Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.050600

ARTICLE

Orbit Weighting Scheme in the Context of Vector Space Information Retrieval

Ahmad Ababneh[1], Yousef Sanjalawe[2], Salam Fraihat[3,*], Salam Al-E'mari[4] and Hamzah Alqudah[5]

[1]Department of Computer Science, Faculty of Information Technology, American University of Madaba, Amman, 11821, Jordan

[2]Department of Cybersecurity, Faculty of Information Technology, American University of Madaba, Amman, 11821, Jordan

[3]Artificial Intelligence Research Center (AIRC), College of Engineering and Information Technology, Ajman University, P.O. Box 346, Ajman, 13306, United Arab Emirates

[4]Information Security Department, Faculty of Information Technology, University of Petra, Amman, 11196, Jordan

[5]Department of Data Science and Artificial Intelligence, Faculty of Information Technology, American University of Madaba, Amman, 11821, Jordan

*Corresponding Author: Salam Fraihat. Email: s.fraihat@ajman.ac.ae

Received: 11 February 2024 Accepted: 05 June 2024 Published: 18 July 2024

ABSTRACT

This study introduces the Orbit Weighting Scheme (OWS), a novel approach aimed at enhancing the precision and efficiency of Vector Space information retrieval (IR) models, which have traditionally relied on weighting schemes like tf-idf and BM25. These conventional methods often struggle with accurately capturing document relevance, leading to inefficiencies in both retrieval performance and index size management. OWS proposes a dynamic weighting mechanism that evaluates the significance of terms based on their orbital position within the vector space, emphasizing term relationships and distribution patterns overlooked by existing models. Our research focuses on evaluating OWS's impact on model accuracy using Information Retrieval metrics like Recall, Precision, Interpolated Average Precision (IAP), and Mean Average Precision (MAP). Additionally, we assess OWS's effectiveness in reducing the inverted index size, crucial for model efficiency. We compare OWS-based retrieval models against others using different schemes, including tf-idf variations and BM25Delta. Results reveal OWS's superiority, achieving a 54% Recall and 81% MAP, and a notable 38% reduction in the inverted index size. This highlights OWS's potential in optimizing retrieval processes and underscores the need for further research in this underrepresented area to fully leverage OWS's capabilities in information retrieval methodologies.

KEYWORDS

Information retrieval; orbit weighting scheme; semantic text analysis; Tf-Idf weighting scheme; vector space model

1 Introduction

Information Retrieval (IR) models have long been at the heart of the modern information-driven world, this means that any improvement in the IR implies advancements in search engines, recommendation systems, and vast databases. Various IR models were experienced and developed in IR. The Vector Space Model (VSM) represents one of the significant matching and retrieval models in the IR field. It represents documents and queries as vectors in a multi-dimensional space, facilitating

the determination of relevance through spatial relationships such as cosine similarity. Impartment research efforts were proposed to enhance the VSM model, but there remains potential for enhancing its effectiveness through the integration of novel weighting schemes.

The Orbit Weighting Scheme (OWS) has emerged as a significant weighting mechanism within text mining [1,2]. The concept of "orbit weighting" stems from the idea of leveraging orbital dynamics, where entities rotate in a manner that defines their relationship with a central object. This approach can potentially introduce new dimensions of relationships beyond the linear dimensions commonly used in VSM. By viewing terms and documents as entities in orbit, we can introduce gravitational effects, velocities, and relative positional information, thus offering potentially richer representations.

The landscape of IR has been permanently evolving, looking for ever-more precise methods to determine the relevance of documents to user queries [3]. While traditional IR models like the VSM employ static representations of terms and documents, a more dynamic approach emerges with the concept of "orbit weighting." By visualizing terms and documents as entities revolving around a central theme or query focus, the orbit weighting methodology presents a departure from linear relevance evaluations. Instead, it offers a holistic view where relevance is determined not only by the proximity of a term or document to a query but also by its 'orbital' trajectory and interactions with other terms and documents.

Incorporating orbit weighting into IR systems could redefine the understanding of term-document relationships. Traditional term Frequency-Inverse Document Frequency (TF-IDF) models, for instance, assign static weights based on term occurrences. In contrast, an orbital approach could factor in the temporal evolution of a term's significance, its interplay with other terms, and its changing relevance over time, reflecting a document's evolving context. This dynamic representation might identify subtle nuances in relevance that static models might overlook, ensuring more contextually aligned results for user queries. The potential implications of applying orbit weighting in IR are vast. Not only could it lead to a more nuanced understanding of document relevance, but it could also offer insights into trends, emerging topics, and the ebb and flow of information themes over time. As IR systems strive for greater accuracy and context-awareness, the integration of such dynamic, orbital perspectives might well herald the next frontier in the quest for optimal search and retrieval mechanisms.

The OWS creates semantic spaces within texts, where each space encompasses verbs and nouns frequently appearing together. Within these spaces, nouns act as the center, while verbs revolve in predefined orbits. OWS prioritizes verbs by their proximity to the core noun and assigns those verbs appearing with various nouns in distinct semantic spaces to farther orbits. A pivotal aspect of OWS is its ability to omit terms found in distant orbits during the weighting process, categorizing them as stopwords. This method streamlines the efficiency by curbing the terms requiring weighting, thereby minimizing the inverted index size. Additionally, OWS addresses semantic challenges by identifying the text's inherent semantic spaces and sidelining universally occurring terms.

In this paper, we delve into the theoretical principles of the orbit weighting scheme, investigate its application in the VSM context, and examine its practical implications and advantages. Also, we provide compelling evidence of its benefits. We present the flowchart for computing the weight of term t in Fig. 1 and show visual and numerical results in Figs. 2–8. We summarize the specifications of the datasets in Table 1, and we show the impact of OWS on retrieval accuracy and efficiency in Tables 2–9. Three datasets from Arabic (Kalimat, 242 data) and English (Blog Authorship) were used to test the effectiveness of our methodology. Also, important IR relevancy measures (recall, precision, IAP, average IAP, MAP, ratio of reduction in the inverted index size, average retrieval time, and time and

space complexity) were collected from three different experiments. The achieved results validate the effectiveness of our OWS-based retrieval and show the superior performance of the OWS, as evidenced by a recall rate of 54%, a mean average precision of 81%, and a substantial reduction of 38% in the dimensions of the inverted index.

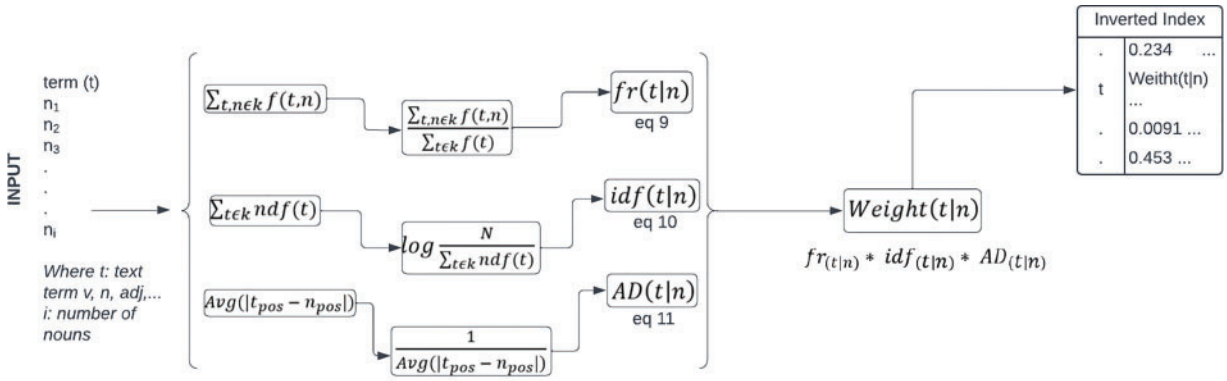


Figure 1: Flowchart for computing the weight of term t

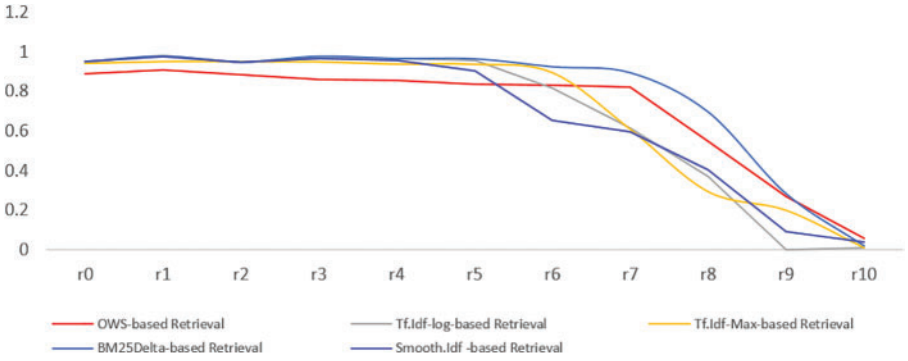


Figure 2: The recall-precision curves, Experiment 1

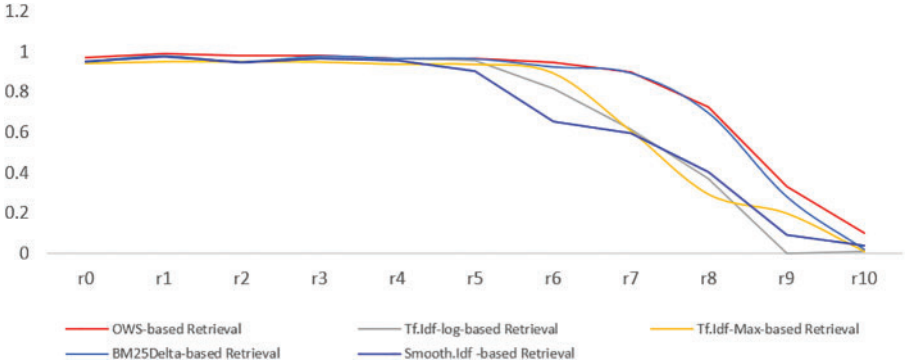


Figure 3: The recall-precision curves, Experiment 2

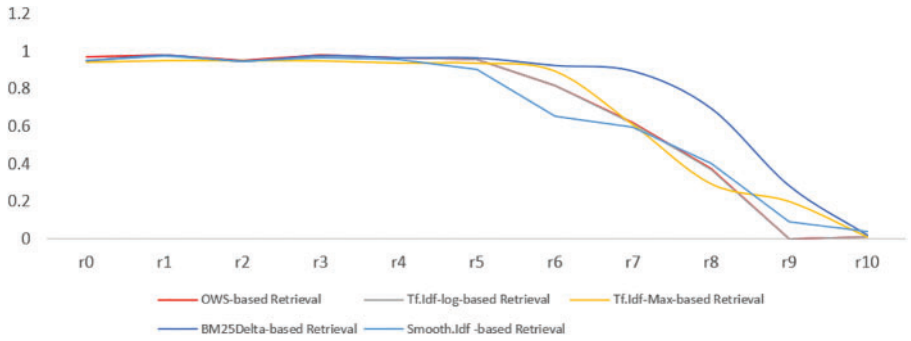


Figure 4: The recall-precision curves, Experiment 3

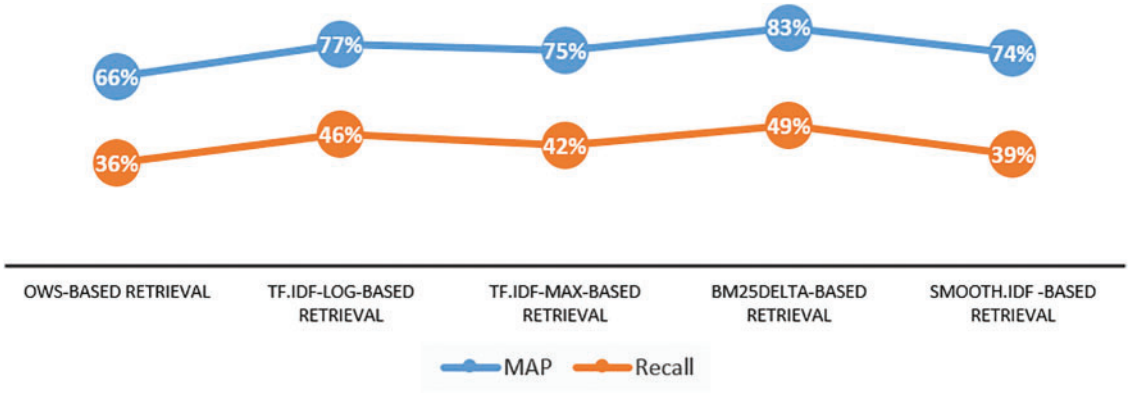


Figure 5: Recall and MAP final results, Experiment 1

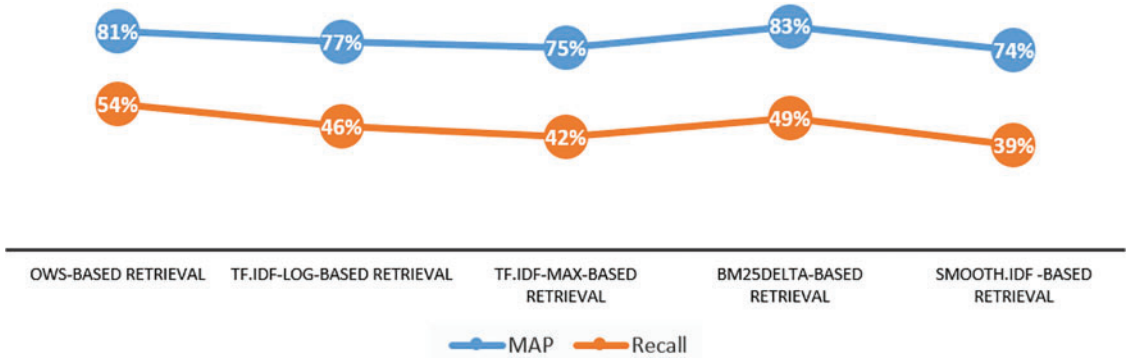


Figure 6: Recall and MAP final results, Experiment 2

Through extensive experimentation and analysis, we aim to establish the efficacy of the orbit weighting scheme in enhancing the retrieval performance of VSM-based systems. The significance of this research lies not only in enhancing the robustness of existing VSM-based systems but also in pioneering a new wave of thought around how spatial relationships can be reimagined and harnessed in IR. The fusion of orbital dynamics with VSM may pave the way for future methodologies that rethink how we perceive and utilize spatial relationships in computational models.

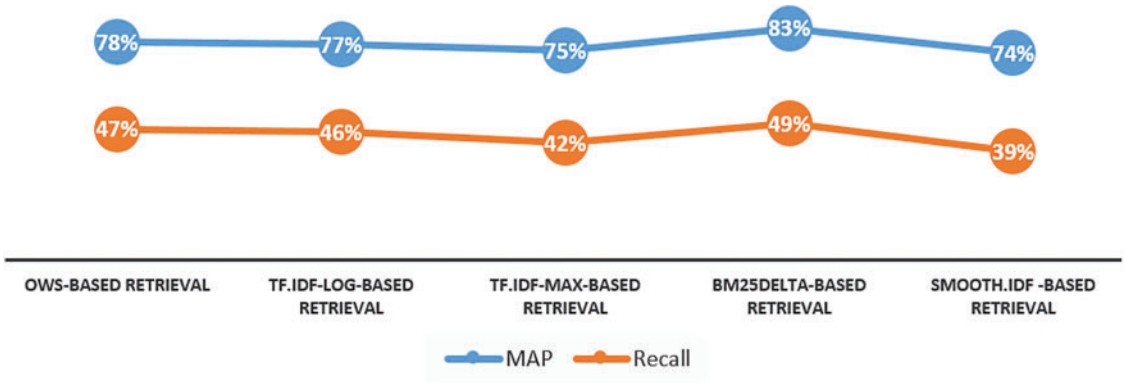


Figure 7: Recall and MAP final results, Experiment 3

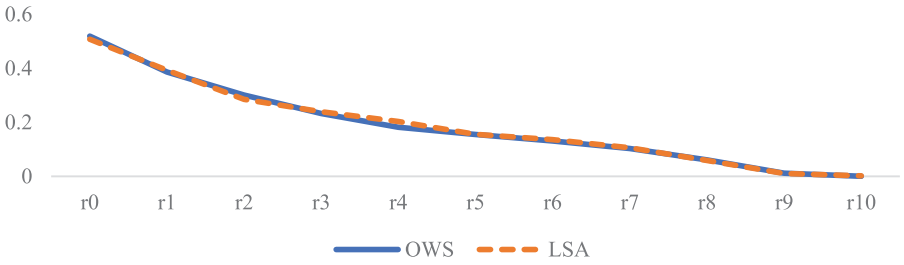


Figure 8: Recall-precision curves of the LSA based retrieval and OWS based retrieval of Experiment 1

Table 1: Specifications of benchmark datasets

Dataset	Number of documents	Language	Number of terms	Number of queries	Relevancy judgment
Kalimat data corpus*	20,290	Arabic	6,000,000	100	Automatic
242 data corpora	242	Arabic	20,000	60	Manual
Blog authorship	681,288	English	140,000,000	100	Automatic

Note: * The corpus is available free at http://www.lancaster.ac.uk/staff/elhaj/corpora.html (accessed on 23/04/2024).

Table 2: Query, documents similarity, q: "Second World War", Experiment 2

OWS-based retrieval		Tf.Idf-log-based retrieval		Tf.Idf-log-based retrieval		BM25 Delta-based retrieval		Smooth.Idf-based retrieval	
Doc id	Cosine Sim	Doc id	Cosine Sim	Doc id	Cosine Sim	Doc id	Cosine Sim	Doc id	Cosine Sim
370	0.204	689	0.205	370	0.204	689	0.213	689	0.208
689	0.202	370	0.201	689	0.202	264	0.195	370	0.191
332	0.194	332	0.182	332	0.194	332	0.194	332	0.188

(Continued)

Table 2 (continued)

OWS-based retrieval		Tf.Idf-log-based retrieval		Tf.Idf-log-based retrieval		BM25 Delta-based retrieval		Smooth.Idf-based retrieval	
Doc id	Cosine Sim	Doc id	Cosine Sim	Doc id	Cosine Sim	Doc id	Cosine Sim	Doc id	Cosine Sim
357	0.189	264	0.179	357	0.189	370	0.191	264	0.186
347	0.187	391	0.176	347	0.187	328	0.088	342	0.179
373	0.178	274	0.171	373	0.178	343	0.083	394	0.176
264	0.176	699	0.169	264	0.176	344	0.074	373	0.176
342	0.175	328	0.090	342	0.175	347	0.053	349	0.172
391	0.173	343	0.072	391	0.173	335	0.042	693	0.171
394	0.172	344	0.066	394	0.172	125	0.037	343	0.077

Table 3: The retrieved sets of documents of the query "The Second World War", Experiment 1

OWS-based retrieval	Tf.Idf-log-based retrieval	Tf.Idf-log-based retrieval	BM25 Delta-based retrieval	Smooth.Idf-based retrieval
370	689	689	689	370
689	370	370	370	689
332	332	332	332	332
357	264	264	264	357
347	391	391	342	347
373	274	274	394	373
264	699	699	373	264
342	328	328	349	342
391	343	343	693	391
394	344	344	343	394
699	125	347	328	699
692	373	692	344	349
693	65	380	347	693
274	380	12	692	274
343	12	110	335	343
328	110	106		328
344	224	13		344

Table 4: Query 5 recall/precision values, Experiment 3

Queryid	OWS			TfIdf-log			TfIdf-log			BM25Delta			SmoothIdf		
	Doc Ret	R	P	Doc Ret	R	P	Doc Ret	R	P	Doc Ret	R	P	Doc Ret	R	P
5	90	0.059	0.143	234	0.059	0.333	90	0.059	0.167	234	0.059	0.333	90	0.059	0.143
5	194	0.118	0.154	43	0.118	0.222	234	0.118	0.250	43	0.118	0.200	194	0.118	0.154
5	203	0.176	0.100	90	0.176	0.250	78	0.176	0.111	90	0.176	0.250	203	0.176	0.100
5	234	0.235	0.105	194	0.235	0.182	89	0.235	0.095	203	0.235	0.138	234	0.235	0.105
5	78	0.294	0.125	203	0.294	0.116	201	0.294	0.098	78	0.294	0.125	78	0.294	0.125

(Continued)

Table 4 (continued)

Queryid	OWS			TfIdf-log			TfIdf-log			BM25Delta			SmoothIdf		
	Doc Ret	R	P	Doc Ret	R	P	Doc Ret	R	P	Doc Ret	R	P	Doc Ret	R	P
5	201	0.353	0.136	78	0.353	0.120	236	0.353	0.109	89	0.353	0.143	201	0.353	0.136
5	188	0.412	0.149	89	0.412	0.125	188	0.412	0.113	194	0.412	0.163	188	0.412	0.149
5	89	0.471	0.157	201	0.471	0.121	203	0.471	0.127	188	0.471	0.154	89	0.471	0.157
5	236	0.529	0.101	188	0.529	0.122	187	0.529	0.138	201	0.529	0.134	236	0.529	0.101
5	187	0.588	0.109	187	0.588	0.119	82	0.588	0.137	236	0.588	0.122	187	0.588	0.109
5	82	0.647	0.112	82	0.647	0.112	194	0.647	0.145	187	0.647	0.125	82	0.647	0.112
5	43	0.706	0.120	236	0.706	0.117	43	0.706	0.141	82	0.706	0.135	43	0.706	0.120
5	79	0.765	0.112	79	0.765	0.106	200	0.765	0.098	79	0.765	0.100	79	0.765	0.112
5	217	0.824	0.095	217	0.824	0.081	79	0.824	0.103	217	0.824	0.085	217	0.824	0.095
5	200	0.882	0.082	200	0.882	0.075	217	0.882	0.099	200	0.882	0.083	200	0.882	0.082

Table 5: IAP of the query "شبكات الحاسب الآلي computer networks" in Experiment 2

Recall level	Precision OWS	Precision TfIdf-log	Precision TfIdf-max	Precision smoothIdf
r0	1	1	1	1
r1	0.9	0.833333	1	1
r2	0.645833	0.690476	0.928571	0.732143
r3	0.651515	0.683333	0.881944	0.788889
r4	0.527864	0.6125	0.766238	0.766234
r5	0.351073	0.29499	0.621345	0.575188
r6	0.220784	0.267829	0.578755	0.512213
r7	0.179173	0.18124	0.339286	0.485294
r8	0.168708	0.134724	0.155996	0.302569
r9	0.163261	0.124183	0	0.129252
r10	0	0	0	0.090909

Table 6: Average IAP for all queries, Experiment 2

Recall level	Precision OWS	Precision TfIdf-log	Precision TfIdf-Max	Precision BM25Delta	Precision smoothIdf
r0	0.969654233	0.949870881	0.939843232	0.950057656	0.950136999
r1	0.988762901	0.978816802	0.948815643	0.978894612	0.978628982
r2	0.983287557	0.948560034	0.948563298	0.94914406	0.948772514
r3	0.98220316	0.97730351	0.947735431	0.978083051	0.968228319
r4	0.966595183	0.966507019	0.936507449	0.966018111	0.957025263
r5	0.965077849	0.955434776	0.93543432	0.965143307	0.905574045
r6	0.947573112	0.816016105	0.89323	0.924974198	0.655087248
r7	0.898484746	0.617299396	0.607293221	0.894709619	0.59455581

(Continued)

Table 6 (continued)

Recall level	Precision OWS	Precision TfIdf-log	Precision TfIdf-Max	Precision BM25Delta	Precision smoothIdf
r8	0.724940841	0.372808419	0.292804443	0.695971459	0.403994767
r9	0.333323342	0	0.198893434	0.283223624	0.092457167
r10	0.102329822	0.009900099	0.009900099	0.01980198	0.03960396

Table 7: MAP, Experiment 2

MAP	OWS	TfIdf-log	TfIdf-Max	BM25Delta	SmoothIdf
	0.4673839	0.4200002	0.429321	0.4102211	0.4192292

Table 8: The ratio of reduction in the inverted index size. Experiments 1, 2, and 3

	2orbit-OWS	3orbit-OWS	4orbit-OWS	TfIdf-log	TfIdf-Max	BM25Delta	Smooth.Idf
Reduction	54%	38%	19%	0%	0%	0%	0%

Table 9: Average retrieval time (ms)/260 queries

	Exp1 (2orbit-OWS)	Exp2 (3orbit-OWS)	Exp3 (4orbit-OWS)
OWS index	33.34	49.231	58.65101
Tf.Idf-Log-index	76.098	76.098	76.098
Tf.Idf-Max-index	76.098	76.098	76.098
BMDeltaindex	76.098	76.098	76.098
Smoothindex	76.098	76.098	76.098

The contribution to this research can be summarized in two points:

- Changing the mode of the VSM from being a statistical-based to a semantic-based information retrieval model, by utilizing a semantic-based weighting scheme to weight the terms and create the necessary vectors.
- Increase the efficiency of retrieval by decreasing the size of the inverted index, which indicates a decrease in the number of terms used to represent the document.

In IR, it is paramount to employ renowned relevancy metrics to gauge the retrieval model's accuracy [4]. Metrics such as precision, recall, MAP, and IAP, among others, are indispensable. However, alongside these conventional measures, our research also monitors the size of the inverted index as a key indicator of the improvements achieved in retrieval efficiency.

To achieve our objectives, we used a research methodology that comprises three key phases: data preparation, model architecture, and performance evaluation.

- Data Preparation: This phase involves detailing the datasets utilized in the study. We outline the sources of the data set.
- Model Architecture: In this stage, we formulate the architectural and mathematical foundations of the OWS model. We provide insights into the design, structure, and algorithms of the model, elucidating its underlying principles.
- Performance Evaluation: The performance of the models is assessed using proposed evaluation formulas. This entails measuring the relevancy and efficiency of the models in achieving their objectives. We analyze the results obtained and evaluate the effectiveness of the models based on predetermined metrics.

The structure of this paper unfolds as follows: Section 2 delves into a comprehensive exploration of the OWS weighting scheme. Section 3 then presents an overview of other prominent weighting schemes within the realm of IR. The VSM model, serving as the backdrop where these weighting schemes are employed, is elucidated in Section 4. We conducted three distinct experiments as part of our research. The setup, outcomes, and insights from these experiments are elaborated in Sections 5. This paper is concluded in Section 6.

2 Research Background

The field of IR has witnessed significant advancements in recent years, with the development of novel techniques and methodologies aimed at enhancing the accuracy and efficiency of document retrieval systems [5]. Among these innovations, the OWS has emerged as a noteworthy concept, introducing a fresh perspective to the domain of vector space information retrieval. OWS represents a departure from conventional term weighting schemes, such as tf-idf, by introducing a unique approach that connects nouns to their semantic contexts through the examination of singular verbs. This literature review section delves into the extensive research conducted on OWS, exploring its application, implications, and its comparative performance against established models in information retrieval. Through this section, we aim to provide a comprehensive understanding of the OWS framework and its significance in the evolving landscape of information retrieval methodologies [6].

The OWS emerged in 2019 and stands as a pioneering framework in the realm of Natural Language Processing (NLP) and IR, offering a fresh and innovative approach to document representation and retrieval. Unlike traditional methods like tf-idf, OWS takes a unique perspective by bridging the gap between nouns and their semantic contexts, predominantly by scrutinizing singular verbs. The system employs three key parameters—Verb_Noun Frequency, Verb_Noun Distribution, and Verb_Noun Distance—to assign weights to terms, thereby reflecting their significance in a given document. This sophisticated scheme has garnered significant attention in the field, reflecting its potential to enhance the accuracy and efficiency of information retrieval systems [6].

OWS establishes associations between nouns and their semantic contexts by examining singular verbs within each context. The weight of a term is determined by taking into account three essential parameters: Verb_Noun Frequency, Verb_Noun Distribution, and Verb_Noun Distance. The Verb_Noun Distribution parameter is mathematically formulated to depict the semantic relationship between a noun and a specific set of verbs that exclusively appear in the context of that noun. The authors conducted a comparative evaluation of these novel models against well-established models in the field, including Skip-Gram, Continuous Bag of Words, and GloVe [7].

The Noun Based Distinctive Verbs (NBDV) is a synonyms extraction model developed in 2021 and based on OWS. The NBDV hired the innovative OWS as a replacement for the traditional tf-idf method. The NBDV model underwent testing in both Arabic and English languages, resulting in a 47% recall and 51% precision in dictionary-based assessment, as well as a 57.5% precision in evaluations performed by human experts. When compared to synonym extraction based on tf-idf, the NBDV achieved an 11% increase in recall and a 10% increase in precision. Regarding efficiency, our findings reveal that, on average, the extraction of synonyms for a single noun necessitates the consideration of 186 verbs, and in 63% of instances, the count of singular verbs was less than 200. It can be inferred that the developed method is efficient and processes a single operation in linear time [8].

A critical component of understanding OWS's significance lies in its comparative analysis against established information retrieval models. These comparisons have consistently demonstrated OWS's ability to outperform its counterparts, with improved recall, precision, and synonym extraction capabilities. This performance boost highlights OWS as a viable and promising alternative to conventional techniques, positioning it as a potential game-changer in the field. OWS leverages the power of semantic context, a critical facet of its significance in modern IR methodologies. By associating nouns with singular verbs within specific contexts, OWS creates a nuanced understanding of term importance that extends beyond traditional frequency-based metrics. This deepens the understanding of word relationships within documents, enabling more precise retrieval and enhancing the overall quality of search results. In a landscape where semantic understanding is paramount, OWS's focus on context-driven term weighting places it at the forefront of information retrieval advancements [9].

The applicability of OWS transcends language barriers, making it a versatile framework that has been successfully tested and implemented in both Arabic and English languages. OWS's ability to process information efficiently is another factor underlining its significance. With empirical data showing that synonym extraction for a single noun can often be accomplished with a limited number of singular verbs, OWS offers a time-efficient approach to information retrieval. In 63% of cases, this number remains below 200, indicating scalability and cost-effectiveness in large-scale retrieval systems. This efficiency not only contributes to improved retrieval times but also holds promise for applications in real-time and high-throughput environments, further solidifying OWS's importance in the evolving landscape of information retrieval methodologies [7,10].

Vector space retrieval methodologies encompass a range of approaches that offer unique perspectives on document representation and relevance judgment. The term frequency-inverse document frequency (tf-idf) model [11], for instance, calculates weights for terms based on their frequency in documents and their rarity across the entire document corpus. Therefore, the tf-idf gives the terms that are frequent within a document and rare in the corpus higher weights. Latent Semantic Analysis (LSA) uses the singular value decomposition factorization technique to capture the latent semantic relationships between the terms and documents [12]. The LSA allows for document similarity beyond literal term matches. Latent Dirichlet Allocation (LDA) [13], on the other hand, is a probabilistic generative model that assigns documents to topics and words to topics; this enables the discovery of latent thematic structures within a corpus. These established schemes provide valuable benchmarks for evaluating the effectiveness of OWS in enhancing retrieval accuracy and efficiency.

However, in contrast to traditional vector space retrieval schemes, such as tf-idf, LSA, and LDA, the OWS introduces a distinct methodology that bridges semantic contexts with document representation. While tf-idf primarily relies on term frequencies and document specificity, OWS emphasizes the relationship between nouns and their semantic contexts through the analysis of singular verbs. LSA neglects the order and syntactic structure of words and this limits its ability to

capture nuanced semantic relationships, also, LSA faces important challenges related to the time and space complexity when processing huge text [2]. The semantic-centric approach of OWS enables it to capture more nuanced semantic relationships within documents and offers a richer understanding of document content and relevance. By contextualizing OWS within the context of existing vector space retrieval methodologies, we can gain insights into its unique contributions and its implications for advancing information retrieval techniques.

The domain of OWS in the context of vector space information retrieval has seen limited scholarly attention, with relatively few studies conducted thus far. This scarcity of research in the area represents a compelling research gap that warrants exploration and investigation. The limited existing literature suggests that there is ample room for novel research to advance our understanding of the OWS and its applications in the context of vector space information retrieval. Addressing this research gap can provide valuable insights into the untapped potential of OWS, shed light on its effectiveness, and pave the way for new developments in information retrieval methodologies. By conducting further research in this domain, scholars have the opportunity to contribute to the evolution of information retrieval techniques and enhance the field's knowledge base.

3 Preliminaries

In this section, the innovative OWS and its intricate mechanisms are proposed in Subsection 3.1 while Subsection 3.2 provides an overview of other well-established weighting schemes that contribute to the evolving landscape of information retrieval methodologies. These subsections collectively expand our understanding of the complexities and nuances involved in term weighting and its impact on the effectiveness of information retrieval models.

3.1 Orbit Weighting Scheme

The Orbit Weighting Scheme is a promising weighting model in the field of text mining [1,3,7]. It links the nouns of a particular text to their semantic space by examining the singular verbs in each semantic context. The OWS tries to establish strong noun-verb relationships by adopting the traditional statistical parameters to discover this relationship. The weight of the term is determined by considering three parameters: VerbsNoun Frequency, VerbsNoun Distribution, and VerbsNoun Distance. The VerbsNoun Distribution parameter is mathematically formulated to depict the semantic relation between the noun and a certain set of verbs that only appear in the context of this noun. The OWS creates a vector for each noun n, the vector contents are numerical weights of the verbs that appear in the semantic context of the noun:

$$\vec{n} = \left(w_{v1,} w_{v2,} w_{v3,}, \ldots, w_{vi,}\right) \tag{1}$$

where $w_{v1,}$ is the weight of the verb *v1* in the space of (n), and (i) is the number of verbs that appeared in the semantic space of (n) in the whole corpus. Assume that (S_n) is a set of verbs that appeared in the semantic space of (n), such that: ($S_n = n_{v1}, n_{v2}, n_{v3}, \ldots, n_{vi}$), then, for each verb ($v$), the weight of ($v$) is determined by considering the following parameters:

- VerbsNoun Frequency ($fr_{(v|n)}$): the number of times the verb v appeared in the semantic space of n in the whole corpus. This parameter can be seen as the $f\ (v, n)$ that appears in the Pointwise Mutual Information (PMI) definition [14].
- VerbsNoun Distribution ($idf_{(n|v)}$): the number of noun spaces that contain the verb (v) in the whole corpus.

- VerbsNoun Distance ($AD_{(v|n)}$): the average distance between the verb v and the noun n in all the (v, n) occurrences in the semantic space of n.

To define the three parameters mathematically, assume that t refers to any term belonging to the space K, n refers to any noun that belongs to K, v refers to any verb that belongs to K, and N is the number of subspaces in K.

The **VerbsNoun Frequency** ($fr_{(v|n)}$) identifies the verbs that commonly appear with a specific noun. The $fr_{(v|n)}$ is computed as follows:

$$fr = \sum_{v,n\in k} f(v,n) \tag{2}$$

But some verbs are common and appear intensively, and others are specific and appear in certain domains and platforms, thus, we normalize the $fr_{(v|n)}$ by dividing it by the total number of times the verb (v) appeared in K:

$$fr\,(v|n) = \frac{\sum_{v,n\in k} f(v,n)}{\sum_{v\in k} f(v)} \tag{3}$$

The normalization degrades the importance of the common verbs because the denominator in Eq. (3) will be high for such verbs, and this decreases the weight and shifts the verb to outer orbits.

VerbNouns distribution is the number of orbiting spaces that contain the verb (v) in the whole corpus.

$$ndf\,(v) = \sum_{t\in k} f(n|v) \tag{4}$$

where fr(n|v) is the number of distinctive nouns that appeared with v. To dampen the effect of the $ndf\,(v)$, we normalized it as follows:

$$idf\,(n|v) = log\frac{N}{\sum_{t\in k} ndf\,(v)} \tag{5}$$

VerbNouns distance is the average distance between the verb v and the noun n in all occurrences of (v, n) in the semantic space of n.

$$AD\,(v) = \frac{1}{Avg\left(\left|v_{pos} - n_{pos}\right|\right)}, \forall f(v,n) \tag{6}$$

where v_{pos} is the position of the verb v, n_{pos} is the position of the noun n, and $f(v,n)$ represents any occurrence of v and n in the semantic space of n. After computing the value of each parameter, the weight of the verb v in the semantic space of n is computed, as shown in Eq. (7):

$$Weight\,(v|n) = fr_{(v|n)} * idf_{(n|v)} * AD_{(v|n)} \tag{7}$$

Eq. (7) summarizes the OWS weighting scheme, where $fr_{(v|n)}$ is the frequency of v with respect to n, $idf_{(n|v)}$ is the number of n with respect to v, and $AD_{(v|n)}$ is the average distance between v and n.

The philosophy behind the OWS is to picture the semantic meaning of a noun n as an orbiting space in which the noun is the main object (placed in the core), and the semantically related verbs are satellites that circulate in fixed orbits around the center. This picture assumes that the semantic meaning of a noun is determined by the set of verbs that always spin in the orbits of its semantic space. To specify the range of weights that should be included in each orbit. OWS assumes that all the weights

are located between the interval (*MINw*, *MAXw*). Therefore, the orbit range is assumed to be:

$$Orbit = \frac{MAXw - MINw}{y} \tag{8}$$

where y is the number of verbs that are weighted, $MAXw$ is the weight of the verb that appears in the inner orbit, and the $MINw$ is the weight of the verb that appears in the outer orbit. The number of orbits for semantic spaces is user-defined, and in this research, we tested the retrieval effectiveness at three orbit levels: two, three, and four. The founder of OWS used three as the number of orbits [1]. Accordingly, we consider three orbits as the baseline and test the richness of the semantic spaces given several orbits that are less than three and greater than three. The OWS weighs the verbs in the semantic space of a noun n and the other parts of speech, such as the adjectives, which cannot be manipulated by Eqs. (3), (5)–(7).

In this research, we propose to change Eqs. (3), (5)–(7) to process any term in the space of n (Fig. 1), this creates an adaptive OWS model that model any term t spans in the semantic space of n. the adaptive OWS uses the Eqs. (9)–(12) for all parts of speech:

$$fr\,(t|n) = \frac{\sum_{t,n\epsilon k} f(t,n)}{\sum_{t\epsilon k} f(t)} \tag{9}$$

$fr(t/n)$ represents a normalized term–noun frequency using the same logic of Eq. (3).

$$idf\,(t|n) = log\frac{N}{\sum_{t\epsilon k} ndf\,(t)} \tag{10}$$

$Idf\,(t|n)$ represents the number of orbiting spaces that contain the term (t), based on the same logic of Eq. (5).

$$AD\,(t|n) = \frac{1}{Avg(|t_{pos} - n_{pos}|)}, \forall f\,(t,n) \tag{11}$$

$AD(t|n)$ is the average distance between the term t and the noun n in all occurrences of (t, n) in the semantic space of n, based on the same logic of Eq. (6).

$$Weight\,(t|n) = fr_{(t|n)} * idf_{(t|n)} * AD_{(t|n)} \tag{12}$$

$Weight(t|n)$ is the weight of the term t in the semantic space of n, based on the same logic of Eq. (7).

The flowchart of Fig. 1 summarizes the main steps and actions necessary to weight t using the OWS based on Eqs. (9)–(11). The final output is the weight of the term t with respect to a noun space centered by n. Then, the computed weighted will be used as the weight entry of the term t in the inverted index.

The above mathematical interpretation shows the possibility of integrating the OWS based retrieval, with its orbital dynamics, in the context of the VSM-based information retrieval. In the operational framework of OWS, the semantic spaces for each noun support the terms weighing within the VSM. This conceptual framework enables the VSM model to interpret the weight assignments for the text terms within these semantic spaces as gravitational influences, noting that the closer terms to the nouns exert stronger influences on the central noun. Mathematically, parameters such as $fr(t|n)$, $idf\,(t|n)$, and $AD(t|n)$ quantify the strength and nature of these gravitational relationships and directly inform the weighting of terms within the VSM. To exactly map the three parameters to the context of the VSM-based retrieval, the OWS interprets them as follows:

- The $fr(t|n)$ or the noun-term frequency parameter reflects the frequency of interactions between a specific noun and its orbiting terms within the VSM space. The $fr(t|n)$ boosts the term frequency by giving it a semantic flavor and making it play a crucial role in the weighting scheme. For example, suppose we have a semantic space centered around the noun "computer". The fr(t|"computer") parameter indicates how certain terms like "software", "hardware", and "programming" appear in the space of "computer" (fr("software"|"computer."), fr("hardware"|"computer"), fr("programming"|"computer")). The fr("software"|"computer.") value was the highest and this suggests a strong association with the noun "computer" and reflects the significance of the term "software" in the context of computer-related documents.
- The $idf(t|n)$ or noun-term distribution parameter quantifies the extent to which a term is distributed across different semantic spaces within the VSM, analogous to the spread of influence across various dimensions of the vector space. For example, in the semantic spaces that focus on medical data. The idf(t|n) parameter indicates how certain terms like "Aspirin" are distributed across different spaces centered by "anatomy" "pharmacology" and "pathology"(idf("Aspirin"|"anatomy"), idf("Aspirin"|"pharmacology", and idf("Aspirin"|"pathology")). The high values of idf("Aspirin"|"anatomy") and idf("Aspirin"|"pathology") indicates that it is prevalent across various medical topics, while the low value of idf("Aspirin"|"pharmacology") suggests a more specialized distribution.
- The AD(t|n) or the noun-term distance parameter captures the spatial relationship between terms and nouns within the VSM, and this reflects the proximity of terms within the vector space and their impact on document retrieval. Consider a semantic space centered on the noun "environment". The AD(t|n) parameter captures the spatial relationship between certain terms like "pollution" and "biodiversity" with respect to the noun "environment" (AD(pollution|environment) and AD(biodiversity|environment)). The small value of AD (pollution|environment) compared to "biodiversity" signifies closer proximity to the central noun(environment) and indicates a stronger influence on the documents within the environmental contexts.

The theoretical foundation and mathematical formulations of the OWS in the operational framework provide a clearer understanding of how OWS enriches the representation of semantic relationships within the VSM, thereby enhancing the effectiveness of IR systems in capturing document relevance.

3.2 Other Weighting Schemes

As aforementioned, this subsection provides an overview of other well-established weighting schemes that contribute to the evolving landscape of information retrieval methodologies. The chosen weighting schemes are widely recognized in the field of information retrieval and have undergone extensive testing. They serve as reliable benchmarks for comparison [9,11,15–18].

3.2.1 Tf-Idf Weighting Scheme

In the field of IR, no standard weighting scheme is found [19]. A well-known weighting scheme was proposed by Salton et al. [11]. Salton's weighting scheme is called Tf-Idf, and it is still significant and appears in recent research efforts in the field of information retrieval and natural language processing [9,15]. In the Tf-Idf, the weight is computed using Eq. (13):

$$w_{t,d} = (1 + log f_t)\, log \frac{N}{df_t} \tag{13}$$

where $w_{t,d}$ is the weight of the term t in text d, df_t is the frequency of the term t in text d, df_t is the number of text segments containing t, and N is the number of text segments in the corpus (text segment could be a document or query). In Eq. (13), the df_t is normalized using the $(1 + logf_t)$ normalization scheme, another normalization scheme in Eq. (14), that provides significant weights based on Salton et al. [11], is to normalize the term frequency by the maximum term frequency of t in d.

$$f_{t,d} = 0.5 + \frac{0.5 \,.\, tf}{\max(tf)} \tag{14}$$

In the Tf.Idf weighting scheme, the term that frequently appears in a certain document, and is distributed over a few numbers of documents, takes more weight than the term that appears in every document. Thus, stopwords and common nouns and verbs that appear everywhere in the text, and do not represent concepts or topics, gain insignificant weights.

3.2.2 Smooth.Idf Weighting Scheme

Smooth.Idf (SIF) is an enhanced weighting scheme [16,17]. It computes the frequency of certain term t in document $d(f_t)$ relative to the total number of terms in the corpus (T). Eq. (15) has been used to calculate the term frequency for a certain term.

$$tf_t = \frac{f_t}{\mathrm{T}} \tag{15}$$

Then, the SIF computes the weight (w_t) of the term t based on Eq. (16):

$$w_t = \frac{a}{\mathrm{a} + tf_t} \tag{16}$$

where a is a parameter that has been generally set to 10^{-4}.

3.2.3 SMART and BM25 Delta Weighting Scheme

The BM25 weighting scheme is a probabilistic model that has been developed by Stephen Robertson (1971). The BM25Delta [18] is a variant of BM25 that uses more advanced term weighting methods. According to Paltoglou et al., the SMART and BM25Delta weight of term i is computed as shown in Eq. (17):

$$w_i = \frac{(k_1 + 1) \,.tf_i}{\mathrm{k} + tf_i} . \log\left(\frac{\left(N_{c1} - df_{i,c1} + 0.5\right) . \left(df_{i,c2} + 0.5\right)}{\left(N_{c2} - df_{i,c2} + 0.5\right) . \left(df_{i,c1} + 0.5\right)}\right) \tag{17}$$

where N_{c1} is the total number of training documents in class *c1* and $df_{i,c1}$ is the number of training documents in class $c1$ that contain term i, N_{c2} is the total number of training documents in class *c2* and $df_{i,c2}$ is the number of training documents in class *c2* that contain term i, and K is defined by the following formula:

$$k = k_1 (1 - b) + b\frac{dl}{\mathrm{avg\_dl}} \tag{18}$$

where *avg_dl* is the average number of terms in all documents, k_1 and b are tuning parameters of 1.2 and 0.95 values, respectively.

3.3 Proposed Model

The IR processes consist of several steps that include text preprocessing, statistical weight computations, inverted index construction, query-documents matching, and the generation of a ranked list of documents. In the proposed model, we built an IR system based on the VSM model, which was proposed in the field of IR by Salton et al. [11]. It is the most widely used model in information retrieval and natural language processing [20–22]. The retrieved set of documents from the VSM model is ranked according to the cosine similarity value. The VSM allows partial matching between the query and text documents, which adds more flexibility to the retrieval process.

The VSM is an algebraic model for matching documents and queries [19]. It has a robust mathematical foundation in which the documents and queries are depicted as vectors in multidimensional space. The components of each vector are a set of terms' weights that reflect the importance of these terms in the document, as follows:

$$\vec{d}_j = \left(w_{1,j}, w_{2,j}, w_{3,j}, \ldots, w_{n,j,}\right) \tag{19}$$

$$\vec{Q} = \left(w_{1,q,} w_{2,q}, w_{3,q}, \ldots, w_{n,q,}\right) \tag{20}$$

where $\vec{d}_j$ is the vector of document j in the collection, $w_{1,j}$ is the weight of the term 1 in document j, $\vec{Q}$ is the vector of query Q, and $w_{1,q}$ is the weight of term 1 in the query. After computing the weights and preparing the documents' vectors, VSM calculates the similarity between each document and the user query by computing the cosine of the angle between the vectors that represent them as shown in Eqs. (21) and (22) [23].

$$\text{sim}\left(\vec{d}_j, \vec{Q}\right) = \cos\left(\vec{d}_j, \vec{Q}\right) = \frac{\vec{d}_j.\vec{Q}}{|\vec{d}_j|.|\vec{Q}|} \tag{21}$$

$$\text{sim}\left(\vec{d}_j, \vec{Q}\right) = \frac{\sum_{i=1}^{t} w_{d_j}.w_{Q_i}}{\sqrt{\sum_{i=1}^{t} w_{d_j,i}^2}.\sqrt{\sum_{i=1}^{t} w_{Q_i}^2}} \tag{22}$$

where $\vec{d}_j$ is the vector of document j, $\vec{Q}$ is the vector of query Q, w_{d_j} is the weight of the term i in d_j, w_{q_i} is the weight of the term i in Q, and t is the number of terms in the whole corpus.

Eq. (23) is applicable for all the previously discussed weighting schemes except the OWS. The *OWS* computes the weights of the nouns and the terms that have distinctive semantic relations with each noun. This participates in reducing the length of the d_j vector. We reformulate the d_j vector in a new vector *OWS* d_j to discard the set of less significant verbs (from the point view of *OWS*):

$$OWSd_j = \left(w_{n1,} wt_{1|n1}, wt_{2|n1}, \ldots, wt_{k|1,} wt_{1|n2}, wt_{2|n2}, \ldots, wt_{h|n2}, \ldots, w_{nj}, wt_{1|nj}, wt_{2|nj}, \ldots, wt_{h|nj}\right) \tag{23}$$

where w_{nj} represents the weight of n_j and $wt_{h|nj}$ is the weight of the term t_h in the context of n_j. The length of $OWSd_j$ vector is smaller than the length of d_j vector because the OWS considers only the terms that circulate in the first three orbits of the semantic space of the noun n, this implies that the terms that have low weights (shifted the fourth orbit or above) will not be considered as part of $OWSd_j$ vector. In addition, representing the document vector using $OWSd_j$ reduces the size of the inverted index because the set of terms that need to be indexed for $OWSd_j$ vector is less than the set of terms of d_j vector. The change in vector representation will not affect the similarity computation, but a slight

change is required:

$$cos\,(\vec{OWSd_j}, \vec{Q}) = \frac{\sum_{i=1}^{s} w_{d_{j,i}} . w_{Q_i}}{\sqrt{\sum_{i=1}^{s} w_{d_{j,i}}^2} . \sqrt{\sum_{i=1}^{t} w_{Q_i}^2}} \tag{24}$$

where $\vec{d_j}$ is the vector of document j, $\vec{Q}$ is the vector of query Q, $w_{d_{j,i}}$ is the weight of the term i in $\vec{d_j}$, w_{Q_i} is the weight of the term i in Q, and s is the number of terms in $OWSd_j$ (s is less than t). In the IR model, we have introduced several enhancements and contributions to the traditional VSM, which is the cornerstone of information retrieval and natural language processing. We based our system on the VSM framework, originally proposed by Salton et al., which has been a fundamental model in this field. The VSM allows for partial matching between queries and documents, adding a layer of flexibility to the retrieval process, and making it more adaptive to various query types and information needs.

One of our significant contributions lies in adapting the VSM to accommodate the OWS, which is a novel approach to term weighting. OWS captures the semantic relationships between nouns and verbs in the text, resulting in a more precise and context-aware representation of documents [24]. By employing OWS, our model generates a more compact representation of documents, significantly reducing the vector dimensionality, and thus enhancing efficiency in indexing and retrieval without compromising retrieval quality. Furthermore, we refined the similarity computation process to match the changes introduced by OWS. This ensures that our model remains compatible with traditional term weighting schemes while taking advantage of the enhanced precision achieved by OWS. These adjustments maintain the core simplicity and effectiveness of the VSM while adding the nuance and power of OWS. Our contribution lies in the effective amalgamation of these elements, providing a balanced, robust, and efficient information retrieval framework for addressing modern information retrieval challenges.

Overall, the proposed model builds upon the established VSM foundation while introducing OWS and refining the retrieval process to create an enhanced information retrieval system that adapts to the growing complexities and demands of information retrieval in the digital age. This model opens up exciting possibilities for improved search results, streamlined document indexing, and optimized retrieval performance in various domains.

4 Experiments and Results

In this section, we delve into a comprehensive analysis through three experiments aimed at measuring the impact of the innovative OWS within the context of the VSM retrieval framework. Our primary objective is to assess how OWS-based retrieval stacks up against other conventional weighting schemes. These experiments provide valuable insights into the effectiveness of OWS when applied to different numbers of orbits in the semantic space of each noun.

4.1 Experimental Design

Experiment 1 undertakes a comparative evaluation by pitting OWS-based retrieval against other weighting schemes in a scenario where each noun operates within two orbits of semantic space. Experiment 2 extends the comparison to nouns existing within three orbits, while Experiment 3 takes it a step further with four orbits. The choice of the number of orbits in semantic space serves as a crucial variable that helps us explore the intricacies of the OWS approach and its adaptability. To facilitate these experiments, we created various indices, each tailored to a specific weighting scheme and

orbit configuration. These include the 2orbits-OWSindex, 3orbits-OWSindex, and 4orbits-OWSindex, which employ the OWS weighting scheme. Additionally, we constructed indices like Tf-Idf-Log-index, Tf-Idf-Max-index, BMDeltaindex, and Smoothindex, which employ different traditional weighting schemes. These meticulously designed experiments and indices form the foundation of our in-depth investigation into the efficacy of OWS within the VSM retrieval model.

In summary, three experiments were conducted to measure the effect of the OWS weighting scheme in the context of the VSM retrieval model. The three experiments are based on the following seven inverted indexes:

- 2orbits-OWSindex: a complete index. The weights of the terms in the index were determined using the OWS weighting scheme and the number of orbits is two (Eq. (12)).
- 3orbits-OWSindex: a complete index. The weights of the terms in the index were determined using the OWS weighting scheme and the number of orbits is three (Eq. (12)).
- 4orbits-OWSindex: a complete index. The weights of the terms in the index were determined using the OWS weighting scheme and the number of orbits is four (Eq. (12)).
- Tf.Idf-Log-index: a complete index. The weights of the terms in the index were determined using Tf.Idf weighting scheme with log normalization (Eq. (13)).
- Tf.Idf-Max-index: a complete index. The weights of the terms in the index were determined using Tf-Idf weighting scheme with max normalization (Eq. (14)).
- BMDeltaindex: a complete index. The weights of the terms in the index were determined using the BM25 Delta weighting scheme (Eq. (16)).
- Smoothindex: a complete index. The weights of the terms in the index were determined using the Smooth-Idf scheme (Eq. (17)).

4.2 Experiment Datasets

In this section, we turn our attention to the heart of our experimental analysis: the datasets that serve as the testing grounds for evaluating the various weighting schemes under scrutiny. In our quest to measure the impact of these schemes on relevance measurements, we have meticulously selected extensive datasets from both the Arabic and English languages. Several criteria were considered for choosing the datasets from Arabic and English languages:

1. The size of the selected datasets is suitable to test the effectiveness of the OWS weighting scheme in the information retrieval applications. Totally, the number of documents reaches 70,1820.
2. The datasets maintain diversity in the topics. The set of topics includes religious, financial, technological, sports, health, and political data. The diversity fosters the required variety to generate different semantics spaces.
3. The selected datasets commonly used in the field of text mining and information retrieval, and this facilitates the comparison with other models and ensure consistency in evaluation metrics [4,19–21].

The datasets play a pivotal role in our research and offer diverse perspectives on the effectiveness of different weighting strategies. The datasets, including Kalimat data corpus, 242 data corpus, and Blog Authorship, come with varying characteristics, from language and size to the source of relevancy judgment. This diversity allows us to conduct comprehensive and informative experiments that shed light on the performance of the discussed weighting schemes in distinct linguistic and contextual settings. Table 1 summarizes the specifications of these datasets.

In the experiments we used 260 queries, the queries are diverse in length and topics. For the 242 data corpus, the dataset includes 60 queries with a manual assessment of relevancy by the founder [2]. For Kalimat and Blog Authorship datasets, we formulated queries with different lengths; 2, 3, 4, and terms; and we considered the different topics of the datasets. Automatic and Manual relevancy for each query: the set of relevant documents is prepared manually for corpus 242. The founders of the corpus listed the set of documents that match each query (60 queries). For example, relevant documents for the query "جمعية الحاسبات السعودية Saudi Computers Association" were 45, 46, 47, 48, 57, 55, 56, 77, 78, 80, 204, 206, 207, 208, 209, 210, 211, 212, 214, and 230. The relevancy judgment has been determined automatically using full-text indexing over the VSM model. The model was implemented in [2] and obtained 95% MAP and 88% recall.

In our experiments, we considered several criteria to form the queries:

1. The query topics: the datasets cover a diversity of topics that include and not limited to religious, financial, technological, sports, health, and political topics. The queries cover all these topics; 43 queries for the religious topic, 49 for financial topic, 50 for technology topic, 29 sports topics, 44 for health topic, and 45 political topics. We tried to balance the numbers of queries as the datasets contains predominantly balanced number of documents for each topic.
2. The query language: we used dataset from Arabic and English languages, so we created 100 queries for English dataset, and 160 for the Arabic datasets. The number of queries was specified based on the dataset size.
3. Query length: we diversify the query length to be 2, 3, and 4 terms. The selection was based on important statistics mentioned on statista.com (accessed on 15/04/2023) that showed that the online query of length 2 words dominated 38%, 3 words 24.3%, and 4 words 11.5% of the users' queries. Also, another important statistic appeared in Semrush.com (accessed on 15/04/2023) that showed that most Google search queries are 3 to 4 words long.
4. Blind selection: to avoid human biased, we choose our queries for the English dataset and for the Kalimat dataset without investigating the documents contents. Regarding the 242-dataset the queries were manually prepared by the founder.

4.3 IR Relevancy Measures Used in the Experiments

Important and well-known relevancy measures were used in the three experiments. Let D_{rel} be the set of relevant documents of query q, D_{ret} be the set of retrieved documents, and D_{rr} be the set of relevant retrieved documents, then:

$$Precision_q = \frac{D_{rr}}{D_{ret}} \tag{25}$$

$$Recall_q = \frac{D_{rr}}{D_{rel}} \tag{26}$$

We consider precision and recall as unranked relevancy measures. If we need to be more accurate and measure the quality of the retrieved list of documents, we should use another set of relevancy measures such as the IAP. The IAP divides the recall into 11 recall levels and tries to answer the question, what is the precision when the recall level is r. The IAP is an effective tool for estimating the harmony between recall and precision. For example, if at recall level 0.5, the precision value was 0.9, this means that 90% of the retrieved list is relevant and at 90% precision, we retrieved 50% of the relevant document. The IAP is a significant indication of the retrieved list quality. If the precision value was high at a high recall level, this implies that the set of irrelevant documents in the retrieved list is small.

$$IAP = max_{i:s \leq recall_i \leq i+1} Precision \tag{27}$$

MAP is another ranked relevancy measure used in the IR field. To compute the MAP, we need to compute the precision pi when each relevant document is retrieved based on query q, and then take the average of pi values ($average_{pi}$). Then, the MAP is computed based on Eq. (28):

$$MAP = Average(average_{pi}), \forall queries \tag{28}$$

Also, the MAP gives significant induction about the quality of the retrieved list. Simply, in comparison to the relevant documents that are listed lower, the higher-ranking ones add more to the average. Besides the relevancy measures mentioned in this section, we considered the inverted index size and the retrieval time metrics. These measures help us to estimate the performance of the IR system. Regarding the inverted index size, we investigated the number of weighted terms in the inverted index and considered the amount of reduction achieved. Regarding the retrieval time, we averaged the response time for each run in each experiment.

4.4 Hardware and Software Specifications

The three experiments were executed on Intel (R) Core (TM) i5-7200U CPU @ 2.50 GHz 2.71 GHz, with installed RAM is 32 GB. Python is used to build the information retrieval application and the dataset documents, queries, and relevancy assessment were read from csv files. Hardware specifications can assist the reader in gauging the potential impact of delay time. The delay time of IR systems is influenced by factors such as hardware architecture, network configuration, and server load. However, since our experiments were conducted on a single device, we can disregard considerations related to network and server specifications.

4.5 Experiments Results

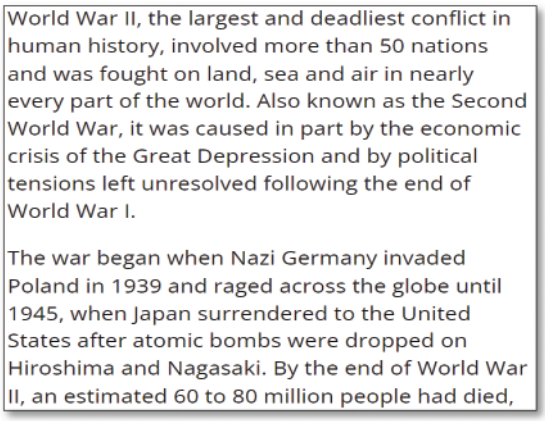
As previously mentioned, several experiments were carried out to determine the impact of various weighting schemes on the relevancy of the VSM information retrieval systems, and to compare the results of the traditional weighting schemes with those of the OWS weighting scheme. We gathered the following data from all the experiments to standardize how we evaluate the impact of these models, and the three experiments' outcomes have been combined for additional analysis and review. (Query, Documents) similarities: We gathered the similarity values between the query and the documents that had been represented in the seven indices. The similarity values were utilized to get the collection of relevant documents.

Table 2 presents the similarity values, in Experiment 2, between the query "Second World War" and the documents retrieved using the same VSM model but with different weighting scheme. Each row represents a document, and the columns correspond to different retrieval models, including OWS-based Retrieval, Tf.Idf-log-Based Retrieval, Tf.Idf-log-Based Retrieval, BM25 Delta-Based Retrieval, and Smooth.Idf-Based Retrieval. The cosine similarity values indicate the degree of similarity between the query and each document. The higher values indicate greater similarity. For example, in the OWS-based Retrieval, document 370 has the highest similarity value (0.204), which indicates a relatively strong relevance to the query. Similarly, in other retrieval models, different documents are ranked based on their cosine similarity values, reflecting their relevancy to the query.

A ranked retrieved list: the set of retrieved documents for each query is collected and ranked according to the similarity value between the query and the set of documents in the corpus. Such data facilitates precision and recall calculations. Table 3 shows a ranked list of documents that were retrieved for the query "The Second World War".

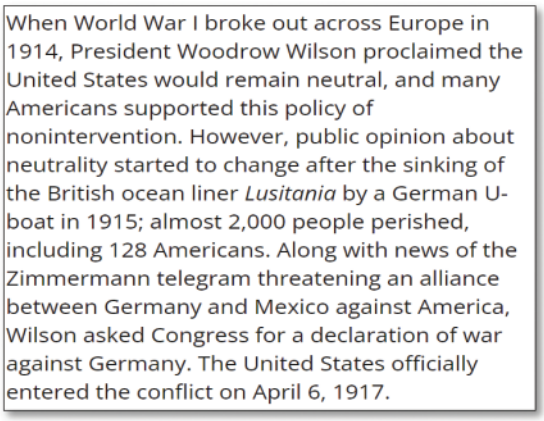
Table 3 provides a ranked list of retrieved documents for the query "Second World War" using the retrieval models in Experiment 2. The row represents a document and the columns correspond to different retrieval models, including OWS-based Retrieval, Tf.Idf-log-Based Retrieval, Tf.Idf-log-Based Retrieval, BM25 Delta-Based Retrieval, and Smooth.Idf-Based Retrieval. The documents are ranked based on their similarity values to the query. The most relevant documents appeared at the top of the list. For example, in the OWS-based Retrieval, document 370 is ranked first, which indicates that it is considered the most relevant document to the query based on the OWS weighting scheme. Conversely, in other retrieval models, such as Tf.Idf-log-based Retrieval, document 689 is ranked first, which suggests it is the most relevant document according to that weighting scheme. These ranked lists allow for the comparison of retrieval performance across different models proposed in this research and provide insights into the effectiveness of each weighting scheme in retrieving relevant documents.

In Tables 2 and 3, the OWS-based Retrieval puts the document 370 on the top of the retrieved list, with 0.204 similarity value. Document number 370 talks about the history of World War I and World War II. The other retrieval models put document 689 on the top of the retrieved list and the document mainly talks about the USA's participation in World War I.


World War II, the largest and deadliest conflict in human history, involved more than 50 nations and was fought on land, sea and air in nearly every part of the world. Also known as the Second World War, it was caused in part by the economic crisis of the Great Depression and by political tensions left unresolved following the end of World War I.

The war began when Nazi Germany invaded Poland in 1939 and raged across the globe until 1945, when Japan surrendered to the United States after atomic bombs were dropped on Hiroshima and Nagasaki. By the end of World War II, an estimated 60 to 80 million people had died,

Part of document 370


When World War I broke out across Europe in 1914, President Woodrow Wilson proclaimed the United States would remain neutral, and many Americans supported this policy of nonintervention. However, public opinion about neutrality started to change after the sinking of the British ocean liner *Lusitania* by a German U-boat in 1915; almost 2,000 people perished, including 128 Americans. Along with news of the Zimmermann telegram threatening an alliance between Germany and Mexico against America, Wilson asked Congress for a declaration of war against Germany. The United States officially entered the conflict on April 6, 1917.

Part of document 689

Precision values: We measured new precision values each time the model retrieved relevant documents. These precision values facilitate the IAP calculations and generate the recall-precision cure (Figs. 1–3). Table 4 shows the recall/precision values for the query "الأنظمة الخبيرة Expert Systems", in Experiment 3.

Interpolated Average Precision (IAP): The IPA traces the relevant retrieved documents in the retrieved list concerning 11 recall levels. In our experiments, we collected the IAPs for each query. Table 5 shows the IAP of the query "شبكات الحاسب الآلي Computer Networks" in Experiment 2.

In Table 5, the 90% precision value that emerged with r1 (precision of OWS-based retrieval) was attained when the recall value was more than or equal to 0.1 and less than 0.2. The average IAP of Experiment 2 for all queries is displayed in Table 6 (260 queries).

The Mean Average Precision (MAP): The MAP measures the quality of the retrieved set; if the retrieved set contains a sufficient number of relevant documents and the relevant documents appear

at the top of the retrieved list, then the value of the MAP will be high. Table 7 shows the MAP that was obtained in Experiment 2.

The number of terms in the inverted index: We measured the sizes of the inverted indexes created in the three experiments. The number of words that make up the inverted index was counted. We are interested in how the inverted index's size was reduced and how this reduction influenced the retrieval relevancy. However, because of the three weighting schemes, Tf.Idf, Smooth-Idf, and BM25Delta, process all the terms in the corpus, it is significant to observe that the sizes of the inverted indexes for them are identical (with 0% reduction). Table 8 shows that only the size of the inverted index of the OWS-based retrieval is reduced.

The average retrieval time: in the three experiments, we computed the average retrieval time for each experiment by dividing the total retrieval time by 260 (the number of queries). As shown in Table 9, the Tf.Idf-Log, Tf.Idf-Max, BMDelta, and Smooth based retrievals have identical retrieval times since the inverted index reduction for them is 0, and they all use the VSM retrieval model (note that the difference between these retrieval models lies in the term weights, not the retrieval model itself). The retrieval time is significantly improved in Experiment 1, and this comes in line with the 54% reduction in the inverted index size.

As we specified in the introduction section, the experimental evaluation aimed to assess the effectiveness of various weighting schemes (including the OWS) within VSM-based information retrieval. The results, as summarized in Tables 2 to 9, provide important performance insights of different weighting schemes across multiple efficiency and relevancy metrics. The analysis of document similarity revealed notable differences in the retrieval effectiveness of various weighting schemes. For instance, the OWS generated competitive similarity values, particularly in Experiment 2, where it outperformed traditional weighting schemes such as Tf.Idf-log and BM25 Delta-based retrievals. Furthermore, the retrieval rankings presented in Table 3 underscored the impact of weighting schemes on document relevance and showed that the OWS-based retrieval prioritized documents that have high relevancy to the query "Second World War". This observation suggests that OWS effectively captures the semantic relationships between terms and enhances document retrieval accuracy.

The evaluation of precision values and the IAP in Tables 4 to 7 provided further insights into the performance of weighting schemes across different recall levels. The findings indicate that OWS achieved competitive precision values, particularly in Experiment 3, where it exhibited superior recall-precision trade-offs compared to other weighting schemes. The inverted index size reduction and retrieval time in Tables 8 and 9 highlighted the efficiency gains achieved by OWS-based retrieval. The OWS-based retrieval demonstrated a significant reduction in inverted index size, which improved the retrieval time and enhanced the efficiency of information retrieval systems. Overall, the experimental results highlighted the effectiveness of OWS in improving document retrieval accuracy and efficiency within VSM-based information retrieval systems.

5 Analysis and Discussion

This segment of our study delves into the comprehensive evaluation of the OWS by comparing its relevancy and efficiency outcomes with those of established and widely used weighting schemes. Our analysis encompasses a spectrum of relevancy metrics, including recall, precision, interpolated average precision, and MAP, while also scrutinizing the size of the inverted index—a critical indicator of the system's efficiency. To gauge the practical impact of the OWS weighting scheme on IR system performance, we closely examine the degree of reduction in the inverted index size, connecting these findings with the recall-precision curve and the overall recall achieved for each weighting scheme. In

the context of relevancy, the discussion revolves around the outcomes observed in the three conducted experiments. Our presentation of recall-precision curves in Figs. 2–4, capturing precision behavior at 11 recall locations, reveals the distinctive performance of the OWS-based retrieval approach (depicted by the red curves) against other prevalent weighting methods (represented by other curves). Each figure corresponds to one of the three experiments (Experiments 1 to 3), offering a granular perspective on precision behavior.

5.1 Comparisons with Statistical Based Data Models

In this section, we evaluate the OWS based retrievals against the statistical techniques described in Section 3.2. Upon initial inspection, it becomes evident that all retrieval methods, including OWS, demonstrate an ability to retrieve a significant proportion of relevant documents, as indicated by the relatively marginal variations among the curves. However, when we delve into the specifics, Experiment 1, illustrated in Fig. 2, presents a notable scenario. Here, we employed a configuration with two orbits to represent the semantic space of each term, which substantially reduced the size of the inverted index (a remarkable 54% reduction, as presented in Table 8). It is at this juncture that a nuanced story emerges.

When examining the recall and MAP outcomes, as indicated in Fig. 5, the OWS-based retrieval appears to produce comparatively lower relevancy results, with recall and MAP values of 36% and 66%, respectively, trailing behind other established weighting schemes. The employment of two orbits effectively results in the exclusion of terms located in the third orbit and beyond, which inevitably reduces the inverted index's size. While this reduction may seem advantageous from an efficiency perspective, it ultimately compromises the index's informativeness. This trade-off emphasizes the nuanced dynamics involved in tuning the OWS weighting scheme and hints at the need for a more fine-tuned approach to achieve optimal results.

Fig. 3 unfolds the intricacies of Experiment 2, involving the utilization of three orbits for each noun in the OWS weighting scheme. This experiment brings to the forefront a fascinating narrative, one that revolves around the capacity of retrieval methods to maintain proximity to the ideal relevancy line, up to r7, corresponding to a 70% recall level. In this context, both OWS-based and BM25 Delta-based retrievals stand out as frontrunners, demonstrating the most significant relevancy outcomes. What's particularly intriguing is that the OWS-based and BM25 Delta-based retrievals exhibit noteworthy resilience, showcasing a relatively modest decline in precision beyond r8. To further illuminate these findings, Fig. 6 serves as a valuable companion, shedding light on the MAP and recall levels for each weighting scheme investigated in Experiment 2. The overarching trend reinforces the convergence in relevancy findings between the BM25Delta and OWS-based retrievals. Fig. 6 unveils a remarkable achievement by the OWS-based retrieval, attaining an impressive 81% MAP while successfully retrieving 54% of the entire relevant documents (recall = 54%). In tandem, the BM25 Delta-based retrieval secures an 83% MAP at a recall level of 49%. These outcomes cast a favorable light on the OWS-based retrieval, underscoring its remarkable performance, particularly when juxtaposed with the fact that OWS methodologies simultaneously managed to reduce the inverted index size by a substantial 38% (as evident in Table 8).

As we venture into Experiment 3, where the OWS weighting scheme was tested with four orbits, the relevancy outcomes in Figs. 4 and 7 reveal a distinctive narrative. Notably, these findings showcase a convergence with the Tf.Idf-log weighting schemes. This outcome is indeed rational, given that employing four orbits led to the reduction of a relatively small portion of the text (19%, as corroborated by Table 8). Furthermore, two key parameters of the OWS weighting scheme, namely ($fr_{(v|n)}$, $idf_{(n|v)}$), share analogous logic with Tf.Idf-log parameters, particularly pertaining to log normalization. Experiment

3 offers a unique perspective, suggesting that increasing the number of orbits might render the OWS weighting scheme more inclined toward a statistical approach, as opposed to purely semantic-based.

These results not only affirm the findings of OWS creators regarding the optimal number of orbits within the OWS weighting system, as documented by [4], but also provide tangible evidence supporting the utility of OWS, particularly in scenarios that demand three orbits. The implications of our study are far-reaching, with potential applications in domains such as text mining and information retrieval. The nuanced interplay between the number of orbits and the weighting scheme's performance underscores the flexibility and adaptability of the OWS model in various contexts.

The significance of this paper's contributions cannot be overstated, as they herald a transformative shift in the traditional VSM framework for information retrieval. Firstly, by introducing a semantic-based weighting scheme in place of the conventional statistical-based approach, the paper fundamentally alters the VSM's mode of operation. This shift marks a pivotal advancement, as it empowers the VSM to harness the latent semantic dimensions of text, thereby facilitating a more nuanced and context-aware understanding of documents and queries. This strategic move represents a pivotal step forward in improving the precision and recall of the information retrieval process. Secondly, the paper's success in reducing the size of the inverted index is a game-changer for the efficiency of retrieval systems. The diminished index size signifies a notable decrease in the number of terms required to represent each document, which, in turn, translates into reduced computational and storage overhead. This achievement underscores the practicality and scalability of the proposed model, making it a pioneering solution with the potential to revolutionize the landscape of information retrieval systems. Together, these contributions lay the foundation for a superior information retrieval framework that seamlessly balances precision, recall, and efficiency.

5.2 Comparisons with Semantic Based Data Models

In this section, we compared the relevancy outcomes of the OWS-based retrieval with other competing models that are capable to explore the semantic meaning of the text. Furthermore. It is important to mention that, in this part of our experiments, we used the same conditions and setting used in Experiment 1 since it obtained the most significant results in terms of relevancy and performance.

5.2.1 Latent Semantic Analysis

The Latent Semantic Analysis (LSA) is an NLP powerful technique to capture the semantic relationships between the text terms. The LSA is commonly used in several NLP tasks, such as information retrieval, document clustering, and text classification. The main contribution of the LSA in the NLP and IR fields is the ability to overcome the limitations of traditional weighting schemes by capturing the semantic structure of text [12].

The LSA analyzes the relationships between the terms and the documents and projects them into a lower-dimensional semantic space. It shares important idea with the OWS, the LSA assumes that the words that have similar meanings tend to appear in similar contexts within documents, but it uses the Singular Value Decomposition (SVD) technique to lower the dimensionality of the sematic space to acceptable dimensions. LSA suffers from two problems; the "bag of words" assumption, where the words' order and syntactic structure are disregarded, and the insignificant time and space complexity of processing huge text corpus [12]. Despite this limitation, LSA remains a powerful tool for capturing semantic relationships, and it can be employed to assess relevancy and performance in comparison to the OWS.

To perform the required comparison between the LSA and OWS, we created LSA based retrieval in which we used the same conditions and setting of Experiment 1. LSA is used to weight the text terms and the VSM model is used to perform the required matching and ranking, samples of the weights generated by the LSA is listed below:

Arabic term	Stem	W
عالج	علج	0.026047
دراسه	درس	0.040315
موضوعين	وضع	0.023493
هامين	هام	0.095853
مجال	جلي	0.030022
استخدامات	خدم	0.027608
تقنيه	قنن	0.030753
معلوماتيه	علم;	0.016933
خدمه	خدم	0.027608
ترجمه	ترجم	0.046586

Fig. 8 shows the Recall-Precision Curves of the LSA based retrieval and the OWS based retrieval with 2 orbits structure (Experiment 1). The curve shows the convergent relevancy results, and this proves the significant impact of OWS in the relevancy of IR systems. However, the time complexity of the OWS creates the difference, as shown in [6], the time complexity of OWS can be estimated by two parameters N (the number of nouns) and t (the total number of terms). It seems that in the worst case, the complexity will be O($t * N$), but the worst case only occurs if for every noun, all the other terms appear in its context space, and in human languages this thing will never happen. Considering the fact that the number of terms that could be found in the first and second orbit of the semantic space of the noun is small, this makes the number of terms tends to be constant number, leaving the complexity on O(N). Whereas, in [25], the authors showed the time complexity of the LSA is the minimum of $\{t^2d, td^2\}$ where t is the number of terms and d is the number of documents.

5.2.2 Word2Vec GloVe and FastText

After comparing the OWS-based retrieval with LSA-based retrievals that use the SVD factorization technique to generate semantic weights, we compare the OWS-based retrieval with competing models that feature semantic understanding and dynamic weighting mechanisms such as Word2Vec, GloVe, and FastText. Word2Vec, GloVe, and FastText are text representation models that are capable of creating vector representations for individual text terms. The generated vectors are known as word embeddings, and these embeddings represent the words in a continuous vector space where semantically similar words have similar vector representations.

Word2Vec is a popular word embedding model in natural language processing. The model generates dense vector representations (embeddings) for words in a continuous vector space. The model suggests that the meaning of a word can be inferred from its context. Word2Vec learns these representations by processing a large corpus of text data and adjusting the word vectors to predict the context of a target word within a given window size [26]. Global Vectors for Word Representation (GloVe) is another popular word embedding model that is designed to learn word

embeddings by leveraging global statistical information from a corpus of text data. The GloVe focuses on predicting context or target words based on local word co-occurrences and incorporates global word co-occurrence statistics into the training process. The GloVe was built based on the idea that word co-occurrence frequencies contain valuable information about the relationships between words and their meanings [27].

FastText is an extension of Word2Vec that introduces subword (or n-gram) information into word embeddings. It was designed to capture the morphological structure and meaning of words to solve the out-of-vocabulary word problem. FastText uses a shallow neural network to predict the probability of a word appearing in a context given its surrounding words or subwords [28]. We chose these three models to compare with because they are intensively used in literature and produced by reputable founders (Word2Vec produced by Google, GloVe produced by Stanford University, and FastText produced by Facebook's AI Research (FAIR) lab).

We implemented the three models using python as follows:

- Word2Vec: we used the Word2Vec model from gensim.models.
- GloVe: we used en_vectors_web_lg pre-trained GloVe vectors imported from Spacy. The pre-trained vectors are built for English words; therefore, we used Blog Authorship; the English language dataset for this part of our experiment.
- FastText: we used the FastText model from gensim.models.

To perform the required comparison between these three models and OWS, we created Word2Vec-based retrieval, GloVe-based retrieval, and FastText-based retrieval models, in which we used the same conditions and settings as in Experiment 1. However, as these three models find the vector representation of a single term, we need to aggregate the word vectors to their centroid to compute the term weight and then apply the cosine similarity [29]. In this part of our experiments, we chose to use Word centroid similarity to find the similarity between the query and the document because it aggregates the weights and then computes the cosine similarity, and this comes in harmony with our assumption of using the cosine similarity in the OWS-based retrieval.

Fig. 9 demonstrates that the Recall and MAP values for the OWS retrieval were significant and exceeded the performance of both Word2Vec and GloVe retrievals. Also, the figure reveals the OWS method approached the outcomes of the FastText retrieval model (4% difference in recall and 2% difference in precision), which exhibited the highest relevancy scores. Fig. 10 asserts the findings of Fig. 9, as the OWS recall-precision curve closely paralleled the FastText curve and surpassed the Word2Vec and GloVe retrievals.

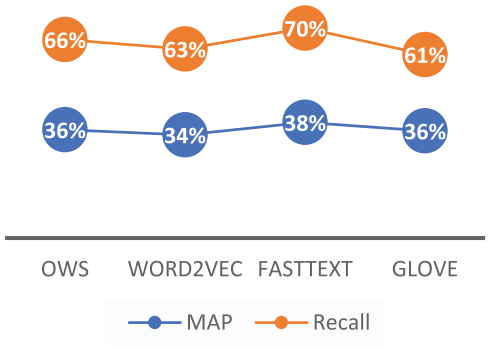


Figure 9: MAP: Recall-precision: OWS-based retrieval *vs.* Word2Vec, FastText, and GloVe based

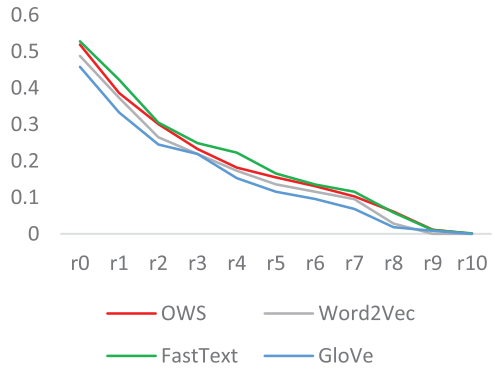


Figure 10: Recall-precision: OWS-based retrieval *vs.* Word2Vec, FastText, and GloVe based

5.3 Limitations and Challenges of OWS

Implementing the OWS in real-world IR systems holds significant promise for enhancing search precision and efficiency. It is essential to address the inherent challenges and limitations that accompany OWS. A primary concern is the model's reliance on statistical parameters, such as term frequency and inverse term frequency, adapted to probe semantic meanings without guaranteeing genuine semantic exploration. The weighting schemes, as discussed in Sections 3 and 5, are statistical and do not delve into the semantic substance of the text, necessitating a comparison with more semantically aware or dynamically adaptable weighting schemes.

Limitations:

1. Semantic Exploration: The main drawback of OWS lies in its statistical approach to semantic meaning, potentially missing deeper semantic connections.
2. Focus on Nouns: OWS's scheme exclusively focuses on nouns and ignores the other parts of text and this leads to an incomplete representation of the text.
3. Definition: The original definition of OWS investigates the semantic meaning of the noun based on the set of verbs that revolve in the semantic space of that noun without considering the other part of speech like adjectives.
4. Orbit Determination: There was no clear procedure or technique to determine the number of orbits, and because of this limitation, we were compelled to test three indexes in this research. This limitation is critical because increasing the number of orbits means increasing the number of terms to be processed, and this may negatively affect the time and space complexity of the OWS.

Challenges:

1. Integration with Existing Models: The potential difficulty of integrating OWS with models reliant on traditional Tf.Idf weighting, like the skip-gram or Continuous Bag-of-Words models, presents compatibility and optimization hurdles.
2. Computational Overhead: Implementing OWS can significantly increase computational demands due to its intricate processing of semantic relationships, necessitating robust computational resources.
3. Scalability: OWS's scalability is under scrutiny, especially in handling expansive datasets without compromising performance.

4. Domain and Language Adaptability: The limitations of the OWS degraded the adaptability to different domains and languages because they collectively limit the ability to accurately capture and represent the linguistic characteristics of the text.
5. User Interaction and Search Result Presentation: OWS's influence on how users interact with search systems and perceive results is a critical area for exploration. Adjustments in user interface design and result presentation strategies may be required to accommodate OWS's nuanced search outputs, ensuring clarity and user satisfaction.

The comprehensive evaluation of the OWS weighting scheme against the statistical and semantic-based models reveals significant findings.

1. The experimentation across three scenarios with varying numbers of orbits demonstrates the balance between efficiency and relevancy. We noted that reducing the number of orbits leads to efficient inverted index but comes at the cost of lowered recall and MAP values. However, as the number of orbits increases, the OWS model tends to converge with traditional statistical approaches and provides a dynamic balance between semantic exploration and computational efficiency. This highlights the flexibility of the OWS model and proves its potential in information retrieval.
2. The comparison with competing semantic-based models like LSA, Word2Vec, GloVe, and FastText proves the significance of OWS in achieving noteworthy relevancy outcomes. The OWS outperforms these models in recall and MAP values. However, challenges such as the integration with existing models, computational overhead, scalability, and domain adaptability signifies the need for further refinement and exploration.
3. The findings present a transformative shift in the traditional VSM framework and position the OWS as a pioneering solution that balances the relevancy and efficiency in information retrieval field.

5.4 Trending Topics

1. In the realm of image copy detection, recent advancements have focused on improving the accuracy of identifying illegal copies of copyrighted images. Notably, methods such as those described in [30] utilize a global context verification scheme to enhance the discriminability of local image features and reduce false matches—an approach that resonates with the foundational principles of our OWS in text-based information retrieval. Similar to the overlapping region-based global context descriptor (OR-GCD) used in image copy detection, which leverages rich global context information of SIFT features for more reliable verification, OWS introduces a dynamic, context-aware weighting mechanism that assesses the significance of terms based on their 'orbital' position within the vector space of a document. This similarity underscores a shared objective: both methodologies aim to enhance the precision of search and retrieval systems by incorporating a deeper understanding of the contextual relationships within the data—whether visual or textual. By adopting a similar global context-aware approach, OWS can potentially reduce inaccuracies in text retrieval that result from traditional static weighting models like tf-idf and BM25. Moreover, the efficient verification methods applied in image copy detection, such as the use of random verification for fast image similarity measurement, can inspire analogous strategies in OWS to accelerate the verification process in large-scale information retrieval systems without sacrificing accuracy. The integration of such global context verification strategies from image copy detection into text-based retrieval systems not only paves the way for cross-disciplinary innovations but also

enhances the robustness and efficiency of OWS. This could lead to significant improvements in how both partial and full duplicates are detected in textual data, drawing from the successes of their application in image processing."

2. In recent advancements in image steganography, a novel coverless approach has been developed in [31], where secret information is transmitted not by modifying an image, but by using natural image databases to find partial duplicates that can act as stego-images. This method, which divides database images into non-overlapping patches and indexes them based on extracted features, shares a conceptual similarity with our OWS in text-based information retrieval. Like the innovative steganographic technique that utilizes visually similar patches from natural images, OWS leverages semantic 'orbits' of terms within documents to enhance retrieval effectiveness without altering the original text data.

   In OWS, terms are weighted and positioned based on their semantic and contextual relevance to other terms, akin to how stego-images are selected based on the similarity of patches to parts of a secret image. This methodology ensures that the natural structure of the data, whether text or images, is preserved, thereby increasing resistance to analysis and detection, much like how coverless steganography evades traditional steganalysis tools. The principle of utilizing inherent patterns in data, without alteration, provides a robust framework for enhancing security and efficiency. In OWS, this approach could further be refined by adopting image steganography's techniques of feature extraction and partial matching, potentially increasing the accuracy and stealthiness of semantic search and retrieval in large databases. Moreover, the use of natural patterns for embedding or retrieving information can inspire new methods in information retrieval to ensure data integrity and prevent manipulation, drawing from steganography's goal to maintain the cover image's authenticity. Thus, the integration of concepts from image steganography into text-based information retrieval could open new avenues for creating more secure, efficient, and robust systems in handling and retrieving information across various digital platforms.

3. In the field of image search, particularly in handling partial-duplicate recognition, recent innovations have focused on enhancing the discriminability of local features through advanced verification schemes [31], such as the proposed region-level visual consistency verification. This method enhances the Bag-of-Visual-Words (BOW) model by identifying and verifying Visually Consistent Region (VCR) pairs between images, which significantly increases search accuracy and efficiency. This approach mirrors the principles behind our OWS in text-based information retrieval, where the focus is on improving the discriminability of semantic features within text documents.

   Just as the region-level verification scheme utilizes mapping and matching of local features to verify visual consistency, OWS employs a dynamic weighting mechanism that evaluates the significance of terms based on their semantic relationships and distribution patterns within the vector space. Both methods aim to address the challenges posed by traditional models—in the case of image search, the BOW model, and in text retrieval, static models like tf-idf and BM25. These traditional methods often fail to capture the nuanced relationships that are crucial for identifying partial duplicates or relevant documents accurately.

   Moreover, the integration of compact gradient descriptors and convolutional neural network descriptors for verifying visual consistency in image search can be paralleled with OWS's approach to using advanced semantic analysis techniques to assess term relevance more accurately. The proposal to use fast pruning algorithms to enhance search efficiency in image databases also aligns with similar strategies in OWS where less relevant or outlier terms

are dynamically deprioritized to streamline search processes. By exploring these analogous strategies between image and text retrieval systems, our research into OWS can benefit from understanding and possibly adapting image search's region-level verification techniques. This could lead to more sophisticated semantic parsing and matching algorithms that could further refine the accuracy and efficiency of information retrieval systems, especially in large-scale environments where partial duplicates or closely related documents need to be identified swiftly and accurately.

6 Conclusion

In this research, we embarked on an insightful journey to explore the practical implications of OWS within the Vector Space IR model. To substantiate our findings and gauge the real-world impact of these schemes, we conducted a series of rigorous experiments employing vast Arabic and English datasets, encompassing millions of terms and an array of 260 queries. Our investigations centered on the assessment of relevancy and efficiency, a pivotal facet of IR system performance. The experiments yielded compelling results that underscore the profound influence of OWS-based retrieval on precision and recall. This enhancement directly contributes to the overall effectiveness of IR systems, promising practical advantages in information retrieval tasks. Notably, OWS's inverted index showcased its efficiency, with a remarkable reduction in size—approximately 38% smaller than conventional document-based inverted indices. This size reduction translates to significant benefits, ranging from efficient memory and storage usage to faster query-term matching. Delving deeper into the heart of our findings, three key benefits came to the forefront. Firstly, OWS-based weighting schemes exhibited superior relevancy measures, outperforming traditional Tf.Idf-log, Tf.Idf-Max, and Smooth.Idf-based retrievals and semantic based retrieval using the LSA, GloVe, and Word2Vec based retrieval. These results underscore the merit of incorporating semantic dimensions into the weighting scheme to improve recall and mean average precision (MAP) in relevance evaluation.

Secondly, our experimentation revealed an efficiency improvement achieved through OWS weighting. It yielded a more concise and informative inverted index. Remarkably, even the three orbits OWS weighting technique demonstrated either equal or superior relevancy results compared to approaches that indexed the entire text. This gain in efficiency was achieved while significantly reducing the size of the inverted index by 38% and reducing the average retrieval time from 76.098 to 33.34 ms. Lastly, the delicate balance of orbit selection was unveiled through our experiments. We observed that careful tuning is essential for achieving optimal outcomes. Fewer orbits compromised the relevancy of the retrieved list, while an excessive number of orbits risked distorting the semantic dimensions intrinsic to OWS. These insights underscore the importance of fine-tuning OWS parameters for optimal performance in IR systems. In conclusion, the experiments affirm the compactness and informativeness of the inverted index constructed with the OWS model. The evidence, based on recall, precision, MAP, IAP, execution time, and the size of the inverted index, underscores the promising potential of OWS as a valuable tool for enhancing the efficacy of IR systems in practical applications.

In the future, we aim to explore several key research questions to deepen our understanding; here are specific research questions we plan to investigate:

- How effectively can OWS be integrated with other weighting schemes without compromising retrieval accuracy and efficiency across various NLP tasks?
- How does incorporating OWS into machine learning models, particularly those utilizing word embeddings like Word2Vec, GloVe, and FastText, enhance their semantic understanding and, consequently, their performance on NLP tasks?

- What are the optimal techniques for incorporating OWS into various AI and machine learning models to maximize retrieval accuracy and efficiency?
- How effectively can OWS be integrated into multilingual environments to provide a more comprehensive evaluation of OWS performance?
- How does the efficacy of OWS integration vary across different domains within text mining and NLP, such as legal documents, medical records, or social media content?

Acknowledgement: The authors appreciate Artificial Intelligence Research Center (AIRC), College of Engineering and Information Technology, Ajman University, Ajman and American University of Madaba, Jordan, for their administrative support.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm their contribution to the paper as follows: study conception and design: Ahmad Ababneh and Yousef Sanjalawe; data collection: Salam Fraihat and Salam Al-E'mari; analysis and interpretation of results: Ahmad Ababneh and Yousef Sanjalawe; data collection: Salam Fraihat, Salam Al-E'mari and Hamzah Alqudah; draft manuscript preparation: Ahmad Ababneh and Yousef Sanjalawe. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] A. H. Ababneh, J. Lu, and Q. Xu, "New synonyms extraction model based on a novel terms weighting scheme," *J. Inf. Organ. Sci.*, vol. 45, no. 1, pp. 171–221, 2021. doi: 10.31341/jios.45.1.9.

[2] A. Ababneh, "The enhancement of Arabic information retrieval using Arabic text summarization," Ph.D. dissertation, University of Huddersfield, 2019.

[3] A. Y. Nanehkaran, J. Chen, S. Salimi, and D. Zhang, "A pragmatic convolutional bagging ensemble learning for recognition of Farsi handwritten digits," *J. Supercomput.*, vol. 77, no. 11, pp. 13474–13493, 2020. doi: 10.1007/s11227-021-03822-4.

[4] J. Izquierdo-Domenech, J. Linares-Pellicer, and I. Ferri-Molla, "Virtual reality and language models, a new frontier in learning," *Int. J. Interact. Multimed. Artif. Intell.*, vol. 8, no. 5, pp. 46–54, 2024. doi: 10.9781/ijimai.2024.02.007.

[5] V. Rengasamy, T. Y. Fu, W. C. Lee, and K. Madduri, "Optimizing word2vec performance on multicore systems," in *Proc. Seventh Workshop Irregul. Appl.: Arch. Alg.*, 2017, pp. 1–9.

[6] J. Schler, M. Koppel, S. Argamon, and J. W. Pennebaker, "Effects of age and gender on blogging," in *AAAI Spring Symp.: Comput. Approaches Anal. Weblogs*, 2006, vol. 6, pp. 199–205.

[7] M. Mosbah, "A novel practical algorithm for strong and weak synonyms extraction with simple equality operation of web operational machine translation systems results," *Int. J. Knowl. Eng. Data Min.*, vol. 7, no. 3–4, pp. 271–285, 2022. doi: 10.1504/IJKEDM.2022.126072.

[8] A. E. Maredj, M. Sadallah, and N. Tonkin, "Enhancing multimedia document modeling through extended orbit-based rhetorical structure: An approach to media weighting for importance determination," *Knowl. Inf. Syst.*, vol. 66, no. 2, pp. 1–25, 2023. doi: 10.1007/s10115-023-01984-6.

[9] A. Onan and M. A. Toçoğlu, "A term weighted neural language model and stacked bidirectional LSTM based framework for sarcasm identification," *IEEE Access*, vol. 9, no. 1, pp. 7701–7722, 2021. doi: 10.1109/ACCESS.2021.3049734.
[10] B. Zhang, *et al.,* "Construction of English translation model based on neural network fuzzy semantic optimal control," *Comput. Intell. Neurosci.*, vol. 20, no. 1, pp. 17–23, 2022. doi: 10.1155/2022/9308236.
[11] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, no. 11, pp. 613–620, 1975. doi: 10.1145/361219.361220.
[12] C. Ormerod *et al.,* "Automated short answer scoring using an ensemble of neural networks and latent semantic analysis classifiers," *Int. J. Artif. Intell. Educ.*, vol. 33, no. 3, pp. 467–496, 2023. doi: 10.1007/s40593-022-00294-2.
[13] H. Jelodar *et al.,* "Latent Dirichlet allocation (LDA) and topic modeling: Models, applications, a survey," *Multimed. Tools Appl.*, vol. 78, no. 11, pp. 15169–15211, 2019. doi: 10.1007/s11042-018-6894-4.
[14] K. Church and P. Hanks, "Word association norms, mutual information, and lexicography," *Comput. Linguist.*, vol. 16, no. 1, pp. 22–29, 1990.
[15] F. Ghahramani, H. Tahayori, and A. Visconti, "Effects of central tendency measures on term weighting in textual information retrieval," *Soft Comput.*, vol. 25, no. 11, pp. 7341–7378, 2021. doi: 10.1007/s00500-021-05694-5.
[16] V. Gupta, A. K. Saw, P. P. Talukdar, and P. Netrapalli, "Unsupervised document representation using partition word-vectors averaging," in *ICLR, 2019 Conf.*, New Orleans, LA, USA, 2018.
[17] A. Onan, "Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks," *Concurr. Comput.: Prac. Exp.*, vol. 33, no. 23, pp. e5909, 2021. doi: 10.1002/cpe.5909.
[18] G. Paltoglou and M. Thelwall, "A study of information retrieval weighting schemes for sentiment analysis," in *Proc. 48th Ann. Meet. Assoc. Comput. Linguist.*, 2010, pp. 1386–1395.
[19] R. Baeza-Yates and B. Ribeiro, *Modern Information Retrieval*. Essex, England: ACM Press, 1999. Accessed: Dec. 21, 2023. [Online]. Available: https://web.cs.ucla.edu/~miodrag/cs259-security/baeza-yates99modern.pdf
[20] S. Dai, Q. Diao, and C. Zhou, "Performance comparison of language models for information retrieval," in *Art. Intell. Appl. Innov.: IFIP TC12 WG12. 5-Second IFIP Conf. Art. Intell. Appl. Innov. (AIAI2005)*, Beijing, China, Springer, 2005, vol. 2, pp. 721–730.
[21] J. N. Singh and S. K. Dwivedi, "A comparative study on approaches of vector space model in information retrieval," *Int. J. Comput. Appl.*, vol. 975, pp. 8887, 2013.
[22] S. Luo, Y. Wang, X. Feng, and Z. Hu, "A study of multi-label event types recognition on Chinese financial texts," in *Inf. Syst.: Res., Develop., Appl., Educ.: 11th SIGSAND/PLAIS EuroSymp. 2018*, Gdansk, Poland, Springer, Sep. 20, 2018, pp. 146–158.
[23] H. Schütze, C. D. Manning, and P. Raghavan, *Introduction to Information Retrieval*. Cambridge, UK: Cambridge University Press, 2008, vol. 39.
[24] E. Hanandeh, "Building an automatic thesaurus to enhance information retrieval," *Int. J. Comput. Sci. Issues (IJCSI)*, vol. 10, no. 1, pp. 676, 2013.
[25] Z. He, S. Deng, X. Xu, and J. Huang, "A fast greedy algorithm for outlier mining," in *Adv. Knowl. Discov. Data Mining: 10th Pacific-Asia Conf., PAKDD 2006*, Berlin Heidelberg, Springer, Apr. 9–12, 2006, pp. 567–576.
[26] M. Xue, "A text retrieval algorithm based on the hybrid LDA and Word2Vec model," in *2019 Int. Conf. Intell. Transp., Big Data Smart City (ICITBS)*, IEEE, 2019, pp. 373–376.
[27] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proc. 2014 Conf. Empir. Methods Nat. Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.
[28] I. Khasanah, "Sentiment classification using fasttext embedding and deep learning model," *Proc. Comput. Sci.*, vol. 189, pp. 343–350, 2021. doi: 10.1016/j.procs.2021.05.103.
[29] L. Galke, A. Saleh, and A. Scherp, "Word embeddings for practical information retrieval," in *Informatik 2017 Gesellschaft für Informatik*, 2017, pp. 2155–2167.

[30] Z. Zhou, Y. Wang, Q. Wu, C. Yang, and X. Sun, "Effective and efficient global context verification for image copy detection," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 1, pp. 48–63, 2016. doi: 10.1109/TIFS.2016.2601065.

[31] Z. Zhou, Y. Mu, and Q. Wu, "Coverless image steganography using partial-duplicate image retrieval," *Soft Comput.*, vol. 23, no. 13, pp. 4927–4938, 2019. doi: 10.1007/s00500-018-3151-8.