BOT: a Token-Based Bag of Words Model for Arabic Biographical Texts Clustering

Ebtihal Nooreldeen [1], and Anna Maria Zanaboni [2]

[1] College of Computer Science and Information Technology, Sudan University of Science and Technology, Khartoum, Sudan, (e-mail: ebtihal.nooreldeen@gmail.com)

[2] Computer Science Department and Data Science Research Center (DSRC), Università degli Studi di Milano, Milan, Italy, (e-mail: zanaboni@di.unimi.it)

Received: 10/03/2026
Accepted: 02/06/2026

***Abstract*— Information retrieval often yields an overwhelming number of results, making effective data utilization challenging. Clustering techniques enhance navigation by grouping similar results. However, Arabic text clustering research often neglects semantic relationships. This study introduces the Bag of Tokens (BOT) model, which integrates semantic information, unlike the traditional Bag of Words (BOW) model. BOT relies on entities and relations extracted from text, enabling both improved clustering and dimensionality reduction. The approach was evaluated on an Arabic text dataset from the Wikipedia Monolingual Corpora, focusing on Sudanese politicians' biographies. Results demonstrated that BOT achieves lower dimensionality and enhances cluster coherence and separation across various clustering algorithms and distance metrics compared to BOW.**

***Index Terms*—** Arabic Text Clustering, Semantic Representation, Bag of Words.

المستخلص - غالبًا ما يُنتج استرجاع المعلومات عددًا هائلًا من النتائج، مما يجعل الاستفادة الفعّالة منها أمرًا صعبًا. وتُسهم تقنيات التجميع في تسهيل تصفح هذه النتائج من خلال تجميع الوثائق المتشابهة في مجموعات مترابطة. ومع ذلك، فإن معظم الأبحاث المتعلقة بتجميع النصوص العربية لا تولي اهتمامًا كافيًا للعلاقات الدلالية بين الكلمات والكيانات. تقدم هذه الدراسة نموذج Bag of Tokens (BOT) **، الذي يدمج المعلومات الدلالية في تمثيل الوثائق، بخلاف نموذج** Bag of Words (BOW) **التقليدي. ويعتمد نموذج** BOT **على الكيانات والعلاقات المستخرجة من النص، مما يساهم في تحسين جودة التجميع وتقليل أبعاد فضاء السمات في الوقت نفسه. وقد تم تقييم النموذج باستخدام مجموعة بيانات للنصوص العربية من** Wikipedia Monolingual Corpora **، مع التركيز على السير الذاتية للسياسيين السودانيين. وأظهرت النتائج أن نموذج** BOT **يحقق تمثيلًا أقل أبعادًا، كما يحسن تماسك المجموعات ودرجة الفصل بينها عبر مجموعة متنوعة من خوارزميات التجميع ومقاييس المسافة، مقارنةً بنموذج** BOW **.**

I. INTRODUCTION

In the realm of natural language processing, the challenge of effectively representing and understanding the nuances of diverse languages has been a central focus. The clustering of Arabic documents—especially in the context of biographical texts—has received increasing attention due to the need to automatically organize large collections of data, such as historical archives, encyclopedias, and biographical databases.

The challenges posed by Arabic's rich morphology include the use of roots, affixes, and word forms that can result in a high degree of inflection and word variation [1, 2]. These complexities make traditional clustering methods less effective, as they often fail to account for the semantic variations that define different document categories, including biographies. Additionally, these morphological variations increase the complexity of creating index terms in information retrieval [3].

Research on clustering Arabic text is less extensive than in other languages. Most studies emphasize the role of morphological analysis on clustering outcomes. The primary challenges in text clustering include high dimensionality, sparsity of text documents, and data representation.

Traditional models like Bag of Words (BOW) have proven effective in many contexts, and have long been used to represent textual data by counting the frequency of words, yet they ignore word order, syntactic structure, and, critically, the deeper semantic meaning that often holds the key to understanding text in its full context. This becomes particularly problematic when

processing languages with complex morphologies, such as Arabic. In traditional BOW models, the feature space grows significantly with the number of distinct words, and high-dimensional feature spaces often lead to inefficiency and poor performance due to the sparse nature of the data, making clustering difficult. The document clustering process involves three main steps: document representation, similarity calculation, and grouping. The focus of the present work is mainly on document representation, with the following objectives: appropriateness for Arabic language, compact document representation, incorporation of semantics into the document representation, and enhanced clustering performance. The proposed document representation method is called Bag of Tokens (BOT) since it takes into account semantic tokens rather than words as the BOW does.

In previous studies, most datasets have been based on news articles. Arabic datasets were either built from scratch [4, 5], or existing datasets were used [6, 7, 8, 9, 10, 11]. We were interested in the biography domain since its richness in relations is not present in other domains. The biography domain contains cultural and historical information; analyzing this information through clustering can lead to extracting useful knowledge such as similarities between biographies and gaps in certain fields of knowledge. Examples of datasets of biographies are Pantheon1.0 [12] and Wikipedia Monolingual Corpora (WMC) [13]. Pantheon1.0 consists of 11,341 biographies taken from Wikipedia, supporting more than 25 languages including Arabic. It comprises files in the TSV (tab-separated values) format where each row contains characteristics for one biography, such as: person's name, article ID in Wikipedia, birthplace, and gender. To obtain text files, it is necessary to search for and download the articles based on the characteristics provided in the TSV files. The Wikipedia Monolingual Corpora is a textual dataset from the Wikipedia articles, covering 30 languages including Arabic. It consists of 2.6 gigabyte, 709,674 articles, 2,067,569 paragraphs, and 131,120,680 tokens. This dataset was constructed in XML files, with one file for each language. The approach presented in this paper was evaluated on an Arabic text dataset from the Wikipedia Monolingual Corpora, focusing on Sudanese politicians' biographies.

The rest of this paper is organized as follows: Section II reviews related work, the proposed method is illustrated in Section III, experiments are presented in Section IV, results are discussed in Section V, and the conclusion is provided in Section VI.

II. RELATED WORK

Research on clustering in the Arabic language is limited compared to similar studies in other languages, yet it focuses on several key areas: the application and comparison of clustering techniques for Arabic texts, the impact of preprocessing on clustering performance, the advantages of clustering in information retrieval, the strategies for managing high-dimensional data, and the development of novel clustering methods.

Representation models can be categorized into two main types: Classical Representation Models and Distributed Representation Models. Classical Representation Models rely on word frequency and treat words as independent and discrete entities. In contrast, Distributed Representation Models consider the context of words to capture their semantics. The second category includes approaches like Word2Vec, GloVe, FastText, as noted by [14]. A categorization that can be done from another point of view is presented in [15], where the most commonly used document representation methods can be classidied as Vector Space Model, Probabilistic Topic Model, and the Statistical Language Model.

Methods for measuring semantic similarity can be categorized into four main groups: *Co-occurrence-based methods*, that use the Bag-of-Words model for text representation, measuring similarity through cosine similarity; *Statistical corpus-based methods*, that represent words based on their context within a corpus, capturing meaning from their distribution patterns; *Lexical database-based methods*, that rely on resources like WordNet to determine semantic relationships between words; *Word embedding-based methods*, that derive meaning from the context of words in large datasets, utilizing techniques such as Word2Vec, GloVe, or FastText. These various approaches help in understanding how words, sentences, and documents are semantically related [16],[17].

The representation we are proposing can be considered a classical model and aligns with the Vector Space Model; semantic similarity is measured through a Co-occurrence-based method.

Accordingly, in the following short literature review, we consider related works only within the scope of classical and Vector Space Model representations.

The representation of a complete document can cause high dimensionality and noise issues, reducing the efficiency of document clustering and increasing the running time of the clustering process. In [18] the focus is on enhancing clustering efficiency in high-dimensional data, while high dimensionality is addressed by improving document representation in [7, 10, 4];the research in [5] addresses both document representation and improving clustering efficiency.

In [7] Latent Semantic Analysis is recommended for document summarization and BOW for its representation; the work also shows that stemming may negatively impact clustering; in [10] a modification to BOW through semantic grouping of similar words model was useful for dimensionality reduction. The method in proposed [10], however, was not evaluated for clustering purposes. A hybrid method that combines Bayesian Vectorizer and K-means was discussed in [5], where documents were represented by their categories instead of individual words; an improvement in clustering performance and reduction in runtime were reported. In [4] agglomerative hierarchical clustering was applied to Arabic documents, using keyphrases to represent documents; in that work a similarity measure based on the common lemma form of keyphrases (keyphrases sharing the same root are considered as the same word) was introduced, which yielded better clustering results than using the full text.

In [18] Grey Wolf Optimization is combined with Self-Organizing Maps to improve K-means clustering for high-dimensional data; in that work traditional data representations like CountVectorizer and Term Frequency-Inverse Document Frequency (TF-IDF) were used rather than semantic representations; the model was evaluated news datasets such as Modern Standard Arabic and New Arabic Dataset for Text Classification and an improvement over conventional Self-Organizing Maps and K-means methods was found.

Data mining techniques help to reduce high dimensionality, as shown in [19], [20] and [21]. In [19] the FPMax algorithm is used to extract Maximal Frequent Wordsets, which represent recurring word sets instead of individual words; Maximal Frequent Wordsets are then used to calculate document similarity and to form clusters; the method was evaluated to the CNN Arabic corpus and Open-Source Arabic Corpora, with good results. High dimensionality is addressed in [20] using frequent item sets to define clusters, where each cluster is represented by shared words and documents are organized by topic relevance through hierarchical clustering; the methodology included TF-IDF following stemming and was compared with other techniques, such as Frequent Itemset Hierarchical Clustering, on multiple datasets, with good results. Frequent Itemset Hierarchical Clustering is used in [21] together with N-gram analysis for Arabic, achieving better results than for European languages. Agglomerative hierarchical clustering together with suffix trees was used to extract key phrases from documents enhanced clustering quality on the Arabic Contemporary Corpus which includes 278 documents [6].

Three dimensionality reduction techniques, namely Principal Component Analysis, Nonnegative Matrix Factorization, and Singular Value Decomposition were combined with K-means in [22]; experiments on a dataset containing 7,232 Arabic news articles and 7,293 English documents from Reuters, all represented using TF-IDF, shown that Principal Component Analysis outperformed the other two techniques in accuracy and normalized mutual information.

Tools like WordNet are often used to calculate semantic similarity between words, addressing polysemy and synonymy [23]. Semantic representation of Arabic documents is addressed by integrating the Resource Description Framework (RDF) with DBpedia, allowing for easier entity extraction and linking [24]; RDF triples are extracted with the support of the Named Entity Recognition available in GATE[25], using predefined rules; this approach has not been evaluated in the context of information retrieval in general or document clustering in particular.

In clustering Arabic web pages documents were represented by semantic class features [26]; verbs were identified and categorized using VerbNet, then documents were vectorized into semantic class vectors utilizing either density or a probability distribution for weighting, then they were clustered and annotated with key topics. The method was evaluated on a dataset of 753 documents from Arabic online newspapers and outperformed the standard K-means algorithm.

Restricting the focus just on classical representation models, the studies present in the literature [27] reveal a gap in the semantic representation of Arabic documents for clustering, particularly regarding the use of semantic tokens in document representation. Table 1 summarizes these gaps.

III. PROPOSED METHOD

In the proposed approach, documents are represented as bag of tokens (BOT): tokens are extracted though annotations of entities and relations, and the representation of documents takes into account semantic aspects of text. This way of representing documents is more compact and allows for enhanced clustering performance.

In order to produce a BOT representation, the following phases are necessary: entities extraction, relations extraction and document representation, as shown in Figure 1. Some preprocessing is conducted to facilitate subsequent steps, as follows. *Resetting*, where all previous annotations in the data set are deleted. *Normalization*, where some letters in Arabic written in different shapes are unified to one shape, for example, different shapes إ، أ، آ are unified to letter ا. *Tokenization*, where the text is partitioned into tokens.*Tagging*, where words are tagged with the appropriate part of speech tag. In our experiments, we used Stanford Tagger (found in Stanford CoreNLP in [25]) because it supports the Arabic language.

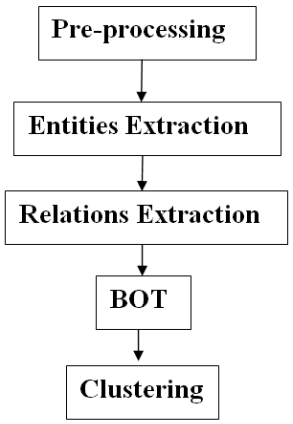

Figure 1: Overall System

A. *Entities Extraction*

Entities are the main topics mentioned in text documents, and are extracted from the dataset through annotation. Annotations can be performed with the aid of gazetteers and annotation rules, often expressed as regular expressions. In our experiments, we annotated entities using our constructed gazetteer and a specific set of rules.

Table 1: Related Work vs. BOT

Aspect	Related Work	Gap / Limitation	BOT
Dataset Usage	news corpora, Arabic web pages, Reuters, CNN Arabic corpus, etc.	Lack of a unified semantic representation adaptable across datasets	can be applied to different Arabic text datasets
Representation	BOW, TF-IDF, keyphrases, frequent itemsets	Limited semantic understanding	Uses semantic Bag-of-Tokens (BOT)
Features Used	Individual words or statistical features	Ignores structured semantic information	Uses entities and relations annotations
Semantic Information	Partial use of WordNet, RDF, DBpedia, VerbNet	Semantic resources not integrated into clustering-oriented representation	Directly incorporates semantic aspects into document representation
Dimensionality	High-dimensional representations	Causes sparsity, noise, and longer clustering time	More compact representation
Noise Reduction	Limited handling of noisy terms	Full-text representations include irrelevant words	Uses extracted semantic tokens only
Clustering Performance	Improved using optimization or feature reduction methods	Representation itself still weak semantically	Enhances clustering quality through semantic representation
Existing Studies	Clustering algorithms and preprocessing	Limited focus on semantic document modeling	Focuses on semantic-aware representation model
Evaluation	Some semantic methods not evaluated for clustering	Lack of clustering-oriented semantic representation	Evaluated specifically for document clustering
Novelty	Traditional or partially semantic methods	No combined entity–relation token representation	Combines entities and relations in BOT model

B. Relation Extraction

There are various methods for extracting relationships between entities. The rule-based approach relies on predefined rules to identify relationships. The statistical approach exploits machine learning techniques to infer relationships from data. A hybrid approach combines both rule-based and statistical methods for enhanced accuracy.
In our work, we adopted the rule-based method.

C. Bag of Tokens (BOT)

Previous research has primarily utilized the BOW model to represent text data. This method, however, has notable drawbacks, such as high dimensionality and a lack of semantic information in the representation. The proposed BOT representation considers only tokens that contribute to semantic meaning, such as entities and relations. Additionally, feature importance is weighted using frequencies rather than TF-IDF as usually done with the Bag of Words representation.

The corpus of N documents is represented as the ***BoT matrix***, an $N \times F$ matrix where F is the total number of features in the corpus. Each row d_i is an F-dimensional vector representation of document d_i.
Two layers of features are considered: the first layer contains M primary features $(P_1, P_2, \ldots, P_M)$, and the second layer contains a variable number of secondary features associated to each primary feature.

The primary features encompass all entity and relation types in the corpus, while the secondary features for a given primary feature are the unique tokens appearing within its occurrences in the corpus.

In the BoT matrix, primary features are implicit, while secondary features are explicitly represented. Therefore, we refer to secondary features simply as *features* in the following discussion.

Each element (d_i, f_j^k) of the BoT matrix represents the frequency of the j-th feature relative to the k-th primary feature in document d_i. Feature frequencies are normalized with respect to their corresponding primary feature, ensuring that the sum of all feature frequencies for a given primary feature equals 1.

Formally, let H be the matrix of absolute feature frequencies, with $h(d_i, f_j^k)$ denoting its elements. The elements of the BoT matrix, denoted by $h'(d_i, f_j^k)$, are computed as equation 1 and Algorithm 1 show.

$$h'(d_i, f_j^k) = \frac{h(d_i, f_j^k)}{\sum_s h(d_i, f_s^k)} \tag{1}$$

This normalization ensures that feature frequencies are relative to their respective primary feature distributions.

Algorithm 1 BOT algorithm

Require: $D = d_1, d_2, ..., d_N$ ▷ Documents
Require: $P_1, P_2, ..., P_M$ ▷ Primary features

Initialize h' with 0 values
for $i = 1$ to N **do**
  for $k = 1$ to M **do**
    $S_k \leftarrow$ all secondary features of type P_k in d_i
    for f_j in S_k **do**
      $h[i][f_j^k] \leftarrow$ frequency of f_j in d_i
      $h'[i][f_j^k] = \frac{h[i][f_j^k]}{|S_K|}$
    end for
  end for
end for
return(h') ▷ BOT matrix

D. Clustering performance evaluation

The clustering phase is performed using a clustering algorithm, whose quality has to be evaluated. Here we present a method for assessing cluster quality through extensive similarity evaluations, called *Optimized Similarity Clustering* (OSC). It is comparable to the Davies-Bouldin score and the Calinski-Harabasz index, as it emphasizes intra-cluster cohesion and inter-cluster separation. However, unlike the Davies-Bouldin score and the Calinski-Harabasz index, which rely on distance metrics, OSC utilizes similarity measures. This is particularly important because distance metrics can be ineffective for data with a high number of zeros. OSC evaluates clustering performance in three distinct phases.

In the first phase, *Internal Similarity* (*ISim*) is calculated in order to measure the cohesion within clusters. The proposed internal similarity measure *ISim* is defined in equation 2 for centroid-based clustering algorithms and equation 3 for non-centroid-based algorithms. The presence of clusters containing exactly one document is penalized, by using a penalization parameter $\alpha \in [0, 1]$ so that the presence of one-document clusters is taken into account or progressively ignored as α increases.

$$ISim = \frac{1}{N}\sum_{i=1}^{K}\left(\frac{1}{T_i}\sum_{j=1}^{T_i} sim(d_{j,i}, A_i)\right) + (1-\alpha)\frac{N-K}{N} \quad (2)$$

$$ISim = \frac{1}{N}\sum_{i=1}^{K}\left(\frac{1}{T_i}\sum_{1\le r<q\le T_i} sim(d_{r,i}, d_{q,i})\right) + (1-\alpha)\frac{N-K}{N} \quad (3)$$

Where

- *sim* is the cosine similarity in the range $[0, 1]$
- K is the number of clusters containing more than one document.
- N is the total number of clusters
- α is a penalty factor, in the range $[0, 1]$
- T_i is the number of documents in cluster (*i*)
- A_i is the centroid of cluster (*i*)
- $d_{j,i}$ is the j-th document in cluster (*i*)
- $d_{r,i}, d_{q,i}$ is the r-th and q-th documents in cluster (*i*), $d_{r,i} \neq d_{q,i}$

Note that $0 < ISim \leq 1$, since:
a) when all clusters contain more than one document ($K = N$) and have the best cohesion it holds that

$$ISim = \frac{1}{N}\left[\sum_{i=1}^{K}\left(\frac{1}{T_i}\sum_{j=1}^{T_i} sim(d_{j,i}, A_i)\right)\right] = \frac{N}{N} = 1$$

b) when all clusters contain exactly one document ($K = 0$) it holds that

$$ISim = (1-\alpha)\frac{N}{N} \leq 1$$

c) in all the other cases

$$\frac{1}{N}\left[\sum_{i=1}^{K}\left(\frac{1}{T_i}\sum_{j=1}^{T_i} sim(d_{j,i}, A_i)\right)\right] + (1-\alpha)\frac{N-K}{N} < 1$$

In the second phase, *External Similarity* (*ESim*) is computed in order to evaluate the separation between clusters as shown in equation 4 for centroids-based algorithms and equation 5 for non-centroids-based algorithms.

$$ESim = \frac{2}{N(N-1)}\sum_{i=1}^{N}\left(\sum_{j=i+1}^{N} sim(A_i, A_j)\right) \quad (4)$$

$$ESim = \frac{1}{P}\sum_{i=1}^{N}\sum_{k=1}^{T_i}\sum_{j=i+1}^{N}\left(\sum_{a=1}^{T_j} sim(d_{k,i}, d_{a,j})\right) \quad (5)$$

Where

- *sim* is the cosine similarity, in the range $[0, 1]$
- $d_{k,i}$ is the k-th document in cluster (*i*)
- $d_{a,j}$: The a-th document in cluster (j)
- P is the total number of pairwise similarities between documents in different clusters, calculated by $\sum_{1\le i<j\le N} T_i \cdot T_j$.

In the third phase, the Euclidean distance between the pair of internal and external similarity (*ISim*, *ESim*) and their optimal values $(1, 0)$ is calculated, as indicated in equation 6. This performance measure is called Optimized Similarity Culstering(OSC), and small values indicate better quality clusters.

$$OSC = \sqrt{((ISim - 1)^2 + (ESim - 0)^2)} \quad (6)$$

IV. Experiments

In order to evaluate the proposed method we run experiments on the biographies of Sudanese politicians, using articles written in Arabic, with one article for each person. The articles were extracted from Wikipedia Monolingual Corpora [13] using BaseX software. The tools used in this work include General Architecture for Text Engineering (GATE) [25] for text processing, Java Annotation Patterns Engine (JAPE) [25] for rules writing, BaseX [28] for preparing the dataset, and Rapid Miner [29] and Clustering Toolkit (CLUTO) [30] for clustering data.

A. Entities Extraction

Entities are the main topics mentioned in text documents; each domain has common entities. We studied the biographies dataset and determined six types of entities: Person, Location, Geo-political Entity (GPE), Organization, Facility, and Date. Entities were extracted from the dataset by annotating them, through a gazetteer (described in Section A) and a set of rules (in form of regular expressions).

The Gazetteer

In order to have a high-quality dataset for testing, we manually annotated entities. These manual annotations were subsequently used as a ground truth to evaluate other gazetteers. Table 2 shows an example of annotations, the corresponding entries in the gazetteer, and some possibilities for annotations ("will annotate" column) using the entries in the gazetteer. The goodness of a gazetteer is related to the number of annotation possibilities. In building our gazetteer we considered the following issues:

- Entries for Person have the granularity of tokens; in this way the annotation for a Person entity, that may consist of two tokens or more, can be correctly achieved with the aid of some combination rules. For example, (Mohammed Ahmed محمد احمد) can be annotated as Person by applying a rule that combines the two entries (Mohammed محمد) and (Ahmed احمد). This approach allows for a good flexibility since the possible combinations are recognized by rule application (in the example there are four possible combinations as shown in "will annotate" column in Table 2).

- Entries for Date are just names of days and months, then rules are used to construct some patterns in form of regular expressions.

- In some case lists of keywords are useful; for example, each token preceded by the keyword (City مدينه) is annotated as a Location entity.

B. Relations Extraction

Three types of relations are usually relevant in a biographies dataset: Birth (the birth of the person); Educational (educational stages of the person); Employment (the person's occupation, in terms of work or career). The first entity occurring in all types of relation is the Person; this entity may be mentioned explicitly or implicitly. The second entity in all types of relation may be Location, GPE, Organization, Facility, Date or any combination of them. We employed the rule-based method, which consists of two phases: pattern extraction and rule construction. Table 3 shows an example of the pattern and rule construction.

Patterns Extraction

All instances of the three relation types in our dataset were manually annotated. These annotations were later used to evaluate automatic relation extraction. To construct patterns for these relations, sample instances were necessary. Therefore, we selected a subset of the manual annotations (29 documents) for pattern construction and extracted multiple patterns for each relation type.

Rules Construction

Each pattern is converted to a rule. Some points were considered to generalize the rules:

- Keywords: different words are frequently repeated in the same type of relation. These words, having similar meanings within the same type of relation, can be used interchangeably. Hence, they are considered as keywords for those relations. Some keywords were found in the patterns, while others were added manually.

- Flexibility: some words in the patterns are not necessary for certain relation instances (i.e., entities, tokens, prepositions or punctuations). These words are considered as optional, increasing the number of extracted relations which leads to generalization of the rules.

- Entity patterns: As mentioned previously, the second argument in the relations may be a combination of the entities.

Table 2: Example of gazetteer entries

Entries in Gazetteer	Will Annotate
Khartoum city مدينه الخرطوم	1. Khartoum city مدينه الخرطوم
1. Mohammed محمد 2. Ahmed احمد	1. Mohammed محمد 2. Ahmed احمد 3. Mohammed Ahmed محمد احمد 4. Ahmed Mohammed احمد محمد

Table 3: Patterns and Rules Creation from Manual Annotation

Sentence already manually annotated	محمد تخرج من جامعة السودان Mohammed graduated from Sudan University
Pattern	{Person}{Token.string==”تخرجgraduated”} {Token.string==”منfrom”} {Facility}
Rule	Rule: Educational1 (({Person}{EduKey}({Token.string==”من”})?{EntitiesPattern})):tag −−> :tag.Educational = { Name = Educational, rule = Educational1}
Will annotate	1. محمد تخرج من جامعة السودان Mohammed graduated from Sudan University 2. علي انضم لجامعة السودان Ali joined Sudan University 3. علي انضم لجامعة السودان عام ١٩٨٠ Ali joined Sudan University in 1980 4. علي انضم لجامعة السودان بمدينة الخرطوم عام ١٩٨٠ Ali joined Sudan University in Khartoum in 1980

C. The Bag of Tokens

The primary features include all types of entity and relations in the corpus. In the considered dataset the above mentioned nine primary features were: P_1 = Person, P_2 = Location, P_3 = Gpe, P_4 = Organization, P_5 = Facility, P_6 = Date, P_7 = Birth, P_8 = Educational, P_9 = Employment.

As already mentioned, each primary feature P_k is expanded into m_k secondary features. The total number of features F is equal to $(m_1 + m_2 + \cdots + m_M)$, where in our case $M = 9$.

The number of secondary features in each P_j in the considered dataset was 474, 85, 472, 469, 205, 388, 170, 426 and 472 for Person, Gpe, Location, Organization, Facility, Date, Birth, Educational and Employment respectively.

An example of representing two documents d_1 and d_2, considering just ”Person”, is shown in Table 4 (frequencies) and Table 5 (normalized frequencies).

The values in Table 4 are normalized using equation 1, which involves dividing each frequency by the total sum of all features in ”Person”, which is 43. Document d_1 contains 19 occurrences of Person, and document d_2 contains 13 occurrences of Person. There are 27 unique tokens mentioned as part of Person occurrences in d_1 or d_2. These tokens represent the secondary features in Person, and are the columns of Table 4 and Table 5.

D. Clustering

In our experiments the two centroid-based algorithms K-means and K-medoids and the non-centroid-based algorithm Repeated Bisection Refinement(RBR) were considered. K-means and K-medoids were used with three distance measures, namely Euclidean, Manhattan and Cosine, while RBR relies solely on Cosine similarity [31]. Each algorithm was tested with a different number of clusters, ranging from 2 to 11. Both the K-means and K-medoids algorithms were configured to run 10 times, each starting with a different random initialization, each run allowing up to 100 optimization steps. In the RBR algorithm K clusters were constructed by $K - 1$ repeated bisections. The bisections aim to optimize a specific criterion function in [30].

V. RESULTS

Performance results were evaluated for each of the four main phases of the proposed system, namely entity extraction, relation extraction, document representation, and similarity calculation and clustering. We compared results obtained by BOT with results obtained by BOW, since the aim of the work was to improve the performance of BOW representation by adding semantics to it.

To validate the automatic annotations of entities and relations, ground-truth annotations are essential.

Table 4: Example of secondary feature representation in Person (two documents): absolute frequencies

	ابراهيم	احمد	الامام	الخالق	الرحمن	الرحيم	السيد	الصادق	الصديق	العزيز	الفاضل	الكريم	اله	المهدي	الهادي	جعفر	جناح	عبد	عبود	علي	فكتوريا	مبارك	محجوب	محمد	مصطفى	ميرغني	وامين
d_1	0	1	10	0	3	0	2	1	2	0	1	0	1	8	3	0	1	4	0	2	1	0	1	1	1	0	0
d_2	1	0	0	1	0	1	0	0	0	1	0	3	1	0	0	1	0	7	1	0	0	1	2	2	0	9	1

Table 5: Example of secondary feature representation in Person (two documents): relative frequencies

	ابراهيم	احمد	الامام	الخالق	الرحمن	الرحيم	السيد	الصادق	الصديق	العزيز	الفاضل	الكريم	اله	المهدي	الهادي	جعفر	جناح	عبد	عبود	علي	فكتوريا	مبارك	محجوب	محمد	مصطفى	ميرغني	وامين
d_1	0	0.02	0.2	0	0.07	0	0.05	0.03	0.05	0	0.02	0	0.02	0.2	0.07	0	0.02	0.09	0	0.05	0.02	0	0.02	0.02	0.02	0	0
d_2	0.02	0	0	0.02	0	0.02	0	0	0	0.02	0	0.07	0.02	0	0	0.02	0	0.2	0.02	0	0	0.02	0.05	0.05	0	0.2	0.02

The manual annotations serve two purposes: first, to build patterns and rules for relations; and second, to serve as ground truth in evaluating the automatic annotations. This manual annotation process is time-consuming and tiring, therefore a small number of documents in our corpus was used for evaluation, namely 81. The evaluation involves comparing the system annotations (annotation set B) with the manual annotations (annotation set A). The statistics for our dataset are shown in Table 6.

Table 6: Occurrences in the dataset used for testing

Document	81
Birth	82
Educational	163
Employment	635
Facility	156
Gpe	200
Location	515
Organisation	277
Person	984
Sentence	2052
Token	63884

There are five possible outcomes when comparing two sets of annotations.

- Matching with concordance (MC): the text has the same annotations in both sets A and B.
- Matching with disconcordance (MD): The text has two different annotations in both sets A and B.
- Only A (OA): The annotation is present only in set A.
- Only B (OB): The annotation is present only in set B.
- Overlapping (O): Two annotations of the same type share some text.

A confusion matrix was utilized to compare two sets of annotations. Three versions of the confusion matrix were computed: V_1, V_2, and V_3. The first version, V_1, evaluates all annotation comparison cases, namely MC, MD, OA, OB, and O. Both OA and OB are not considered in V_2 and V_3, and additionally in V_2 the case O is excluded.

Each version of the confusion matrix was assessed using four metrics: Precision (P), Recall (R), F-measure (F), and Cohen's kappa (C). The general equations for these metrics include a numerator comprising MC and O and a denominator consisting of MC, O, MD, OA, and OB. By excluding certain elements from the denominator, the values of the metrics (P, R, F, and C) can be increased. Consequently, V_1 yields lower scores compared to the later versions.

To calculate these metrics, two different methods were employed: the macro summary method, which gives more weight to high-frequency annotation types, and the micro summary method, which treats all annotation types equally. A single Cohen's kappa value was computed for each version of the confusion matrix.

A. *Evaluation of Entities Extraction*

Different gazetteers were compared to the manual annotations: GATE[25], WikiFan[32], NileAPg[33], ANERcorp[34], and our gazetteer. Table 7 shows the entity types in each gazetteer.

Micro-summary consistently outperformed macro-summary for all the considered gazetteers, except for ours that showed a very similar performance for the two metrics.

Table 7: Included entities in different gazetteers: P (Person), Gpe, L (Location), R (Organization), F (Facility) and D (Date). Used symbols: "+" if the entity is present in the gazetteer, "-" if it isn't.

Gazetteer	**P**	**Gpe**	**L**	**R**	**F**	**D**
GATE	+	+	+	+	+	+
WikiFANE	+	+	+	+	-	-
NileAP	+	+	-	-	-	-
ANER	+	-	+	-	-	-
Our gazetteer	+	+	+	+	+	+

The macro summaries from our gazetteer (across all versions and all metrics) outperformed those of other gazetteers, as illustrated in Figures 2, 3, and 4. The performance metrics of our gazetteer were closely aligned with those of the GATE gazetteer. In terms of micro summaries, both achieved the highest scores when considering confusion matrix versions V_2 and V_3. Considering the V_1 version, our gazetteer's showed the best performances.

B. *Evaluation of Relations Extraction*

The three types of relationships are compared in Figure 5. In Figure 6, micro summaries are contrasted with macro sum-

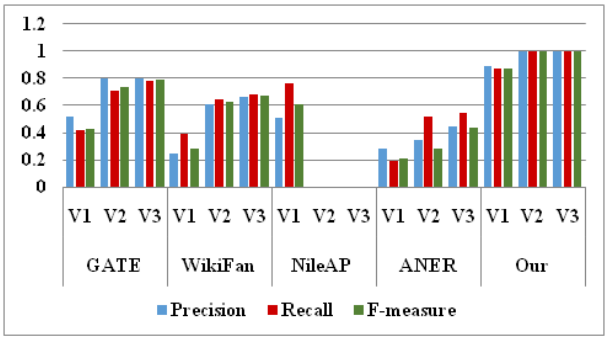

Figure 2: Macro Summary for Entities Extraction

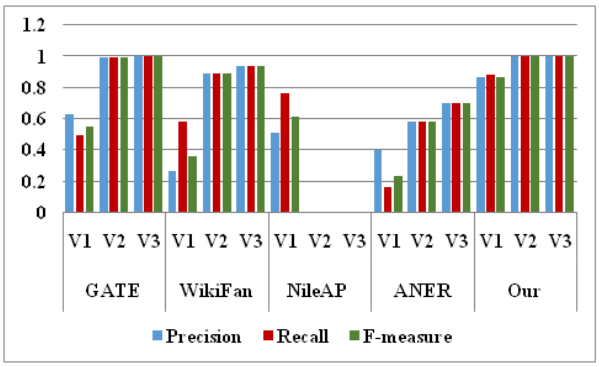

Figure 3: Micro Summary for Entities Extraction

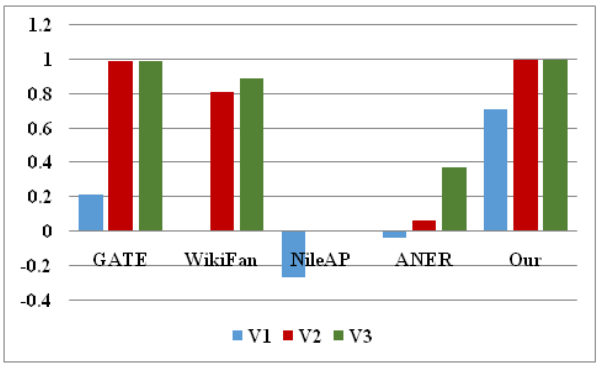

Figure 4: Cohen Kappa for Entities Extraction

maries. Figure 7 presents the evaluation of the extracted relationships using Cohen's Kappa statistic.

In the second and third versions of the confusion matrices, all the metrics have a value of 1 because these versions do not include the cases labeled OA and OB and the case MD is not found. The best results in the first version are observed in the Birth relation, with score 0.99 for all the considered metrics, followed by Educational relation with values between 0.84 and 0.86; Employment relation showed performances between 0.81 and 0.86.

Employment and Educational relations have the highest frequencies, so their impact on the micro summary is stronger if compared to the Birth relation. However, their performance metrics are lower than those of Birth. As a result, the considered performance measures (P, R, F_1) decrease from (0.88, 0.89, 0.89) in the macro summary to (0.83, 0.87, 0.85) in the micro summary.

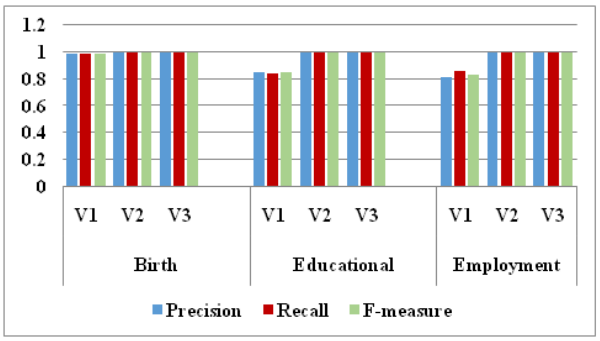

Figure 5: Results of Relations Extraction

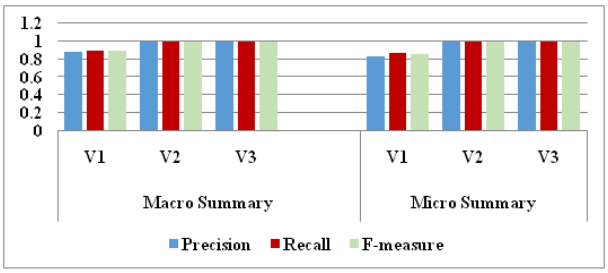

Figure 6: Macro vs. Micro in Relations Extraction

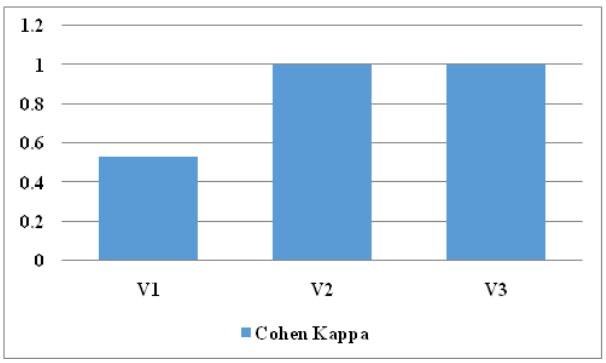

Figure 7: Cohen Kappa for Relations Extraction

C. *Evaluation of BOT*

We conducted a comparison between the BOT and BOW representations in three aspects: dimensionality, semantics, clustering cohesion and separation, as shown in Table 8.

The BOT model effectively reduced dimensionality associated with the BOW model by approximately 80%. This reduction in the number of features is due to the semantic-based

Table 8: Comparison between BOW and BOT document representations

	BOW	**BOT**
Dimensions	81×15845	81×3161
cell(i, j)	TF-IDF of term j in document i	Frequency of term j in document i
Semantic	Term-based.	Considers meaningful information (entities and relations).
Zero-valued cells	97.4%	96.68%

nature of the BOT model, in contrast to the term-based BOW model.

Both methods produce sparse data matrices, and the BOT matrix is a little less sparse than the BOW one. In the BOT model, there are $247,529$ zero-value cells (96.68%), while the BOW model contains $1,249,558$ zero-value cells (97.4%).

D. Evaluation of Clustering

Clustering results, in terms of clustering quality, were compared when using BOT or BOW representations.

Results for Centroid algorithms

The results of the OSC analysis using centroid-based algorithms are presented as a function of the number of clusters (with a fixed value for the the penalization parameter) and also as a function of the penalization parameter (with a fixed value for the number of clusters). Penalization of single-document clusters with the highest value ($\alpha = 1$) is shown in Figure 9, where BOT exhibits lower OSC values compared to the BOW model in most cases. This suggests that the BOT model has better clustering performance, as a lower OSC indicates higher intra-cluster similarity (*ISim*) and lower inter-cluster similarity (*ESim*). In contrast, when considering single-document clusters as all the other clusters ($\alpha = 0$, Figure 8), the best OSC values are achieved by BOT when using Cosine similarity. The BOW model outperforms BOT when using Euclidean and Manhattan distances. It is important to note that the low OSC values for the BOW model when $\alpha = 0$ do not accurately reflect its clustering performance. This discrepancy arises from the frequent occurrence of single-document clusters in the BOW model, particularly when using Euclidean and Manhattan distances. In these cases, one cluster contains the majority of the documents, while the other clusters consist mainly of single-document entries. This situation leads to high *ISim* values and artificially low OSC values. Therefore, the low OSC values for the BOW model at $\alpha = 0$ should not be interpreted as an indication of better clustering performance.

Figure 10 displays OSC values when $K = 11$, with OSC plotted against α in the range $[0, 1]$. The OSC remains stable across all values of α when single-document clusters are not present, as shown in some cases of Cosine similarity in Figure 10. In contrast, in the presence of the single-document clusters, OSC increases with higher values of α, which indicates more penalization. The OSC increases more significantly with increasing α values in the BOW model compared to the BOT model, which shows lower OSC values at the highest levels of α.

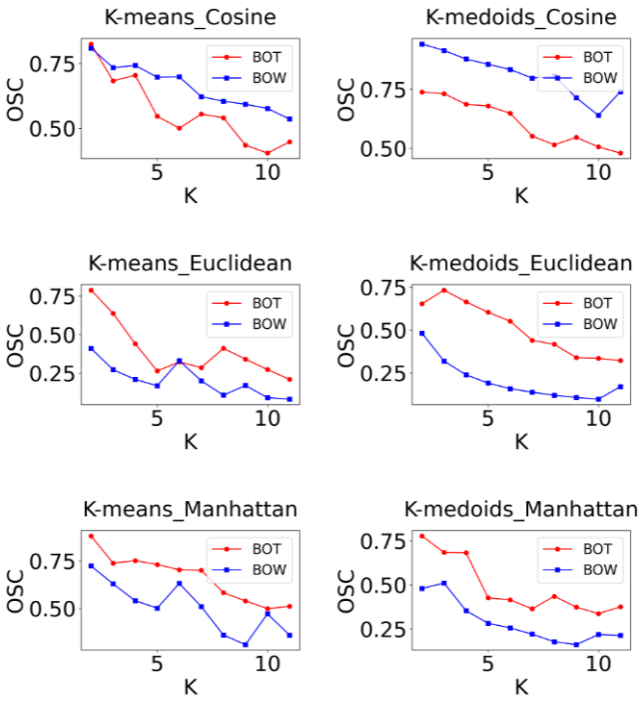

Figure 8: OSC for Centroid Algorithms with $\alpha = 0$

Results for non-Centroid algorithms

The non-centroid algorithm RBR performances assessed through OSC are depicted in Figures 11 and 12. This algorithm utilizes one of the seven optimization criteria as shown in Table 9.

Some criteria, such as L_1 and L_2, aim to maximize *ISim*, while others, like E_1, G_1, and G_1p, focus on minimizing *ESim* and thereby indirectly maximizing *ISim*. Additionally, criteria such as H_1 and H_2 adopt a hybrid approach by combining elements of L_1, L_2, and E_1. Certain criteria, including E_1 and G_1p, emphasize the contribution of larger clusters by incorporating cluster size into their calculations.

Both models exhibit a general decrease in OSC as K increases, except for L_1 and G_1, which show some fluctuations. Overall, BOT consistently outperforms BOW, maintaining lower OSC values across all K values. The results are similar across all criteria (except G_1) for the two extreme cases of α, primarily because single-document clusters are rare. However, one criterion, G_1, shows different behavior for α values; specifically, when $\alpha = 0$ and $K > 6$, a single-document cluster emerges, and BOW outperforms BOT in this case. Fluctuations are more pronounced in G_1 compared to other criteria, but the BOT model still performs better than the BOW model.

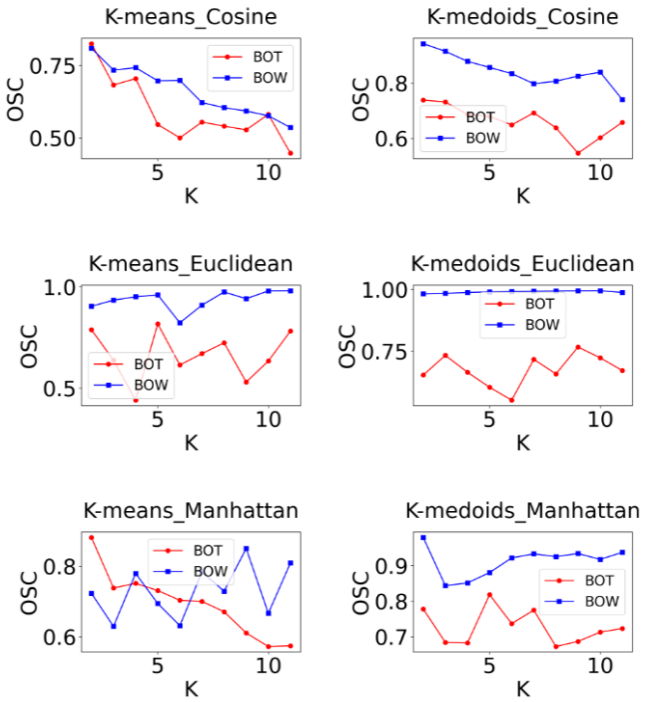

Figure 9: OSC for Centroid Algorithms with $\alpha = 1$

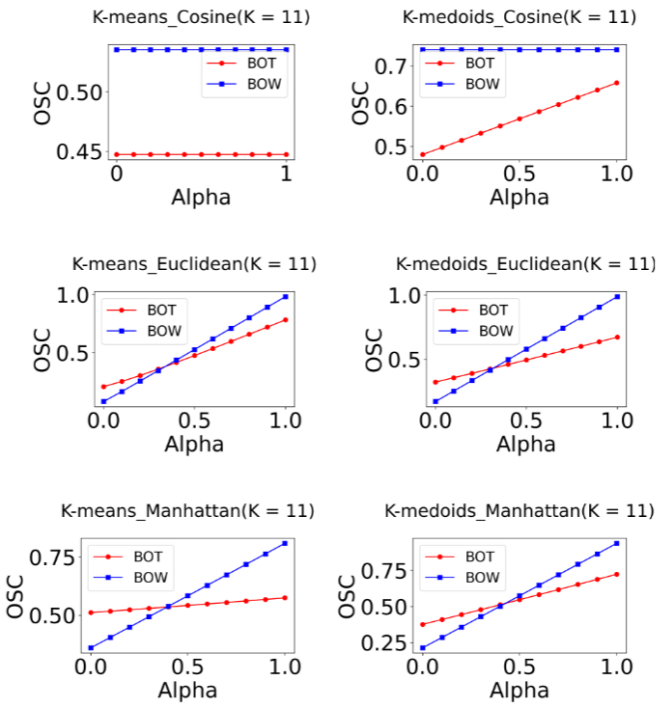

Figure 10: OSC with $K = 11$ for centroid algorithms and varying values of $alpha$

Single-document clusters are not common with the RBR algorithm, resulting in stable OSC across different values of α as shown in Figure 13, except for G_1, When they appear, the BOT demonstrates better OSC for higher values of α.

Table 9: The mathematical definition of clustering criterion functions in RBR. The notation in these equations is as follows: k is the total number of clusters, S is the total objects to be clustered, S_i is the set of objects assigned to the i-th cluster, n_i is the number of objects in the i-th cluster, and represent two objects, and sim(v, u) is the similarity between objects v and u. [30]

L_1	maximize $\sum_{i=1}^{k} \frac{1}{n_i} \left(\sum_{v,u \in S_i} \text{sim}(v,u) \right)$
L_2	maximize $\sum_{i=1}^{k} \sqrt{\sum_{v,u \in S_i} \text{sim}(v,u)}$
E_1	minimize $\sum_{i=1}^{k} n_i \frac{\sum_{v \in S_i, u \in S} \text{sim}(v,u)}{\sqrt{\sum_{v,u \in S_i} \text{sim}(v,u)}}$
G_1	minimize $\sum_{i=1}^{k} \frac{\sum_{v \in S_i, u \in S} \text{sim}(v,u)}{\sum_{u,v \in S_i} \text{sim}(v,u)}$
G'_1p	minimize $\sum_{i=1}^{k} n_i^2 \frac{\sum_{v \in S_i, u \in S} \text{sim}(v,u)}{\sum_{v,u \in S_i} \text{sim}(v,u)}$
H_1	maximize $\frac{L_1}{E_1}$
H_2	maximize $\frac{L_2}{E_1}$

VI. CONCLUSION

Clustering Arabic text documents remains an area that requires further exploration, and challenges in text document clustering are related to semantic representation and high dimensionality management. The traditional Bag of Word (BOW) model struggles with these issues, as it relies primarily on word occurrences without capturing underlying relationships. The proposed Bag of Tokens (BOT) model provides a more structured representation by focusing on entities and their relationships within documents rather than just individual word occurrences.

To evaluate the effectiveness of BOT, a comparative analysis was conducted using a manually annotated corpus of Arabic documents related to Sudanese politicians. Manual annotation was employed to ensure a more accurate comparison between the two models.

An alternative performance metric, Optimized Similarity Clustering (OSC), was introduced, based on Cosine similarity to better handle sparse datasets, and allowing for a penalization of single-element clusters. Results demonstrated that BOT effectively reduced the high dimensionality inherent in BOW and that it achieved better cluster coherence and separation across various clustering algorithms and distance metrics.

These findings highlight the advantages of integrating semantic information into document representation.

The presented study has to be considered as a pilot study, whose results are promising. Future work could further explore the applicability of BOT across different Arabic corpora and other languages to assess its broader impact.

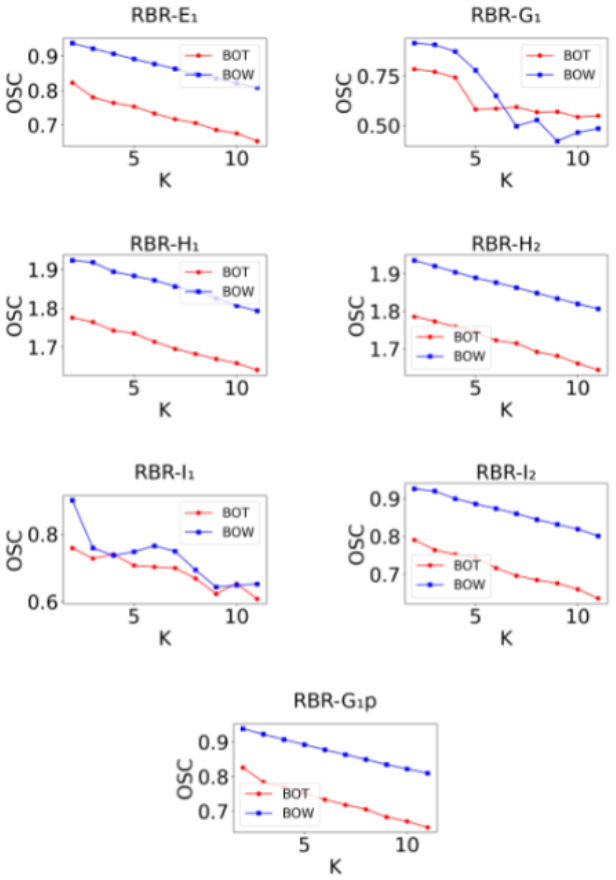

Figure 11: OSC with $alpha = 0$ for RBR using different criteria measures

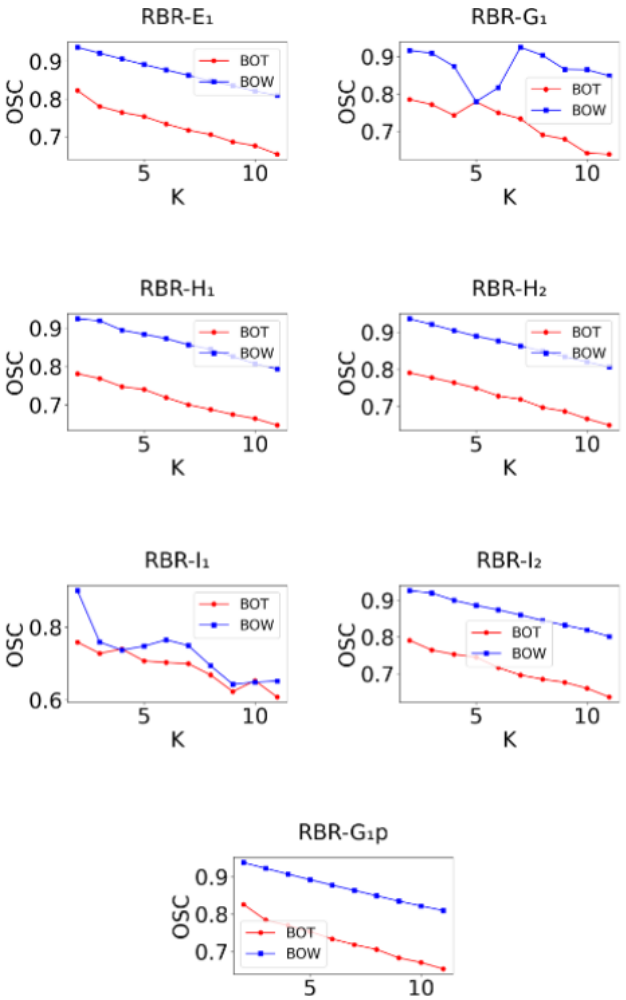

Figure 12: OSC with $alpha = 1$ for RBR using different criteria measures

VII. REFERENCES

[1] Fawaz S. Al-Anzi and Dia Abuzeina, "Stemming Impact on Arabic Text Categorization Performance : a Survey," *in 5th international conference on information & communication technology and accessibility (ICTA)*, 2015, pp. 1–7.

[2] Mostafa Sayed, Rashed K Salem, and Ayman E Khder. (2019). "A survey of Arabic text classification approaches." *International Journal of Computer Applications in Technology*. [Online]. 59(3), pp. 236–251. Available: https://www.inderscienceonline.com/doi/abs/10.1504/IJCAT.2019.098601.

[3] Yaser A Al-Lahham. (2021). "Index term selection heuristics for Arabic text retrieval." *Arabian Journal for Science and Engineering*. [Online]. 46(4), pp. 3345–3355. Available: https://link.springer.com/article/10.1007/s13369-020-05022-3.

[4] Moufeda Hussein, Abdelwahab Alsammak, and Tarek Elshishtawy, "Keyphrase-Based Hierarchical Clustering for Arabic Documents," *in Proceedings of the 10th International Conference on Informatics and Systems - INFOS '16*, 2016, pp. 61–67.

[5] Hanan M. Alghamdi, Ali Selamat, and Nor Shahriza Abdul Karim. (2014). "Improved text clustering using k-mean bayesian vectoriser." *Journal of Information & Knowledge Management*. [Online]. 13(03), pp. 1450026. Available: https://www.worldscientific.com/doi/abs/10.1142/s0219649214500269.

[6] Hanane Froud, Issam Sahmoudi, Abdelmonaime Lachkar, et al., "An efficient approach to improve Arabic documents clustering based on a new keyphrases extraction algorithm," *in Second International Conference on Advanced Information Technologies and Applications (ICAITA-2013)*, 2013, pp. 243–256.

[7] Hanane Froud, Abdelmonaime Lachkar, and Said Alaoui Ouatik. (2013). "Arabic text summarization based on latent semantic analysis to enhance arabic documents clustering." *arXiv preprint arXiv:1302.1612*. [Online]. Available: https://arxiv.org/abs/1302.1612.

[8] Osama A Ghanem and Wesam M Ashour. (2012). "Stemming Effectiveness in Clustering of Arabic Documents." *International Journal of Computer Applications*. [Online]. 49(5), pp. 1–6. Available: https://www.academia.edu/download/63671801/pxc3880674.pdf.

[9] Abdessalem Kelaiaia and Hayet Merouani. (2016). "Clustering with Probabilistic Topic Models on Arabic Texts: A Comparative Study of LDA and K-Means." *The International Arab Journal of Information Technology*. [Online]. 13(2), pp. 332–338. Available: https://www.ccis2k.org/iajit/PDF/Vol.13,%20No.2/6146.pdf.

[10] Arfat Awajan, "semantic vector space model for reducing arabic text dimensionality," *in 2015 Fifth International Conference on Digital Information and Communication Technology and its Applications (DICTAP)*, 2015, pp. 129–135.

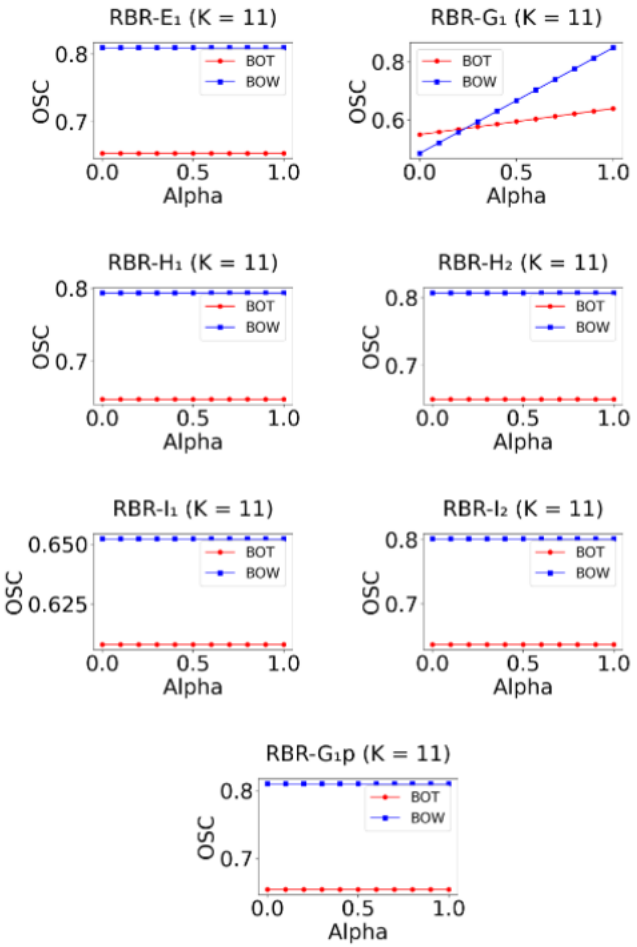

Figure 13: OSC with $K = 11$ for the RBR algorithm, varying alpha values on the X-axis from 0 to 1.

[11] Diab Abuaiadah. (2016, Dec.). "Using bisect k-means clustering technique in the analysis of Arabic documents." *ACM Transactions on Asian and Low-Resource Language Information Processing*. [Online]. 15(3), pp. 1–13. Available: https://dl.acm.org/doi/abs/10.1145/2812809.

[12] Amy Zhao Yu et al. (2016). "Pantheon 1 . 0 , a manually veri fi ed dataset of globally famous biographies." *Scientific data*. [Online]. 3(1), pp. 1–16. Available: https://www.nature.com/articles/sdata201575.

[13] Peter Kolb. (2018). "Wikipedia Monolingual Corpora–Linguatools." [Online]. Available: https://linguatools.org/tools/corpora/wikipedia-monolingual-corpora/.

[14] Rahhali Dounia, Khaider Yasssine, and En Nahnahi Noureddine, "Exploring text representation impact on K-means based arabic text documents clustering," *in 2022 International Conference on Intelligent Systems and Computer Vision, ISCV 2022*, 2022,

[15] Ksh Singh, H Mamata Devi, Anjana Kakoti Mahanta, et al. (2017). "Document representation techniques and their effect on the document Clustering and Classification: A Review." *International Journal of Advanced Research in Computer Science*. [Online]. 8(5), Available: https://www.researchgate.net/profile/Ksh-Singh-5/publication/356282949_Document_representation_techniques_and_their_effect_on_the_document_Clustering_and_Classification_A_Review/links/621af382579f1c04171d0045/Document-representation-techniques-and-their-effect-on-the-document-Clustering-and-Classification-A-Review.pdf.

[16] Marwah Alian and Arafat Awajan. (2020). "Semantic similarity for english and arabic texts: a review." *Journal of Information & Knowledge Management*. [Online]. 19(04), pp. 2050033. Available: https://www.worldscientific.com/doi/abs/10.1142/S0219649220500331.

[17] Niyaz Mohammed Salih and Karwan Jacksi. (2020). "State of the art document clustering algorithms based on semantic similarity." *Jurnal Informatika*. [Online]. 14(2), pp. 58–75. Available: https://pdfs.semanticscholar.org/3894/0587856efae6550b54a133ca4ebe141d1b18.pdf.

[18] Souad Larabi-Marie-Sainte, Mashael Bin Alamir, and Abdulmajeed Alameer. (2023). "Arabic Text Clustering Using Self-Organizing Maps and Grey Wolf Opimization." *Applied Sciences*. [Online]. 13(18), pp. 10168. Available: https://www.mdpi.com/2076-3417/13/18/10168.

[19] Khitam.A Salman and Hussein.K Khafaji. (2024). "A New Algorithm for Arabic Document Clustering Utilizing Maximal Wordsets." *Revue d'Intelligence Artificielle*. [Online]. 38(3), pp. 805–813. Available: https://www.researchgate.net/profile/Khitam-Salman-3/publication/381776623_A_New_Algorithm_for_Arabic_Document_Clustering_Utilizing_Maximal_Wordsets/links/6681cad5714e0b0315386040/A-New-Algorithm-for-Arabic-Document-Clustering-Utilizing-Maximal-Wordsets.pdf.

[20] Benjamin C.M. Fung, Ke Wang, and Martin Ester, "Hierarchical Document Clustering Using Frequent Itemsets," *in Proceedings of the 2003 SIAM international conference on data mining*, 2003, pp. 59–70.

[21] Haytham S Al-sarrayrih and Riyad Al-Shalabi, "Clustering arabic documents using frequent itemset-based hierarchical clustering with an N-grams," *in Proc. Int. Conf. Inf. Technol.(ICIT)*, 2009, pp. 113.

[22] A A Mohamed. (2020). "An effective dimension reduction algorithm for clustering Arabic text." *Egyptian Informatics Journal*. [Online]. 21(1), pp. 1–5. Available: https://doi.org/10.1016/j.eij.2019.05.002.

[23] Rowaida Khalil Ibrahim, Subhi Rafeeq Mohammed Zeebaree, and Karwan Fahmi Sami Jacksi. (2019). "Survey on semantic similarity based on document clustering." *Advances in Science, Technology and Engineering Systems*. [Online]. 4(5), pp. 115–122. Available: https://www.researchgate.net/profile/Subhi-Zeebaree/publication/335700827_Survey_on_Semantic_Similarity_Based_on_Document_Clustering/links/5e1b708d299bf10bc3a9056a/Survey-on-Semantic-Similarity-Based-on-Document-Clustering.pdf.

[24] Gehad Zakria et al., "Semantic Representation Extraction from Unstructured Arabic Text," *in Proceedings of the 8th International Conference on Software and Information Engineering*, 2019, pp. 222–226.

[25] Hamish Cunningham et al. *Developing Language Processing Components with GATE Version 8 (a User Guide) For GATE version 8.4.2-snapshot (development builds)*. South Yorkshire, UK: University of Sheffield., 2017. Available: https://gate.ac.uk.

[26] Hanan M. Alghamdi, Ali Selamat, and Nor Shahriza Abdul Karim. (2014). "Arabic web pages clustering and annotation using semantic class features." *Journal of King Saud University - Computer and Information Sciences*. [Online]. 26(4), pp. 388–397. Available: http://dx.doi.org/10.1016/j.jksuci.2014.06.002.

[27] Ebtihal Nooreldeen and Anna M. Zanaboni, "Classical Vector Space-based Semantic Representation and Dimensionality Reduction for Arabic text Clustering: a survey", Journal of Engineering and Computer Science, 2026, in press.

[28] BaseX. (2015). *BaseX Documentation*. Available: `http://docs.basex.org/wiki/Main_Page`. (accessed 3.8.20).

[29] Altair Engineering Inc. (2024). *Altair AI Studio*. Data science and machine learning platform.

[30] George Karypis. (2003). "CLUTO: A Clustering Toolkit." *University of Minnesota, Department of Computer Science*. [Online]. pp. 71. Available: `http://oai.dtic.mil/oai/oai?verb=getRecord%5C&metadataPrefix=html%5C&identifier=ADA439508`.

[31] Ricardo. Charu C. Aggarwal. Chandan K. Reddy Maronna. (2016). "Data clustering: algorithms and applications." *Statistical Papers*. [Online]. 57(2), pp. 565. Available: `https://www.proquest.com/openview/e6e82b87dfd872cbf404b9c81d04495d/1?pq-origsite=gscholar&cbl=31177`.

[32] Fahd Alotaibi and Mark G Lee, "Automatically Developing a Fine-grained Arabic Named Entity Corpus and Gazetteer by utilizing Wikipedia," *in Proceedings of the Sixth International Joint Conference on Natural Language Processing*, 2013, pp. 392–400.

[33] Omnia Zayed, Samhaa El-beltagy, and Osama Haggag, "An Approach for Extracting and Disambiguating Arabic Persons ' Names using Clustered Dictionaries and Scored Patterns," *in International Conference on Application of Natural Language to Information Systems*, 2013, pp. 201–212.

[34] Yassine Benajiba, Paolo Rosso, and José Miguel Benedi Ruiz, "ANERsys: An arabic named entity recognition system based on maximum entropy," *in International Conference on Intelligent Text Processing and Computational Linguistics*, 2007, pp. 143–153.