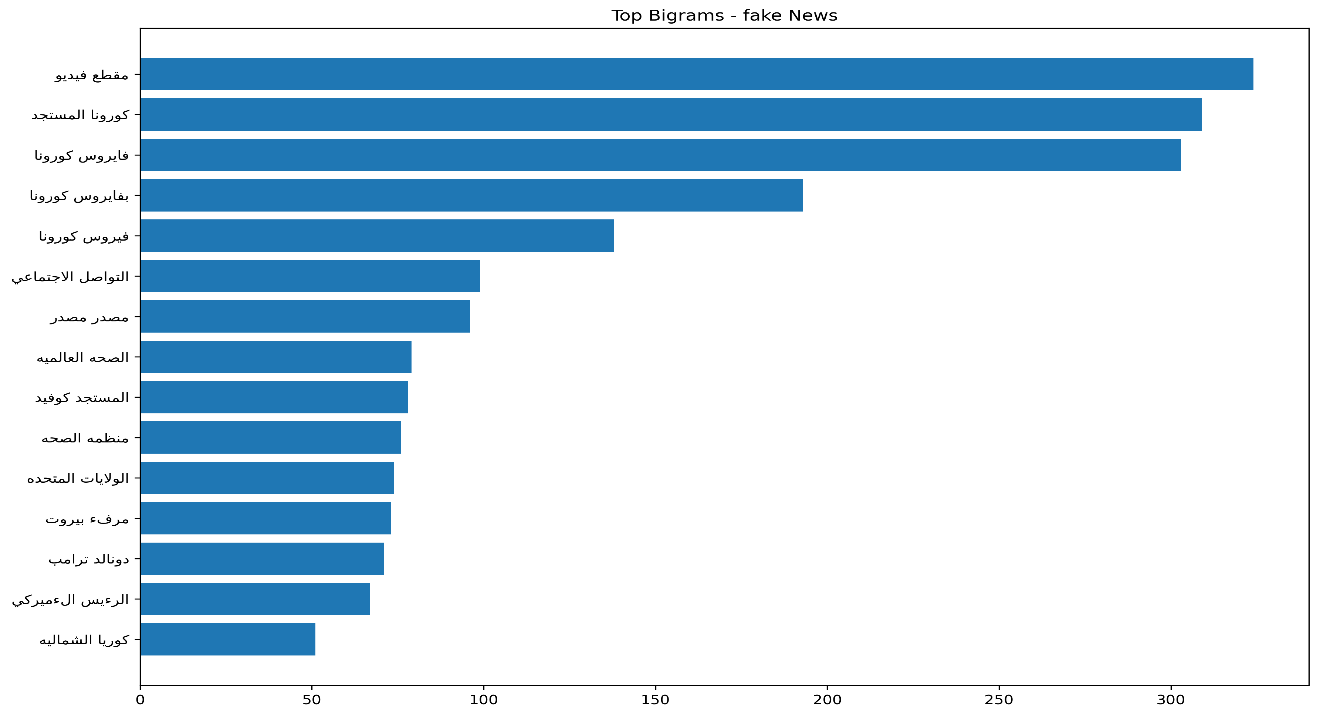
We can notice that the data imbalanced but, in the TF,-IDF we

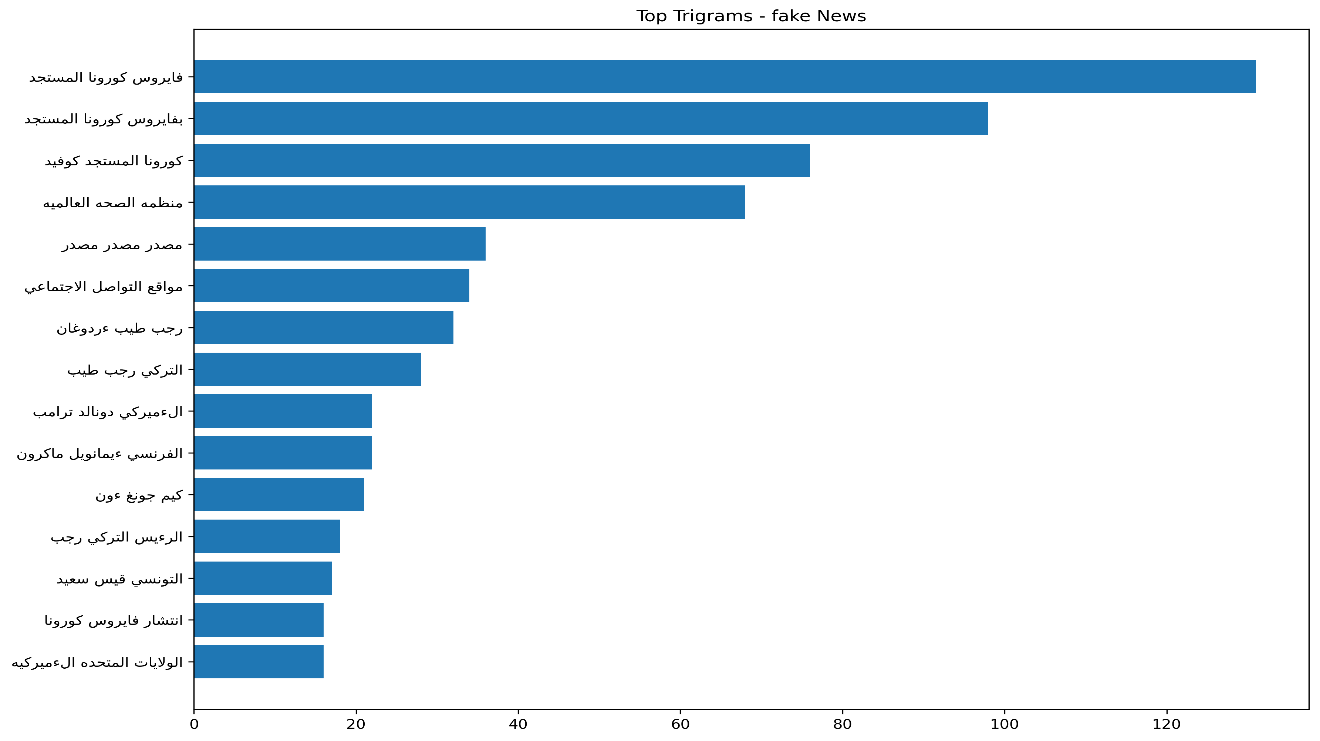
don’t Have to balance data and we have consider that

[false, partly false, unverifiable, sarcasm] all of them as fake news

And just the True as real news’s.

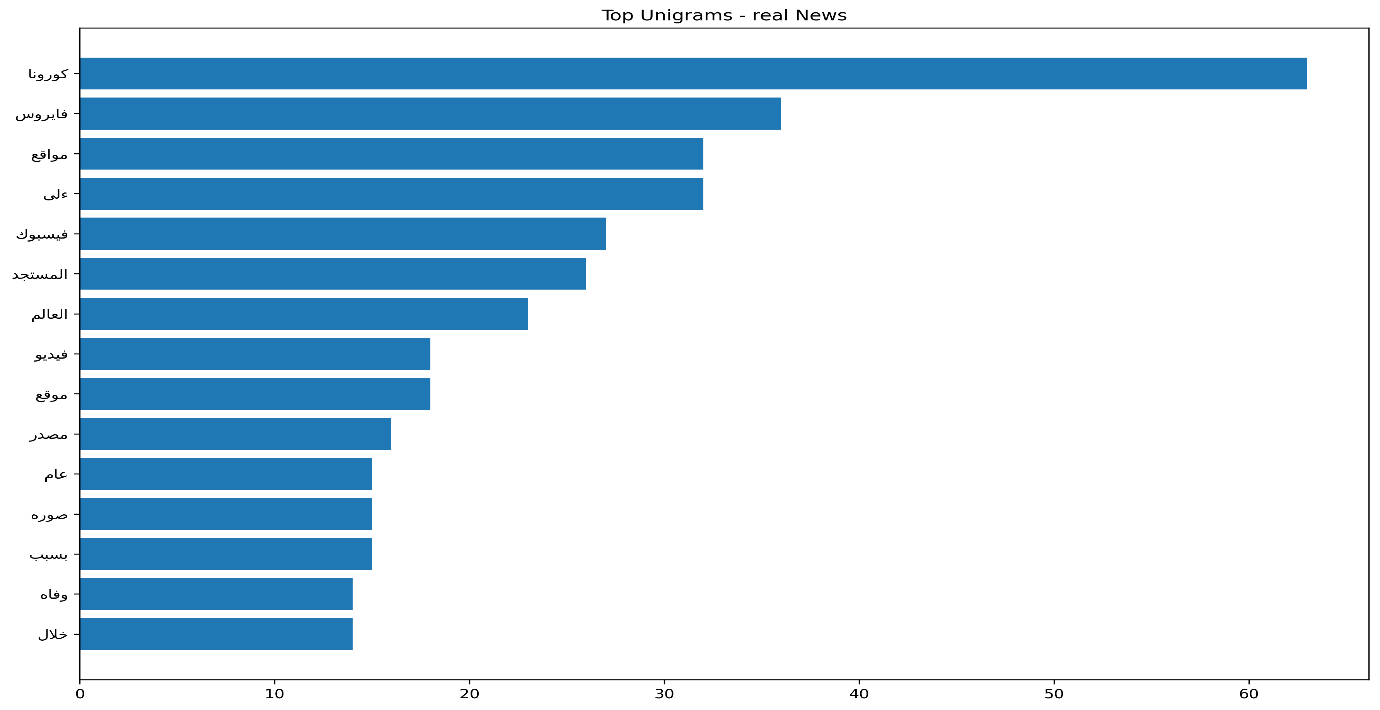


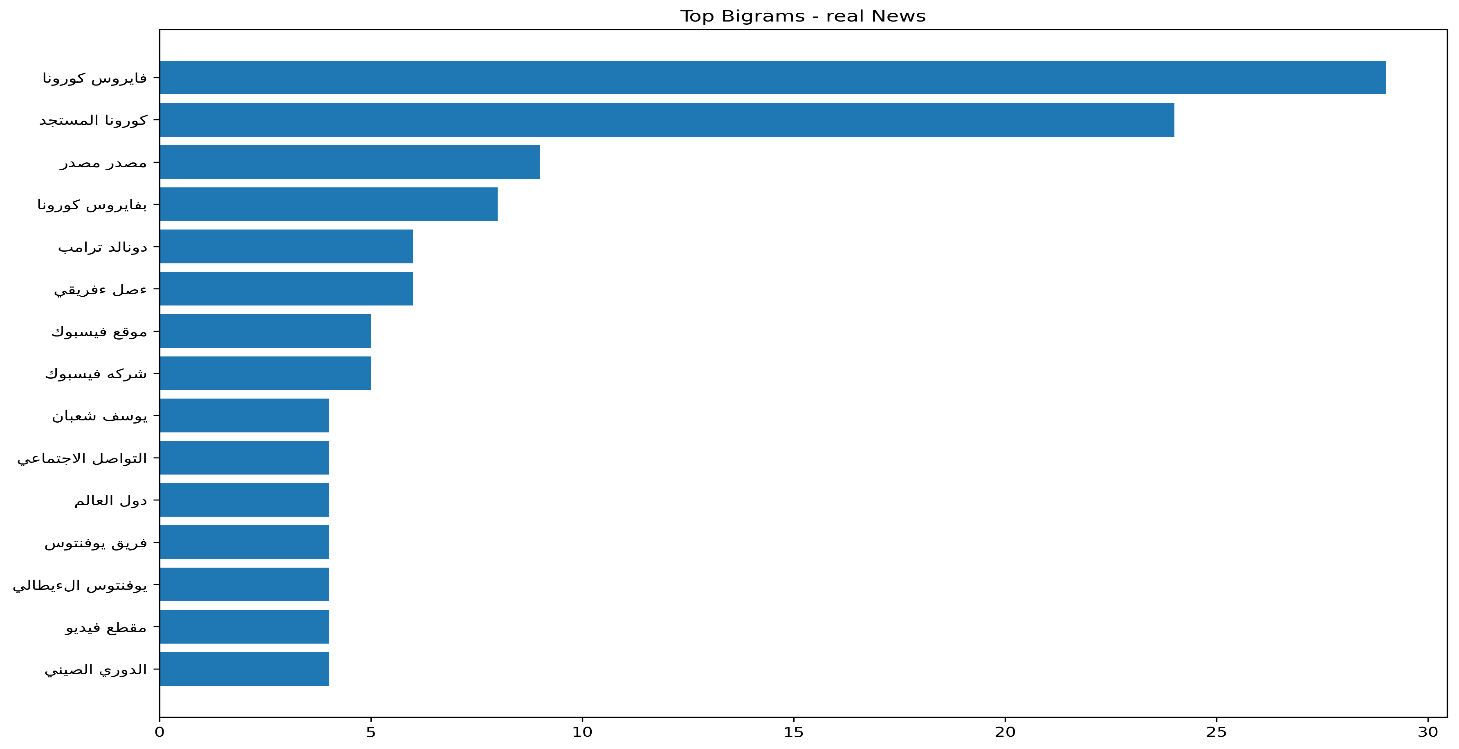


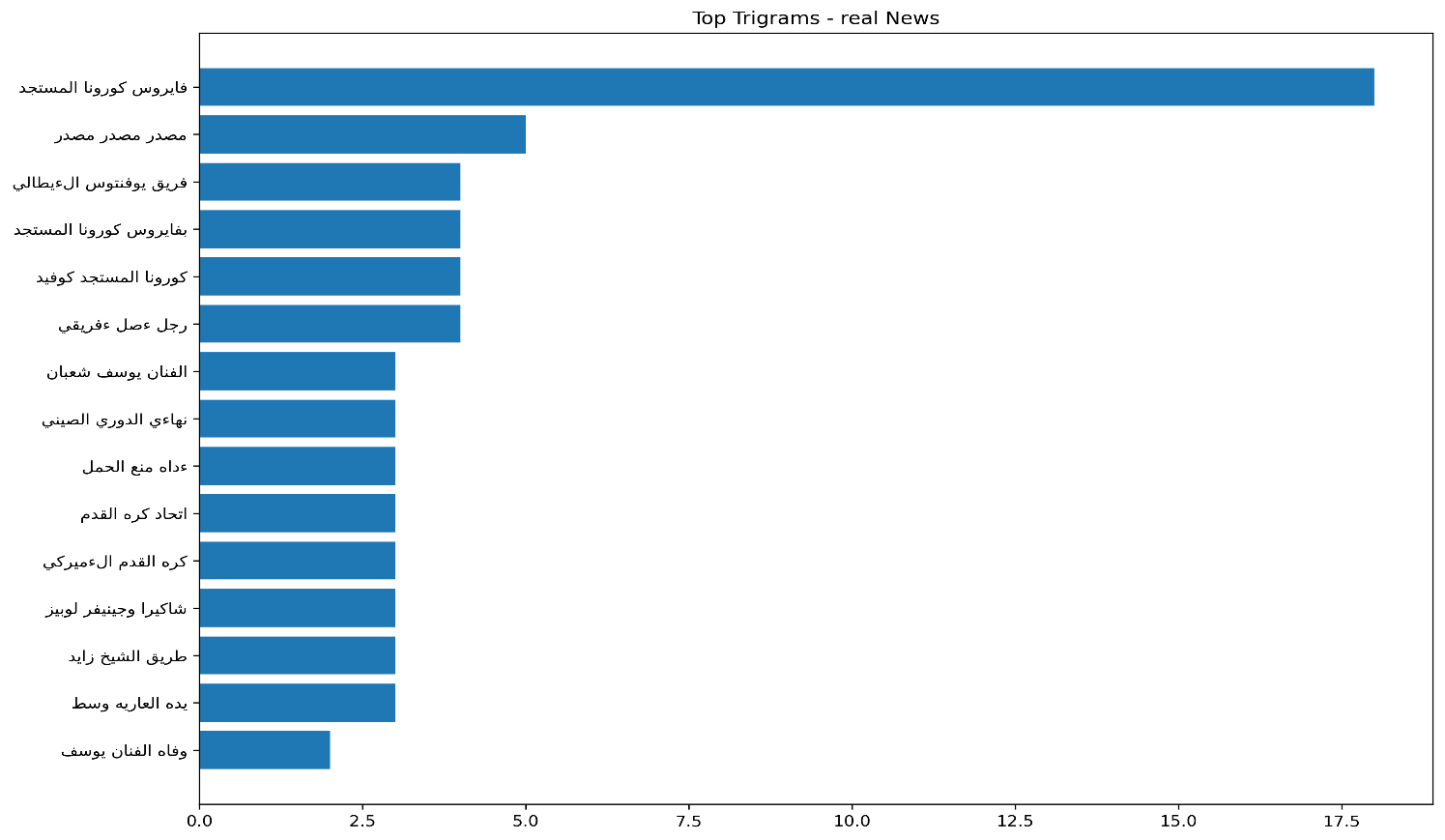


And this is the result of the n-gram with the 3 types on the fake news’s label.

And here is the cloud plot for the fake news’s.







****

**Patterns and Insights**

1. Label Distribution & Text Length

Imbalanced Dataset: Fake news dominates the dataset which may reflect real-world misinformation trends but could introduce bias in analysis.

Longer Fake News: Fake articles have significantly longer text lengths.

2. Unigram Analysis:

Fake news:   
dominated by trends and multimedia sources like:

صورة، مقطع، كورونا، مواقع، ترامب.

Real news:

Contains more details like the source or the year :

مصدر, وفاه, خلال, عام

3.Bigram Analysis:

Fake news:

Repetition of the source word: مصدر مصدر

President’s mention like: الرئيس الأمريكي, الرئيس الكوري

Unstructured words like:المستجد كوفيد

Real news:

Structured words like: الدوري الصيني, دول العالم

Trigram Analysis:

Fake news:

More President’s mentions and countries names like:

رجب طيب اردوغان, كيم جونغ اون, التونسي قيس سعيد,

Real news:

Sport news like:

اتحاد كرة القدم, كرم القدم الأمريكي, فريق يوفينتوس الإيطالي

Actors and names like :

وفاة الفنان يوسف, شايرا وجينيفر لوبيز

**Challenges and Limitations**

1. **Class Imbalance**: The dataset was heavily skewed toward fake-news labels, and we don’t know if we must use resampling techniques or not in the case of TF-IDf or even what is the best resampling technique over or under sampling in the textual data so it was a big challenge .
2. **Noisy Metadata**: Embedded URLs and source fields sometimes bled into the text, requiring extra regex cleaning.
3. **Arabic Orthographic Variants**: Inconsistent use of alef and hamza forms initially split identical words into separate tokens, which require more data normalization and preprocessing techniques.
4. **Arabic NLP tools: few sources and pre-modules for handling and working with the Arabic languages and so we had to search more.**

**Conclusion**

As an Arabian students we are looking for update and construct many project in Natural Language Processing in the Arabic language and this project was very interesting one to implement and test one the most famous techniques (n-grame) in NLP and its has shown the strength of it and give a high accuracy 97% which is very good result.