

# Generating Sense Inventories for Ambiguous Arabic Words

Marwah Alian<sup>1</sup> and Arafat Awajan<sup>1,2</sup>

<sup>1</sup>King Hussein School of Computing Sciences, Princess Sumaya University for Technology, Jordan

<sup>2</sup>Information Technology College, Computer Science Department, Mutah University, Jordan

**Abstract:** *The process of selecting the appropriate meaning of an ambiguous word according to its context is known as word sense disambiguation. In this research, we generate a number of Arabic sense inventories based on an unsupervised approach and different pre-trained embeddings, such as Aravec, Fasttext, and Arabic-News embeddings. The resulted inventories from the pre-trained embeddings are evaluated to investigate their efficiency in Arabic word sense disambiguation and sentence similarity. The sense inventories are generated using an unsupervised approach that is based on a graph-based word sense induction algorithm. Results show that the Aravec-Twitter inventory achieves the best accuracy of 0.47 for 50 neighbors and a close accuracy to the Fasttext inventory for 200 neighbors while it provides similar accuracy to the Arabic-News inventory for 100 neighbors. The experiment of replacing ambiguous words with their sense vectors is tested for sentence similarity using all sense inventories and the results show that using Aravec-Twitter sense inventory provides a better correlation value.*

**Keywords:** *Word sense induction, word sense disambiguation, arabic text, sense inventory.*

*Received February 25, 2021; accepted March 7, 2021*  
<https://doi.org/10.34028/iajit/18/3A/8>

## 1. Introduction

Semantic similarity has an important role in different applications of Natural Language Processing (NLP) [4]. Ambiguous words affect the semantic similarity between two texts because the similarity score between texts depends on the similarity of their context words to determine if the two texts are similar or not [1, 23].

A single word that may have different meanings is called an ambiguous word, and the process of detecting the appropriate meaning of an ambiguous word is known as Word Sense Disambiguation (WSD) [17]. The context of an ambiguous word consists of the words surrounding the ambiguous target word.

The ability to define what a word means with respect to its meaning is one of the most difficult issues in NLP. Ambiguity is common across all languages, but it has greater challenges in Semitic languages such as Arabic [5].

According to Alian *et al.* [6], WSD approaches can be categorized into knowledge-based, supervised, unsupervised, and hybrid approaches. In knowledge-based approaches, the different meanings of an ambiguous word are extracted from a dictionary or a lexicon. Supervised approaches use training annotated corpus and testing sets, unsupervised approaches have no training set and instead use word context and clustering algorithms, and hybrid approaches merge the different methods.

One of the unsupervised approaches is word sense

induction, which represents words as a graph and then uses a clustering algorithm to group similar words in the graph. Each cluster is considered as a sense. Our research uses one of the word sense induction approaches to build a sense inventory for Arabic based on pre-trained embeddings. The sense inventory is used in WSD for sentences with ambiguous words from the Arabic paraphrasing benchmark [3]. The sense inventory is then evaluated using the retrieved senses in terms of accuracy measure [2].

Four sense inventories are generated and tested using Aravec-Twitter, Aravec-Wiki, Fasttext, and Arabic-News pre-trained embeddings. Another experiment is conducted to show the benefit of using the sense representation by replacing the sense representation in the sentence and comparing this new representation of the sentence by the use of pre-trained embedding of an ambiguous word in the sentence. Also, for these four sense inventories, an experiment is performed by replacing the vector of the retrieved sense instead of the ambiguous word vector to represent the sentence representative vector. Next, the similarity between sentences is measured and compared to the human evaluation using Pearson correlation.

This paper is organized in five sections as follows. Section 2 reviews the previously proposed work related to Arabic WSD. Section 3 explains the sense induction algorithm used for constructing the sense inventory while section 4 discusses the experiment and results. Then, section 5 presents the conclusion.

## 2. Related Work

Different approaches are proposed for Arabic WSD using word representation. For example, Alian *et al.* [5] used Wikipedia as a lexical resource and a Vector Space Model as a representational approach to texts. Cosine similarity is then used to measure the relatedness between Wikipedia's retrieved senses and the text that has an ambiguous word.

Hadni *et al.* [13] utilized two external resources, Arabic WordNet (AWN) and English WordNet (WN), to translate terms that cannot be found in AWN using a machine translation system. The nearest concept for the ambiguous word is chosen based on the number of relationships between concepts in the same local context. The authors evaluated their approach using naïve Bayesian and support vector machine. The proposed approach achieves an accuracy of 0.732 using Wu and Palmer's [24] measure with support vector machine.

Representing words as vectors in the distributional space has attracted researchers in different NLP applications and provided promising results. Arabic WSD based on word embedding is one of these applications. For example, Laatar *et al.* [16] proposed a WSD method based on word embedding where the word embedding is learned using the skip-gram model [19]. The similarity is measured between context vector and sense vectors, where the context vector is computed using word embeddings that appear in the context of an ambiguous word. The definitions of senses are retrieved from a dictionary, and the most similar definition vector to the context vector is selected as the appropriate sense. This approach achieves an accuracy of 0.78.

In addition, Alkhatlana *et al.* [7] utilized two embedding methods, Word2vec and GloVe, to generate global contexts of words and extract the synsets of ambiguous words from AWN. They constructed a test dataset to be used for the WSD task. The sense vector is obtained based on the retrieved AWN synset and then the cosine similarity between context vector and sense vector is computed. The most similar sense vector is considered as the correct sense for an ambiguous word.

Pelevina *et al.* [22] proposed an approach for learning word sense embeddings. This approach provides a sense

inventory from word embeddings by applying a clustering algorithm to an ego-network or word graph with related relationships. They used two WSD methods where the vectors for context words are taken from the matrix of word embeddings or the matrix of context representative vectors, the first method used the probability of sense in a context while the second method used the similarity between the sense representative vector and the context vector.

Then, Chang *et al.* [10] introduced an efficient graph-based approach for word sense induction which constructs global non-negative vector embeddings. Then they used clustering for the generated graph to get senses of each ambiguous word. The experiment was conducted using three datasets and the results show similar or better sense clusters compared to other methods such as Pelevina *et al.* [22].

Logacheva *et al.* [18] proposed a new unsupervised WSD approach based on the work of Pelevina *et al.* [22]. This approach depends on pre-trained embeddings and does not need any external annotated corpus. In this approach, a semantic graph is constructed for words in the vocabulary of the pre-trained embedding model and then the graph is clustered into subgraphs according to the similarity between word vectors. Each subgraph represents a sense. Next, a retrofitting approach is used to make the sense vector in the direction of the ambiguous word. The authors used Fasttext embeddings to build sense inventories for 158 languages, including Arabic.

A comparison between the previously discussed approaches that works with Arabic is given in Table 1. The comparison includes the authors, publication year, category of approach, corpus or dataset used, sense inventory, evaluation metric and results.

In this research, we apply the approach of Logacheva *et al.* [18] who used pre-trained embeddings of Aravec. WSD is then applied to 86 sentences from the Arabic paraphrasing benchmark using the senses extracted from sense inventories. The results are compared to the results of the senses retrieved from the Fasttext inventory.

Table 1. Comparison between approaches used for WSD.

Ref	Year	Approach category	Dataset/Corpus/embeddings	Sense Inventory	Similarity measure	Evaluation metric	Results
Alian <i>et al.</i> [5]	2016	Knowledge based	External dataset	Wikipedia	Cosine similarity	N/A	-
Hadni <i>et al.</i> [13]	2016	Knowledge based	Essex Arabic Summaries Corpus (EASC)	AWN, WN.	Wu and Palmer's	Accuracy	0.732
Laatar <i>et al.</i> [16]	2017	Semi-supervised	Historical Arabic Dictionary Corpus,	Almu-Jam-Alwasit dictionary	Cosine similarity	Precision	0.78
Alkhatlana <i>et al.</i> [7]	2018	Semi-supervised	collected from Arabic News	AWN	Cosine similarity	N/A	-
Logacheva <i>et al.</i> [18]	2020	Unsupervised	Fasttext pre-trained embeddings	Created from embeddings	Cosine similarity	N/A for Arabic	-

## 3. Word Sense Disambiguation

Polysemy is defined as the existence of many possible

meanings for a word. These meanings are called senses, and the word will be called a multi-sense word. It is one drawback of word representation, and it can be solved

using sense representation techniques. Our work will be based on context and sense representation to disambiguate a word and find the similarity.

WSD is used to find the proper meaning of a word with an uncertain meaning given its context [14, 22]. Owing to the ambiguity in human languages, a word may represent different meanings in two contexts. These meanings are identified in sense inventories in separate units, known as senses [15]. For example, the word “حكيم Hkym” in the following sentences has different meanings according to its context:

The boy went to a doctor for treatment.  
The man was wise in his choice.

The different senses of the word “حكيم Hkym” in Arabic are a person with wisdom ذو الحكمة, doctor الطبيب, and philosopher الفيلسوف. In the first sentence, the word “حكيم Hkym” means “doctor الطبيب” while in the second sentence it means “wise ذو الحكمة.” The context “for treatment” helps identify that the sense of “حكيم Hkym” is a doctor not a wise man.

The process of disambiguation of polysemy words depends on building a sense inventory to retrieve the senses of the ambiguous word and then selecting the most similar sense to the context of an ambiguous word. We benefit from the work of Logacheva *et al.* [18] in building an Arabic sense inventory using two different Aravec pre-trained embeddings. One is trained on Twitter and the other is trained on Wikipedia.

The algorithm of Logacheva *et al.* [18] consists of two main concepts: the first relies on graph-based word sense induction, and the second is graph filtering using vector operations for word vectors.

### 3.1. Word Sense Induction

Word sense induction depends on finding a list of nearest neighbors for word embedding in the distributional space. The method of constructing the semantic graph is as follows:

For each word  $w$  in the vocabulary:

- Construct the set of N-nearest neighbors (S) for the target word  $w$ . Let S members be  $\{s_1, s_2, \dots, s_n\}$ .
- Construct the set of N-anti-neighbors ( $\Delta$ ) that consists of words that are not similar to the corresponding nearest neighbors of  $w$ , where the vectors of these words is computed as the subtraction between the vector of word  $w$  and its neighbor  $s$ :  $(w - s_i)$ .
- Construct the set  $\bar{N} = \{\bar{s}_1, \bar{s}_2, \dots, \bar{s}_n\}$  that consists of the most similar words to the vectors in  $\Delta$ , but the result may be the target word  $w$  [17].
- The set of anti-pairs consists of  $(s_i, \bar{s}_i)$  but not the target word  $w (\bar{s}_i \neq w)$ .
- These anti-pairs are words that should not be connected in the graph unless both words  $s_i$  and  $\bar{s}_i$  are members in the set of N-nearest neighbors (S).
- Construct the set of vertices of the graph (V) by

adding words from the set (S) and their anti-pair from ( $\bar{N}$ ) only if the word and its anti-pair are part of the set (N) of the target word  $w$ . In other words, only add to the set (V) words that may benefit in separating different senses of  $w$ .

- Construct the set of edges (E): For each word  $s_i$  in the nearest neighbors (S), create a set of nearest neighbors ( $S^i$ ) =  $\{c_1, c_2, \dots, c_n\}$  and add the edge between word  $s_i$  and the nearest neighbor  $c_j$  if  $c_j$  is not an anti-pair of  $s_i$ .

There is no edge between a word  $s_i$  and its anti-neighbor in the graph because they belong to different senses.

Then a clustering algorithm is used to get the senses of an ambiguous word. These steps are shown in Figure 1, while clustering is described in the following section.

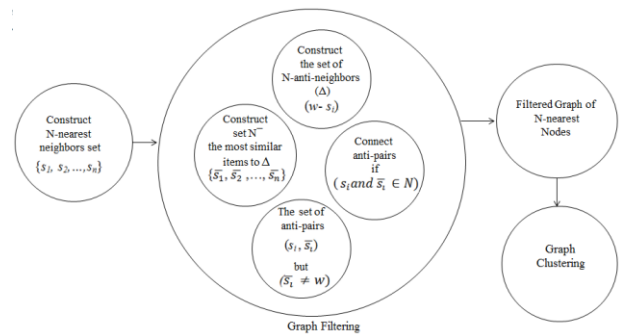


Figure 1. Steps of word sense induction and graph clustering.

### 3.2. Clustering

The constructed graph is clustered into subgraphs where each subgraph represents a sense of the target word. The average of the word embeddings in each subgraph represents the vector of the sense. Retrofitting is also applied to the sense vector.

Each cluster represents a sense of the target word, and the computed sense vector represents the keyword of that sense. Each sense of the target word with its keyword and cluster is saved to the sense inventory.

### 3.3. Disambiguation

Sense vectors are used for WSD in Arabic text by extracting the senses of the ambiguous word from the sense inventory and then computing the context vector by averaging the vectors of context words that are most similar to the ambiguous target word. The cosine similarity is computed between the sense vector and the context vector. Then the most similar sense will be selected as the correct sense.

## 4. Experiment and Results

To evaluate the disambiguation approach, four sense inventories are generated and tested on sentence similarity. The first one is generated from Twitter Aravec, the second one uses Wiki Aravec, the third sense inventory is generated using Arabic-News vectors trained by Altowayan and Tao [9], and the last one is

generated by Logacheva *et al.* [18] from pre-trained Fasttext vectors. The experiments are conducted to build sense inventories based on word semantic graphs with N-neighbors. N is tested for 50, 100, and 200.

#### 4.1. Dataset

Arabic paraphrasing benchmark [20] is used in the WSD experiment. This benchmark is constructed based on the transformation rules for Arabic [8, 11], such as permutation, deletion, and addition. These rules are applied to the structure of a sentence to produce a new sentence. The benchmark consists of 1010 sentence pairs labeled for similarity and paraphrasing. In our experiment, we used 86 sentences containing ambiguous words.

#### 4.2. Aravec Pre-Trained Embeddings

AraVec [20] is an Arabic distributed word embedding model that is trained using different resources and is available online with different dimensions. The word embeddings are obtained using the Word2vec skip-gram and Continuous Bag Of Words (CBOW) models [19].

Aravec-Twitter model is trained on Arabic tweets with a vocabulary size of 145,428 and dimensions of 100 and 300 for each word vector. The document size is 66,900,000.

Aravec-Wiki model is trained using 1,800,000 documents from World Wide Web pages with Arabic content. It has vector dimensions of 100 and 300 and a vocabulary size of 662,109.

#### 4.3. Fasttext Pre-Trained Embeddings

Arabic Fasttext embeddings are provided by Grave *et al.* [12]. These embeddings result from training on Wikipedia and Common Crawl corpus. They use an extension of the Fasttext model with subword information. This model is available online and has a dimension of 300 for word vector.

#### 4.4. Arabic-News Pre-Trained Embeddings

Altowayan and Tao [9] build their corpus from news articles with Arabic content based on local Arabic newspapers and the international Arabic news from CNN and BBC. They trained the corpus using the Word2vec CBOW model to learn word embeddings with a window size of 10 and a vector dimension of 300. The vocabulary size of the learned embeddings is 159,175.

#### 4.5. Results and Discussion

The retrieved senses are evaluated by an expert who provides each selected sense with a label as correct or incorrect. For ambiguous words that have no sense in the sense inventory, an unknown label is given. The number of target words to be disambiguated is 139.

Accuracy is measured as the correct senses from the

total senses, where the unknown senses are excluded from the total number of senses as in Equation (1):

$$\text{Accuracy} = \frac{\text{Correct senses}}{\text{Total senses} - \text{unknown senses}} \quad (1)$$

Tables 2, 3, and 4 compare the results of each sense inventory for 50, 100, and 200 neighbors, respectively, in terms of correct, incorrect, unknown senses, and accuracy.

Table 2. Results of sense inventories for 50-Neighbors.

Sense Inventory	Correct	Incorrect	Unknown	Accuracy
Aravec-twitter	45	49	45	0.479
Aravec-Wiki	22	30	87	0.423
Fasttext	56	68	15	0.451
Arabic-News	36	79	24	0.313

Table 3. Results of sense inventories for 100-Neighbors.

Sense Inventory	Correct	Incorrect	Unknown	Accuracy
Aravec-Twitter	36	83	20	0.303
Aravec-Wiki	38	70	31	0.352
Fasttext	42	77	20	0.353
Arabic-News	36	83	20	0.303

Table 2 shows that the accuracy of the sense inventory that is constructed based on Aravec-Twitter pre-trained embedding provides the best accuracy value of 0.48.

Table 3 shows that the accuracy of using Fasttext is better than that of Aravec-Twitter inventory for 100 neighbors, but it is similar to the accuracy achieved by the Aravec-Wiki inventory, while the accuracy of Arabic-news inventory is similar to that of Aravec-Twitter inventory.

The results of 200 neighbors in Table 4 show that the accuracy of the Aravec-Twitter-based inventory provides the least number of unknown senses. Fasttext-based inventory provides a very similar accuracy value as the Aravec-Twitter inventory. Arabic-news pre-trained embeddings provides the lowest accuracy value of 0.255.

Table 4. Results of sense inventories for 200-Neighbors.

Sense Inventory	Correct	Incorrect	Unknown	Accuracy
Aravec-twitter	60	69	10	0.465
Aravec-Wiki	53	72	14	0.424
Fasttext	54	62	23	0.466
Arabic-News	28	82	29	0.255

#### 4.6. Word Sense Disambiguation

We use 86 sentences with ambiguous words from the Arabic paraphrasing benchmark [3] to evaluate WSD with the retrieved senses from the sense inventory.

We apply the algorithm of Logacheva *et al.* [18] to the pre-trained word embeddings, that have been used to construct the Arabic sense inventories, for the disambiguation process.

The two Aravec models are Twitter and Wikipedia. The Twitter model has 1,476,715 vocabularies, but there is a limit for the vocabulary used in the experiment of Logacheva *et al.* [18]. The Wiki model has 662,109 vocabularies. The experiments show that the number of

vocabulary affects the clusters of each sense.

Sense inventories are generated from Aravec, Arabic-News embedding vectors with N-neighbors (50, 100, and 200). The keyword of each sense cluster is generated as in the approach of Logacheva *et al.* [18] by determining the centroid of the cluster as the mean of word vectors that belong to the sense cluster. The resulted vector is shifted via a retrofitting approach to be in the direction of the word vector.

We compute the scores for sentence similarity after replacing ambiguous words with the selected sense vectors retrieved from the generated 200-neighbors Aravec sense inventory.

The four generated sense inventories are tested on sentence similarity: Aravec-Twitter, Aravec-Wiki, Arabic-News, and Fasttext inventory. Table 5 shows the correlation of replacing an ambiguous word with its retrieved sense from each inventory to measure the sentence similarity compared to human annotations.

Table 5. Correlation results of WSD from different sense inventories.

Sense inventory	Pearson Correlation
Twitter Aravec	0.399
Wiki Aravec	0.222
Fasttext	0.318
Arabic-News	0.29

## 5. Conclusions

This paper presents a disambiguation approach for Arabic words that uses the word sense induction approach to build a sense inventory for Arabic words. An evaluation for three sense inventories is provided, where these inventories are based on four different pre-trained embeddings, namely, Aravec-Twitter, Aravec-Wiki, Fasttext, and Arabic-News embeddings.

In the experiment of 50 neighbors sense inventory, the Aravec-Twitter sense inventory achieves the best accuracy of 0.47, whereas in the 100 neighbors experiment, the Fasttext sense inventory provides better accuracy value.

In the case of 200 neighbors, the Aravec-Twitter and Fasttext sense inventories achieve very similar accuracy values.

Similarity between sentences is measured after replacing the ambiguous word vector with the retrieved sense vector and then the results are evaluated using Pearson correlation. However, in the case of paraphrasing identification task, the polysemy problem still has to be studied. This task requires more analysis of semantic similarity and material resources to evaluate the effect of WSD.

## References

- [1] Alian M. and Awajan A., "Semantic Similarity Approaches- Review," in *Proceedings of The International Arab Conference on Information Technology*, Werdanye, pp. 1-6, 2018.
- [2] Alian M. and Awajan A., "Sense Inventories for Arabic Texts," in *Proceedings of The International Arab Conference on Information Technology*, Giza, pp. 1-4, 2020.
- [3] Alian M., Awajan A., Al-Hasan A., and Akuzhia R., "Towards building Arabic paraphrasing benchmark," in *Proceedings of The 2<sup>nd</sup> International Conference on Data Science, E-learning and Information Systems*, Dubai, pp. 1-5, 2019.
- [4] Alian, M. and Awajan A., "Semantic Similarity for English and Arabic Texts: A Review," *Journal of Information and Knowledge Management*, vol. 19, no. 4, 2020.
- [5] Alian M., Awajan A., and Al-Kouz A., "Word Sense Disambiguation for Arabic Text Using Wikipedia and Vector Space Model," *International Journal of Speech Technology*, vol. 19, no. 4, pp. 857-867, 2016.
- [6] Alian M., Awajan A., and Al-Kouz A., "Arabic Word Sense Disambiguation-Survey," in *Proceedings of The International Conference on New Trends in Computing Sciences*, Amman, pp. 236-240, 2017.
- [7] Alkhatlana A., Kalita J., and Alhaddad A., "Word Sense Disambiguation for Arabic Exploiting Arabic WordNet and Word Embedding," in *Proceedings of The 4<sup>th</sup> International Conference on Arabic Computational Linguistics*, Dubai, pp. 50-60, 2018.
- [8] AlKouli M., *Transformation Rules for Arabic Language ( qwAEd tHwylyAh llgAh AlErbyAh)*, Dar Al-Falah, 1999.
- [9] Altowayan A. and Tao L., "Word Embeddings for Arabic Sentiment Analysis," in *Proceedings of the International Conference on Big Data (Big Data)*, Washington, pp. 3820-3825, 2016.
- [10] Chang H., Agrawal A., Ganesh A., Desai A., Mathur V., Hough A., and McCallum A., "Efficient Graph-based Word Sense Induction by Distributional Inclusion Vector Embeddings," *arXiv preprint arXiv:1804.03257*, 2018.
- [11] Chomsky N., *Syntactic Structure*, The Hague Mouton Publishers, 1957.
- [12] Grave E., Bojanowski P., Gupta P., Joulin A., and Mikolov T., "Learning Word Vectors for 157 Languages," in *Proceedings of The International Conference on Language Resources and Evaluation*, 2018.
- [13] Hadni M., El Alaoui S., and Lachkar A., "Word Sense Disambiguation for Arabic Text Categorization," *The International Arab Journal of Information Technology*, vol. 13, no. 1A, no. 1A, pp. 215-222, 2016.
- [14] Ide N. and Véronis J., "Word Sense Disambiguation: The State of the Art," *Computational Linguistics*, vol. 24, no. 1, pp. 1-40, 1998.

- [15] Jurgens D., "An Analysis of Ambiguity in Word Sense Annotations," in *Proceedings of the 9<sup>th</sup> International Conference on Language Resources and Evaluation*, Reykjavik, pp. 3006-3012, 2014.
- [16] Laatar R., Aloulou C., and Belguith L., "Word Sense Disambiguation of Arabic Language With Word Embeddings As Part of The Creation of A Historical Dictionary," in *Proceedings of the International Workshop on Language Processing and Knowledge Management*, Sfax, 2017.
- [17] Laatar R., Aloulou C., and Bilguith L., "Word Sense Disambiguation of Arabic Language With Word Embeddings As Part of The Creation of A Historical Dictionary," in *Proceedings of The 8<sup>th</sup> International Conference on Computer Science and Information Technology*, Amman, 2018.
- [18] Logacheva V., Teslenko D., Shelmanov A., Remus S., Ustalov D., Kutuzov A., Artemova E., Biemann C., Ponzetto S., and Panchenko A., "Word Sense Disambiguation for 158 Languages using Word Embeddings Only," *arXiv preprint arXiv:2003.0665*, 2020.
- [19] Mikolov T., Sutskever I., Chen K., Corrado G., and Dean J., "Distributed Representations of Words And Phrases and Their Compositionality," *Neural Information Processing Systems*, pp. 3111-3119, 2013.
- [20] Mohammad A., Eissa K., and El-Beltagy S., "Aravec: A Set of Arabic Word Embedding Models for Use in Arabic Nlp," *Procedia Computer Science*, vol. 117, pp. 256-265, 2017.
- [21] Navigli R., "Word Sense Disambiguation: A Survey," *ACM Computing Surveys*, vol. 41, no. 2, pp. 1-69, 2009.
- [22] Pelevina M., Arefyev N., Biemann C., and Panchenko A., "Making Sense of Word Embeddings," in *Proceedings of The 1<sup>st</sup> Workshop on Representation Learning for NLP*, Berlin, pp. 174-183, 2016.
- [23] Srivastava S. and Govilkar S., "A Survey on Paraphrase Detection Techniques for Indian Regional Languages," *The International Journal of Computer Applications*, vol. 163, no. 9, pp. 0975-8887, 2017.
- [24] Wu Z. and Palmer M., "Verb Semantics and Lexical Selection," in *Proceedings of The 32<sup>nd</sup> Annual Meeting of the Associations for Computational Linguistics*, Stroudsburg, pp. 133-138, 1994.



**Marwah Alian** is a PhD candidate in Princess Sumaya University for Technology since 2015. She received her B.Sc. degree in Computer Science from Hashemite University in 1995 while her MS.c degree was received in Computer Science in 2007 from Jordan University. Her research interest is in the fields of e-learning systems, data mining and Natural language processing.



**Arafat Awajan** is a Full Professor and the president of Mutah University. He was teaching at Princess Sumaya University for Technology (PSUT). He received his PhD degree in Computer Science from the University of Franche - Comte, France in 1987. He has held various administrative and academic positions at the Royal Scientific Society and Princess Sumaya University for Technology. Head of the Department of Computer Science (2000 -2003) Head of the Department of Computer Graphics and Animation (2005 -2006) Dean of the King Hussein School for Information Technology (2004 - 2007) Director of the Information Technology Center, RSS (2008 -2010) Dean of Student Affairs (2011 - 2014) Dean of the King Hussein School for Computing Sciences (2014 -2017) He is currently the vice president of the university (PSUT). His research interests include: Natural Language Processing, Arabic Text Mining and Digital Image Processing.