Sharif University of Technology

Masoud Tahmasbi Fard

Student ID: 402200275

EE577: Massive Data Analytics

Assignment #1- Introduction to PySpark

December 19, 2023

Table of Contents

1 Data Preparation

To begin the analysis, the PySpark libraries were installed on Google Colab to allow for distributed processing capabilities. The Persian news dataset was then downloaded from Google Drive for use in this analysis.

A Spark session was initialized to enable access to PySpark functionality. The news data files were stored in JSON format across several files. Using the spark session, these JSON files were read into a Resilient Distributed Dataset (RDD) called *news-rdd* to facilitate parallel data processing.

An example of one element of the *news-rdd* is shown below:

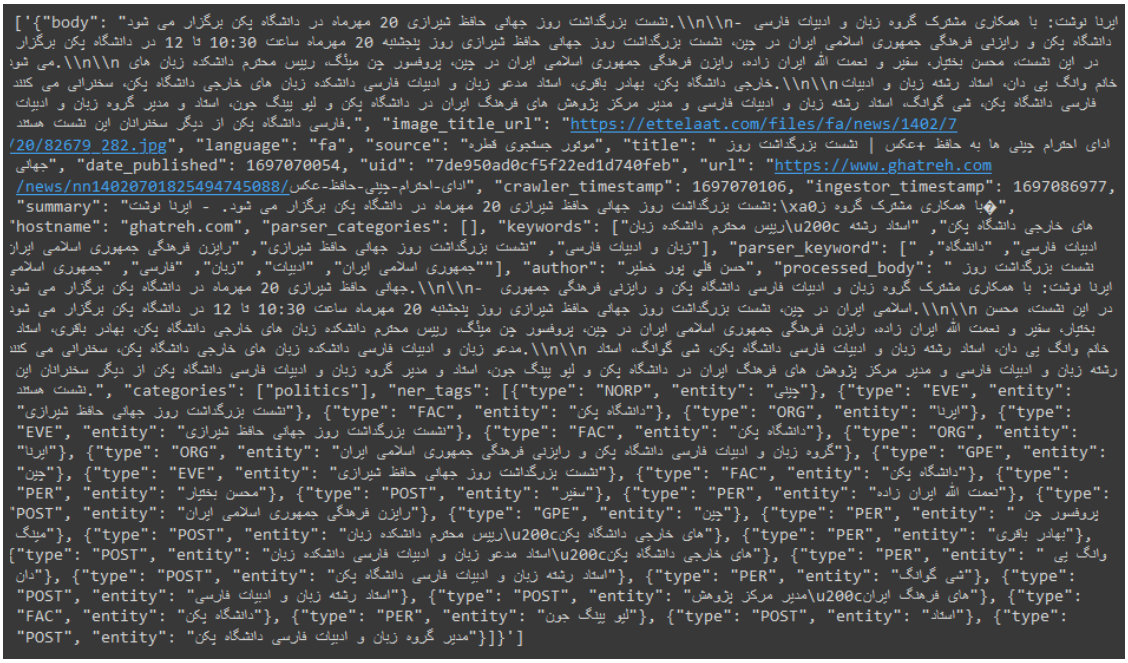
['{"body": "ایرنا نوشت: با همکاری مشترک گروه زبان و ادبیات فارسی -\\n\\n.نشست بزرگداشت روز جهانی حافظ شیرازی 20 مهرماه در دانشگاه پکن برگزار می شود\\n\\n دانشگاه پکن و رایزنی فرهنگی جمهوری اسلامی ایران در چین، نشست بزرگداشت روز جهانی حافظ شیرازی روز پنجشنبه 20 مهرماه ساعت 10:30 تا 12 در دانشگاه پکن برگزار در این نشست، محسن بختیار، سفیر و نعمت الله ایران زاده، رایزن فرهنگی جمهوری اسلامی ایران در چین، پروفسور چن مینگ، رییس محترم دانشکده زبان های \n\\n\\.می شود خانم وانگ پی دان، استاد رشته زبان و ادبیات \n\\n\\.خارجی دانشگاه پکن، بهادر باقری، استاد مدعو زبان و ادبیات فارسی دانشکده زبان های خارجی دانشگاه پکن، سخنرانی می کنند فارسی دانشگاه پکن، شی گوانگ، استاد رشته زبان و ادبیات فارسی و مدیر مرکز پژوهش های فرهنگ ایران در دانشگاه پکن و لیو پینگ جون، استاد و مدیر گروه زبان و ادبیات فارسی دانشگاه پکن از دیگر سخنرانان این نشست هستند."., "image_title_url": "https://ettelaat.com/files/fa/news/1402/7/20/82679_282.jpg", "language": "fa", "source": "موتور جستجوی قطره", "title": " نشست بزرگداشت روز ادای احترام چینی ها به حافظ ‌عکس | جهانی", "date_published": 1697070054, "uid": "7de950ad0cf5f22ed1d740feb", "url": "https://www.ghatreh.com/news/nn14020701825494745088/ادای-احترام-چینی-حافظ-عکس", "crawler_timestamp": 1697070106, "ingestor_timestamp": 1697086977, "summary": "ایرنا نوشت: با همکاری مشترک گروه ز\xa0:نشست بزرگداشت روز جهانی حافظ شیرازی 20 مهرماه در دانشگاه پکن برگزار می شود. - ایرنا نوشت",", "hostname": "ghatreh.com", "parser_categories": [], "keywords": ["رییس محترم دانشکده زبان\u200c ,"استاد رشته های خارجی دانشگاه پکن", "زبان و ادبیات فارسی", "نشست بزرگداشت روز جهانی حافظ شیرازی", "رایزن فرهنگی جمهوری اسلامی ایراز ادبیات فارسی", "دانشگاه", "جمهوری اسلامی ایران", "ادبیات", "زبان", "فارسی", "جمهوری اسلامی""], "author": "حسن قلی پور خطیر", "processed_body": " نشست بزرگداشت روز ایرنا نوشت: با همکاری مشترک گروه زبان و ادبیات فارسی دانشگاه پکن و رایزنی فرهنگی جمهوری -\\n\\.جهانی حافظ شیرازی 20 مهرماه در دانشگاه پکن برگزار می شود در این نشست، محسن \n\\n\\.اسلامی ایران در چین، نشست بزرگداشت روز جهانی حافظ شیرازی روز پنجشنبه 20 مهرماه ساعت 10:30 تا 12 در دانشگاه پکن برگزار می شود بختیار، سفیر و نعمت الله ایران زاده، رایزن فرهنگی جمهوری اسلامی ایران در چین، پروفسور چن مینگ، رییس محترم دانشکده زبان های خارجی دانشگاه پکن، بهادر باقری، استاد خانم وانگ پی دان، استاد رشته زبان و ادبیات فارسی دانشگاه پکن، شی گوانگ، استاد \n\\n\\.مدعو زبان و ادبیات فارسی دانشکده زبان های خارجی دانشگاه پکن، سخنرانی می کنند رشته زبان و ادبیات فارسی و مدیر مرکز پژوهش های فرهنگ ایران در دانشگاه پکن و لیو پینگ جون، استاد و مدیر گروه زبان و ادبیات فارسی دانشگاه پکن از دیگر سخنرانان این نشست هستند."., "categories": ["politics"], "ner_tags": [{"type": "NORP", "entity": "چینی"}, {"type": "EVE", "entity": "نشست بزرگداشت روز جهانی حافظ شیرازی"}, {"type": "FAC", "entity": "دانشگاه پکن"}, {"type": "ORG", "entity": "ایرنا"}, {"type": "EVE", "entity": "نشست بزرگداشت روز جهانی حافظ شیرازی"}, {"type": "FAC", "entity": "دانشگاه پکن"}, {"type": "ORG", "entity": "ایرنا"}, {"type": "ORG", "entity": "گروه زبان و ادبیات فارسی دانشگاه پکن و رایزنی فرهنگی جمهوری اسلامی ایران"}, {"type": "GPE", "entity": "چین"}, {"type": "EVE", "entity": "نشست بزرگداشت روز جهانی حافظ شیرازی"}, {"type": "FAC", "entity": "دانشگاه پکن"}, {"type": "PER", "entity": "محسن بختیار"}, {"type": "POST", "entity": "سفیر"}, {"type": "PER", "entity": "نعمت الله ایران زاده"}, {"type": "POST", "entity": "رایزن فرهنگی جمهوری اسلامی ایران"}, {"type": "GPE", "entity": "چین"}, {"type": "PER", "entity": " پروفسور چن مینگ"}, {"type": "POST", "entity": "رییس محترم دانشکده زبان\u200c"}, {"type": "PER", "entity": "بهادر باقری"}, {"type": "POST", "entity": "های خارجی دانشگاه پکن\u200c"}, {"type": "PER", "entity": " وانگ پی دان"}, {"type": "POST", "entity": "استاد رشته زبان و ادبیات فارسی دانشگاه پکن"}, {"type": "PER", "entity": "شی گوانگ"}, {"type": "POST", "entity": "استاد رشته زبان و ادبیات فارسی"}, {"type": "POST", "entity": "مدیر مرکز پژوهش های فرهنگ ایران\u200c"}, {"type": "FAC", "entity": "دانشگاه پکن"}, {"type": "PER", "entity": "لیو پینگ جون"}, {"type": "POST", "entity": "استاد"}, {"type": "POST", "entity": "مدیر گروه زبان و ادبیات فارسی دانشگاه پکن"}]}']

Figure 1: An element of the *news-rdd*

2 Preprocessing

To prepare the raw news data for analysis, multiple preprocessing steps were applied using PySpark transformations and custom functions.

First, the *news-rdd* which contained the raw JSON data was parsed into a new RDD called *parsed-rdd*.

Two preprocessing functions were then defined and applied to clean and normalize the data. The *persian-only* function removed any non-Persian characters and unicode strings, while the *txt-cleansing* function standardized punctuation and removed special characters.

By applying these functions to the *parsed-rdd*, a *clean-news-rdd* RDD was produced with cleaned and normalized Persian news text. An example element is shown below:

```
print(clean_news_rdd.first())
```

```
{'body': ['گرفت', 'مردمان', 'شما', 'تیک', 'فوتبال', 'درباره', 'این', 'نگه', 'سبز', 'خود', 'گزارش', 'دقیقه', 'پای', 'مستطیل', 'نساجی',
```

Figure 2: An element of the *clean-news-rdd*

Next, to improve search and analysis, a *stop-remove-lemmatizer* function leveraged the hazm library to remove stop words and lemmatize the terms. A stopwords text file gathered for Persian was attached and utilized to remove common words. Applying this function resulted in the *processed-news-rdd* which contained cleaned, normalized, and simplified news content. An example row after these preprocessing steps is:

```
print(processed_news_rdd.first())
```

```
{'body': ['مرد', 'نمیآید', 'داده', 'برتر', 'تارتار', 'تیک', 'تلویزیون', 'سگو', 'برایمان', 'فوتبال', 'مهدی', 'میخندیم', 'وصفناشدنی', 'ادامه',
```

Figure 3: An element of the *processed-news-rdd*

The full Persian text was now ready for next-step analysis and modeling with unnecessary characters, stop words and unlemmatized terms removed through PySpark and hazm data manipulation.

3 Exploration

3.1 What are the titles and urls of 5 longest news?

To identify the longest news stories, a custom *word-count* function was implemented to return a tuple containing the news length as number of words, along with the title and url. This

leveraged the built-in Python len function to count words in each news excerpt.

The *word-count* function was applied to the *clean-news-rdd* using a *map* transformation, creating keyword-length tuples for each row. The results were reduced by the keyword to consolidate news with same titles.

Finally, the *reduceByKey* method was used to sort the RDD by length and then the the *takeOrdered* method was utilized to retrieve the top 5 longest news. (For simplicity in further analyses, the *top* method was utilized instead of *takeOrdered* to find top results based on the given ordering).

This returned the following 5 longest news based on word count:

Figure 4: titles and urls of 5 longest news

3.2 What are the 20 most frequent words?

A custom *words-KeyValue* function was created to tokenize each news article and output tuples of *(word, 1)* to count individual word frequencies.

This function was applied using a *flatMap* transformation to split all text into discrete words and counts. The results were reduced by the word to sum all 1s and consolidate unique terms. Finally, the *sortByKey* method sorted the aggregated word counts RDD and returned the 20 most common tokens overall, visualized in the following bar chart:

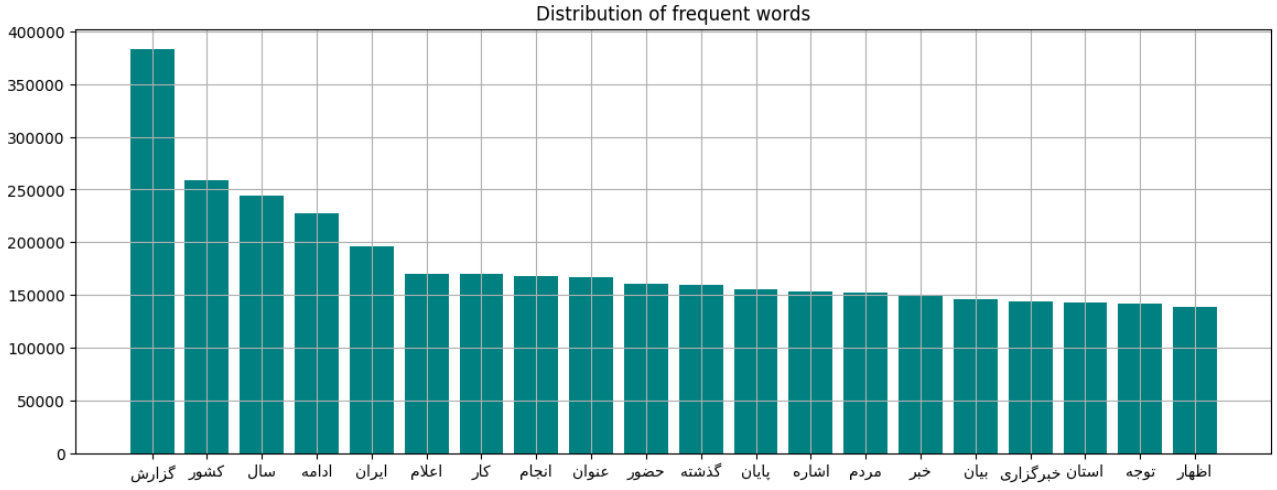

Figure 5: 20 most frequent words

To properly display the Persian text, the *bidi.algorithm* and *arabic-reshaper* libraries were employed to render the words.

Frequent terms match expectation for a Persian news dataset, indicating meaningful text analysis.

3.3 Find the *word clouds* of the news

To visualize prominent topics, *word clouds* were generated from the keyword tags associated with each news.

A *keywords-KeyValue* function was implemented that checked if a given keyword existed in an news' keywords or not.

This function was applied using map and reduce transformations on the full dataset to count the occurrences of each keyword across all news stories.

The results were then scored and sorted by importance according to:

$$\text{score} = 2^{\log_{10}(\text{count})}$$

Finally, the scores were normalized by the maximum value to derive weights. Generating word clouds using these weights produced the following visualizations, sized by significance:

```
[(1.0, 'رژیم صهیونیستی'),
 (0.9153835361235024, 'اسرائیل'),
 (0.8212246921831099, 'جمهوری اسلامی ایران'),
 (0.7451676515885112, 'غزه'),
 (0.7394615313215953, 'لیگ قهرمانان آسیا'),
 (0.7363438588848646, 'تیم ملی'),
 (0.7239635116902534, 'آموزش و پرورش'),
 (0.6955555918790012, 'جمهوری اسلامی'),
 (0.6675887614508967, 'فلسطین'),
 (0.6580118424770381, 'ایالات متحده'),
 (0.655299914935207, 'حماس'),
 (0.6533581120125306, 'اسلام'),
 (0.6400008319358191, 'طوفان الاقصی'),
 (0.6399173808248128, 'استقلال'),
 (0.6394579473932916, 'رئیس جمهور'),
 (0.6250389114260585, 'سیستان و بلوچستان'),
 (0.6229149527751284, 'بانک مرکزی'),
 (0.6116172477928972, 'سازمان ملل'),
 (0.5958543790760519, 'لایحه برنامه هفتم توسعه'),
 (0.5955093942849485, 'مجلس شورای اسلامی')]
```

Figure 6: *word clouds*

Figure 7: *word clouds*

3.4 Plot a timeline which shows total count of news per each day

To examine trends over time, a timeline was generated showing the total number of articles published each day.

The *date-published* field containing the date for each news story was extracted as the key. The

value was set to 1 to represent one occurrence per article.

These *(date, 1)* tuples were mapped across the dataset and then reduced by the date key to sum all instances published on each day.

The following figure showing the resulting time series plot is provided to visualize the trends over time in news volume.

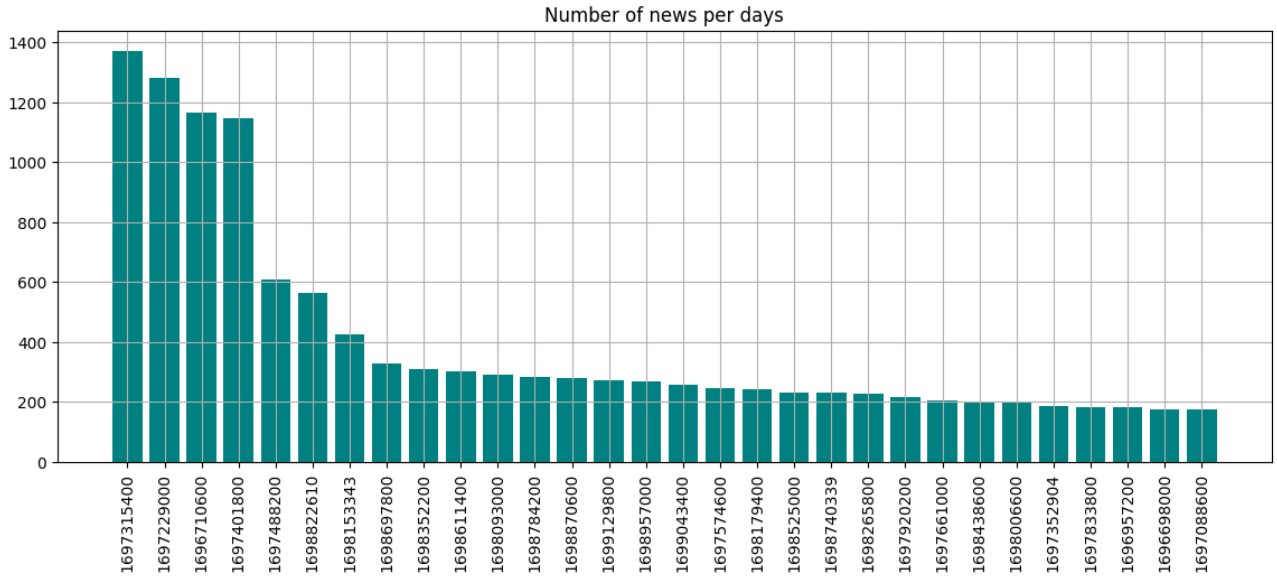

Figure 8: *word clouds*

3.5 Plot a timeline which shows total count of news per each day that have at least one of the word clouds

To narrow down on highly relevant and trending news cycles, the time series analysis was extended to only count articles containing at least one keyword from the significant terms identified in the word clouds.

The same date extraction and counting process was applied, but now combined with a check if any word cloud words appeared in the keywords for that article. So the date tally only incremented if overlapping semantic keywords existed.

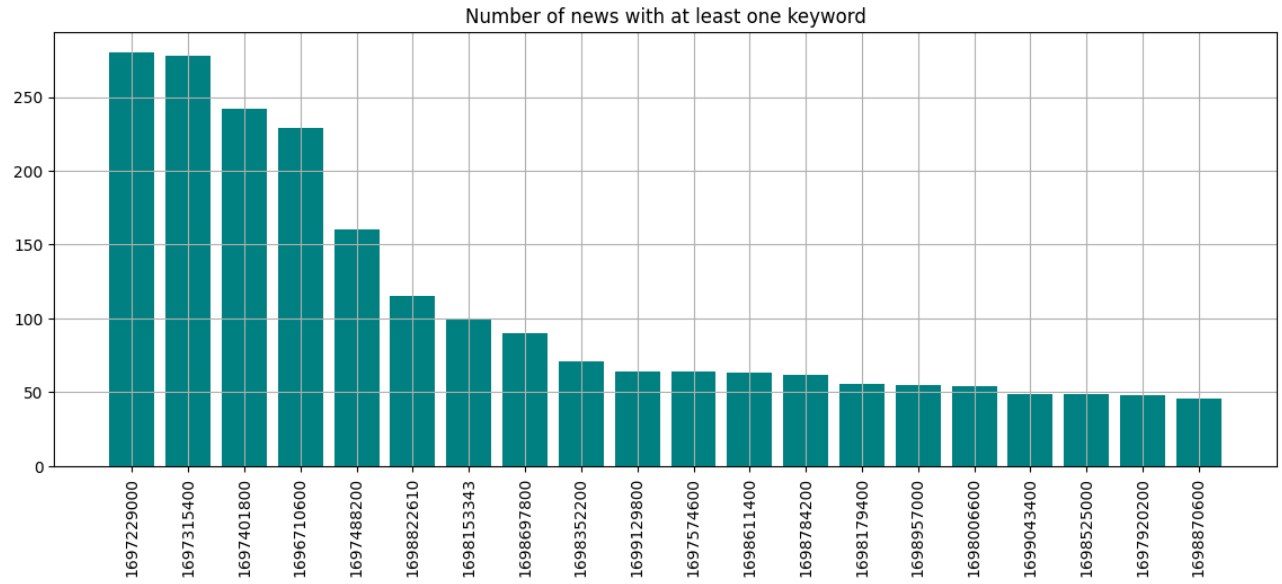

Figure 9: Number of news per each day with at least one keyword

As evident by comparing the unfiltered and filtered date volumes, the peaks in overall news volume correlate strongly to increases in semantically relevant articles containing trending terms. The strong correlation between the unfiltered and filtered date volumes implies the word clouds effectively identified hot topics and keywords that saw greater prevalence during major Persian language news cycles. This determines that the high news volume on certain dates was largely driven by significant coverage of trending subjects captured in the word cloud visualization.

4 A-Priori Algorithm

The next analysis sought to uncover sets of words that commonly co-occur together across the news stories using association rule learning. Specifically, the goal was finding pertinent 3-word combinations that appear together frequently.

To accomplish this, the A-priori algorithm was implemented in PySpark. The A-priori technique works by first identifying individual items that appear frequently, then combining them iteratively into itemsets, pruning combinations that fall below minimum thresholds.

First, the code calculated individual word frequencies to determine popular terms. The 20 most common words were selected as candidates to build word pairs from.

An *A-prior-2tuples* function was applied with map and reduce to count all possible 2-word combinations among these candidates, scoring based on co-occurrence. The top 50 pairs were extracted, including sets like:

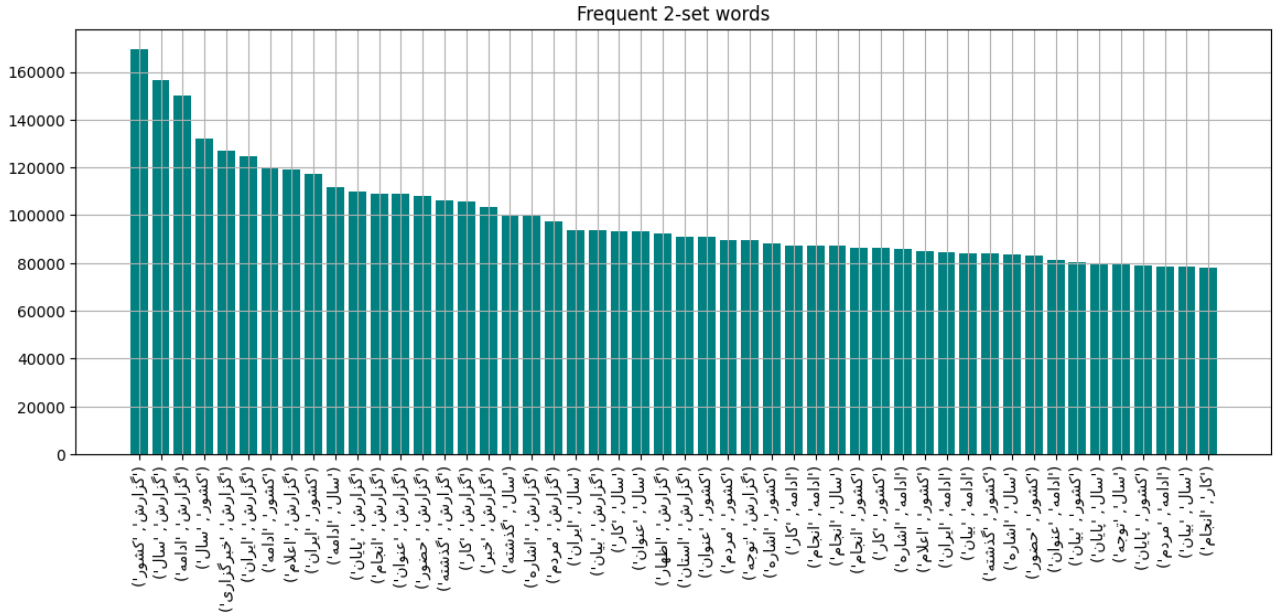

Figure 10: Top 50 pairs

Then to generate 3-element tuples, the frequent word pairs were combined with the top single words in a constrained manner. The *A-prior-3tuples* function ensured new triplets met the Apriori monotonic condition - all 2-word subsets must also be frequent. Applying ranking and filtering resulted in pertinent 3-word associations like:

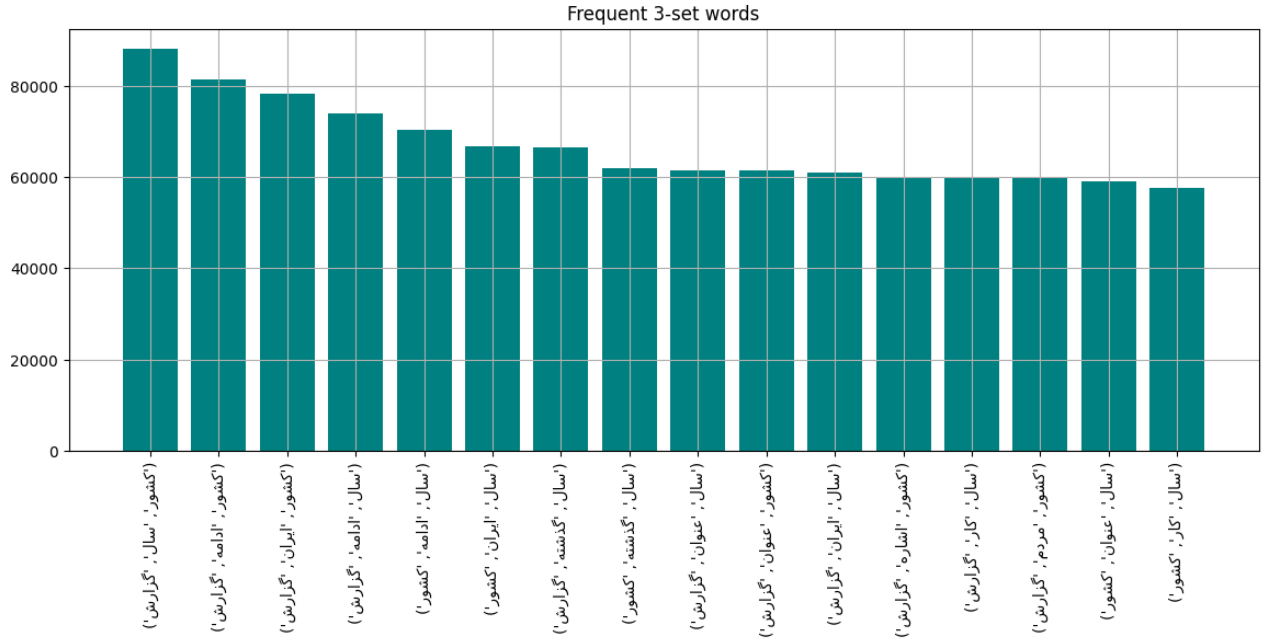

Figure 11: Frequent 3-set words

The output demonstrates meaningful multi-word terms around linked concepts.

4.1 TF-IDF

To evaluate the relevance of the identified 3-word phrase associations, a TF-IDF analysis was conducted.

TF-IDF stands for Term Frequency-Inverse Document Frequency, and is often used in text mining to score the importance of words/phrases while balancing their frequency and specificity. The TF component measures how often a term occurs in an article, while the IDF adjusts for common words across the corpus.

The scoring is defined by:

$$\text{score} = f_{t,d}/len(d) \times log(1 + \frac{N}{n_t})$$

This equation was applied to score documents containing the top 3-term phrases from Apriori. A threshold was set to filter for news stories strongly featuring the target phrase words according to TF-IDF.As an example, the following relevant articles were extracted with TF-IDF > 0.06:

Figure 12: News with highest TF-IDF score for the first word

Figure 13: News with highest TF-IDF score for the second word

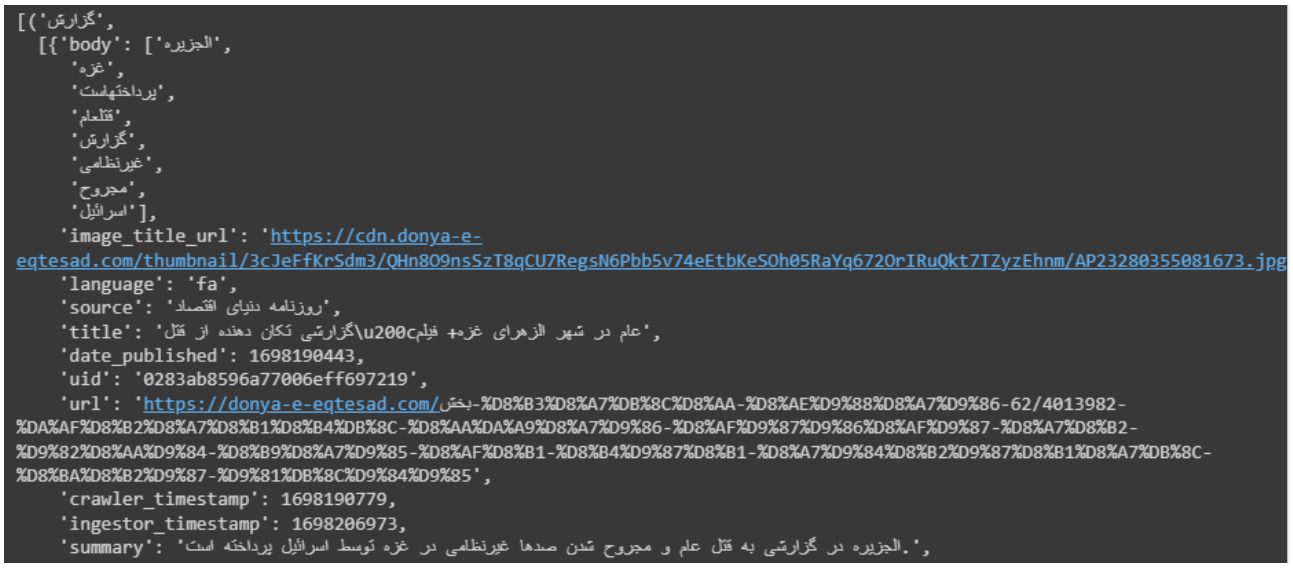

Figure 14: News with highest TF-IDF score for the third word

As it can be seen, some of the selected news stories for different phrase words are related, while others feature the terms in different contexts. This can be attributed to the varying semantic usage and polysemy of the high-level keywords.

However, an interesting observation is that due to the IDF component penalizing frequency in more documents, the TF-IDF model tends to highlight words even in shorter news stories. So

despite lower raw term counts, major terms stand out in specific contexts.

Also as expected, the high count of a word across all documents results in low TF-IDF scores for those very frequent terms. With the same threshold, there was only one news story selected for the third word in the phrase which had highest overall frequency, while multiple news belonged to the other two less common words. This demonstrates the impact of inverse document frequency damping scores for widely used words.

This inverse document frequency effect explains why most of the top-scoring selections are relatively short articles where the key phrase words make up a significant portion of the content. To avoid overemphasizing words from smaller documents, an alternative normalization approach could involve dividing the term frequencies by the maximum word count per document before applying TF-IDF weighting. This normalized frequency would prevent article length bias and improve generalization of the selections.