

## New Synonyms Extraction Model Based on a Novel Terms Weighting Scheme

**Ahmad Hussein Ababneh**

*ahmad.ababneh@aum.edu.jo*

*Computer Science Department*

*American University of Madaba, Madaba, Jordan*

**Joan Lu**

*J.Lu@hud.ac.uk*

*School of Computing and Engineering*

*University of Huddersfield, Huddersfield, UK*

**Qiang Xu**

*Q.Xu2@hud.ac.uk*

*School of Computing and Engineering*

*University of Huddersfield, Huddersfield, UK*

### Abstract

The traditional statistical approach in synonyms extraction is time-consuming. It is necessary to develop a new model to improve the efficiency and accuracy. This research presents a new model in synonyms extraction called Noun Based Distinctive Verbs (NBDV). During the documents' numerical representation phase, the NBDV replaces the traditional tf-idf weighting scheme with a novel weighting scheme called the Orbit Weighing Scheme (OWS). The OWS links the nouns to their semantic space by examining the singular verbs in each context. The weight of the term is determined by considering the three parameters: Verb\_Noun Frequency, Verb\_Noun Distribution, and Verb\_Noun Distance. The Verb\_Noun 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. We compared the new models with important models in the field, such as the Skip-Gram, the Continuous Bag of Words, and the GloVe model. The NBDV model was tested on both Arabic and English languages, and the results showed 47% recall and 51% precision in the dictionary-based evaluation and 57.5% precision in the human experts' evaluation. Comparing with the synonyms extraction based on tf.idf, the NBDV obtained 11% higher recall and 10% higher precision. Regarding the efficiency, we found that on average, the synonyms extraction of a single noun requires the process of 186 verbs, and in 63% of the runs; the number of singular verbs was less than 200. It is concluded that the developed method is efficient and processes the single run in linear time.

**Keywords:** Automatic Synonyms Extraction, Cosine Similarity, Orbit Weighting Scheme, Semantic Context Analysis, Vector Space-based Extraction.

## 1. Introduction

The languages are rich in synonyms or semantically related words. This merit gives the writer the ability to describe the same entity using different words and yields interesting and more vivid text [1]. However, the existence of synonyms confuses the text mining systems that employ the exact matching approaches such as the Boolean models or the statistical models [2]. Therefore, in the text mining fields, it is necessary to develop Automatic Synonyms Extraction (ASE) systems to identify the synonyms for the text mining applications.

In many Natural Language Processing (NLP) and Information Retrieval (IR) publications, the semantic investigation of the text contents was improved by hiring a semantic dictionary (such as the synonyms dictionary) in the investigation process [3], [4], [5]. The semantic dictionaries, with what they contain of synonyms, are valuable tools. These tools improve the precision and the recall of the NLP and IR applications, for example, in the field of text categorization, Barak et al. in [3] used the synonyms investigation to increase the recall from 71% to 92%, and in the field of Information retrieval, Dinh and Lynda [4] showed that the synonyms investigation participated in increasing the Main Average Precision (MAP) by 5.61%. Also, The ASE supports the term weight which is necessary to determine the importance of the word in a particular context [5]

### 1.1. OWS Importance

Recently, important publications in the field of ASE used the statistical approach and gained significant precision [6], [7], and [8], but they did not consider the efficiency and the time required for their systems tends to be long. For example, Leeuwenberga et al. in [7] used a bag of word model called relative cosine similarity to extract the term synonyms. In their work, the construction of the terms-terms weighted matrix is expensive in terms of space and time, and the repetitive computations add more delay. Minkov and Cohen [6] used a path constrained graph, and the problem with this graph is the high time required to construct the graph and the space needed to store the graph. The graph stores each term in the corpus with all existing edges that link this term to the other terms found in the corpus, add to this the time needed to follow all the paths that lead to the terms. Henriksson et al. [9] manipulated two efficient distributional hypothesis models over two large corpora to investigate more semantic relations between the terms, but even that the recall was reasonable (47%), it was obtained in a very low precision (8%). In this case, the low precision means that the size of the answer set was large, and only 8% of the answer was correct. Henriksson et al. used the random permutation and random indexing to construct the required semantic spaces, and those two techniques do not have the accuracy of more advanced statistical techniques such as the Latent Semantic Analysis (LSA). The LSA comes under the Global Matrix Factorization methods [10]; it maps the terms-documents space to a terms-concepts space or a concepts-documents space. The time complexity of the LSA is high because the LSA captures the meaning of the term by creating a sophisticated network in different contexts and huge datasets [11].

In the literature of synonyms extraction, the Pointwise Mutual Information (PMI) mathematical model is used to measure the association between two terms [12, 13, 14, 15]. The PMI of  $x$  and  $y$  considers the number of times  $x$  and  $y$  occurred together,  $f(x, y)$ , and the frequency of  $x$ ,  $f(x)$ , and the frequency of  $y$ ,  $f(y)$ . The PMI is computed using the following equation:

$$PMI(x, y) = \log_2 \frac{f(x, y)}{f(x) * f(y)}$$

Where  $f(x, y)$ ,  $f(x)$ , and  $f(y)$  are normalized by the number of terms ( $N$ ) in the whole corpus.

$$PMI(x, y) = \log_2 \frac{\frac{f(x, y)}{N}}{\frac{f(x)}{N} \frac{f(y)}{N}}$$

$$PMI(x, y) = \log_2 \frac{f(x, y) * N}{f(x) * f(y)}$$

The PMI does not demand the adjacency of  $x$  and  $y$ , and the researchers used the PMI with different window sizes, for example, Inkpen in [15] used a content window of size 2, whereas, Yu et al. in [14, 16] used a content window of size 4.

The PMI is a normalized frequency, which inflates for the low frequency of  $f(x, y)$ . And, the PMI does not consider the semantic association between  $x$  and  $y$ . In addition, with the plain definition of the PMI, how the PMI distinguishes between the verbs, nouns, or any other parts of speech, and how the PMI finds the relation between the different parts of speech that establish the semantic context of the text. All the standard techniques (such as the PMI and Chi-square [17]) that examine the co-occurrences of the terms see the terms as random variables without any semantic meaning. Therefore, measuring the co-occurrence of two terms is not sufficient to find a true relationship between two terms and to distribute the terms over their semantic contexts.

To solve the problem of the PMI and the similar techniques, and to link the terms to their context, the researchers adapted the Vector Space Model (VSM) to be applicable for the synonyms extraction. For example, the VSM model was used to build two synonyms extraction models, the skip-gram (SG) model and the Continuous Bag-of-Words (CBoW) model [18]. Besides considering the frequency of the examined terms, the VSM measures the distribution of the terms over the text segments by using the inverse term frequency in the tf.idf weighting scheme. However, the traditional VSM model in synonyms extraction is time-consuming, especially in real applications such as query expansions and text mining [6], [11], [19]. In traditional tf.idf models, a vector of weights is created for each term in the text. This vector consists of the weights of every term in the corpus with respect to the term being processed, which yields several weighting processing steps equals the square number of the terms. Mikolov et al. in [18] proposed hierarchical softmax and negative sampling approaches to prevent quadratic time.

Pennington et al. in [20] combined the matrix factorization methods with the shallow window-based methods to produce the pioneering GloVe model. The GloVe model produces global vectors for the corpus terms, and through these vectors, the similarities between the terms can be determined. This model can be used in the field of synonyms extraction because it can give accurate similarity between two words, as the authors showed in table 2 in [20]. In Addition, the GloVe model is considered efficient because it processes only the non-zero elements in the co-occurrence matrix. However, the basic assumption of the GloVe model requires the process of the term  $i$  with respect to every term that occurs in the context of  $i$ . To avoid  $O(|V|^2)$  time complexity, where  $V$  is the number of vocabularies, Pennington et al. placed a tight bound on the number of nonzero elements in the co-occurrence matrix.

Excluding the stopwords, the vector space models equalize the importance of the verbs, adjectives, and nouns in determining the weights of a specific term and in exploring the relationship between two terms. From our point of view, this is not accurate and may disperse the extraction process. For example, it is hard to find a set of adjectives that are associated with a specific set of nouns, and the vast majority of the adjectives appear with objects that belong to different semantic contexts, whereas, we can find a set of verbs that normally appears in a specific semantic context, think about the verb “*park*” for the context that talks about the vehicles, and the verb “*bark*” that directly link the picture toward certain kind of animals (dogs or wolves). The design of the OWS weighting scheme addresses this critical issue. The OWS is based on the claim that processing the verbs increases the efficiency because it neglects the processing of the other parts of speech, and increases the precision because it will not process the text’s parts that disperse the extraction process.

The use of verbs as a semantic relation investigator between two concepts was used by Imsombut in [17], Imsombut experienced the semantic relations extraction for the Thai language. Imsombut used the standard definition of the tf.idf to find the weights of the verbs in which, the tf is the frequency of the verb  $v$  in the collection, and the idf is the distribution of the verb  $v$  over the text segments. The standard definition of tf.idf does not correlate the verbs with the nouns (or concepts); it correlates the verbs with their statistical context. We found two problems of the tf.idf used in [17], (1) the processing of all the verbs found in the corpus, and as we know that some verbs are general verbs and can be used with a verity of objects and subjects in different domains (think about the verbs buy, walk,...), and (2) the inverse verb frequency did not consider the concept (the number of concepts appeared with the verb  $v$ ); it considered the document or the text segment (the number of documents the verb  $v$  appeared in).

## 1.2. Research Aims and Contributions

This research aims to enhance the weighting scheme in a way that processes the meaningful units and correlates the verbs with the nouns in semantic and robust relations. The paper contributions are summarized in the following points:

- Orbit Weighting Scheme: the paper discusses the efficiency and accuracy enhancement of the synonyms extraction through the amendment of the

traditional weighting schemes in the VSM-based text extraction. The enhanced weighing scheme or the Orbit Weighting scheme represents a developed version of the traditional tf-idf that remedies the time penalty of weighing the contents of text and investigates the semantic context of the verbs and nouns. The OWS computes the term weight based on the semantic investigation that uses clear parameters and robust procedure.

- Semantic Context Investigation: the OWS links a group of nouns or semantically related nouns with a set of verbs that are only used with these nouns. This linking transfers the extraction of the noun synonyms from the traditional statistical context (tf.idf model) to the semantic context space. The nouns' co-occurrences are represented by semantic relations instead of simply counting their occurrences.
- Efficiency and precision improvement: the use of the OWS will reduce the number of processed terms in every single run of the synonyms extraction, which will participate in enhancing the efficiency (see Table 7, and Figure 15, 16). And, the semantic investigation of the text contents will participate in increasing the precision and provide more accurate synonyms (see Figure 9, 10, 11, 12 and Table 4, 5, 6).

The paper sections are organized as follows: Section 2 surveys the research in the field of statistical synonyms extraction. Section 3 describes the NBDV method of extraction with the weighting schemes and the similarity model. Section 4 depicts the experimental environment and presents the results. Section 5 evaluates the relevancy and efficiency of the NBDV results, and section 6 discusses the factors that affected the extraction process and mentions the advantages and disadvantages of the NBDV method. Finally, section 7 draws the conclusions and future work.

## **2. Related Work**

### **2.1. Synonyms Sets Appearance and their Influence**

The idea of constructing a lexical database came from a group of psychologists and linguists who aimed to find an informative way to search the English dictionary [21]. They manually collected the synonyms and stored them in a lexical database. The produced databases grouped words based on their meanings (the synonym relation), and the grouped words are called synsets. The synsets linked together via super-subordinate relation in which the general objects that belong to a certain synset were linked to a more specific object that belongs to another synset. Also, the relation was transitive and allowed the link of the general synsets with the parts of the specific synsets [22].

Researchers in Computational linguistics are interested in investigating the synonyms of the words, and they arranged them in a special kind of dictionaries called WordNets [23]. The WordNet is an extensive database that stores the words together with their synonyms. It is a concepts dictionary that groups words based on their meanings to produce the synsets. The first appearance of the WordNets was at

Princeton University, and it was performed manually for the English language [21]. According to Miller et al., the WordNet aimed to facilitate the searching in the dictionary and to substitute the regular word searching (which is typically done by the alphabetical ordering of words) by the concepts searching.

The EuroWordNet was developed for eight European languages [24]. EuroWordNet employed the same model used to construct the Princeton WordNet and added two new contributions, the hiring of the Base Concept, and the addition of new relations with a precise way to clarify the relations among the synsets [24]. AWN is the WordNet that was developed for the Arabic Language [25]. Elkateb in [25] benefited from the model used in Princeton WordNet and EuroWordNet, but they faced a real challenge related to the morphological structure of the Arabic Language. To solve this challenge, Elkateb combined the Interlingual Index used in EuroWordNet with a suggested upper merged ontology [25].

In many NLP publications, the semantic investigation of the text contents was improved by hiring a semantic dictionary such as the synonyms dictionary [26], [27], [28], and [29]. The term weight of a given term was computed based on its parameters (for example, term frequency and inverse term frequency) and the parameters that can be obtained from its synonyms. In the field of text classification, Scott and Matwin in [26] used the WordNet and computed the weight of a term by dividing the number of occurrences of the term's synset (taken from WordNet) in the document over the document length. Bloehdorn and Hotho in [27] used WordNet to generalize the terms to their concepts and employ them in the classification process instead of individual terms. Bloehdorn et al. in [30] mapped the terms to their super concept using the WordNet. Another semantic text repository used to improve the semantic text classification was Wikipedia [31] and the Open Direct Project [32].

Text Categorization is another field of text mining that utilized the WordNet semantic dictionary, in [28] the authors used WordNet to build training data and in [3] the authors supplemented the Latent Semantic Analysis with the concepts that were extracted from the WordNet. In [29] Luo et al. proposed a weighting scheme that considers the semantic similarity value between the term and the interpretations of the name of the category, and the interpretations are taken from the WordNet. The major benefit they gained was the smaller training data required to distribute the uncategorized documents over their categories.

In the field of Information retrieval (IR), the semantic investigations of the term synonyms and the generalization of the terms to concepts during the weighting and indexing process were addressed by many researchers [4]. Dinh and Tamine used the semantic meaning to solve the ambiguities that are generated from the regular tf-idf weighting. They mapped the terms to specific concepts taken from MeSH (semantic dictionary for medical data) and then correlated the concept to the correct domain. In this case, the IR system can capture the correct meaning of the term because it knows the concept underlying the term and to which domain the term belongs. Dinh and Tamine built an IR system to implement their idea, and they gained a noticeable improvement in the relevancy of their system comparing with two tf-idf IR systems. However, Dinh and Tamine used specific domain knowledge (biomedical documents),

which enabled them to reduce the problem space (a small number of concepts and a small number of domains).

## 2.2. Synonyms Extraction Techniques

In the literature of the ASE, three main extraction techniques can be derived from the published research:

- The statistical techniques over monolingual corpora, such as the Point Mutual Information [14], the Vector Space Model with cosine similarity or relative cosine similarity [7], [33], [34], [35], and [36].
- The translation techniques among different languages over the bilingual or multilingual dictionaries (the words that share the same interpretations are synonyms) [37], [38], and [39].
- The linguistic analysis techniques that syntactically and semantically parse the corpus or the dictionary to extract synonyms [6], [9], [8], [40], and [41].

In general, the VSM based models in text mining and information retrieval process the whole corpus and make detailed computations for every term [19]. These computations include the weighting of the terms which requires two things: (1) Specifying the required parameters such as the term frequency (tf) and the inverse term frequency (idf). (2) Applying a weighting scheme such as the log frequency and the tf-idf weighting schemes. The best-known formula used in the information retrieval or text mining of the tf-idf scheme was the one proposed by Salton and McGill [42]:

$$w_{t,d} = (1 + \log f_{t,d})(\log \frac{N}{dft})$$

Where  $w_{t,d}$  is the weight of the term  $t$  in text  $d$ ,  $f_{t,d}$  is the frequency of the term  $t$  in text  $d$ ,  $dft$  is the number of text segments contains  $t$ ,  $N$  is the number of text segments in the corpus, text segment could be document, query, paragraph, or sentence.

After specifying the weight for each term, the vector space model computes the similarity between the vectors. The vectors represent the weights of the terms in the text (if it was a text mining application) or the weights of the terms in the user query and in the set of documents (if it was an information retrieval application). The cosine similarity is computed by using the following equation [2]:

$$\begin{aligned} \text{sim}(\vec{v1}, \vec{v2}) &= \cos(\vec{v1}, \vec{v2}) = \frac{\vec{v1} \cdot \vec{v2}}{|\vec{v1}| \cdot |\vec{v2}|} \\ \text{sim}(\vec{v1}, \vec{v2}) &= \frac{\sum_{i=1}^t wv1_i \cdot wv2_i}{\sqrt{\sum_{i=1}^{|\vec{v1}|} wv1_i^2} \cdot \sqrt{\sum_{i=1}^{|\vec{v2}|} wv2_i^2}} \end{aligned}$$

Where  $\vec{v1}$  is the vector of text segment  $v1$ ,  $\vec{v2}$  is the vector of the text segment  $v2$ ,  $wv1_i$  is the weight of the term  $i$  in  $v1$ ,  $wv2_i$  is the weight of the term  $i$  in  $v2$ ,  $t$  is the number of terms in the whole corpus.

The cosine similarity measures the angle between the vectors; if the angle is small (high cosine value), then the similarity is high which implies high resemblance between the two text segments (two documents, document and query, document and summary, two lists of synonyms).

Many research publications employed the VSM model in the synonyms extraction [7], [33], [34]. The tf-idf weighting scheme and the cosine similarity are adapted to reflect the terms to terms relations instead of the query to documents relations in the IR [7]. Analog to the VSM model in IR, the CBoW and SG model were developed in the field of synonyms extraction and hired the cosine similarity to find the similarity between specific term and all the terms found in the corpus [18].

Chen and Lynch [34] used the Vector Space Model to extract the synonyms by computing the cosine similarity between the terms found in a large corpus. They collected the nouns (called the descriptors) and computed the cosine similarity between all the descriptors that have a frequency greater than 3. Before Chen and Lynch, Crouch [33] built an automatic thesaurus dictionary to expand the user query in IR research. Crouch used the VSM and represented the terms as dimensions, and the documents were vectors in the term dimension space. The author built thesaurus classes by combining similar documents in one cluster. Then, Crouch extracted the terms that had low document frequencies from each cluster to form the thesaurus classes. The aim was not the synonyms by themselves, but to expand the user query terms by supporting terms found in a similar document.

Leeuwenberga et al. in [7] emphasized the idea that the simple cosine similarity hurt the precision because it combines synonyms, hypernyms, and hyponyms in the synonym set. Leeuwenberga et al. proposed to consider the top ten similar words and included them in the calculation of the similarity to obtain more accurate similarities. They divided the simple cosine similarity between  $w$  and  $w_q$  words by the summation of the cosine similarities of the top ten words similar to  $w$ . Leeuwenberga et al. obtained 12% precision value (for both English and German).

To solve the problem of computation penalty, Zhang and Wang [43] used the Word2Vec model based on CBoW and SG model to map the relations among the corpus terms, and they used the cosine similarity to find the similarity and spectral clustering to identify the synonyms. Henriksson et al. [9] developed a model for extracting synonyms in biomedical data, and they aimed to enhance the distributional hypothesis model that is deeply used to extract the related meaning words. Henriksson et al. enlarged the scale of the semantic relations among the terms by incorporating two distributional models (instead of one as usual in the distributional hypothesis) and two corpora instead of one large corpus. The aggregation of multiple models and corpora enabled the authors to create more semantic spaces which enriched the relations between the terms [9].

In [35], Lakshmi and Baskar presented two novel term weighting schemes TF-RTF and TF-RFST. The shared idea between the TF-RTF and TF-RFST was to combine the term frequency and the frequencies of the terms that share the same meaning. The set of semantically related words was determined by the Fuzzy C-Means clustering algorithm. Alsmadi and Hoon [36] proposed a new supervised term

weighting scheme (called SW) for weighting the terms that appear in short text messages.

Recently, to build an ontology for the Arabic Language, Benabdallah et al. depended on automatically extracted patterns (called them markers) to find the semantic relations between statistically selected terms [8]. AlMaayah et al. in [5] produced a synonym set for the terms of AL Quran AL Kareem (the holy book for Muslims). AlMaayah et al. produced the synsets by linking the Quran's terms with their meanings that were obtained from a traditional dictionary. The authors successfully improved the recall of the semantic search by 27% compared with a baseline system. Table 1 shows the list of references that use the statistical methods with their models and some accuracy results that were collected from their publication.

All the mentioned publications in this section either require heavy access to terms or relations database or require massive computational operations over all the terms found in a huge corpus. In this paper, the weighting and extracting of the terms and synonyms are not based on a store dictionary, and the extraction of the semantic relations between the nouns is accomplished by only considering the verbs. Also, not all the verbs will be processed. The verbs that have a large term frequency or appeared with a large number of nouns will be neglected.

### 3. NBDV Synonyms Extraction Method

The NBDV is developed in this investigation. It is a vector space-based synonyms extraction method that considers three aspects during the synonyms extraction process:

- Making the synonyms extraction completely statistical. This means that during all the phases of extraction, the NBDV does not use a database of stored synonyms, meanings, or patterns. The database approach is language-dependent, and it restricts the synonyms extraction process with the contents of the database without having the ability to gain more information from the huge text found on the internet. Also, it requires a huge effort and a lot of time, which may long for years, to build a database that contains all the nouns' synonyms.
- Processing the nouns as meaningful units, not as a bag of words (as in the CBoW and SG models) and capturing the noun's meaning by precisely collecting the verbs that are specific to a group of nouns (the parameters used to identify the verb uniqueness is described in section 3.1).
- Reducing the problem domain by weighting the parts of the corpus that are related to the noun being processed.

The NBDV method uses unsupervised learning to extract nouns synonyms. Definition1 gives a simple interpretation of the NBDV method:

**Definition 1:** assume that  $S_{n_1}$  and  $S_{n_2}$  (Verb\_Noun adjacent lists) are the sets of verbs adjoin the nouns  $n_1$  and  $n_2$  such that:  $\bar{S}_{n_1} = \{n_1v_1, n_1v_2, n_1v_3, \dots, n_1v_i\}$ , and  $S_{n_2} = \{n_2v_1, n_2v_2, n_2v_3, \dots, n_2v_j\}$ . Where  $n_1v_1$  is the first verb adjacent to the noun  $n_1$ , and  $n_2v_1$  is the first verb adjacent to the noun  $n_2$ . Both  $i$  and  $j$  are positive

integers greater than 1. Then,  $n_1$  and  $n_2$  are synonyms if  $|S_{n_1} \cap S_{n_2}| \geq c$  (Threshold value)

**Example:** if  $n_1$  is “سيارة” (car) and  $n_2$  is “مركبة” (vehicle),

$$\text{and } S_{سيارة} = \left\{ \begin{array}{l} \text{تسير, ينتج, يشتري, يقلع, يصنّدم, يركن, يركب, يصلح, يقود} \\ \text{... يبيع, يدهس, يسلك,} \end{array} \right\} \Rightarrow S_{car}$$

$$= \left\{ \begin{array}{l} \text{drive, repair, ride, park, bump, take off, buy,} \\ \text{produce, walk, walk, tread, sell, ...} \end{array} \right\}$$

$$\text{and } S_{مركبة} = \left\{ \begin{array}{l} \text{ينتج, يقلع, يحلق, يبحر, يطير, يصنّدم, يركب, يصلح, يقود} \\ \text{... يسقط, يغرق, يرسو, يهبط, يركن} \end{array} \right\} \Rightarrow S_{vehicle}$$

$$= \left\{ \begin{array}{l} \text{drive, repair, ride, bump, fly, sail, fly, crash,} \\ \text{produce, park, land, land, sink, fall ...} \end{array} \right\}$$

Assume that  $c = 5$ , then

سيارة and مركبة are synonyms because  $|S_{سيارة} \cap S_{مركبة}| = 7 > c$

Definition 1 presents the basic and simple idea of the NBDV method. It considers two nouns as synonyms if they share more than  $c$  verbs. But, the selection of the verbs that can group the nouns into semantically related groups is more complicated. The verbs such as “buy”, “produce”, and “walk” are general verbs and can be found with a wide range of nouns, so they cannot be used as distinguishing verbs. Section 3.1 depicts the criteria that have been used to select the distinctive verbs and explains how these criteria have been employed to weight the verbs in the proposed NBDV method.

Two aspects should be mentioned before proceeding to detail the NBDV model.

- The nouns targeted by the NBDV method are the common nouns, not the proper or the entity nouns because the latter mostly do not have synonyms.

Project Name	Ref	Statistical Model	Recall	Precision
<b>Knowledge base creation based on automatic creation of a semantic network</b>	[34]	Cosine similarity (CBoW and SG model) Cluster Algorithm	27% CBoW 35% SG	62% CBoW 66% SG
<b>Minimally Supervised Approach for Automatic Synonym Extraction</b>	[7]	Relative cosine similarity model	7% German 12% English	12% German 12% English
<b>Automatic synonym extraction using Word2Vec and spectral clustering</b>	[43]	Word2Vec model/ CBoW and SG model/ Cosine Similarity/ Spectral Clustering	74% Manual evaluation	80% Manual evaluation

<b>Synonym extraction and abbreviation expansion with ensembles of semantic spaces</b>	[9]	Enhanced distributional hypothesis model	47%	8%
<b>Extraction of terms and semantic relationships from Arabic texts for automatic construction of an ontology</b>	[8]	Learning Extraction Markers TF-idf weights	84%	76%
			Manual judgment of synonyms, domain-specific corpus	
<b>Graph-based similarity measures for synonym extraction from parsed text</b>	[6]	Path Constrained Graph model	MAP = 59%	
<b>Finding Synonyms Using Automatic Word Alignment and Measures of Distributional Similarity</b>	[38]	Distributional Similarity model	13%	23%
<b>Towards an automatic extraction of synonyms for Quranic Arabic</b>	[5]	TF-idf weights (VSM)	35%	33%

Table 1. Summary Of The Statistical Models With Their Accuracy Found In The Related Work

- The NBDV model can be seen as a synonyms extraction or more generally as a collector of the semantically related words because it combines the terms that normally share one semantic context. This point is important to mention because, in the evaluation of the relevance of the output set, the evaluator should not make an exact match between the automatically generated set and the answer set that is taken from the base dictionary. Also, considering the NBDV as a semantic word collection model makes it more supportive to the other fields of the information systems such as the query expansion in the information retrieval systems.

The NBDV method includes two phases, the weighing phase that uses the OWS scheme, and the synonyms detection phase that uses the cosine similarity between the vectors that are generated from the first phase to decide if two nouns are synonyms or not.

### 3.1. The Orbit Weighting (OWS) Phase

The OWS is used in the weighting phase of the NBDV to replace the traditional tf-idf weighting schemes. The OWS is designed for nouns because the nouns are the primary concern of the text mining applications. Mostly, the query terms in the IR field, and the class and category names in text categorization, and the concept/entity in entity recognition, are nouns.

In the OWS, the nouns that should be processed have a semantic relation with the noun that the user wants to find its synonyms. In each run, the similarities are computed between the nouns that share distinctive verbs. Our claim states that the nouns that share a set of specific verbs are more likely to be synonyms. For example, the noun car and automobile have some special verbs that distinguish them from the other nouns such as the verbs “park”, “crash”, and “drive”.

The reason for selecting the verbs as distinguishing factors is that the other parts of speech are normally used with a wide range of nouns. For example, the adjectives are used in the languages to describe objects of wide ranges and different domains.

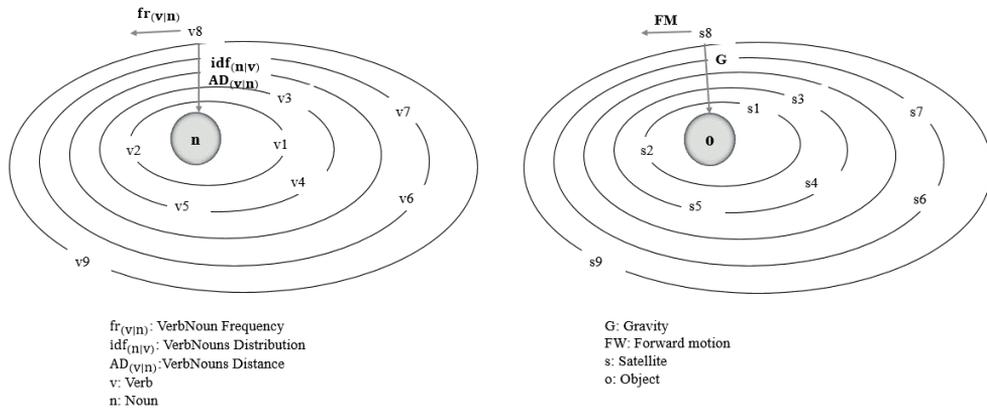


Figure 1. Orbit Representation for the Noun-verbs Relationships (OS(n))

### 3.1.1. Kinetic Relationship

Figure 1 depicts the idea behind the OWS weighting. The philosophy behind the OWS is to picture the semantic meaning of a noun  $n$  [OS(n)]<sup>1</sup> 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 allows determined by the set of verbs that always spin in the orbits of the OS(n).

In Physics, the orbit of an object results from the balance of the forward motion of an object and the gravity (FM and G in Figure 1). This balance creates a robust structure of the space. The reason why we want to depict the text as a space of objects and orbits comes from the fact that the text is a combination of objects (nouns, verbs, and others) and relationships between these objects. Both the forward motion and the gravity can be represented statistical in the text, the term frequency and the term distribution of a certain term can represent the forward motion, and the singularity of a group of verbs to a noun or a set of nouns, which represent the semantic meaning, can represent the gravity. All we need is to find a mathematical scheme of all the forces that affect the verb.

<sup>1</sup> Though out this paper, we will abbreviate the orbiting representation of the noun  $n$  as OS(n)

The OWS sees the nouns and verbs relationship as a kinetic representation in which the verbs are always actions that working interactively and specifically with the nouns. We can represent the verb as an object that holds a specific amount of energy and according to this amount of energy, the verbs round in a specific orbit around the noun. The energy of the verb represents the balance between the semantic relationships and the statistical parameters that can be obtained from a huge corpus.

Based on our assumption, the nouns are grouped in groups (subspaces) according to the meanings of the verbs that are placed in their orbits. In this representation, we depict the semantic context of the noun as a subspace, and this subspace consists of the noun and the verbs that work kinetically with that noun.

The importance of the orbits in this representation is to depict that not all the verbs have the same degree of importance or the same amount of kinetic energy, and the verbs that are close to the center have high energy that increase their effectiveness in determining the subspace of the noun. Note that this representation forces that the effects of the verbs decrease as we move from the inner to the outer orbits. Accordingly, the verbs that appear in the outer orbits are general verbs that are circulating in the outer orbits of the vast majority of the nouns.

In Figure 1, the verbs  $v_1$  and  $v_2$  (could be the verbs “يقود drive” and “يركن park”) spin in the inner orbit, and their contributions are greater than the contributions of the verbs  $v_8$  and  $v_9$  (could be the verbs “يشترى buy” and “يبيع sell”) that are shifted to circulate in the outer orbit.

### 3.1.2. The Mathematical Representation of the OWS

In the OWS, the noun  $n$  is represented as a vector, and the vector of the noun is a set of weights of the verbs that appeared in the OS( $n$ ) in the corpus, and it can be represented as follow:

$$\vec{n} = (w_{v1}, w_{v2}, w_{v3}, \dots, w_{vi})$$

Where  $w_{v1}$  is the weight of the verb  $v_1$  in the space of  $n$ , and  $i$  is the number of verbs appeared in the OS( $n$ ) in the whole corpus.

The weights that compose the noun vector  $\vec{n}$  are computing by the OWS weighting scheme. The OWS computes the weight of a specific verb and decides whether this verb belongs to the  $n$  subspace and in which orbit is spinning. Thus, we can see the weight of the verb  $v$  with respect to the noun  $n$  as the amount of energy hold by  $v$ .

The OWS weights the verbs based on their singularity to a group of nouns. The singularity represents the verb energy and the energy is determined using three parameters, (1) the number of times the verb appeared in the OS( $n$ ) ( forward motion force), (2) the number of orbiting spaces the verb appeared in (gravity force), and (3) the average distance between the verb and the noun in each time the verb appears in the OS( $n$ ) (gravity force). These parameters represent the forces that are affected  $v$ , and they are necessary to measure the uniqueness of the verb with respect to a specific set of nouns or in anther words in which orbit of these nouns' subspaces the verb is located.

The purpose of combining the three parameters is to neglect the verbs that appear in the large number of OSs. For example, if the verb appeared with a large number of nouns, this implies that the verb is a general verb and the value of the second parameter will be very low (the second parameter is inversely proportional to the number of nouns the verb appeared-with as appeared in equation 4), and in this case, the OWS gives the verb tiny weight and shifts it to circulate in the outer orbits.

In Definition 1, two nouns are synonyms if they shared a certain number of verbs. However, our assumption should be updated to process the verbs according to their semantic relation with the nouns, which the orbiting representation reflects. For example, consider the verb “يشترى” (buy) and the verb “يركن” (park) both of them appeared in example 1 as verbs adjacent to the nouns “سيارة”,”مركبة” (appeared in the OS(سيارة) and OS(مركبة)), the verb "يشترى" is a general verb that can appear in the orbiting spaces of a lot of objects or services, but the verb يركن is more specific and related to fewer spaces such as car or bus. The verb “يركن” should make more contribution in determining that the OS(مركبة) and OS(سيارة) are semantically related spaces. It is crucial to determine which verbs should be considered as distinguishing verbs (circulating in the inner orbits )and have a significant effect in determining the synonyms. Therefore, Definition 1 can be updated as follow:

**Definition 2:** assume that  $S_n$  is a set of verbs that appeared in the OS(n) such that:  $S_n = \{n_{v1}, n_{v2}, n_{v3}, \dots, n_{vi}\}$ , Then, for each Verb  $v$ , the weight of  $v$  is determined by considering the following parameters:

1. *VerbNoun Frequency* ( $\mathbf{fr}_{(v|n)}$ ): The number of times the verb  $v$  appeared in the OS(n) in the whole corpus. This parameter can be seen as the  $f(v, n)$  that appears in the PMI definition [12].
2. *VerbNouns Distribution* ( $\mathbf{idf}_{(n|v)}$ ): The number of nouns spaces that contain the verb  $v$  in the whole corpus.
3. *VerbNouns Distance* ( $\mathbf{AD}_{(v|n)}$ ): The average distance between the verb  $v$  and the noun  $n$  in all the  $(v, n)$  occurrences in the OS(n).

In the NBDV method,  $n_j$  is considered as a synonym to noun  $n$  if the similarity between  $n_j$  and  $n$  exceeded a certain threshold value, and the similarity is computed based on the weights of the shared verbs that are weighted by considering the three parameters, the  $fr_{(v|n)}$ , the  $idf_{(n|v)}$ , and the  $AD_{(v|n)}$ . These three parameters are the identifiers of the OWS.

Based on definition 2, the verb that frequently circulates in the orbits of a certain noun (large value of  $fr_{(v|n)}$ ), and normally located as close as possible to that noun (small value of  $AD_{(v|n)}$ ), and appeared with a small set of nouns (small value of  $idf_{(n|v)}$ ), obtained high weight value and should be spin in the inner orbits of Figure 1.

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

**Parameter 1: VerbNoun Frequency ( $fr_{(v|n)}$ )**

VerbNoun Frequency is the number of times the verb  $v$  appeared in the OS( $n$ ). Parameter 1 or the  $fr_{(v|n)}$  identifies the verbs that commonly appear with a specific noun. The  $fr_{(v|n)}$  is computed in equation 1:

$$fr_{(v|n)} = \sum_{v,n \in K} f(v, n) \dots (1)$$

But, some verbs are general and appear intensively, and others are specific and appear in certain domains and platforms. The normalization of equation 1 is performed by dividing the  $fr_{(v|n)}$  by the total number of time the verb  $v$  appeared in  $K$ :

$$fr_{(v|n)} = \frac{\sum_{v,n \in K} f(v, n)}{\sum_{v \in K} f(v)} \dots (2)$$

The normalization degrades the importance of the general verbs because the denominator in equation 2 will be high for such verbs, and this will decrease the weight and shift the verb to outer orbits.

**Parameter 2: VerbNouns Distribution ( $idf_{(n|v)}$ )**

VerbNouns Distribution is the number of orbiting spaces that contain the verb  $v$  in the whole corpus.

$$ndf(v) = \sum_{t \in K} f(n|v) \dots (3)$$

Where  $f(n|v)$  is the number of distinctive nouns that appeared with  $v$

To dampen the effect of the  $ndf(v)$ , equation 3 is normalized as follow:

$$idf_{(n|v)} = \log \frac{N}{\sum_{t \in K} ndf(v)} \dots (4)$$

VerbNouns Distribution parameter indicates the singularity of the verb. The large value of the  $idf_{(n|v)}$  parameter means that the verb appeared in a large number of orbiting spaces. If the verb distribution among the nouns is high, the effectiveness of the verb in differentiating the nouns will be reduced. We argued at this point that the verbs that appear intensively in the text should be treated as Stopwords because for mathematical computation, they will not add any value. The smaller the value of  $idf_{(n|v)}$  the larger the contribution of  $v$ .

In traditional tf.idf, the inverse document frequency ( $idf$ ) is used to measure the distribution of the term over the whole documents [44]. If the  $idf$  value was high, this implies that the term appeared in a large number of documents, and the weight of the term will be low. In our weighting scheme, the  $idf$  is represented in a practical way in which the verbs that appear everywhere in the text are discarded. The  $idf_{(n|v)}$  is used to measure the distribution of the verbs over all the nouns found in the corpus, which participates in correlating a set of nouns with a specific set of verbs that appeared only with these set of nouns in one semantic space. Similar to  $idf$  used in IR

and NLP applications, if the  $idf_{(n|v)}$  value was high, this implies that the verb appeared with a large number of nouns and the verb will not be beneficial in distinguishing the nouns (the weight of the verb will be very low, and the OWS makes the verb spins in a far orbit).

**Parameter 3: VerbNouns Distance ( $AD_{(v|n)}$ )**

VerbNouns Distance is the average distance between the verb  $v$  and the noun  $n$  in all occurrences of  $(v, n)$  in the OS(n).

$$AD_{(v|n)} = \frac{1}{Avg(v_{pos} - n_{pos})}, \forall f(v, n) \dots (5)$$

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 OS(n).

In the OWS scheme, the verb and the noun are not necessary to be adjacent because, in some cases, certain words (such as Adjectives, adverbs, ...) may come between them. For example, consider the following statements from our corpus:

هو يقود السيارة يقود السائق السيارة يقود السائق المتهور السيارة يقود السائق المخمور برعونة السيارة	He drives the car The driver drives the car The reckless driver drives the car Recklessly, the drunk driver drives the car
---	---

The distance parameter ( $AD_{(v|n)}$ ) imposes that the effect of the verb on the noun becomes stronger if the verb was adjacent to the noun, and this effect is reduced as the noun becomes far away from the verb. In the above example, in sentence number one, the whole concentration will be on the noun car, but in the fourth sentence, it addresses the driver, not the car. So, the  $AD_{(v|n)}$  identifier gives the verb drive with respect to the noun car in the first sentence heavier weight than the verb drive in the fourth sentence with respect to the same noun. The importance of  $AD_{(v|n)}$  parameter lies in showing how close the verb to the noun and normally the adjacency means a robust relationship between the verb and noun. So, the small value of  $AD_{(v|n)}$  implies a large value of the weight of  $v$ .

**3.1.3. Tf-idf Traditional Weighting**

Before proceeding in further details about the employment of the previously mentioned parameters in the weighting equation, the definitions of these parameters in traditional tf.idf are mentioned here as how Imsombut in [17] defined and used them. The purpose is to make the efficiency and accuracy comparisons.

In the traditional tf.idf, the  $tf$  and the  $idf$  are defined as follow:

$$fr_{(v,n)} = \sum_{t \in k} f(v, n)$$

Where  $fr_{(v,n)}$  is the number of times the verb  $v$  appeared with the noun  $n$  in  $K$ ,

The normalization step that appeared in the OWS (equation 2) is not necessary, and we used the traditional normalization as follow:

$$fr(v, n) = \begin{cases} 1 + \log \sum_{t \in k} f(v, n), & \text{if } \sum_{t \in k} count(v, n) > 0 \\ 0 & \text{otherwise} \end{cases}$$

In the traditional tf.idf, we link the noun  $n$  with a text segment, not with the verbs that appeared with  $n$ . The inverse noun frequency of the noun  $n$  ( $idf_n$ ), is the number of documents  $d$  that contains  $n$  and can be defined as:

$$idf_{(n,d)} = \log \frac{N}{idf(n)}$$

Where  $idf(n)$  is the number of documents that contain  $n$ .

The VerbNouns Distance in OWS definition can be employed in the tradition tf.idf using equation 5. This employment unifies the window size and ensures that the same set of verbs are processed in the two weighting schemes (OWS and tf.idf).

### 3.1.4. The Weighting Equation of OWS

After computing the value of each parameter in the OWS, the weight of the verb  $v$  the OS( $n$ ) is computed, as shown in equation 6:

$$Weight(v|n) = fr_{(v|n)} * idf_{(n|v)} * AD_{(v|n)} \dots (6)$$

Equation 6 summarized the OWS weighting scheme, where  $fr_{(v|n)}$  is the frequency of  $v$  with respect to  $n$ ,  $idf_{(v|n)}$  is the number of  $n$  with respect to  $v$ , and  $AD_{(v|n)}$  is the average distance between  $v$  and  $n$ .

Whereas, the weighting equation in the tf.idf weighting scheme will be:

$$Weight(v) = fr(v, n) * idf_{(n,d)} * AD_{(v|n)} \dots (7)$$

Note that in the tf-idf, there is no semantic connection between the noun and the verbs because the  $idf$  is computed relative to the document not to the verbs that normally appear with a set of nouns.

### 3.1.5. The OWS Process Architecture

The OWS weighting process is designed in Figure 2. In Figure 2, the list of verbs that appeared with the noun Nouni are weighted based on equations 2, 4, 5, and 6; then the weights are distributed to the appropriate orbits based on equation 8.

In Figure 2, orbit 1 represents the inner orbit that contains the verbs that gained the highest weights, and orbit  $j$  represents the outer orbit that includes the smallest weights. The OWS generates the output as a linked list, and the lists reflect the orbits' representation of the noun-verbs combinations. Nested linked lists are built for the noun being processed, and the list contains the verbs associated with the noun in the corpus.

Each level in the list represents one orbit, and all the verbs found in one orbit have roughly the same weight value (located with the range specified in equation 8). The verbs that are close to the root node (noun node) have the largest weights, and the weights start to decrease toward the leaf.

The pointer data structure with the dynamic memory allocation is used to represent the OWS output, and the purpose of this representation is to optimize the time and storage space.

The weighting process is designed as follows:

1. Compute the weighting parameters ( $fr_{(v|n)}$ ,  $idf_{(n|v)}$ , and  $AD_{(v|n)}$ ) for each  $v$  appeared in  $OS(n)$ .
2. Compute the weight using equation 6 for each  $v$  appeared in  $OS(n)$ .
3. Construct the vector of  $n$ .

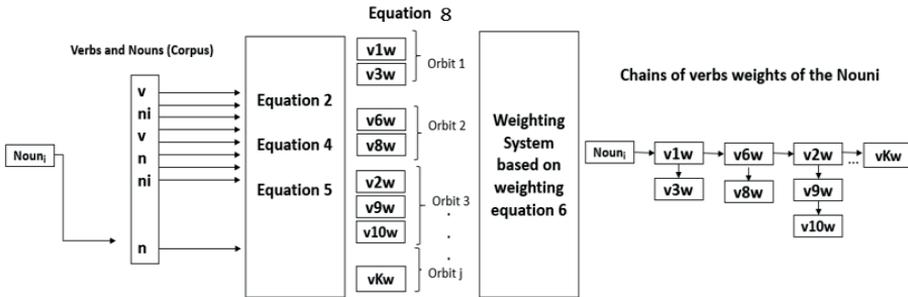


Figure 2. OWS Process Architecture

4. Specify the range of weights that should be included in each orbit. All the weights are located between the interval ( $MINw$ ,  $MAXw$ ). Therefore, the orbit range is assumed to be:

$$Orbit = \frac{MAXw - MINw}{y} \dots (8)$$

Where  $y$  is the number of verbs that are weighted in step 2,  $MAXw$  is the weight of the verb appears in the inner orbit, and the  $MINw$  is the weight of the verb appears in the outer orbit.

5. Extract all the nouns that share the verbs appeared in the first three orbits that are specified in step 4. Here it is important to mention that the NBDV method processes the nouns found in the first, second, and third orbit and three is set as the threshold value of the number of orbits processed by the NBDV method. The identification of this threshold value is made by measuring the number of processed nouns based on the first 13 orbits for 120 value of  $n$  (for 120 runs of the NBDV), and the results are presented in Figure 3. For more illustration, the number of nouns appeared in orbit 3 in our sample is 227, and this number increases to 279 in Orbit 4. It was found that the number of nouns becomes very large after the third Orbit.
6. Using OWS equation 6, compute the weights of each verb located in the first three orbits with respect to each noun extracted in step number 5, and generate a vector of weights for each noun.

At the end of the OWS phase, the NBDV model has a number of vectors equal to the number of nouns shared between the verbs that are located in the first three orbits. The outputted vectors are then transferred to the second phase of the NBDV model.

Regarding the tf.idf, in step number two, the weighting will be accomplished using equation 7, and it will be computed for all the verbs appeared in K, and steps 4,5,6 are not necessary because all the nouns will be processed not parts of them, so no need to specify which nouns should be processed and which noun should be neglected.

### 3.2. Synonyms Detection phase

The purpose of the second phase is to generate the required synonyms set. The input of the synonyms detection phase is a set of vectors generated from the OWS phase. The NBDV investigates the vectors and computes the cosine similarity between the vectors, as shown in Figure 4.

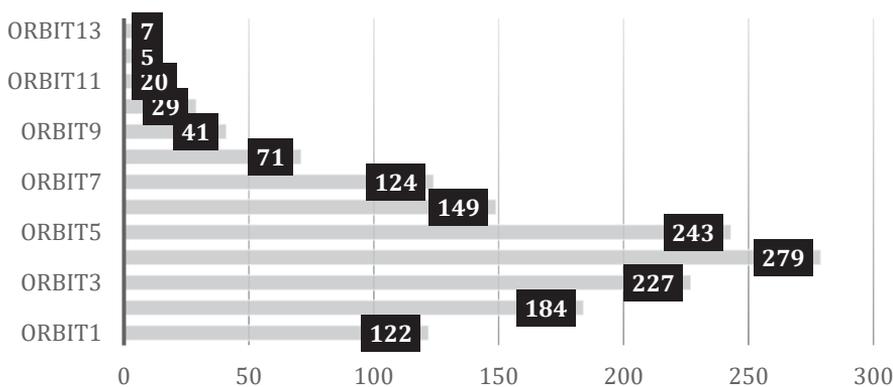


Figure 3. the Number of Processed Nouns at 13 Orbits of the 120 Runs of the NBDV Model

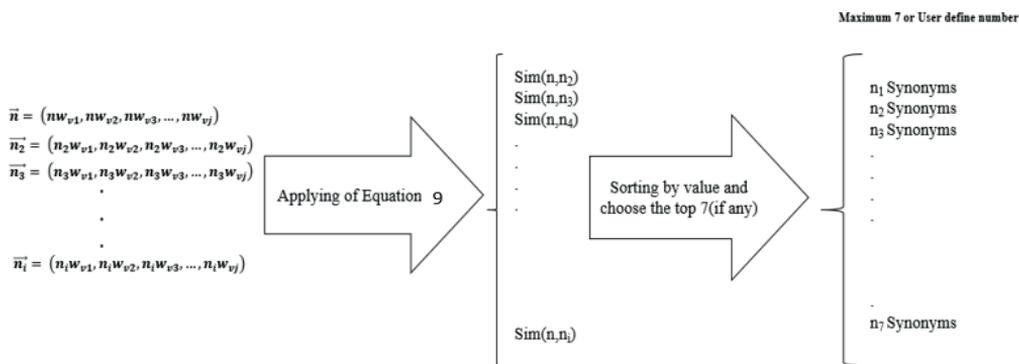


Figure 4. Synonyms Detection Steps Embedded in the NBDV Model

The synonyms detection phase takes the following steps:

1. Find the similarity between  $\vec{n}$  and all the vectors  $\vec{n}_x$  that is produced from the OWS phase using equation 9.

$$\text{sim}(\vec{n}, \vec{n}_x) = \cos(\vec{n}, \vec{n}_x) = \frac{\vec{n} \cdot \vec{n}_x}{|\vec{n}| \cdot |\vec{n}_x|}$$

$$\text{sim}(\vec{n}, \vec{n}_x) = \frac{\sum_{i=1}^j w(nv_i) \cdot w(n_x v_i)}{\sqrt{\sum_{i=1}^j w(nv_i)^2} \cdot \sqrt{\sum_{i=1}^j w(n_x v_i)^2}} \dots (9)$$

Where  $j$  is the number of shared verbs that are identified in the OWS phase,  $w(nv_i)$  is the weight of the verb  $v_i$  with respect to the noun  $n$ , and  $w(n_x v_i)$  is the weight of the verb  $v_i$  with respect to the noun  $n_x$ . The  $\text{sim}(n, n_x) = 1$ , if all the verbs are shared and have the same weights. ( $0 \leq \text{sim}(n, n_x) \leq 1$ ).

2. Sort the similarity values in descending order. Discard all the similarity values that are less than 0.18. The 0.18 value was determined by scanning the generated similarity values of the first 120 nouns. The threshold value was specified by dividing the similarity value over 8 ranges as appearing in Figure 5, and we counted the number of processed noun located in each range. We note that the range of 0.2-0.3 contained the largest number of nouns (213). Then, and to be more specific, we took this range and the range below it ( $0.1 - 0.2$ ) and divided them into seven subranges, as shown in Figure 5. We noted that below 0.18, the number of processed nouns become small (46). Add to this note, the point appears in Table 2, which shows that the average precision was insignificant at average similarity values below 0.2.

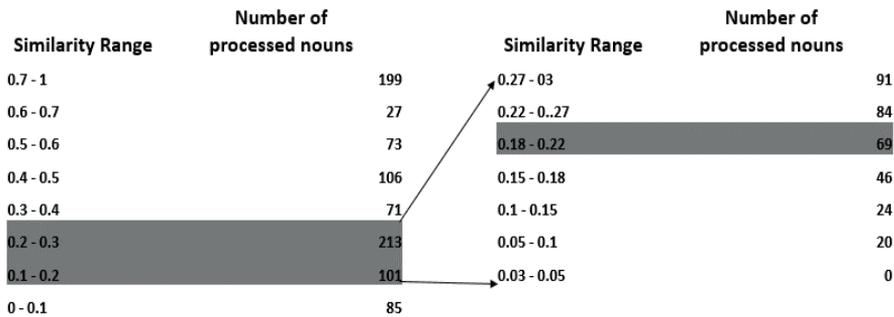


Figure 5. The Number of Processed Nouns Divided into 8 Ranges and 7 Subranges of the Noun Similarity Values.

3. Consider the top similarity values and extract the nouns corresponding to these values as the answer set.

In steps number two and three, the similarity values are sorted, and the top  $n$  values are treated. The value of  $n$  can be user-defined, but in our experiment section,

we chose seven as the value of  $n$  (maximum seven synonyms for each noun). Each similarity value is a value that measures the closeness between two nouns. If the value is large, the NBDV method takes the participating nouns as synonyms.

Back to the orbit representation, the computation function takes the orbit number in consideration because the verbs in the inner orbits should have heavier weights. The equation solves the problem that may arise if a large number of verbs were shared at the outer orbits because their values will be very small to make any difference. This feature distinguishes our equation from the similarity equations used in NLP and IR fields. All the similarity equations such as the Dice's Coefficient, Jaccard's Coefficient, and the Cosine similarity deal with a set of values, not an ordered list, they do not consider the position of the values in the set before computing the similarity.

The selection of seven as the maximum number of synonyms was based on the information that appeared in Table 2, the table traces the precision value of the first 24 top nouns that appeared in the answer set of 120 runs of the NBDV, for example, Top 4 precision is the average precision of all the precision values that were collected from the first four synonyms of the generated answer set in 120 runs. Note that after the top 7, the precision value recorded an abrupt decrease, from 51% to 35%.

The NBDV method is completely statistical and language-independent for the languages that have the following word orders: SVO, VSO, VOS, OVS, but it is not applicable to the languages that have the SOV (like Hindi, Japanese, Korean) and OSV (like Warao) word orders, because the OWS assumes that the verb precedes the noun.

### 3.3. NBDV Algorithm

The algorithm that appears in Algorithm 1 is designed based on the methodology described in sections 3.1 and 3.2. The algorithm accepts two inputs, the noun being processed (called  $x$ ), and a preprocessed huge corpus (called  $K$ ). The preprocessing of the corpus explained in the experiment setting section 4.2. The algorithm creates two dynamic arrays,  $verbs(x)$  stores the verbs adjacent to  $x$ , and  $candidate(x)$  stores the candidate synonyms of the noun  $x$ . The output that represents the synonyms for the noun  $x$  (maximum seven elements) is retrieved in a static array  $S$ .  $S$  is an ordered list sorted by the cosine similarity (equation 9) between the noun  $x$  and the candidate synonym  $x_c$  in  $S$ . In the NBDV algorithm, the OWS process is used twice, the first use is to compute the weights of the verbs adjacent to the main noun  $x$  (stored in  $verbs(x)$ ), and the second use is for each noun  $x_c$  stored in  $candidate(x)$ . The second use of the OWS will not hurt the time complexity because the algorithm performs it only for the verbs stored in  $verbs(x)$  after deleting all the verbs that are not located in orbits 1 -3 (called  $O_1, O_2, O_3$ ).

	<b>Average Precision</b>	<b>Average Similarity</b>
<b>Top 1</b>	0.91	1
<b>Top 2</b>	0.908333333	0.749838303
<b>Top 3</b>	0.75	0.601859652
<b>Top 4</b>	0.791666667	0.515274347
<b>Top 5</b>	0.575	0.423002959
<b>Top 6</b>	0.566666667	0.386232659
<b>Top 7</b>	0.511666667	0.355228858
<b>Top 8</b>	0.358333333	0.326606785
<b>Top 9</b>	0.291666667	0.300332795
<b>Top 10</b>	0.266666667	0.274658761
<b>Top 11</b>	0.175	0.253573794
<b>Top 12</b>	0.166666667	0.246014353
<b>Top 13</b>	0.15	0.236191662
<b>Top 14</b>	0.15	0.220238948
<b>Top 15</b>	0.116666667	0.209058545
<b>Top 16</b>	0.116666667	0.195219259
<b>Top 17</b>	0.083333333	0.180584797
<b>Top 18</b>	0.083333333	0.173604638
<b>Top 19</b>	0.066666667	0.168626283
<b>Top 20</b>	0.058333333	0.165138188
<b>Top 21</b>	0.025	0.161756153
<b>Top 22</b>	0.016666667	0.152375659
<b>Top 23</b>	0.016666667	0.142409236
<b>Top 24</b>	0.016666667	0.139284236

Table 2. the Average Precision and Average Similarity at 24 Levels of the Answer Set of the 120 Runs of the NBDV Synonyms extraction

#### 4. Experiment Preparation

The experiments are designed to measure the performance in terms of efficiency and relevancy of the NBDV method. This implies comparing the results that are generated from the NBDV method in two cases, in the first case, the NBDV will use the OWS to generate the vectors for the verbs, and in the second case, the tf.idf weighting scheme will be used to generate the verbs weights. In both cases, the accuracy measurements (Precision and Recall) and the efficiency measurements (the number of verbs and the number of nouns) should be collected in every single run of the NBDV model.

**ALGORITHM 1: NBDV Extraction**

**Input:** the noun  $x$ , preprocessed Text Corpus  $K$  // *Kalimat corpus with term-tag-stem format*

**Output:** the set  $S$  of synonyms of the noun  $x$  of the form:  $S = \{syn_1, syn_2, \dots, syn_i\}$ , where  $sim(x, syn_1) \leq sim(x, syn_2) \leq \dots \leq sim(x, syn_i)$ , with  $max(i) = 7, j = 1, 2, \dots, max(i)$

**Method:**

**Begin**

Construct  $vec(x) = \{\}$ ; // *vector of  $x$  contains the OWS weights of the verbs adjacent to  $x$*

Construct  $verbs(x) = \{\}$ ; // *creating dynamic array to store the verbs adjacent to  $x$*

Search through  $K$ ;

$c = 0$ ;

For each  $v_i$  appeared within  $i^{th}$  locations of  $x$  // we set 5 as the value of  $i$  "window size"

{

Add  $v_i$  to  $verbs(x)$

Find  $fr(v_i|x)$  // *using equation 2*

Find  $idf(x|v_i)$  // *using equation 4*

Find  $AD(v_i|x)$  // *using equation 5*

Find  $W(v_i|x)$  // *using equation 6*

Add  $W(v_i|x)$  to  $vec(x)$ ;  $vc = vc + 1$ ; //  $vc$  : verbs counter

}

Orbit =  $(max(w) - min(w)) / vc$ ; // *computing the orbit range*

For each  $w \in vec(x)$  do

{

if  $max(w) \geq w \geq (max(w) - orbit)$  add  $w$  to  $O_1$ ; // *inner orbit*

Else if  $(max(w) - orbit) > w \geq (max(w) - 2*orbit)$  add  $w$  to  $O_2$ ;

Else if  $(max(w) - 2*orbit) > w \geq (max(w) - 3*orbit)$  add  $w$  to  $O_3$ ; // *the outer orbit*

Else delete  $w$  from  $vec(x)$ ; delete  $v$  from  $verbs(x)$ ; // *excluded all the verbs that located after orbit 3*

}

Construct  $candidate(x)$  // *creating dynamic array to store the candidate synonyms of  $x$*

For each  $v_i \in verbs(x)$  do

If noun  $x_c$  adjacent to  $v_i$  add  $x_c$  to  $candidate(x)$  // *add noun  $x_c$  the candidate array*

For each  $x_c \in candidate(x)$  do

{

compute the weights of each  $v_i$  verbs ( $x$ ) with respect to  $x_c$

construct  $vec(n_c)$ ;

if  $sim(vec(x), vec(x_c)) < 0.18$  delete  $x_c$  from the  $candidate(x)$  // *similarity equation 9*

}

Sort  $candidate(x)$ ;

$S =$  the first seven element of  $candidate(x)$ ; // *the final synonyms set*

**End**

**4.1. Arabic and English Datasets**

Huge datasets from the Arabic and English languages are used to test the effectiveness of the new weighting scheme. Almost, the English language has been used to test the performance of all the state-of-the-art models such as the SG, CBoW, and GloVe. Therefore, we chose the Common Crawl corpus with 42 billion terms. This corpus is used to compare the performance of the NBDV model with the performance of the GloVe model as showing in section 5.3.

The Arabic language has a well-defined structure with no irregular forms of the structures or the verbs. The feature that pushes toward the use of the Arabic language as the testing language is the flexibility in constructing the sentences. In Arabic, the sentence can be constructed in more than one-word orders, and the writers of the Arabic language have the flexibility to structure the sentences using the form Subject-Verb-Object (SVO) or Verb-Subject-Object (VSO). So, in the experiment, instead of using two different languages with two-word orders (SVO, VSO), we used one language that accepts both word orders.

A huge dataset of Arabic text has been used in the experiment of our model. This corpus collected four datasets of the Arabic language. The corpus contains millions of words. The corpora that included in this corpus include:

Essex Arabic Corpus: This corpus published free on <http://www.lancaster.ac.uk/staff/elhaj/corpora.htm>. The corpus used recently by Al-Radaideh and Bataineh [45].

Kalimat data corpus<sup>2</sup> : Kalimat contains 20,291 Arabic articles. The corpus comprises greater than 6,000,000 terms. The data was taken from Omani newspapers

242 data corpus, the corpus includes 242 Arabic text documents. The corpus was used by many researchers who investigated the Arabic IR field [46].

AH-Dataset corpus: It's a new dataset that comprises a huge number of Arabic words. The data is collected from three Arabic newspaper websites (Aljazeera.net, Alarabiya.net, and Addustour.com) and the Arabic website of Wikipedia. All the necessary preprocess operations (Tokenization, POS tagging, and Lemmatization) were performed using the ALP tool<sup>3</sup>. This tool is developed by Freihath et al. and available free for processing Arabic text. We tested the output of this tool manually for groups of documents, and the accuracy was significant.

We tried to vary the topics and the domain of knowledge, so the selected data talking about health, science, history, art, religion, technology, environment, economic, and financial aspects.

## 4.2. Experiment Setting

This section describes the experiment environment used to test the behavior of the NBDV for synonyms extraction. The experiment was performed on Intel® Core™ i5-7200U CPU @ 2.5GHz processor with 8 GB RAM and Windows 10 OS.

The NBDV method of synonyms extraction is implemented in the VSyn software package (see Figure 6). The purpose of developing the VSyn software is to test the performance of the NBDV. The VSyn was designed and implemented according to the detailed specifications of the NBDV algorithm described in section 3.3. The VSyn allows the user to enter the noun and search for the synonyms.

The VSyn software interface includes three output panels:

- The first one displays the sentences that contain the noun.

<sup>2</sup> The corpus is available free on <http://www.lancaster.ac.uk/staff/elhaj/corpora.htm>

<sup>3</sup>The tool is available online on <http://arabicnlp.pro/alp/>

- The second panel displays the verbs that appeared with this noun sorted by their distance from the noun.
- The third panel displays the generated synonyms.

Also, the software generates an excel sheet that contains all the processed verbs and their obtained weights.

For automatic evaluation with automatic evaluation tools such as the ROUGE tool, the system generates a text file for each processed noun that contains its synonyms list (see Figure 6).

As discussed in the previous section, the Kalimat dataset is used in the experiment. In Kalimat, the terms are already tagged, but preprocessing operations were performed to eliminate the Stopwords, punctuation markers, and special and strange symbols. Also, a simple modification is made to unify all the subtypes under one tag. For example, Kalimat classifies the types of the nouns, for example, اسم الآلة (the nouns that refer to equipments or tools such as key, saw, lathe, fan, radiator, and scalpel), اسم الزمان و المكان (nouns indicate the place and time of the action such as park, airport, appointment), and all the nouns types were unified under the “noun” tag. The same thing was done for the verbs; all verb types were unified under the tag “verb”. After the preprocessing, the text is stored in ( term-tag-stem) format.

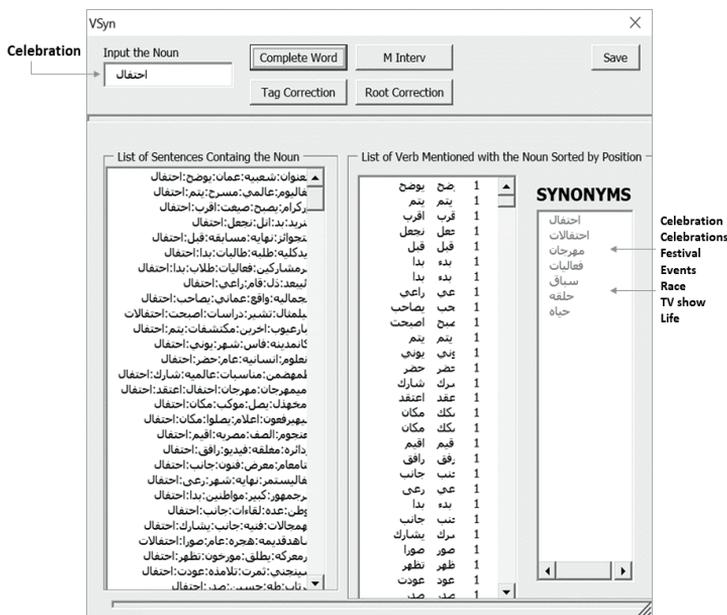


Figure 6. VSyn Interface

The selection of the nouns to be tested in the experiment includes 564 nouns from the Kalimat dataset. The nouns are chosen randomly. The random selection allows the testing of nouns that vary in the number of synonyms they have. For example, some nouns are common and have a large number of synonyms, and some nouns are special and have a small number of synonyms. The NBDV method is significant if it

returns a large number of synonyms for the common nouns and a few numbers of synonyms for the special nouns.

The selected nouns are processed one by one, and the generated synonyms were collected for evaluation. Also, the verbs and nouns processed in each run were collected to measure the processed portion of the corpus and to estimate the time complexity needed to finish every single synonyms extraction operation. The types of the collected results in the experiment with the purpose of each type appear in the next section.

### 4.3. Experiment Results

In the experiment, the following results are collected for evaluation purposes:

1. **The verbs that appeared in each run with their weights.** These results are used in the evaluation to assess the ratio of processed verbs in each run to the total number of verbs in the whole corpus. The number of processed verbs is essential to determine the time complexity of the NBDV, as shown in section 5.3. For example, consider the noun “هجوم”, the list of verbs and their computed weights (using the OWS) came as follow:

شن	بشن	شنا	صد	عنف	جرح	رتل	تورط	زحف	نجا
Past of launch	Present of launch	Past of launch for plural	repulsed	expostulate	hurt	intone	mire	Past of crawl	Past of survive
0.829	0.794	0.543	0.265	0.193	0.17	0.126	0.101	0.0952	0.0914

2. **The maximum and minimum weights of each run:** these are necessary to compute the value of the orbit range in each run (the output of the applying equation 8). For example, for the noun “هجوم”, the maximum weight was 0.829 for the verb (launched), and the minimum weight was 0.0914 for the verb (survived).

$$Orbit = \frac{0.829 - 0.091}{16} = 0.046$$

3. **The distribution of the verbs in the orbits:** as in the following example for the noun “هجوم”:

شن	بشن	شنا	صد	اعنف	جرح	يلطف	رتل	تورط	زحف	نجا
0.829	0.79	0.543	0.265	0.19	0.179	0.15	0.12	0.10	0.095	0.091
Inner orbit		Second orbit	Third orbit	Fourth orbit			Outer orbit			

4. **The set of candidate synonyms.** Similar to the number of verbs, the number of processed nouns in each run is collected to estimate the complexity of the NBDV method. For example, the candidate synonyms of the noun “هجوم”:

هجمات	رئيس	ناس	هجوم	اعتداءات	ناصف	حمله	عمليات	قوات	اعتداء	عدوان
attacks	President	People	assaults	attack	campaign	operations	troop	assault	aggression	

5. **The final set of synonyms** after deleting the candidate synonyms that have low similarity with the main noun: as in the following example for the noun “هجوم”.

هجوم	هجمات	ناس	حملة	عمليات	اعتداء	عملية
attack	attacks	people	campaign	operations	assault	operation

6. **The nouns with their synonyms in one table for all runs.** The outputs of 564 runs of the VSyn were collected in the following format:

Term	Syn1	2Syn	3Syn	4Syn	5Syn	6Syn	7Syn
------	------	------	------	------	------	------	------

These results are important to evaluate the accuracy (Precision and Recall) of our method. The final sets of synonyms were evaluated by the ROUGE evaluation tool and assessed by the manual evaluators. Table 3 shows a sample of our results.

## 5. Evaluation

After collecting the complete results, the results were evaluated in two separate ways. Firstly, by comparing our results with two online sources of Arabic Language synonyms; Almaany<sup>4</sup> and Google Translate. A sample of nouns found in our corpus was randomly chosen, and a comparison between the automatically generated synonyms by VSyn system and the synonyms sets available on those two online sources has been established. Secondly, six Arabic language experts were hired to measure their degree of satisfaction with the accuracy of the VSyn system.

For accuracy comparisons with the other publications in the field of statistical synonyms extraction, we used the recall and precision relevancy measurements that are used to test the accuracy of the well-known models; the CBoW and the SG model. The Recall is the number of correct synonyms of a noun returned by our method relative to the number of actual synonyms number found in the Arabic language for that noun (taken from a base dictionary or an expert knowledge).

$$Recall = \frac{\text{the number of correct synonyms of the noun generated automatically}}{\text{the actual number of synonyms of that noun found in the Arabic Language}}$$

The Precision is the number of correct retrieved synonyms of a noun relative to the total number of generated synonyms by the automatic synonym extraction system.

$$Precision = \frac{\text{the number of correct synonyms of the noun generated automatically}}{\text{the number of synonyms of the noun generated automatically}}$$

Also, to measure the behavior during the synonyms extraction process, and to assess the relevancy at each point a new noun is processed by the NBDV method, two new evaluation schemes are proposed:

<sup>4</sup> <https://www.almaany.com/ar/thes/ar-ar/>

Term	Syn 1	Syn 2	Syn 3	Syn 4	Syn 5	Syn 6	Syn 7
ناس <b>people</b>	ناس people	قوم folk	السكان residents	رسول messenger			
منطقه <b>area</b>	منطقه area	مدينه city	شباب youth	ولايه state	ولايات States	مناطق areas	عديد numerous
مجموعه <b>collection</b>	مجموعه collection	جماعه group	شكل form	وفد delegation	اكثر More	عديد numerous	عام general
ولايه <b>state</b>	ولايه state	منطقه area	عام general	محافظة governorate	مناطق Areas	بحريه marine	مدينه city
شارع <b>street</b>	شارع street	طريق road	منطقه area	وادي valley	مختلف different	ولايه states	
مليار	no synonyms generated						
شركه <b>company</b>	شركه company	شركات companies	وزاره ministry	منظمه organization	مؤسسه institution	عام general	مشاركه participation
عدوان <b>aggression</b>	عدوان aggression	رئيس president	عديد numerous	اعتداء assault	هجوم Attack	هجمات attacks	
كاتب <b>writer</b>	كاتب writer	مؤلف author	كتاب book	كاتبه writer (female)	عمل Work	تاريخ history	عالم world
زعيم <b>leader</b>	زعيم leader	زعيمه leader (female)	يوم day	غزو invasion	رئيس president	عام general	وقت time
بحث <b>research</b>	بحث research	دراسه study	تقديم introducing	تحقيق investigation	شباب Youth	تجديد renewal	عمل work
صور <b>pictures</b>	صوره picture	صور pictures	تاريخ history	تصورات perceptions			
وقت <b>time</b>	وقت time	شكل form	قدر destiny				
معرفه <b>knowledge</b>	معرفه knowledge	تحقيق investigation	وقت time	مجال domain	نور light	نפט petrol	معلومات information
مسلم <b>Muslim</b>	مسلم Muslim	مسلمون Muslims	مسلمين Muslims	عالم world			
طعام <b>food</b>	طعام food	حقوق Fields	حديث speech	غذاء nutriment	بروتينات proteins	ماكولات foods	

Table 3. Samples of Generated Synonyms by our Synonyms Extraction System

- Average Recall at point: the average recall after the processing of the noun number  $i$ :

$$A(R|i) = \frac{\sum_{n=1}^i R_n}{i}$$

- Average Precision at point: the average precision after the processing of the noun number  $i$ :

$$A(P|i) = \frac{\sum_{n=1}^i P_n}{i}$$

The  $A(R|i)$  and  $A(P|i)$  trace the average recall and the average precision trends during the processing of the nouns by NBDV method. The importance of the  $A(R|i)$  and  $A(P|i)$  is to measure the fluctuation on the recall and precision values of the individual runs of the VSyn software. The fluctuation should be in minimum, and this can be obtained when the values of the  $A(R|i)$  and  $A(P|i)$  reach to the final average of the recall and precision after processing a small number of nouns.

The recall and precision are manually and automatically collected, and the automatic evaluation is performed using ROUGE 2.0 tool. ROUGE 2.0 is widely used in literature to test the accuracy of the summarization or extraction systems [47], [48]. It evaluates the system generated extract against a reference or gold extracts (typically the reference extracts are performed manually or taken from a well-known source). To measure the precision and recall using the ROUGE tool, the generated text file from the VSyn software is matched against the reference extract taken from the base dictionary.

The evaluation includes the assessment of the precision and recall values obtained in the experiment and the time complexity necessary to run the NBDV method. But, before evaluating them, the size of the produced answer set from the VSyn system should be investigated.

### 5.1. The Size of the Answer Set Evaluation

The size of the answer set is important because if the number of generated synonyms is always high this means that the parameters specified in the description of the NBDV method are not robust and cannot control the synonyms retrieval process, and if the number of synonyms retrieval is always low this means that the parameters cannot establish a real semantic relationship between the noun and the candidate synonyms.

The number of generated synonyms for each noun was statistically determined and divided into two categories, less than or equal 3 and greater than 3. On average, 20% of the nouns gained between one to three synonyms, and 70% of the nouns gained more than 3 synonyms (the remaining 10% of nouns gained 0 synonyms). Figure 7 shows the ratio of the nouns that gained greater than a certain number of synonyms. For example, from Figure 7, 20% (110 nouns out of 564 nouns experimented) of the nouns gained at least one synonym and at most three synonyms, and 70% of the corpus nouns gained at least four synonyms and at most 7. Figure 8 revealed that from 564 of the tested nouns, 90% gained at least one synonym, and 70% gained at least three synonyms. Almost, the output answer is not empty, and the VSyn system that was

built based on the NBDV method succeeded to return the reasonable number of synonyms.

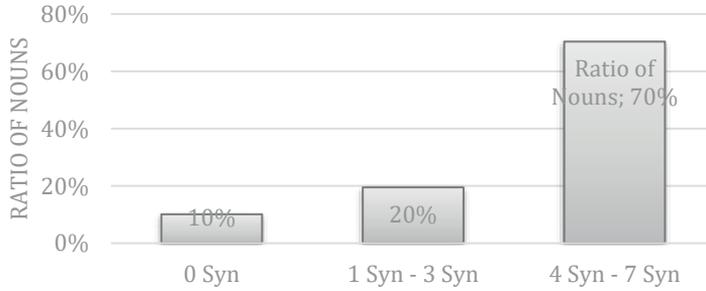


Figure 7. The Ratio of Nouns that gained 0, 1-3, and 4-7 Synonyms.

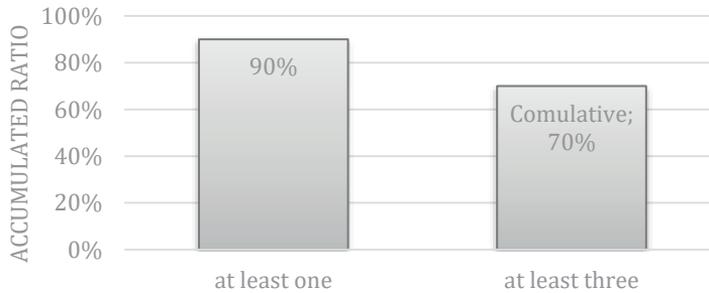


Figure 8. The Accumulative Ratio of Nouns that Gained more than 1 and more than 3 Synonyms

**5.2. Accuracy Assessment Evaluation**

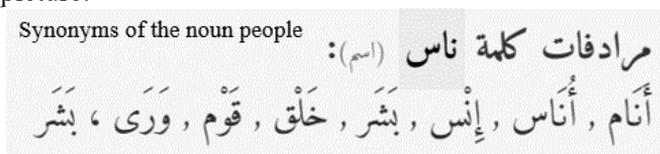
Both recall and precision are used intensely to assess the accuracy of the Natural Language Processing and Information Retrieval applications. The precision indicates the ratio of correctness relative to the answer set, while, the recall gives a broader view and estimates the accuracy relative to the typical answer. However, in such kind of text mining applications, the determination of the typical answer is practically hard, but we can assume that the synonyms set found in the base dictionaries are the ideal answer and prepare our calculation accordingly. Also, Arabic language experts and speakers were hired to take their opinions in the accuracy of the generated answer set. In the manual evaluation, it is hard to ask the experts and the speakers to find the recall because this needs large effort from them to find the optimal synonyms set, so from the manual evaluation, the precision is the only collected relevancy measure. Brief descriptions of the figures and tables appear in this section are as following:

- Figure 9 shows the recall and precision (Almaany dictionary as an optimal answer).

- Figure 10 shows the recall and precision (Google Translate synonyms set as an optimal answer).
- Figure 11 Compares the average recall in the case of using Almaany as a base of evaluation and in the case of using Google Translate as a base of evaluation.
- Figure 12 Compares the average precision in the case of using Almaany as a base of evaluation and in the case of using Google Translate as a base of evaluation.
- Figure 13 Compares the trend of the average recall in case of using the OWS weighting and in case of using tf.idf weighting (Google Translate as a base of evaluation).
- Figure 14 Compares the trend of the average precision in the case of using the OWS weighting and in the case of using tf.idf weighting (Google Translate as a base of evaluation).
- Figure 15: The spearman rank correlation results of the word similarity models over the generated similarities from the NBDV and GloVe.
- Table 4 shows the final average precision and average recall using manual and automatic evaluation strategies.
- Table 5 shows the precision and recall values for the automatically generated synonyms for the first 10 nouns based on OWS and tf.idf weighting schemes.
- Table 6 shows the average precision for the experts' and speakers' manual evaluation.

### 5.2.1. Almaany-Based Evaluation of the NBDV based on OWS Scheme

Almaany online Dictionary contains the meanings, synonyms, and antonyms of the Arabic language words. Almaany is a pioneering online tool that composes a database that was taken from a set of famous Arabic dictionaries including “Lesan Alarab لسان العرب”, “Alraa’d الرائد”, “Alwaseet الوسيط”, “Alghany الغني”, “Modern Arabic Language اللغة العربية المعاصر”, and “Aljaam’a الجامع”. The output of Almaany dictionary looks like the following picture:



The answer set of Almaany usually contains repeated words, so, a simple preprocessing stage to remove the duplicates was performed before the evaluation process. For example, the answer set for the noun “ فوز winning” includes the following synonyms:

انتصار, انتصار, انتصار, ظفر, ظفر, ظهور, غلبه, غلبه, غلبه, غلبه, غلب, غلب, فتح, فلاح, مفازة, منجاه, نجاح, نصر

Note that the Arabic noun “انتصار triumph” repeated three times, the noun “غلبة predominance” repeated five times.

For the manual and automatic evaluation, a group of nouns was inserted into Almaany dictionary, and the generated synonyms were collected in the following format.

Synonyms Generated from Almaany	Synonym Generated from our automatic synonyms finder	Recall	Precision
---------------------------------	--	--------	-----------

For example, for the noun “الناس people”, the results were:

Synonyms Generated from Almaany	Synonym Generated from our automatic synonyms finder	Manual Recall	Manual Precision
انام , اناس , انس , بشر , خلق , قوم , وري , بشر , ساكن	ناس قوم السكان رسول	3/9 = 33%	3/4 = 75%

As mentioned previously, the ROUGE tool is used for automatic evaluation. The ROUGE precision and recall results of Almaany based evaluation were summarized in Figure 9. Figure 9 reveals that the precision was higher than the recall, this means that among the returned set of synonyms for a specific noun, the accuracy was significant (the number of correct synonyms to the number of automatically generated synonyms was high), but the system did not return the sufficient number of synonyms found in the Arabic language (the number of correct synonyms to the number of synonyms found in Almaany was low). Also, the majority of the precision values are confined between 0.4 and 0.5 (the average is 0.46 see Table 4), whereas the recall values fluctuated from 0.05 to 1.0, and the reason for that is the contents of Almaany reference set. Almaany reference sets contain one or two synonyms for certain nouns and contain more than 30 synonyms for other nouns.

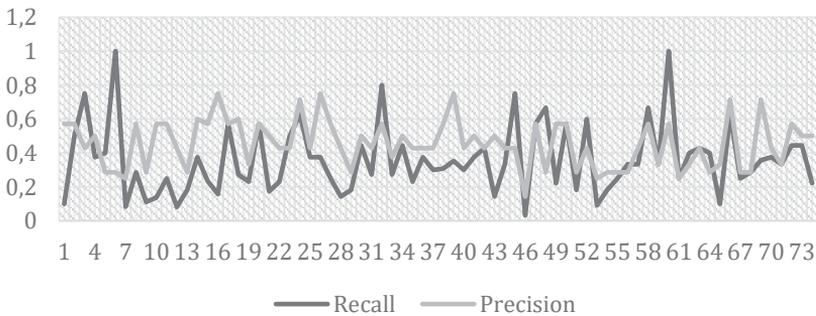


Figure 9. Recall and Precision – Almaany-based Evaluation

### 5.2.2. Google Translate-based Evaluation of the NBDV based on OWS Scheme

The problem we faced in Almaany-Based Evaluation was the contents of the synonyms sets. As described previously, Almaany combines six well-known and ancient dictionaries, and most of its vocabularies are not in use in today's newspapers and journals (the source of Kalimat Dataset). For example, in the synonyms list of the term "people" ("ناس"), the synonym "وَرَى" is not in use in today's language, and the synonym "خَلَق" is not used in these days to refer to "people" (it refers to anything created by God). Therefore, the search for a new source of synonyms that reflects the modern Arabic language is necessary.

Google Translate was the result of this search because it gives a list of modern synonyms for any Arabic term being translated. For example, the term used in our example above ("ناس" people) has the following synonyms in Google Translate: (folk قوم, people ناس, population سكان, society مجتمع, family أسرة). Google Translate uses a statistical machine translation approach to translate the Arabic language, and it collects the meaning and synonyms from a massive number of Arabic articles found on the internet at the time of translation. (Most of these articles are written in modern Arabic vocabularies).

Figure 10 shows the precision and recall curve based on GoogleTranslate synonyms set. In Figure 10, the values of the precision and the recall are convergent, and the recall is more stable comparing with Almaany based recall because the sets of synonyms that were appeared in Google Translate (the base of the comparison) are smaller than the synonyms set of Almaany dictionary and roughly contain the modern Arabic terms. Also, the size and contents of Google Translate synonyms set positively affected the final average precision and average recall.

Table 4 shows the final recall and precision values that were obtained in this experiment. Table 4 shows that the obtained average precision and average recall based on GoogleTranslate are higher than their corresponding values based on Almaany dictionary and the reason for that is the type of language used in GoogleTranslate that uses the same language used in our corpus. Thus, the obtained average precision(51%) and average recall (47%) in GoogleTranslate based evaluation are more significant than the obtained results in the Almaany based evaluation (5% improvement on average precision, 11% improvement on the average recall).

Figure 11 represents the average recall trend  $A(R|i)$ . The vertical axis represents the  $A(R|i)$  values and the horizontal axis represents the id of the processed noun. Figure 12 represents the average precision trend  $A(P|i)$  at each run of the VSyn program. The vertical axis represents the  $A(P|i)$ , and the horizontal axis represents the id of the processed noun. Both figures show the stability of the recall and precision values. In Almaany based evaluation or GoogleTranslate based evaluation, the recall and precision curves converge to their final average values after processing few numbers of nouns (roughly after the noun number 40). The fast convergence gives a good indication of the accuracy of the generated answer set by the VSyn program. If the recall and precision were very low in some cases and very high in other cases (large fluctuations) this requires the process of a large number of nouns to see the

stability. The indication that can be obtained from the curves that in most cases, the precision and recall were close to their average. Also, it is important to note that the recall and precision were higher in the GoogleTranslate based evaluation than in Almaany based evaluation.

Evaluation Type	Weighting Scheme	Average Recall	Average Precision
<b>Almaany Based Evaluation</b>	OWS	36%	46%
<b>Google Translate Based Evaluation</b>	OWS	47%	51%
<b>Google Translate Based Evaluation</b>	Tf.idf	36%	41%
<b>Expert and Speakers Evaluation</b>	OWS	N/A	57.5%

Table 4. Average Precision and Average Recall Using Three Evaluation Strategies

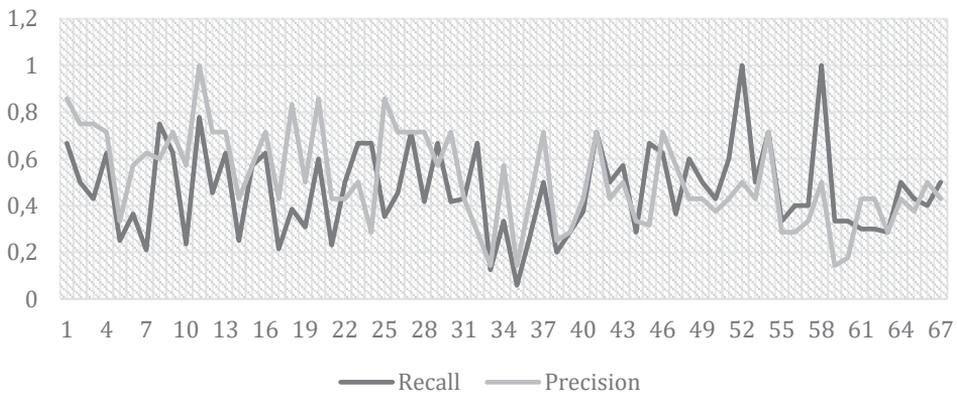


Figure 10. ROUGE Recall and Precision - GoogleTranslate-based Evaluation

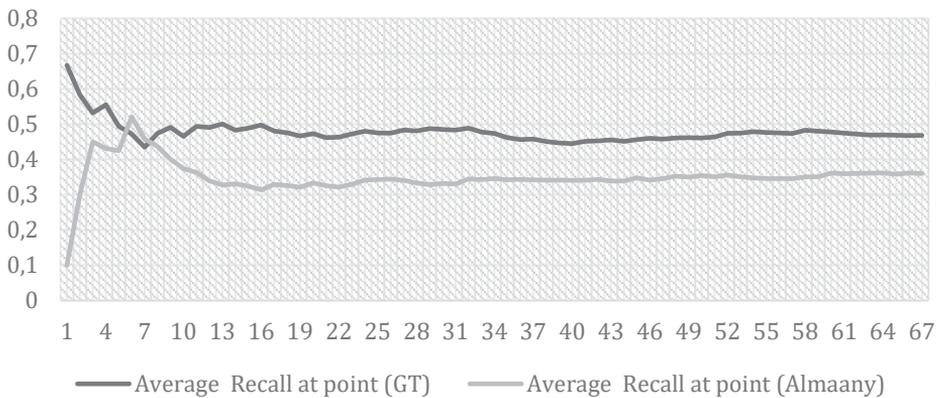


Figure 11. Average Recall Trends at every Noun Processed

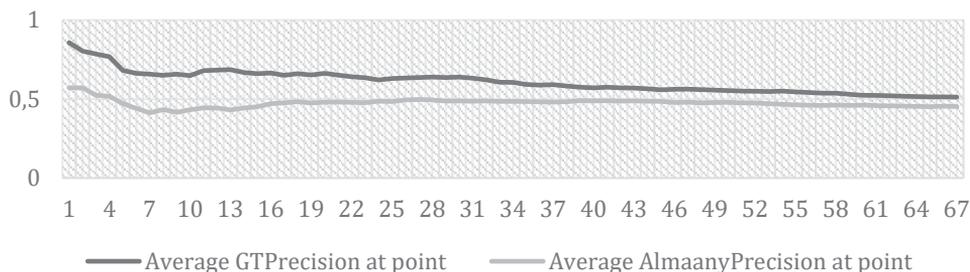


Figure 12. Average Precision Trends at every Noun Processed

5.2.3. Relevancy Comparison between OWS and Tf.idf

This subsection presents the recall and precision evaluation for the NBDV based on the tf.idf weighting scheme, and it offers a comparison between these results and the results that were obtained in the previous section. The precision and recall results based on the GoogleTranslate synonyms set are more significant than the precision and recall results based on the Almaany. Thus, our comparison in this subsection is based on the synonyms set of Google translate as the optimal answer.

Back to Table 4, The NBDV evaluation based on tf-idf achieved 36% average recall and 41% average precision. Table 5 shows a sample of the obtained recall and precision results by the two scheme experienced in this paper. For the most nouns that are mentioned in Table 5, the recall and precision values of the OWS scheme is greater than their equivalent values in the tf.idf scheme, and the difference is significant.

Noun	Recall		Precision	
	OWS	tf.idf	OWS	tf.idf
اجتماع Meeting	0.4167	0.25	0.7143	0.42857
احتفال elebration	0.36364	0.36364	0.57143	0.57143
اعتداء Assault	0.41667	0.16666	0.71429	0.28571
افتتاح Opening	0.21429	0.21429	0.42857	0.42857
انتصار Victory	0.5	0.5	0.42857	0.42857
اولاد Children	0.42857	0.71429	0.375	0.25
بحث Research	0.57143	0.57143	0.57143	0.57143
بطولة Championship	0.23529	0.23529	0.57143	0.57143
بيانات Data	0.66667	0.33333	0.5	0.28571
تاريخ History	0.7500	0.5	0.6000	0.4
جائزة Award	0.3	0.2	0.42857	0.28571

Table 5. Precision and Recall Values for the Automatically Generated Synonyms for the First 10 Nouns (Based on OWS and tf.idf Weighting Schemes)

To trace the complete behavior of the tf.idf recall and precision results, we redraw Figures 11, 12 to reflect a comparison in the trend of the tf.idf recall and precision with the OWS recall and precision. Figure 13 shows the recall trend, and Figure 14 shows the precision trend. From left to right, the trend becomes more stable, and the curves converge to their final mean. The two figures show that the OWS relevancy measurements overtake the tf.idf relevancy measurements.

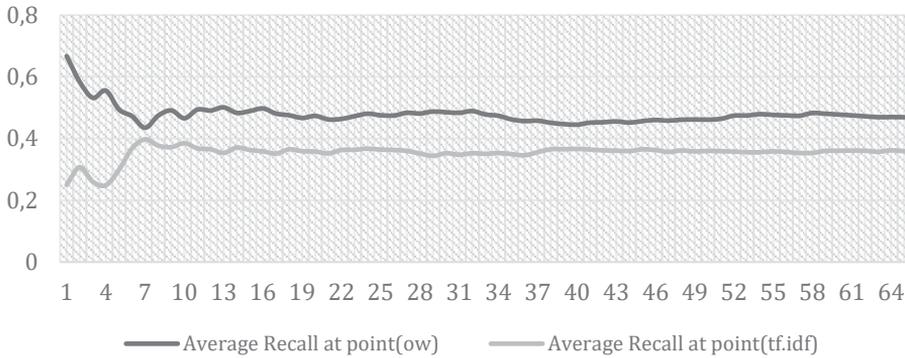


Figure 13. the Trend of the Average Recall Based on OWS and Based on tf.idf

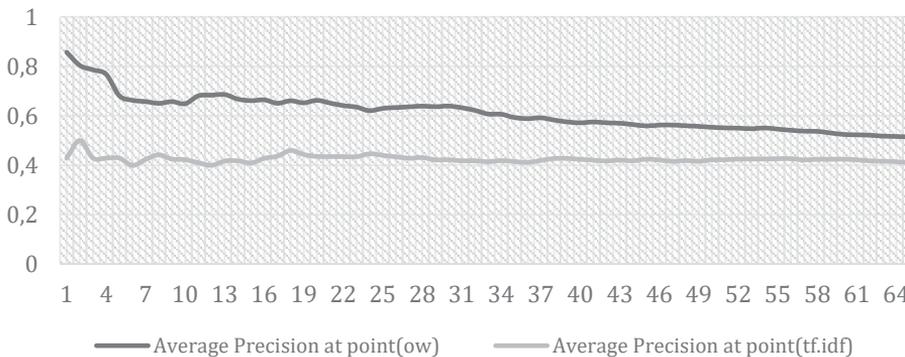


Figure 14. the Trend of the Average Precision Based on OWS and Based on tf.id

#### 5.2.4. Evaluating using Arabic language experts

The results of the OWS recall and precision revealed a significant performance of our system, but it showed that 64% of the synonyms are not returned in Almaany based evaluation. The explanation was that the nature of the base data that were used in the evaluation. However, it is necessary to measure the satisfaction of the native speakers of the modern Arabic language about the accuracy of the synonyms sets returned by the NBDV method. The use of human evaluators has been reported in many research publications [7], [8], and [43].

Two Arabic language experts and four Arabic language speakers from Prince Sattam bin Abdel Aziz University in Saudi Arabia voluntarily evaluated our system. The evaluation was conducted in two groups; each group evaluated 82 nouns' synonyms. From their experience, the experts wrote the number of correct synonyms

that were generated from our system for every noun. The precision was redefined as follow:

*Precision*

$$= \frac{\text{the number of automatically generated synonyms that the expert agreed their correctness}}{\text{total number of automatically generated synonyms of the noun}}$$

Group one of nouns containing 81 noun		Group two of nouns containing 83 noun	
Expert name	Precision	Expert name	Precision
Adeel(expert)	55%	Sadam(expert)	50%
Firas(speaker)	62%	Bassam(speaker)	57%
Sana(speaker)	60%	Nour(speaker)	61%
<b>Average</b>	59%	<b>Average</b>	56%
<b>Average</b>	57.5%		

Table 6. the Average Precision for the Manual Evaluation

Table 6 shows the manual evaluation results. The experts' evaluation showed that the precision was significant; ranging from 50% to 62%. The experts' precision proves the precision values that were gained from GoogleTranslate -based evaluation. Two experts, Sadam and Adeel are Arabic Language specialists who gave 55% and 50% of satisfaction. The other evaluators are native Arabic speakers, and their rate of satisfaction ranges from 57% to 62%.

In [8], [43] the manual evaluation yields 76%, 80% average precision, respectively. The manual evaluation depends on the assessors' point of view and the assessors' knowledge. In the manual evaluation accomplished in this research, the difference between the experts and the native speakers' decisions in whether the words are related words (hyponym, hypernym, plural) or synonyms affected the final precision. For example, for the noun "نوع type" the system returned "نوع, نمط, طراز, رمز symbol" as synonyms, Sana wrote five as correct synonyms, excluded "رمز symbol" whereas Adeel wrote four, and she excluded "رمز symbol" and "فرق sects". Adeel considered "فرق sects" as a hyponym of the word "نوع type". Another difference between the speakers' judgment and the experts' judgment was the plural, the speakers considered the plural as synonyms, for example, that VSyn generated the following synonyms for the word "area":

منطقه	منطقه	مدینه	شباب	ولایه	ولايات	مناطق	عديد
Area	Area	City	youth	State	States	Areas	numerous

Nour considered "areas" as synonymous to "area", whereas, Sadam excluded it. Sadam also excluded the word "states".

### 5.2.5. Recall and Precision Final Results

Table 4 depicted that in Almaany based evaluation, the recall was somehow low (36%), but the precision was significant (comparing with the precision values obtained from the publications in the field as shown in Figure 18). The recall reflects the fact that among all the synonyms found for a noun, 36% of them were returned, whereas the precision demonstrates the fact that among the returned synonyms by our system 46% of them were relevant. Both the Google Translate based evaluation and the manual evaluation showed that more than half of the generated elements in the answer set are correct synonyms, and the Google Translate based evaluation showed that around half of the synonyms found in Google Translate were retrieved by our method.

Regarding the tf.idf recall and precision values, which should be compared with the Google translate OWS recall and precision values, Table 4 shows 11% difference in the recall and 10% in the precision. This difference is significant, and it proves our claim that the accuracy of the synonyms extraction improves by linking the nouns to their semantic context.

### 5.3. Comparison with State-of-the-art Models

The performance of the NBDV model should be compared against the performance of the state-of-the-art models in the same field, such as the CBOW, SG, and GloVe. The evaluation results of these models were mentioned in [10], and to unify the experiment conditions, we used the same dataset that was used [10], and we used the same word similarity tasks and the same statistical models. The following points summaries the aspects we consider to unify the experiment conditions:

- In [10], the authors performed a series of experiments that include: the word analogy task, the words similarity, and the use of GloVe in NER. And, because the OWS is a weighting scheme whose effect directly affected the word similarity between two nouns in the NBDV model, then we choose the comparison of the word similarity to be the base of the comparison between our model and the state-of-the-art models.
- In [10], the results showed that the GloVe model overtook the state-of-the-art models, such as the SVD, SVD-S, SVD-L, CBOW, and SG. Therefore, we need only to compare the word similarity between our model and the GloVe model.
- In [10] the authors used five datasets of the English language and obtained the most significant results on the largest corpora (Common Crawl corpus), which contains 42 billion terms, as appeared in table 3 in [10]. Thus, we chose this corpus in our comparison experiment. Also, the authors employed five word-similarity models, WordSim-353 [49], MC [50], RG [51], SCWS [52], and RW [53]. From these models, we chose the recent two tasks, the SCWS, and RW.

The Spearman rank correlation results of the word similarity models over the generated similarities from the NBDV and GloVe appears in Figure 15. The results

showed that our model’s Spearman rank correlation was less than the GloVe Model’s Spearman rank correlations in SCWS by 0.2 and greater than the GloVe Model’s Spearman rank correlations by 0.5 in RW similarity. These results are roughly convergent, but the time efficiency improvement of the NBDV is more significant as it appears in the next section.

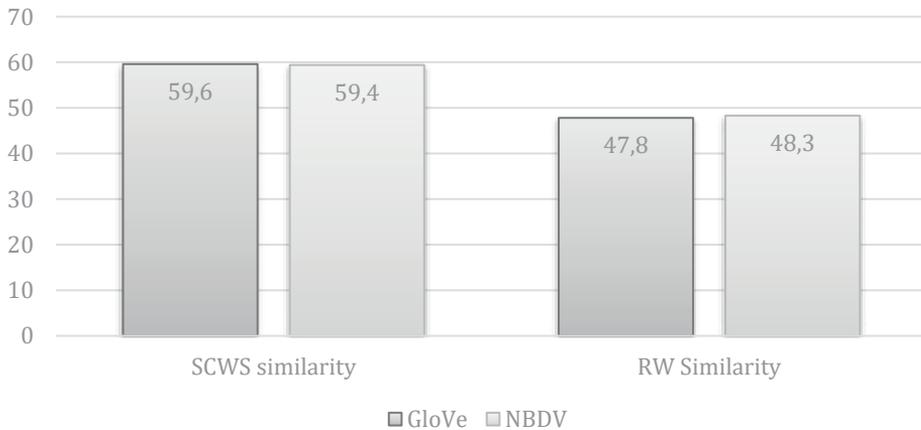


Figure15. The Spearman Rank Correlation Results of the Word Similarity Models over the Generated Similarities from the NBDV and GloVe.

#### 5.4. Time Complexity Analysis

Assuming that the number of nouns in the whole corpus is  $N$ , the number of verbs in the entire corpus is  $v$ , and the number of all terms in the corpus is  $n$ . According to the NBDV method, to find the synonyms of the noun  $x$ , the necessary computational steps are listed in Table 7:

The worst-case time complexity in step 8 is  $O(N.v.n)$ , this occurs if  $y = v$  and  $j = N$ , the meaning of  $y = v$  is that all the verbs in the dataset appeared with the noun  $x$ , and the meaning of the  $j = N$  is that all the nouns in the dataset are sharing the set of verbs stored in  $S_{nv}$ . Actually, these conditions are impossible to happen because in real languages we cannot find one verb that comes with all the nouns or all the nouns share a specific set of verbs, so we can consider  $y, j$  as constant, which means that the total complexity of the NBDV method will be  $O(j.y.n) + O(j.n) \Rightarrow O(n)$ . Figure 16 supports our claim regarding the  $y$  possible value, the number of shared verbs processed by the OWS in every run of the NBDV system was counted, and the maximum obtained value of  $y$  is 829, the average value of  $y$  is 186 verb, and in 63% of the runs, the value of  $y$  was less than 200. Regarding  $j$ , the maximum value of  $j$  recorded in our experiment was 521. Figure 17 shows that in 72% of the NBDV runs, the number of processed nouns is less than 100, and a very small ratio of runs processed a high number of nouns (10% of runs processed more than 200 nouns).

In the NBDV method that hires the traditional tf-idf weighting scheme, the worst case is always occurred, which means that step number eight will always need  $O(N.v.n)$ . Also, the worst case in the Synonyms Detection phase will always be  $O(j.n)$ . In addition, in the tf.idf version of the NBDV model, the values of  $n$  and  $v$  are significant, and we can not ignore them ( the ratio of  $v$  in Kalimat is 30%, and the ratio on  $n$  is 56%).

The NBDV operation		Expected time	Description
OVS phase	1) Extracting the set of verbs $S_{nv}$ adjacent to the noun $n$ , assume that the number of extracted verbs is $y$	$O(n)$	Scanning all the corpus elements with $n$ number of terms.
	2) Computing each parameter mentioned in equations 2,4, and 5	$O(n)$	
	3) Computing the OWS weight	$O(1)$	
	4) Constrcting of the $\vec{x}$ , Repeat step 2 and 4 for each verb $\in S_{nv}$	$O(y.n)$	In the worst-case $O(v.n)$ If $v = y$ .
	5) Computing the orbit range (orbit in equation 8)	$O(1)$	
	6) Dividing the $\vec{x}$ vector components in sets according to the value of orbit and create the $s1,s2,s3,s4$	$O(y)$	
	7) Take $s1, s2, s3$ and extract all the distinctive nouns $x_i$ adjacent to verbs appeared in $s1, s2, s3$ . Assume the number of extracted nouns is $j$	$O(y.n)$	In the worst-case $O(v.n)$ If $v = y$ .
	8) Repeating steps 2,3, and 4 for each noun extracted in 7, computing $\vec{x}_i$	$O(j.y.n)$	In the worst case $O(N.v.n)$ If $y = v$ and $j = N$
Total OVS Complexity		$O(j.y.n)$	The max values of the complexity from step 1 - 8
Synonyms Detection	9) Computing $\text{sim}(\vec{x}, \vec{x}_i)$ , equation 8	$O(j)$	
	10) Extracting the synonyms, if $\text{sim}(\vec{x}, \vec{x}_i) > 0.18$	$O(j.n)$	In the worst case $O(N.n)$ If $j = N$ .

Table 7. Time Complexity Analysis of the NBDV model

Table 7 and Figure 16, 17 depicted the time analysis of the NBDV method and showed the achieved efficiency improvement. To be more accurate, the average number of verbs processed was 186 verbs for each noun, and the maximum number of verbs processed was 839 for the noun “عمل work”. Thus, in our method, the

determination of the semantic relations between a specific noun and the other nouns found in the corpus is performed by processing (weighting) 186 verbs appeared with that noun. Comparing with the tf.idf [17], the improvement in the time consuming came in the terms weighting step and in the similarity computations between the nouns.

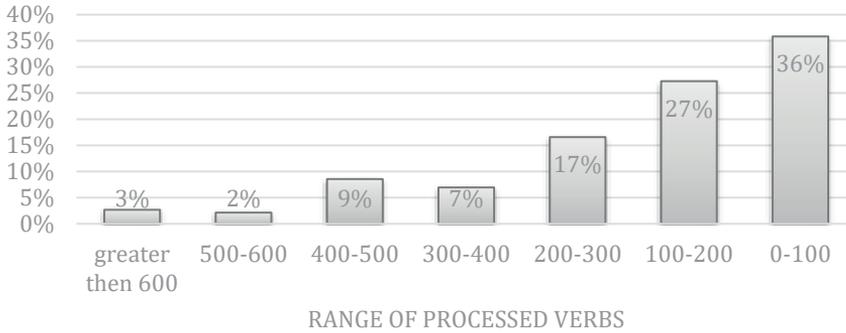


Figure16. The Ratio of Processed verbs in every Run of 564 Runs

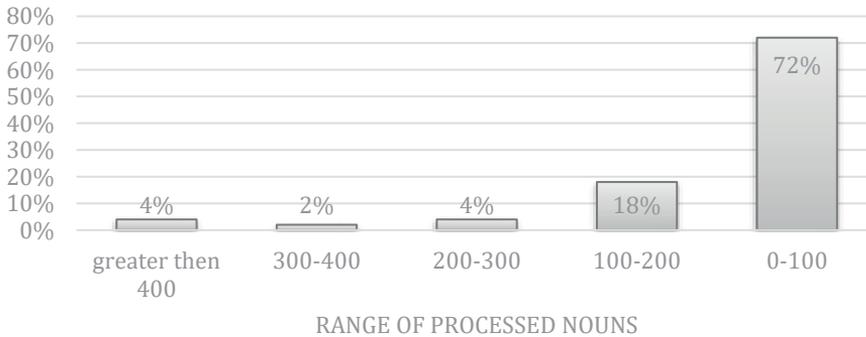


Figure 17. The Ratio of Processed Nouns in every Run of 564 Runs

### 5.5. Relevancy Comparison with Other Publications in the Field

In the previous section, the time complexity analysis of the NBDV methods showed that the method needs  $O(n)$  to extract the synonyms of a specific noun. However, an accuracy comparison of our method with other publications in the field is necessary to show that we have not enhanced the efficiency at the expense of precision. Figure 18 presents a comparison between the results mentioned in Figures 9, 10, 11, 12, and Table 4, with other research in this field. In [9], [7], [38], and [6] the precision calculations are presented in a clear manner. The authors of these references

experienced different approaches of synonyms extraction (Henriksson used enhanced distributional hypothesis model, Leeuwenberga used Statistical approaches with relative cosine similarity, Lonneke used machine translation approach, Lobanova used learning approach, and Minkov used graph-based approach)

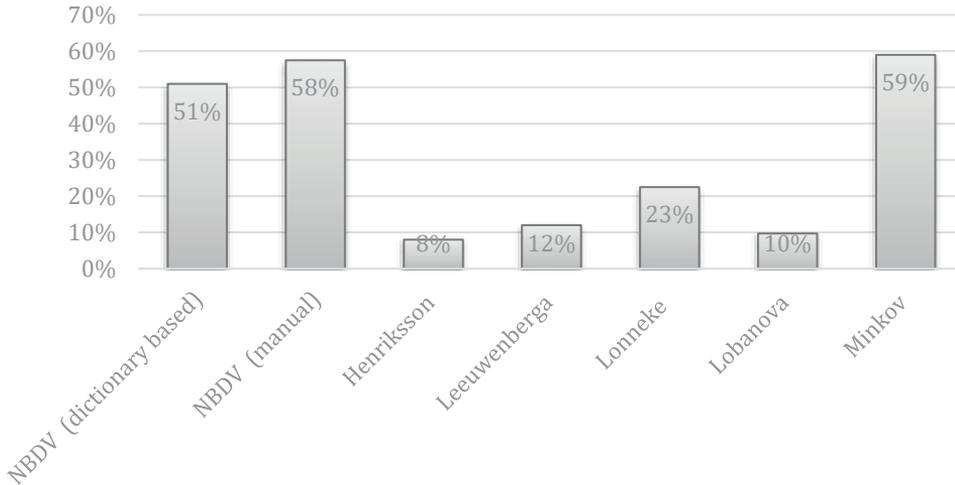


Figure 18. Precision Comparison with Existing Synonyms Extraction Systems

Figure 18 reveals that the precision of the NBDV was significant compared with other statistical methods used for synonyms extraction. The VSyn system obtained 51% average precision in the dictionary-based evaluation in which the automatic synonyms were matched against the synonyms taken from online dictionaries. This precision value was less than the precision obtained by Minkov and Cohen [6] by 8% and greater than the precision of the other systems. Minkov and Cohen [6] used a path constrained graph, and the problem with this graph is the time required to construct the graph and the space needed to store the graph. The graph stores each term in the corpus with all existing edges that link this term to the other terms found in the corpus, add to this the time needed to follow all the paths that lead to the terms. So, the improvement in the time obtained in the NBDV method is much more important than the 8% loss of accuracy, especially that the precision of the NBDV method was more than 50%.

## 6. Discussion

### 6.1. Factors Affected the Precision

The evaluation of the NBDV model based on the OWS weighting scheme shows significant efficiency and accuracy improvement. And, we have strong thought that the relevancy results should be more significant, but some factors affected the

extraction process and caused a kind of degradation in the relevancy values. These factors are summarised in three factors:

The first factor was the existence of a set of nouns that are not synonyms (sometimes antonyms) and shared a set of distinctive verbs. For example, the direction names (north, south, east, and west), the month names, and the currency names. To see the effect of such kind of nouns, consider the generated synonyms set for the noun “جنوب South”, the VSyn system produced the following candidate synonyms:

جنوب      شمال      شماليه      شرق  
 south   north   ...   ...   ...   north   ...   ...   ...   east

The nouns in the sample above represent the four directions names, and they are not synonymous to each other, but the question is why the system returned them. To answer this question, we returned to our results and scanned the verbs appeared with the noun south, see the following samples from our corpus:

<p>تهب الرياح شمالا                  شق طريقه الى الشمال                  يمتد من شمال الكوكب                  تقع شمال الصين                  يبعد نحو 77 كيلو متر الى الشمال                  يبدو في شمال الكرة الارضية                  تتجه نحو الشمال                  تظهر شمال                  التي كانت تتمكن شمالا                  تبدأ قافلة الشمال الاول بدخول</p>	<p>The wind <u>blows</u> to the north  <u>Cleave</u> its road to the north  <u>Extend</u> from the north of the planet  <u>located</u> in the north of China                  about 77 km to the north  <u>Appear</u> in the North of the global                  heading to the north  <u>appeared</u> in the north                  Which was <u>stationed</u> north                  the first north convoy <u>starts</u> to enter</p>
--	---

The replacement of the word north with south, east, or west in these sentences is acceptable, and lexically the sentences remain correct. This means that the verbs (underlined above) are common for these nouns (north, south, east, west) and they have a convergent effect on the weighting computation. Therefore, the NBDV method considered north, south, east, and west as synonyms. The verb “تتجه head” appeared with these four nouns and got significant weights. The weight of this verb with respect to the north was 0.32, and with respect to the south was 0.283, and with respect to the east was 0.2942, and with respect to the west was 0.2678. These weights and the weights of other common verbs produced high similarity between the nouns north, south, east, and west.

The second factor that degraded the precision in the case of the dictionary-based evaluation was the synonyms set of the base dictionary; in some cases, it was empty. See Table 8:

Id	Noun	Almaany Set of synonyms	The Automatically generated synonyms
N6	دينار	Not Mentioned	بيورو,سنه,طريق,وان,ريال,طاقه,,دينار
N17	شركه	Not Mentioned	شركه,شركات,وزاره,منظمه,مؤسسه,عام,مشاركه

N204	بيانات	Not Mentioned	بيانات, معلومات, نتائج, طلاب, كافه, نسبه, مصادر
------	--------	---------------	---

Table 8. Sample of the Nouns that have Empty Synonyms Set in Almaany-based Evaluation

The nouns “دينار Dinar”, “بيانات data”, “شركة corporation” did not have synonyms in Almaany dictionary. In such a case, the precision computation is not possible because no base set of synonyms is found, so we neglected them during the precision computation. The negligence in such cases distorts the results (appeared in Figure 9 and 10) because the precision for the same nouns in the manual evaluation that was accomplished by the Arabic language experts showed reasonable values. See Table 9:

ID	The Automatically generated synonyms	Adeel Evaluation precision	Firas Evaluation precision	Saddam Evaluation precision
N6	يورو, سنه, طريق, وا, دينار, ريال, طاقه	43%	43%	43%
N17	شركه, شركات, وزاره, منظمه, مؤسسه, عام, مشاركه	57%	86%	71%
N204	بيانات, معلومات, نتائج, طلاب, كافه, نسبه, مصادر	57%	71%	57%

Table 9. Precision Evaluation of the Nouns Appeared in Table 8

The third factor that also affected our precision value was the mistakes found in the part of speech tagging produced in the Kalimat dataset, for example, the candidate synonyms set produced by our systems for the noun "الدستور constitution" contains the following: { قرار, قانون, دستور, ايضا, كلمه, ثلاثه, مشروع }, the word "ايضا" is a Stopword means “also” or “as well”. The mistake in this case directly affected the precision because it happened in the synonyms set. Also, some tagging mistakes affected the weighting phase of our system, for example, the verbs list of the noun "حروب" contains the noun “دخان smoke”.

## 6.2. NBDV Advantages and Disadvantages

The NBDV method of synonyms extraction has the following merits:

- **The semantic investigation of the text contents:** the NBDV synonyms extraction method used the distinctive verbs to find the semantic relations that link the nouns with their semantic context. The OWS draws an orbiting space in which the verbs are spin in orbits around the noun located in the center of this orbiting space. The verbs that have strong semantic relation to the center will be placed in the inner orbits, and the first three orbits will be the most important orbits to determine the nouns' semantic context.
- **The efficiency constraints are considered in the NBDV method.** Note that the design of the NBDV involved the use of a new weighting scheme (OWS)

that substituted the traditional tf-idf weighting scheme used in the CBoW and SG models [7], [17], [18]. The tf-idf weighting scheme in ASE is time-consuming and takes  $O(N.v.n)$  as we see in the evaluation section, and the OWS scheme processes the weighing phase in linear time (see Figures 16, 17, and Table 7).

However, the NBDV have the following limitations:

- **NBDV does not solve the synonyms of verbs and adjectives.** In the design of the NBDV for synonyms extraction, we used the verbs to find the synonyms of the nouns, but at this stage, we can not pretend that if a reverse process can help to identify the verbs' synonyms, and we will transfer this question to future research.
- **The NBDV model under the Algorithm 1 design cannot process the word orders that end with the verbs (OSV, SOV).** The OWS scheme, upon which the NBDV method is built, assumes that the verbs precede the nouns (Subject, Object). This drawback prohibited the use of the NBDV in certain languages such as Hindi, Japanese, and Korean. However, simple modifications in the design of Algorithm 1 will cover this problem. For example, building a co-occurrence matrix of the corpus and processing the verbs that precede or follow the noun.
- **The low obtained recall for the ancient nouns of the Arabic language.** The ancient nouns are important because they are found in AL Quran Al Kareem, which is the source of the Islamic religion. The system developed based on the NBDV method succeeded to return only 36% of the synonyms found in the Almaany dictionaries, and this reflects the gap between the shallow vocabulary set used in the Arabic language media and the rich vocabulary set found in the Arabic language literature.

## 7. Conclusion and Future Perspective

### 7.1. Achievements

In this research, an automatic synonyms extraction method is proposed, designed, and implemented. The method uses an unsupervised learning strategy to extract nouns synonyms. The method substituted the traditional tf-idf weighting scheme with an efficient weighting scheme that weights the terms based on their semantic relation with the noun being processed. The targeted contribution of this research is to improve the efficiency of the extraction process and at the same time, obtaining significant precision. This contribution is achieved improving by the following achievements,

**Robust weighing scheme:** the evaluation of the OWS precision and recall showed that the OWS overtakes the traditional tf.idf weighing scheme. The OWS recall was greater than the tf.idf recall by 11%, and the OWS precision was greater than the tf.idf precision by 10%.

**Efficiency improvements:** the average number of terms needed to be processed for each run was 186 (instead of the processing of the whole terms found in the

corpus). This average number is supported by a time complexity analysis that showed that the processing of each run of the NBDV method is accomplished in linear time. This point shows that the semantic context linking was significant and increased the extraction accuracy.

**Significant precision:** the average precision, which was evaluated based on well-known online dictionaries for the Arabic language, was significant (51%), and human experts who showed that 57.5 of the answer set contents are correct proved this precision.

The drawback that emerged during the evaluation was the low recall obtained when we used the Almaany dictionary as the base of evaluation. The NBDV method succeeded to return only 36% of the synonyms found in the Almaany dictionaries, and this reflects the gap between the shallow vocabulary set used in the Arabic language media and the rich vocabulary set found in the Arabic language literature.

## 7.2. Future perspective

The modification and enhancement of the OWS to find the verbs synonyms are necessary, our plans involve the exploiting of the OWS semantic context to investigate the verb synonyms. Also, measuring the effect of the OWS schemes in other text mining applications such as the named entity recognition, pattern recognition, and information retrieval is important to generalize this scheme. And, If the obtained results resemble the results achieved in the synonyms extraction field, then the OWS can replace the traditional tf-idf weighting scheme in the text mining applications which will improve the efficiency of such applications.

## 8. Compliance with ethical standards

**Conflict of interest:** The authors declare that they have no conflict of interest.

**Ethical approval:** All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards.

**Informed consent:** Informed consent was obtained from all individual participants included in the study.

## References

- [1] S. Webb, "The effects of synonymy on second-language vocabulary learning," *Reading in a Foreign Language*, vol. 19, no. 2, pp. 120-136, 2007.
- [2] H. Schütze, C. D. Manning and P. Raghavan, *Introduction to information retrieval*, vol. 39, Cambridge University Press, 2008.
- [3] L. Barak, I. Dagan and E. Shnarch, "Text Categorization from Category Name via Lexical Reference," in *NAACL-Short '09 Proceedings of Human*

- Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume, Boulder, 2009.
- [4] D. Dinh and L. Tamine, "Identification of concept domains and its application in biomedical information retrieval," *Information Systems and e-Business Management*, vol. 13, no. 4, p. 647–672, 2015.
- [5] M. AlMaayah, M. Sawalha and M. Abushariah, "Towards an automatic extraction of synonyms for Quranic Arabic," *International Journal of Speech Technology*, vol. 19, no. 2, 2016.
- [6] E. Minkov and W. W. Cohen, "Graph based similarity measures for synonym extraction from parsed text," in *Workshop Proceedings of TextGraphs-7 on Graph-based Methods for Natural Language Processing Association for Computational Linguistics*, Jeju, Republic of Korea, 2012.
- [7] A. Leeuwenberga, M. Vela, J. Dehdar and J. v. Genabith, "A Minimally Supervised Approach for Synonym Extraction with Word Embeddings," *The Prague Bulletin of Mathematical Linguistics*, vol. 105, no. 1, p. 111–142, 2016.
- [8] A. Benabdallah, M. Abderrahim and M. Abderrahim, "Extraction of terms and semantic relationships from Arabic texts for automatic construction of an ontology," *International Journal of Speech Technology*, vol. 20, no. 2, pp. 289-296, 2017.
- [9] A. Henriksson, H. Moen, M. Skeppstedt, V. Daudaravicius and M. Duneld, "Synonym extraction and abbreviation expansion with ensembles of semantic spaces," *JOURNAL OF BIOMEDICAL SEMANTICS*, vol. 5, no. 6, 2014.
- [10] J. Pennington, R. Socher and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014.
- [11] Z. He, S. Deng and X. Xu, "A Fast Greedy Algorithm for Outlier Mining," Ng WK., Kitsuregawa M., Li J., Chang K. (eds) *Advances in Knowledge Discovery and Data Mining. PAKDD 2006. Lecture Notes in Computer Science, Volume 3918. Sprin*, vol. 3918, 2006.
- [12] K. Church and P. Hanks, "Word association norms, mutual information, and lexicography," *Computational linguistics*, vol. 16, no. 1, pp. 22-29, 1990.
- [13] C. D. Manning and H. Schütze, *Foundations of statistical natural language processing*, MIT press, 1999.
- [14] L.-C. Yu, L. Lung-Hao, Y. Jui-Feng, S. Hsiu-Min and L. Yu-Ling, "Near-synonym substitution using a discriminative vector space model," *Knowledge-Based Systems*, vol. 106, pp. 74-84, 2016.

- [15] D. Inkpen, "A statistical model for near-synonym choice," *ACM Transactions on Speech and Language Processing (TSLP)*, vol. 4, no. 1, p. 2, 2007.
- [16] L. C. Yu, C. H. Wu, R. Y. Chang, C. H. Liu and E. Hovy, "Annotation and verification of sense pools in OntoNotes," *Information Processing & Management*, vol. 46, no. 4, pp. 436-447, 2010.
- [17] A. Imsombut, "A statistical approach for semantic relation extraction," in *In 2009 Eighth International Symposium on Natural Language Processing*, 2009.
- [18] T. Mikolov, K. C. Chen, G. Corrado and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *arXiv preprint arXiv*, vol. 1301, no. 3781, 2013.
- [19] A. Alsaeed, "A survey of term weighting schemes for text classification," *International Journal of Data Mining, Modelling and Management*, vol. 12, no. 2, pp. 237 - 254, 2020.
- [20] J. Pennington, S. Richard and M. Christopher, "Glove: Global vectors for word representation," in Pennington, Jeffrey, Richard Socher, and Christopher Manning. "Glove: GlobalProceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014.
- [21] G. Miller, R. Beckwith, C. Fel, . D. Gross and K. Miller, "Introduction to WordNet: An On-line Lexical Database," *International Journal of Lexicography*, vol. 3, no. 4, p. 235–244, 1990.
- [22] C. Fellbaum, "WordNet and wordnets," *Encyclopedia of Language and Linguistics*. Elsevier, pp. 665-670, 2005.
- [23] C. Fellbaum and P. Vossen, "Connecting the Universal to the Specific: Towards the Global Grid," *Lecture Notes in Computer Science*, vol. 4568, 2007.
- [24] C. Fellbaum and P. Vossen, "Challenges for a multilingual wordnet," *Lang Resources & Evaluation*, vol. 46, no. 2, pp. 313-326, 2012.
- [25] S. Elkateb, W. Black, H. Rodríguez, M. Alkhalifa, P. Vossen, A. Pease and C. Fellbaum, "Building a WordNet for Arabic," in *In Proceedings of the fifth international conference on language resources and evaluation (LREC 2006)*, 2006.
- [26] S. S. Scott and S. Matwin, "Text classification using Word Net hypernyms," 1998.
- [27] S. Bloehdorn and A. Hotho, "Boosting for text classification with semantic features," in *In Proceedings of the MSW 2004 workshop at the 10th ACM SIGKDD conference* , 2004.

- [28] L. Jianqiang, Z. Yu and L. Bo, "Fully Automatic Text Categorization by Exploiting WordNet," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2009.
- [29] Q. Luo, E. Chen and H. Xiong, "A semantic term weighting scheme for text categorization," *Expert Systems with Applications*, vol. 38, no. 10, pp. 12708-12716, 2011.
- [30] S. Bloehdorn, R. Basili, M. Cammisa and A. Moschitti, "Semantic Kernels for Text Classification Based on Topological Measures of Feature Similarity," *Hong Kong*, 2006.
- [31] P. Wang and C. Domeniconi, "Building semantic kernels for text classification using wikipedia," *Las Vegas*, 2008.
- [32] G. Evgeniy and M. Shaul, "Harnessing the Expertise of 70,000 Human Editors: Knowledge-Based Feature Generation for Text Categorization," *Journal of Machine Learning Research*, pp. 2297-2345, 2007.
- [33] C. J. Crouch, "An approach to the automatic construction of global thesauri," *Information Processing and Management*, vol. 26, no. 5, p. 629–640, 1990.
- [34] H. Chen and K. Lynch, "Automatic construction of networks of concepts characterizing document databases," *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETIC*, vol. 22, no. 5, pp. 885 - 902, 1992.
- [35] R. Lakshmi and S. Baskar, "Novel term weighting schemes for document representation based on ranking of terms and Fuzzy logic with semantic relationship of terms," *Expert Systems with Applications*, vol. 137, pp. 493-503, 2019.
- [36] I. Alsmadi and G. K. Hoon, "Term weighting scheme for short-text classification: Twitter corpuses," *Neural Computing and Applications*, vol. 31, p. 3819–3831, 2019.
- [37] D. Lin, S. Zhao, L. Qin and M. Zhou, "Identifying synonyms among distributionally similar words," in *IJCAI*, p. 1492–1493, 2003.
- [38] v. d. P. Lonneke and T. Jorg, "Finding Synonyms Using Automatic Word Alignment and Measures of Distributional Similarity," in *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, Sydney, 2006.
- [39] R. Ageishi and T. Miura, "Automatic Extraction of Synonyms Based on Statistical Machine Translation," in *2010 22nd IEEE International Conference on Tools with Artificial Intelligence*, Arras, France, 2010.
- [40] G. Grefenstette, *Explorations in Automatic Thesaurus Discovery*, Springer US, 1994.

- [41] P. P. Senellart and V. D. Blondel, "Automatic Discovery of Similar Words," in *Survey of Text Mining , Clustering, Classification, and Retrieval*, Springer, 2004.
- [42] G. Salton and M. J. McGill, *Introduction to modern information retrieval.*, McGraw-Hill, 1986.
- [43] L. Zhang, J. Li and C. Wang, "Automatic synonym extraction using Word2Vec and spectral clustering," in *Control Conference (CCC), 2017 36th Chinese*, Dalian, 2017.
- [44] Y.-L. Chen and Y.-T. Chiu, "An IPC-based vector space model for patent retrieval," *Information Processing & Management*, vol. 47, no. 3, pp. 309-322, May 2011.
- [45] Q. A. Al-Radaideh and D. Q. Bataineh, "A Hybrid Approach for Arabic Text Summarization Using Domain Knowledge and Genetic Algorithms," *Cognitive Computation*, vol. 10, no. 4, p. 651–669, 2018.
- [46] E. Hanandeh, "Building an Automatic Thesaurus to Enhance Information Retrieval," *IJCSI International Journal of Computer Science Issues*, vol. 10, no. 1, 2013.
- [47] Y. Sankarasubramaniam, K. Ramanathan and S. Ghosh, "Text Summarization Using Wikipedia," *Information Processing & Management*, vol. 50, no. 3, p. 443–461, 2014.
- [48] F. M. Ba-Alwi, G. H. Gaphari and F. N. Al-Duqaimi, "Arabic Text Summarization Using Latent Semantic Analysis," *British Journal of Applied Science & Technology*, vol. 10, no. 2, pp. 1-14, 2015.
- [49] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman and E. Ruppim, "Placing search in context: The concept revisited," *The concept revisited. ACM Transactions on information systems*, vol. 20, no. 1, pp. 116-131, 2002.
- [50] G. A. Miller and W. G. Charles, "Contextual correlates of semantic similarity," *Language and cognitive processes*, vol. 6, no. 1, pp. 1-28, 1991.
- [51] H. Rubenstein and J. B. Goodenough, "Contextual correlates of synonymy," *Communications of the ACM*, vol. 8, no. 10, pp. 627-633, 1965.
- [52] E. H. Huang, R. Socher, C. D. Manning and A. Y. Ng, "Improving word representations via global context and multiple word prototypes," in *In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, 2012.
- [53] T. Luong, R. Socher and C. Manning, "Better word representations with recursive neural networks for morphology," in *In Proceedings of the*

Seventeenth Conference on Computational Natural Language Learning, 2013.

- [54] A. Lobanova, J. Spenader, T. v. d. Cruys, T. v. d. Kleij and E. T. K. Sang, "Automatic Relation Extraction Can Synonym Extraction Benefit from Antonym Knowledge?," in 2009 Proceedings of WordNets and other Lexical Semantic Resources - between Lexical Semantics, Lexicography, Terminology and Formal Ontologies. , 2009.
- [55] R. Brandow, M. Karl and F. R. Lisa, "Automatic condensation of electronic publications by sentence selection," *Information Processing & Management*, vol. 31, no. 5, pp. 675-685, 1995.
- [56] T. Sakai and K. Sparck-Jones, "Generic Summaries for Indexing in Information Retrieval," in Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, 2001.
- [57] J. M. Perea-Ortega, E. Lloret, L. A. Ureña-López and M. Palomar, "Application of Text Summarization Techniques to the Geographical Information Retrieval Task," *Expert systems with applications*, vol. 40, no. 8, pp. 2966-2974, 2013.