



Recognition Changes in Social State Resting upon Arabic Media

Submitted by:

Shadi Alkeesh Ayman Haj

ID: 324992320

ID: 316255439

Project Source Code:

 [Link to Project Github](#)

Supervisors:

Dr. Renata Avros

Prof. Zeev Volkovich

January 2026

Table of Contents

1	Introduction	2
2	Material and mathematical preliminary	3
2.1	Vector Space Model	3
2.1.1	N-grams model	3
2.2	Feature Selection	3
2.3	Rank Correlation	4
2.4	Wavelets transform	5
2.5	Clustering	5
3	Methodology	6
3.1	Mean Rank Dependency	6
3.2	Change points detection via Haar discrete wavelet transform	6
3.3	Partitioning of texts in homogeneous groups	7
4	The Project	9
4.1	Stage 1: Data Acquisition	9
4.2	Stage 2: Arabic NLP Toolkit & Normalization	9
4.3	Stage 3: N-gram Extraction	9
4.4	Stage 4: Feature Selection (Noise Filtering)	9
4.5	Stage 5: Mean Rank Dependency Calculation	10
4.6	Stage 6: Change Point Detection	10
4.7	Stage 7: Clustering and Visualization	10
5	System Design & Testing	10
5.1	Sequence Diagram: Linguistic Analysis Pipeline	10
5.2	Testing Plan	11
5.2.1	Unit Testing (Component Level)	11
5.2.2	Integration Testing (Pipeline Level)	12
5.2.3	Code Quality Assurance & Robustness	12
5.2.4	Validation & Ground Truth (Sanity Check)	12
6	References	14

1 Introduction

The spread of digitized resources to a high level of the information society allows automatically retrieving data from observations using modern technologies. In this regard, new approaches are required for processing documents on the basis of a scrupulous analysis of the entire linguistic system, for example, the Arabic mass media as a whole . Any mass media may be considered as a system operating according to its own underlying rules, where the system behavior and properties depend on internal features and stimuli as well as on surrounding factors .

One of the possible illustrations is a relationship between the language of the mass media and the society. Mass media systems serve as a feedback mechanism that reflects changes in the natural and social environment through personal perceptions and evaluations of reality . Traditionally, the daily press played this pivotal role. As demonstrated in foundational research by Volkovich et al. (2016), established newspapers like Egypt’s ”Al-Ahram” functioned as primary historical records, effectively acting as a ”mirror” of society where editorial policies reflected official ideology and societal shifts at a high resolution.

However, the modern information landscape has shifted decisively from the 24-hour cycle of the printed press to the real-time dynamics of social network platforms. While the traditional newspaper was imagined as a static mirror reflecting changes in the life of the society, modern social platforms like X (formerly Twitter) serve as a dynamic, real-time mirror. While ”Al-Ahram” captured the events of the ”Arab Spring” through daily editorial filtering , the events of the **Israel-Gaza conflict (commencing October 7th, 2023)** generated an immediate, unfiltered explosion of data. The language used in these platforms has many peculiarities, characterized by slang, dialects, and morphological inconsistencies, which differ significantly from printed editions. This project aims to bridge the gap between traditional journalism analysis and modern social media mining. It focuses on emerging methods for the modeling and visualization of media in Arabic using new quantitative attributes applied to social media data. We upgrade the existing platform—originally designed for the structured language of the press—to handle the chaotic nature of tweets. The project strives to expose discrete markers essential for identifying points signaling changes in the linguistic content connected to fluctuations in the political and security life of the region .

To address the challenge of processing chaotic Arabic text, we adopt the robust **N-gram based Vector Space Model (VSM)**. This approach allows us to treat the continuous stream of tweets with the same mathematical rigor used for analyzing newspaper archives. By utilizing Normalization techniques and a specialized **Arabic NLP Toolkit**, we aim to demonstrate how the ”social mirror” of X reflects the rapid and violent fluctuations of the conflict through changes in the linguistic structure of the Arabic discourse.

2 Material and mathematical preliminary

Our aim is to demonstrate the ways in which significant geopolitical events can be recognized using the proposed methodology within the dynamic environment of social media. As mentioned earlier, to this end we consider a dataset of Arabic tweets collected from the X platform (formerly Twitter) during the Israel-Gaza war, starting from October 7th, 2023. This section outlines the mathematical foundations required for representing these short texts and the specific challenges associated with Arabic text mining.

2.1 Vector Space Model

In this project, we utilize a robust N-gram based version of the common Vector Space Model (VSM). The general VSM representation ignores grammar and the exact order of terms but preserves the variety of terms used in the corpus. Each document (in our case, a tweet) is characterized through a term frequency table against the vocabulary containing all the terms in the corpus. These tables are deemed as vectors in a linear space having dimensionality identical to the vocabulary size.

Processing Arabic texts, particularly from social media, presents unique challenges. The Arabic language is known for the richness of its vocabulary, the cursiveness of its script, and a complex system of verbal conjugation and noun declension. Furthermore, unlike other languages, Arabic utilizes a system of "broken plurals" and a large number of prefixes, suffixes, and particles.

These complexities are amplified in the context of X (Twitter), where users frequently employ local dialects, slang, and non-standard spellings. Traditional "Bag of Words" (BoW) models often struggle here, as they require accurate morphological analysis to be effective. Therefore, to overcome the noise inherent in social media data and the morphological complexity of Arabic, we adopt the N-grams model described below.

2.1.1 N-grams model

An N-gram is a connecting sequence of N characters from a text. In this model, the "terms" are the sequences of symbols occurring in a sliding window of length N. The main reason to use N-grams for our Twitter dataset is that the technique is recognized as being strongly insensitive to recorded mistakes and spelling errors, and it does not require deep linguistic knowledge or dictionaries.

Additionally, applications of the N-gram methodology have a tendency to implicitly filter out words' affixes (prefixes and suffixes). This allows the model to expose, in a formal way, the words' roots, which consist of three letters for the majority of Arabic words. In view of these advantages—robustness to noise and ability to capture linguistic roots—we base our considerations on this methodology.

2.2 Feature Selection

Obviously, a VSM representation resting upon all N-grams appearing in a corpus leads to very sparse, high-dimensional vectors. This challenge is exacerbated in the context of social media data, where the vocabulary size is theoretically infinite due to user-generated content, misspellings, and platform-specific slang.

To construct a reliable model, we must reduce the dimensionality by filtering out "sparse" N-grams—those that arise merely in small fractions of the data with minor occurrences. Such sparse features are "blind" to the bulk of the events and contribute mostly noise. Conversely, "frequent" N-grams exhibit a better ability to separate the documents and capture the underlying linguistic signal.

To quantify the "heaviness" of the N-gram distributions and distinguish between sparse and frequent terms, we calculate the occurrences of actual N-grams across the dataset and quantify the tails of the histograms via the normalized median (V_1), defined as:

$$V_1 = \frac{\text{median}(f_{ij}, j = 1, \dots, m)}{\max(f_{ij}, j = 1, \dots, m)}$$

Where f_{ij} is the frequency of occurrence of an N-gram i in a document (or time window) $j = 1, \dots, m$.

Consistent with the methodology proposed by Volkovich et al., we recognize an N-gram as "frequent" (and thus select it for the model) only if its normalized median exceeds a specific threshold level. For this project, this rigorous selection process ensures that our analysis of the Israel-Gaza

war focuses on stable linguistic patterns rather than transient social media noise.

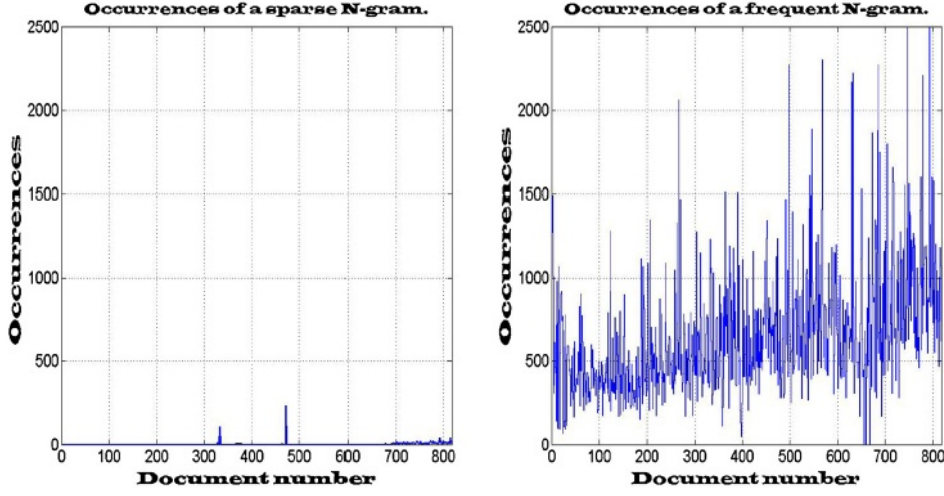


Figure 1: Graphs of the occurrences of a “sparse” N-gram and a “non-sparse (frequent)” one, in 817 issues of the “Al-Ahraam” newspaper collection.

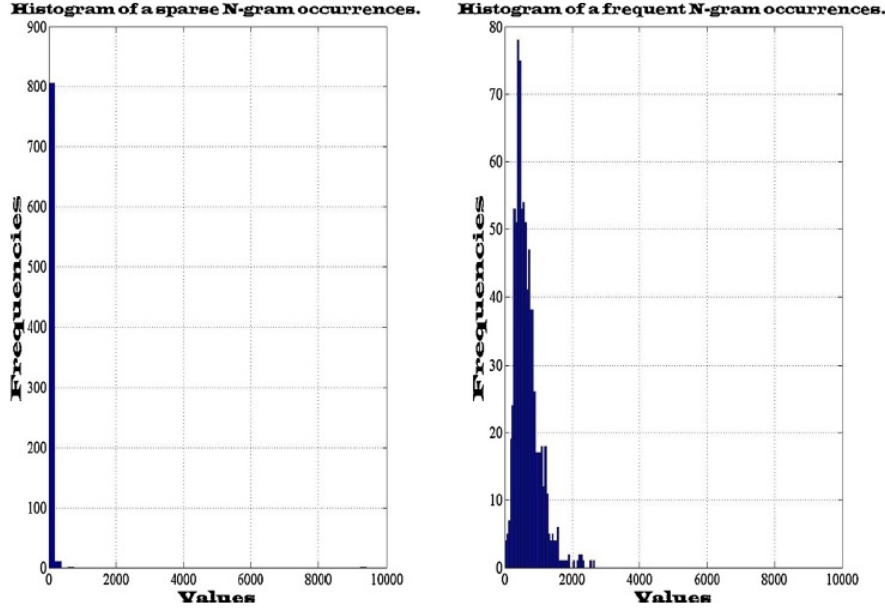


Figure 2: Histograms of occurrences of a “sparse” and a “frequent” 3-gram for the “Al-Ahraam” newspaper collection.

2.3 Rank Correlation

A ranking is an arrangement of items in a given set such that the items are compared between themselves to have unique locations in a hierarchy¹. To quantify the connection between the linguistic histograms of different time periods (e.g., different days of the war), we employ a rank correlation coefficient. This approach treats the N-gram frequencies as ordinal data rather than absolute values.

We specifically utilize Spearman’s rank correlation coefficient (ρ), which assesses how well the relationship between two variables can be described using a monotonic function. The coefficient values range from -1 to +1, where +1 indicates perfect agreement between the ranks (identical

style), 0 indicates no correlation, and -1 indicates complete disagreement (opposite rankings). For a collection of n items (N-grams) with two order scales $X = \{x_i\}$ and $Y = \{y_i\}$, the Spearman value ρ is calculated as:

$$\rho = 1 - \frac{6 \sum_{i=1}^n d_i^2}{n(n^2 - 1)},$$

where d_i represents the difference between the corresponding ranks of each N-gram in the two scales (e.g., the rank difference of a specific term between Day 1 and Day 2 of the war).

Relevance to the Project: In the context of the Israel-Gaza conflict, the volume of tweets fluctuates wildly. By using rank correlation, our model becomes robust to these volume spikes. We are interested in the structure of the discourse (which terms are most dominant relative to others) rather than the raw noise of tweet counts.

2.4 Wavelets transform

The time series generated by social media activity is inherently noisy and non-stationary. Standard signal processing techniques, such as the Fourier transform, assume that the signal properties remain constant over time, which is not the case for dynamic events like the Israel-Gaza war. To address this, we employ the Wavelet transform, which provides a time-frequency representation of the signal, allowing us to analyze localized changes.

Wavelets are constructed from a single original function $\psi(t)$, termed the "mother wavelet," by dilations and shifting. For our discrete dataset, we utilize the **Haar wavelet**, which is the simplest wavelet function resembling a step function. This choice is particularly suitable for detecting sharp transitions (edges) in the data.

The signal f is approximated using a linear combination of basis functions:

$$f = \sum_{k=1}^L b_k \varphi_k + \sum_{j=0}^{\infty} \sum_{k=1}^L b_{jk} \psi_{jk},$$

where:

- b_k are the approximation coefficients (representing the smooth trend).
- b_{jk} are the detail coefficients (representing the fluctuations/noise).

Application to the Project: In our analysis, the "detail coefficients" often correspond to the random noise of social media chatter. By applying a threshold and setting insignificant detail coefficients to zero (a process known as wavelet shrinking or smoothing), we can reconstruct a "denoised" version of the rank correlation graph.

This smoothed signal allows us to identify significant Change Points—moments where the approximation coefficients shift dramatically. In the context of the war, these points signal a fundamental change in the linguistic structure of the discourse, likely triggered by major ground operations or political announcements.

2.5 Clustering

Cluster analysis is the key mechanism intended to recognize meaningful homogeneous groups, named clusters, within the data. In the context of the Israel-Gaza war, our goal is to group sequential days or time windows that share similar linguistic characteristics into distinct "war phases".

To achieve this, we employ the Partitioning Around Medoids (PAM) algorithm (Kaufman & Rousseeuw, 1990). While the K-means algorithm is widely used, PAM is significantly more robust because it minimizes a sum of dissimilarities rather than a sum of squared Euclidean distances³. This distinction is vital for analyzing social media data, which frequently contains outliers and anomalies that could skew a K-means result.

The PAM algorithm operates by identifying k representative items (medoids) within the dataset and iteratively exchanging them with non-medoids to minimize the total distortion.

Determining the Number of Phases: A major challenge in clustering is deciding the number of clusters (k) beforehand. Following the methodology of Volkovich et al., we estimate the optimal number of clusters based on the Change Points detected in the previous step (Section 2.4). The data is partitioned using PAM for a number of clusters ranging between 2 and a maximum limit

derived from the number of change points found plus one.

This approach aligns with the natural assumption that the period between two chronological change points (e.g., between the start of the war and the ground invasion) constitutes a homogenous group, and that similar linguistic "states" may recur at different times during the conflict.

3 Methodology

In this section, we present the mathematical description of the proposed methodology. We introduce a novel similarity measure, named the **Mean Rank Dependency**, which evaluates the connection between a current time window (e.g., a specific day of the war) and its preceding period. Based on this measure, we describe the procedure for detecting significant social fluctuations and partitioning the war timeline into linguistically homogeneous phases.

3.1 Mean Rank Dependency

Traditional similarity methods often evaluate the connection between two specific texts in isolation, disregarding their temporal context. However, public discourse on social media is inherently sequential; a tweet or a daily trend is deeply influenced by the discussions of the previous days. For example, the discourse on "Day 5" of the war is an evolution of the shock and narratives formed during "Days 1-4".

To capture this "memory" of the process, we model the Twitter stream as a time series where each daily aggregation of tweets is affected by its association with numerous earlier ones.

We introduce the Mean Rank Dependency (ZV_T) characteristic. This measure quantifies the mean rank correlation between a current document (daily aggregation) D_i and a set of its T "precursors" (the previous T days), denoted as $\Delta_{i,T} = \{D_j, j = i - T, \dots, i - 1\}$.

The value is calculated as:

$$ZV_T(D_i) = \frac{1}{T} \sum_{D_j \in \Delta_{i,T}} \rho(D_i, D_j)$$

Where:

- $\rho(D_i, D_j)$ is the Spearman rank correlation coefficient (defined in Section 2.3).
- T is the delay parameter (the size of the memory window).

Significance for the Project: By calculating this rolling average of correlations, we smooth out daily anomalies. A high ZV_T value indicates that the discourse is stable and consistent with previous days. A sharp drop in this value suggests a "break" in the narrative—indicating that the events of the current day have caused such a shock that the linguistic structure no longer resembles the recent past. This is our primary indicator for detecting events like the onset of the war or major tactical shifts.

3.2 Change points detection via Haar discrete wavelet transform

Change point detection is a methodology aiming to identify the specific moments when a time series significantly alters its behavior. In our context, these alterations correspond to shifts in the social sentiment or the narrative regarding the war.

To detect these points within the noisy ZV_T graph calculated in the previous step, we utilize the smoothing capabilities of the Haar wavelet transform (detailed in Section 2.4). The procedure is as follows:

1. **Approximation:** A signal approximation of the ZV_T series is calculated at a specific decomposition level (e.g., level 6, as suggested in 1). This filters out high-frequency noise (irrelevant tweet fluctuations) and leaves the "skeleton" of the trend.
2. **Difference Calculation:** We calculate the absolute differences of this approximated signal.
3. **Thresholding:** A jump in the signal is accepted as a significant "Change Point" if its size is substantial relative to the overall maximal jump (e.g., at least half the size of the maximal jump).

These detected points (k^-) serve as markers for potential transitions between different phases of the war (e.g., from "Shock" to "Air Strikes" to "Ground Invasion") and are used to initialize the clustering process.

3.3 Partitioning of texts in homogeneous groups

The final component of the methodology is partitioning the timeline into linguistically homogeneous groups (clusters). To do this accurately, standard Euclidean distance is insufficient because it does not account for the sequential nature of the data.

Therefore, we introduce a novel distance function proposed by Volkovich et al., which measures text diversity through the **Mean Rank Dependency**. The distance between two document aggregations (days) D_i and D_j is defined as:

$$DZV_T(D_i, D_j) = |ZV_T(D_i, \Delta_{i,T}) + ZV_T(D_j, \Delta_{j,T}) - ZV_T(D_i, \Delta_{j,T}) - ZV_T(D_j, \Delta_{i,T})|$$

Interpretation of the Formula: This formula is sophisticated: instead of just comparing Day i to Day j , it compares:

1. How Day i relates to its own past.
2. How Day j relates to its own past.
3. **Crucially:** How Day i would relate to Day j 's past (cross-comparison).

If the documents are identically connected with their own previous neighbors and the previous neighbors of the other document, the distance is zero ($DZV_T = 0$), meaning they belong to the exact same "style" or war phase.

This custom distance matrix is then fed into the **PAM clustering algorithm** (described in Section 2.5) to produce the final visualization of the war's phases.

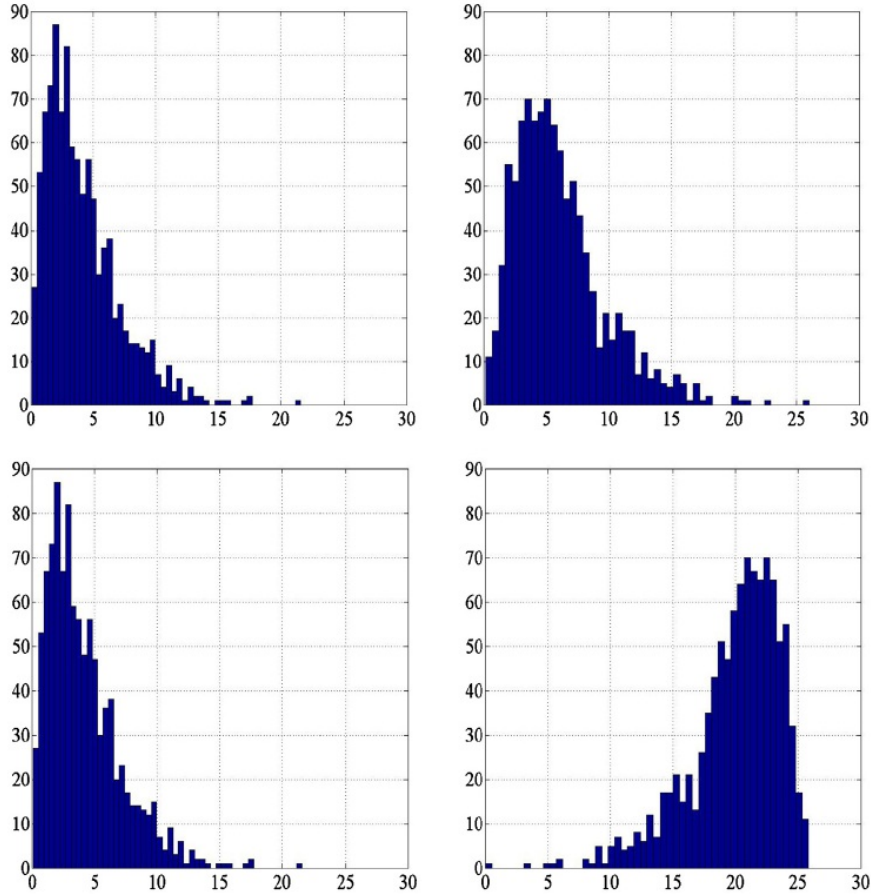


Figure 3: Examples of similar and dissimilar histograms.

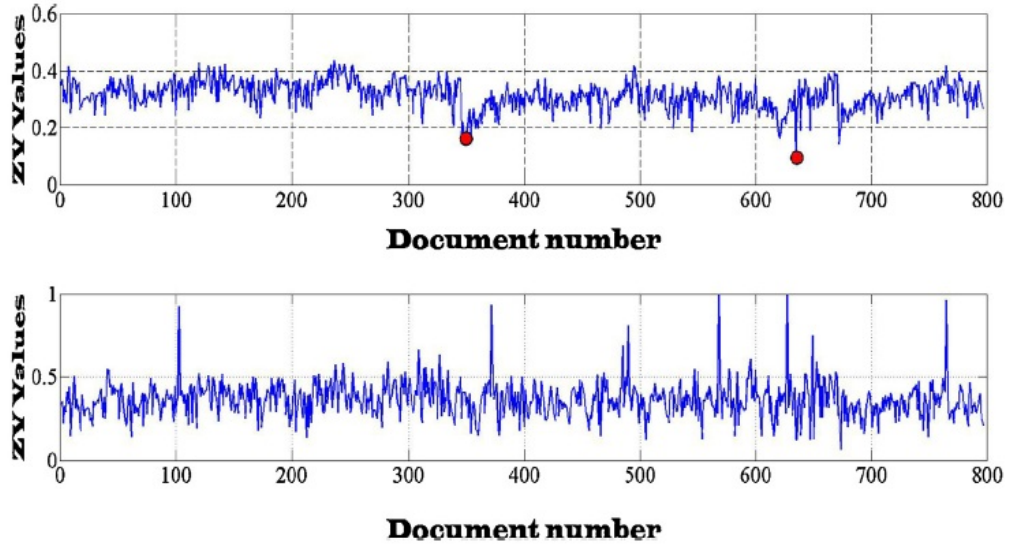


Figure 4: Examples of ZVTgraphs.

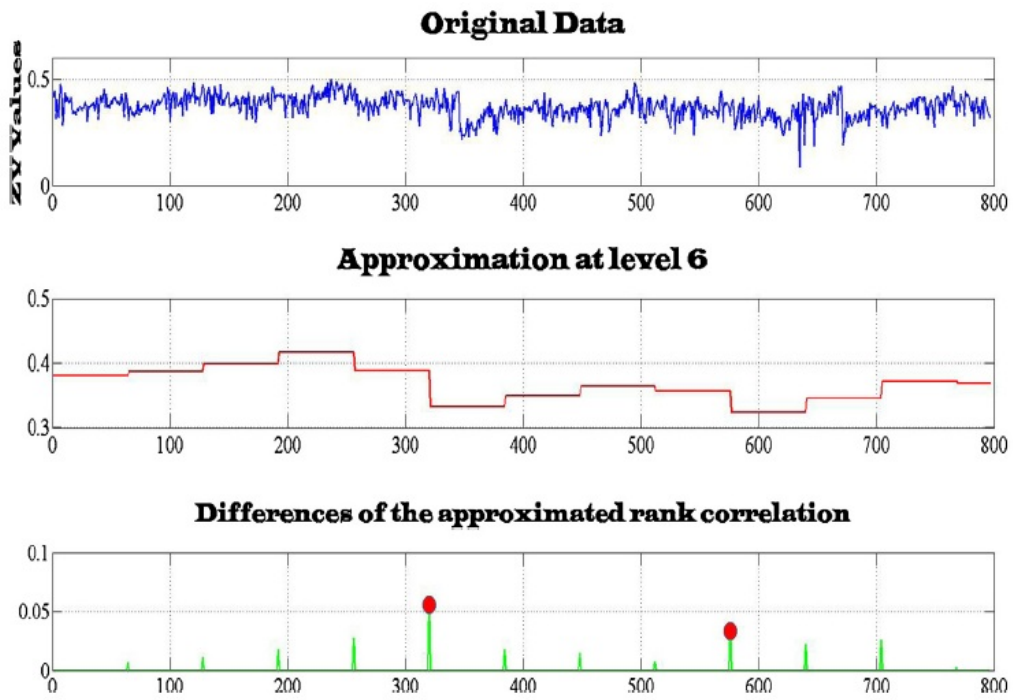


Figure 5: Example of change point detection.

4 The Project

This section details the practical implementation workflow of the project. The analytical pipeline is designed to transform raw, noisy social media data into a structured visualization of the war's linguistic evolution. The process consists of six sequential stages:

Add Flow Chart

4.1 Stage 1: Data Acquisition

Unlike the static newspaper archives used in previous research, this project deals with dynamic, real-time data.

- **Source:** We extract data from the X (Twitter) platform using the official API (or a dedicated dataset).
- **Scope:** The dataset comprises Arabic-language tweets filtered by keywords related to the **Israel-Gaza conflict** (e.g., "Gaza", "Israel", "Al-Aqsa Flood", "Iron Swords").
- **Timeframe:** The data collection begins on **October 7th, 2023**, and continues through the subsequent months of the conflict.
- **Preprocessing:** Basic cleaning is performed to remove URLs, non-Arabic characters, and duplicate spam bots, while retaining the raw morphological structure of the text.

4.2 Stage 2: Arabic NLP Toolkit & Normalization

Processing Arabic text requires a specialized **NLP Toolkit** to handle the language's rich morphology and complex system of prefixes/suffixes.

- **The Toolkit:** We utilize standard libraries for Arabic text processing (such as **PyArabic** or **NLTK**) to construct our preprocessing pipeline.
- **Normalization:** Before analysis, we apply a strict Normalization process to unify the inconsistent orthography typical of social media. This includes:
 - **Alef Normalization:** Unifying all forms of Alef (أ , إ , آ) to a simple bare Alef (ا).
 - **Character Removal:** Removing diacritics (Tashkeel) and the Tatweel character (stretching of letters, e.g., عـــــــ), which are purely stylistic and add noise.
 - **Spelling Correction:** Normalizing ة (Taa Marbuta) to ه (Haa) and ع (Alif Maqsurah) to ي (Yaa) where appropriate.
 - Note: This step is crucial because, in the Vector Space Model, the word "Gaza" written as غزة and غزة would be treated as two different terms without normalization.

4.3 Stage 3: N-gram Extraction

To handle the complexity of Arabic morphology and the informality of tweets:

- We convert the cleaned text into **3-grams** (sequences of 3 characters).
- This choice is based on the linguistic property that the majority of Arabic words are derived from trilateral (three-letter) roots.
- By focusing on 3-grams, we implicitly capture the root meaning of words regardless of the specific prefixes or suffixes added by users (e.g., capturing the root of "bombing" regardless of whether it is written as "and-bombing" or "the-bombing").

4.4 Stage 4: Feature Selection (Noise Filtering)

Social media data is extremely sparse. To avoid analyzing "noise":

- We calculate the frequency distribution of all extracted N-grams.
- We apply the **Normalized Median** (V_1) filter described in Section 2.2.
- Only N-grams that exceed a strict threshold (e.g., the top percentile of "heaviness") are selected for the Vector Space Model. This ensures that our analysis is based only on statistically significant linguistic terms.

4.5 Stage 5: Mean Rank Dependency Calculation

This is the core analytical step where we measure the "pulse" of the discourse:

- We treat the daily aggregation of tweets as a single "document".
- For each day, we calculate the **Mean Rank Dependency** (ZV_T) against a sliding window of the previous T days.
- We use **Spearman's rank correlation** to compare the ranking of the top N-grams.
- **Result:** A time-series graph representing the stability of the public discourse. A high value means the narrative is consistent; a low value indicates a rupture or shock.

4.6 Stage 6: Change Point Detection

To identify specific dates of geopolitical significance (e.g., the start of the ground invasion):

- We apply the **Haar Wavelet transform** to smooth the ZV_T graph.
- We analyze the **approximation coefficients** to filter out daily noise.
- Significant jumps in the smoothed signal are marked as **Change Points**, indicating a fundamental shift in the war's phase.

4.7 Stage 7: Clustering and Visualization

Finally, we visualize the "Phases of the War":

- We use the **PAM algorithm** to group days into clusters.
- The distance between days is calculated using the custom DZV_T **distance function**.
- **Output:** A colored timeline where each color represents a distinct linguistic phase (e.g., "Phase 1: Initial Shock," "Phase 2: Routine of War," "Phase 3: Ceasefire/Hostage Deal"). This visualization provides a clear, data-driven narrative of the conflict's evolution.

5 System Design & Testing

This section presents the logical flow of the system and the testing strategy used to validate the reliability of the proposed methodology.

5.1 Sequence Diagram: Linguistic Analysis Pipeline

The following sequence diagram illustrates the interactions between the User, the System Modules, and the Data Sources during the analysis process.

Description of the Sequence Diagram: The diagram illustrates the sequential interactions between the system components during the analysis of the Israel-Gaza conflict timeline:

1. **Initiation:** The process begins when the Researcher triggers the system by defining a specific timeframe (e.g., Oct 2023 - Dec 2023).
2. **Data Retrieval:** The **System** queries the **X API/Dataset** to retrieve the raw Arabic tweets relevant to the conflict.
3. **Preprocessing Loop:** For each tweet, the system enters a processing loop:
 - The **Normalizer** unifies the Arabic orthography (removing diacritics, unifying Alef forms).
 - The **N-Gram Extractor** converts the cleaned text into 3-gram tokens.
4. **Noise Filtering:** Once processed, the **Filter** component calculates the Normalized Median (V_1) to remove sparse, non-informative terms from the vector space.
5. **Core Analysis:** The **Analyzer** computes the Mean Rank Dependency (ZV_T) across the timeline to measure linguistic stability.

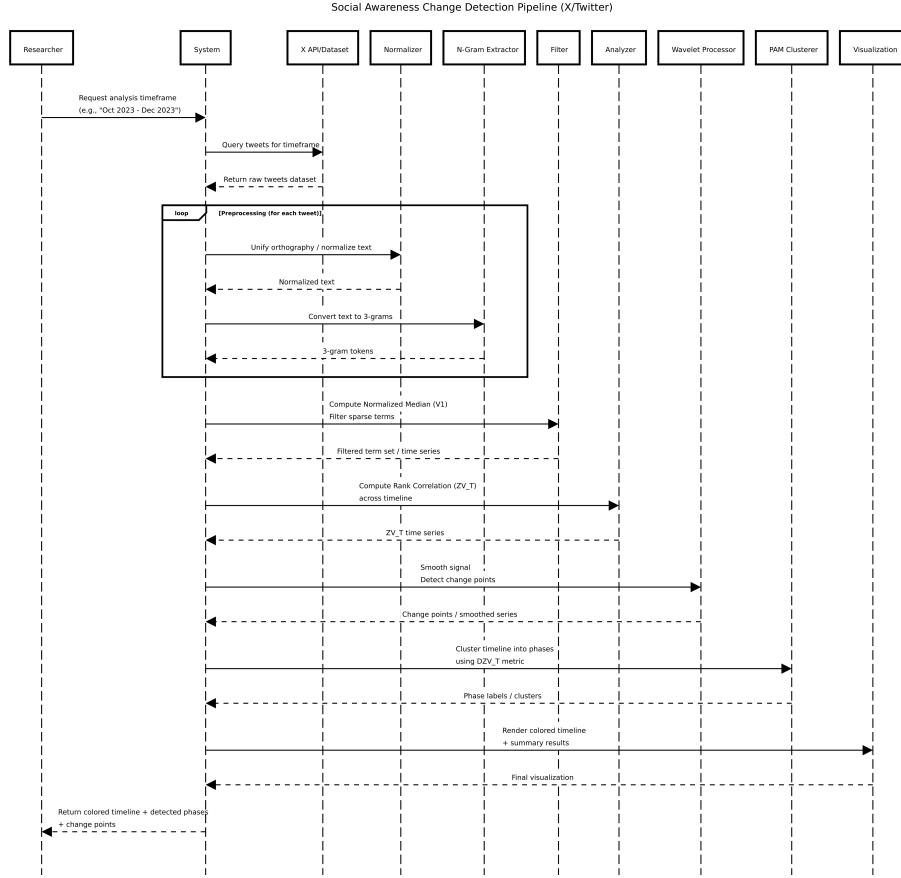


Figure 6: Event-Driven Linguistic Analysis Workflow

6. **Event Detection:** The **Wavelet Processor** smooths the signal and identifies significant "Change Points" (e.g., sudden shifts in discourse).
7. **Clustering:** The **PAM Clusterer** uses the calculated metrics (DZV_T) to group days into linguistically homogeneous phases.
8. **Output:** Finally, the **Visualization** module renders the colored timeline and presents the detected war phases to the Researcher.

5.2 Testing Plan

To ensure the robustness of the system and the validity of the analytical results, we define a comprehensive testing plan consisting of three levels: Unit Testing, Integration Testing, and Ground Truth Validation.

5.2.1 Unit Testing (Component Level)

Each mathematical module is tested in isolation to verify algorithmic correctness.

- **Normalization Test:**

- Input: غَزَّة (Gaza with Shadda/Tashkeel) and غزة (Gaza with Haa).
- Expected Output: Both must map to the identical string غزة.

- **Spearman Correlation Test:**

- Input: Two identical lists of N-grams.
- Expected Output: Correlation coefficient (ρ) must be exactly 1.0.
- Input: Two completely reversed lists.
- Expected Output: Correlation coefficient (ρ) must be -1.0.

- **Metric Properties Test:**

- Check: Verify that the custom distance $DZV_T(D, D) = 0$ for any document D , confirming the semi-metric property.

5.2.2 Integration Testing (Pipeline Level)

Ensuring that data flows correctly between stages without loss or corruption.

- **Dimensionality Check:** Verify that the number of selected features (N-grams) after the V_1 filtering stage is significantly lower than the raw input, typically around 1-10% of total N-grams, ensuring the removal of noise.
- **Data Consistency:** Ensure that the number of days in the final output graph matches the number of days in the input timeframe minus the window size T (since the first T days are required for initialization).

5.2.3 Code Quality Assurance & Robustness

Beyond algorithmic correctness, it is essential to ensure the software codebase is free of syntax errors, follows engineering standards, and is robust against unexpected runtime failures.

- **Static Code Analysis:** To detect potential bugs (such as undefined variables or unreachable code) without executing the program, we utilize static analysis tools compatible with the Python environment (e.g., **Pylint** or **Flake8**).
 - Criterion: The code must adhere to **PEP8** style guidelines and score above a defined threshold (e.g., 8/10 on Pylint), ensuring readability and maintainability.
- **Exception Handling & Robustness:** Given the external dependency on the X (Twitter) API and the chaotic nature of user-generated content, the system includes comprehensive try-except blocks.
 - API Resilience: The system is tested to handle HTTP errors (e.g., 429 Too Many Requests) or timeouts gracefully by implementing retry mechanisms or logging the error without crashing the entire pipeline.
 - Malformed Data: The Preprocessing Loop is tested against null inputs or non-textual data to ensure the Normalizer does not throw unhandled exceptions.
- **Logging Verification:** We verify that the system maintains a detailed log file recording the execution flow. This ensures that if a logical error occurs during the "Change Point Detection" or "Clustering" stages, it can be traced back to the specific timestamp and data batch.

5.2.4 Validation & Ground Truth (Sanity Check)

Since this is a data science project, "passing the code" is not enough; the results must make historical sense. We compare the system's output (detected Change Points) against known real-world events (Ground Truth).

- **Hypothesis:** Significant military or political events should correlate with detected Change Points.
- **Test Case 1: The Outbreak (Oct 7):**
 - Success Criterion: The system must detect a sharp drop in correlation (ZV_T) or a specific Change Point on or immediately after October 7th, 2023.
- **Test Case 2: The Ground Invasion (Oct 27):**
 - Success Criterion: Detection of a phase transition corresponding to the start of ground maneuvers.
- **Test Case 3: The Ceasefire (Nov 24):**
 - Success Criterion: The clustering algorithm should identify the temporary ceasefire period (Nov 24 - Dec 1) as a distinct cluster or a stable period in the smoothed graph, similar to how stability was detected after specific verdicts in previous research.

Table 1: Summary of System Testing and Validation Plan

Test Level	Test Component	Success Criterion / Expected Output
1. Unit Testing (Component)	Normalization	Variants (e.g., Gaza w/ Shadda) map to identical string.
	Spearman Correlation	Identical lists yield $\rho = 1.0$; Reversed lists yield $\rho = -1.0$.
	Metric Properties	Distance $D_{ZVT}(D, D) = 0$ for any document D .
2. Integration (Pipeline)	Dimensionality	Features reduced to $\approx 1\text{--}10\%$ of raw input (noise removal).
	Data Consistency	Output days = Input days – Window Size (T).
3. Code Quality & Robustness	Static Analysis	Adherence to PEP8; Pylint/Flake8 score $> 8/10$.
	API Resilience	Graceful handling of HTTP 429/Timeouts (Retries/Logs).
	Malformed Data	Preprocessing handles nulls/non-text without crashing.
	Logging	Execution flow and errors traced to specific timestamps.
4. Validation (Ground Truth)	Oct 7 (Outbreak)	Detection of sharp drop in correlation or Change Point.
	Oct 27 (Invasion)	Detection of phase transition matching maneuvers.
	Nov 24 (Ceasefire)	Identification of distinct cluster or stable period.

6 References

1. Amelin, K., Granichin, O., Kizhaeva, N., & Volkovich, Z. (2018). Patterning of writing style evolution by means of dynamic similarity. *Pattern Recognition*, 77, 45-64.
2. Avros, R., & Volkovich, Z. (2022). Deep sentimental analysis of the Arabic medieval documents. *Procedia Computer Science*, 207, 709-715.
3. Bernikova, O., Granichin, O., Lemberg, D., Redkin, O., & Volkovich, Z. (2020). Entropy-based approach for the detection of changes in Arabic newspapers' content. *Entropy*, 22(4), 441.
4. Franch, F. (2013). (Wisdom of the Crowds) 2: 2010 UK election prediction with social media. *Journal of Information Technology & Politics*, 10(1), 57-71.
5. Gonzalez, R. C., & Woods, R. E. (2002). *Digital image processing* (2nd ed.). Upper Saddle River, NJ: Prentice Hall.
6. Kaufman, L., & Rousseeuw, P. J. (1990). *Finding groups in data: An introduction to cluster analysis*. Hoboken, NJ: John Wiley & Sons.
7. Kendall, M. G., & Gibbons, J. D. (1990). *Rank correlation methods* (5th ed.). London: Edward Arnold.
8. Soliman, A. B., Eissa, K., & El-Beltagy, S. R. (2017). Aravec: A set of arabic word embedding models for use in arabic nlp. *Procedia Computer Science*, 117, 256-265.
9. Volkovich, Z. (2020). A short-patterning of the texts attributed to Al Ghazali: A "Twitter look" at the problem. *Mathematics*, 8(11), 1937.
10. Volkovich, Z., Granichin, O., Redkin, O., & Bernikova, O. (2016). Modeling and visualization of media in Arabic. *Journal of Informetrics*, 10(2), 439-453.