A Benchmark and Scoring Algorithm for Enriching Arabic Synonyms

Sana Ghanem[1], **Mustafa Jarrar**[1], **Radi Jarrar**[1], **Ibrahim Bounhas**[2,3]
[1]Department of Computer Science, Birzeit University, Palestine
[2]LISI Laboratory of Computer Science for Industrial System, INSAT, Carthage University, Tunisia
[3]JARIR: Joint group for Artificial Reasoning and Information Retrieval, Tunisia
{swghanem, mjarrar, rjarrar}@birzeit.edu
ibrahim.bounhas@isd.uma.tn

Abstract

This paper addresses the task of extending a given synset with additional synonyms taking into account synonymy strength as a fuzzy value. Given a mono/multilingual synset and a threshold (a fuzzy value $[0-1]$), our goal is to extract new synonyms above this threshold from existing lexicons. We present twofold contributions: an algorithm and a benchmark dataset. The dataset consists of 3K candidate synonyms for 500 synsets. Each candidate synonym is annotated with a fuzzy value by four linguists. The dataset is important for (i) understanding how much linguists (dis/)agree on synonymy, in addition to (ii) using the dataset as a baseline to evaluate our algorithm. Our proposed algorithm extracts synonyms from existing lexicons and computes a fuzzy value for each candidate. Our evaluations show that the algorithm behaves like a linguist and its fuzzy values are close to those proposed by linguists (using RMSE and MAE). The dataset and a demo page are publicly available at https://portal.sina.birzeit.edu/synonyms.

1 Introduction and Motivation

Synonymy relationships are used in many NLP tasks and knowledge organization systems. However, automatic synonym extraction is a challenging task, especially for low-resourced and highly ambiguous languages such as Arabic (Darwish et al., 2021). There are some Arabic resources representing synonymy, such as Al-Maknaz Al-Kabīr, Arabic WordNet (Elkateb et al., 2006) and the Arabic Ontology (Jarrar, 2021, 2011); however, these resources are limited in terms of size and coverage (Helou et al., 2016; Al-Hajj and Jarrar, 2021), especially if compared with the English Princeton WordNet (Miller et al., 1990). Building such resources is expensive and challenging (Helou et al., 2014; Jarrar and Amayreh, 2019; Jarrar, 2020). In addition, the notion of synonymy itself is problematic, as it can vary from near (i.e., semantically related) to strict synonymy (Jarrar et al., 2021; Jarrar, 2005). Strict and formal synonymy is used in ontology engineering as an equivalence relation, thus its reflexive, symmetric, and transitive (Jarrar, 2021). A less formal synonymy is used in the construction of synsets in Princeton WordNet, which relies on the substitutionability of words in a sentence: "two expressions are synonymous in a linguistic context c if the substitution of one for the other in c does not alter the truth value" (Miller et al., 1990). For example, (طريق/road) and (شارع/street) are substitutionable in many contexts in Arabic, thus they can be synonyms. As will be reviewed in section 2, different approaches have been proposed for extracting synonyms automatically.

Nevertheless, one of the major challenges in extracting synonyms is that it is hard to evaluate them (Wu and Zhou, 2003) and there are no common evaluation datasets. Moreover, the substitutionability criteria are subjective, because humans do not necessarily agree on synonymy. As will be illustrated later in this paper, if different linguists are given the same words to judge whether they are synonyms, it is unlikely that they will agree on all cases. Thus, instead of relying on "the substitutionability of words in a sentence" as a criterion to judge whether two words are synonyms or not, we propose to model it with a *fuzzy value*. For example, let {confederacy, confederation} be two synonyms in the context of "a union of political organizations", and let "alliance" and "federation" be candidate additional synonyms, our goal is to assign a fuzzy value (e.g., 0.6 and 0.9) to each candidate synonym to indicate how much it is substitutionable, i.e., acceptable to be an additional third synonym.

Using such a fuzzy value is helpful for different application scenarios. For example, when constructing wordnet synsets, synonyms can be extracted with a high fuzzy value, but in the case of less sensitive information retrieval applications, a lower value might be more suitable. In a quality control scenario, one may evaluate a thesaurus by masking each synonym in a synset and assessing if its fuzzy value passes a threshold. Nevertheless,

assigning a meaningful fuzzy value to each synonym in a synset is challenging. Thesauri are typically constructed based on linguists' intuition and without assigning a strength, or a fuzzy, value explicitly. To overcome this challenge, we developed a dataset of 3K synonyms, each assigned with a fuzzy value by four different linguists. We used this dataset to measure how much linguists (dis/)agree on synonymy. The dataset is also used to train our proposed algorithm (i.e., tune its fuzzy model) for extracting synonyms from dictionaries.

Task definition: The task we aim to address is defined as the following: Let S be a set of synonyms, c is a candidate synonym to S, and a dictionary D, our goal is to compute a fuzzy value f to indicate how much c is acceptable to be an addition to S. As will be elaborated in section 4, we assume D to be a set of sets of synonyms, and that S can be mono or multilingual synonyms.

Our main contributions in this paper are a dataset and an algorithm. The dataset was constructed by employing four linguists and giving them 3,000 candidate synonyms and 500 synsets from the Arabic WordNet (Elkateb et al., 2006). Each linguist was asked to score each candidate synonym in a given synset. Our proposed algorithm aims at discovering new candidate synonyms from existing linguistic resources. Given a set of synonyms, the algorithm builds a directed graph, at level k for all words in this set. Cyclic paths in this graph are then detected, and all words participating in these cyclic paths are considered candidate synonyms for the given synset. Each of these candidate synonyms is assigned a fuzzy value, which is calculated based on a fuzzy model that we learned from the dataset and that takes into account the connectivity of the candidate synonym in the graph. The novelty of our algorithm and our dataset is that we treat synonyms as a fuzzy relation. We evaluated the algorithm's fuzzy values by comparing them with the average of the linguists' scores (i.e., as a baseline). The Root Mean Squared Error (RMSE) between the scores of the algorithm and the average of the linguists' scores is 0.32 and the Mean Average Error (MAE) is 0.27. This means that the algorithm was behaving closely to a linguist. To evaluate the accuracy of our algorithm, we used the 10K synsets in Arabic WordNet. We masked the word with (highest, lowest, average, and random) frequency in each synset and used the algorithm to see if it could discover it again with top rank. The achieved accuracy was indeed high. For example, with the average frequency we achieved an accuracy of 98.7% at level 3 and 92% at level 4.

This paper is organized as follows: Section 2 presents related works in the field of synonym extraction. Section 3 overviews the algorithm. Section 4 summarizes and discusses the experimental results. Section 5 concludes the paper and proposes some perspectives.

2 Related Work

In what follows, we overview several approaches have been proposed to extract synonyms or build synsets. We refer to (Naser-Karajah et al., 2021) for a recent survey on this topic.

2.1 Synset Construction

New WordNets may be built by mining corpora and/or monolingual dictionaries as in (Oliveira and Gomes, 2014) for Portuguese. After extracting candidate synonym pairs, authors cluster these pairs into different clusters. Ercan and Haziyev (2019) proposed to build a multilingual synonymy graph from existing resources and wordnets, then used a supervised clustering algorithm to cluster synonyms. In both works, each cluster is then considered a synset. Neural language models, such as word embeddings, were also employed in synonymy extraction and wordnet construction (Mohammed, 2020). For example, Khodak et al. (2017) proposed to construct wordnets using the Princeton WordNet (PWN), machine translation, and word embeddings. A word is first translated into English using machine translation, and these translations are used to build a set of candidate synsets from PWN. A similarity score is used to rank each candidate synset, which is calculated using the word embedding-based method. Similarly, Tarouti and Kalita (2016) used static word embeddings to improve the quality of automatically constructed Arabic wordnet. Furthermore, Al-Matham and Al-Khalifa (2021) proposed to extract Arabic synonyms based on a static word embedding model that was created using Arabic corpora. Cosine similarity, in addition to some filters, were used to extract Synonyms.

2.2 Synonym Graph Mining

Other approaches are proposed to mine a graph from an existing resource(s) in order to discover new synonyms and translation pairs. The structure

of the graph is exploited to compute ranking scores, which reflect how much two terms are likely to be synonyms (Jarrar, 2005). The main hypothesis is that some words, which are not necessarily directly connected with an edge may be semantically close. That is why cycles are widely exploited. Indeed, graphs are generic tools that may be used both for monolingual and bilingual resources and for several types of linguistic resources. For example, Flati and Navigli (2012) proposed an algorithm to find missing synonyms in the Ragazzini-Biagi English-Italian dictionary. A synonymy graph was built using this dictionary, then cyclic and quasi-cyclic paths are detected. Cyclic paths are those that have all edges in the same direction, while quasi cycles should be consecutive reverse edges. The length of a path is used to score the discovered synonyms. Discovering new translation pairs from multilingual dictionaries is also related to synonymy extraction. Villegas et al. (2016) proposed to construct a multilingual translation graph using translation pairs in the Apertium dictionaries. New translation pairs are then extracted from cyclic paths. However, wrong translations might be detected because of polysemy. The authors proposed to score the density of each path and exclude those paths with low densities. Instead of only using density, Torregrosa et al. (2019) proposed to combine it with a multi-way neural machine translation trained with parallel English and Spanish, Italian and Portuguese, and French and Romanian corpora. Their experiment shows a low recall and a reasonable precision ($25\% - 75\%$).

A recent algorithm that uses synonymy graphs was proposed by Jarrar et al. (2021). The idea is to construct an Arabic-English translation graph from a given bilingual dictionary (Jarrar et al., 2019). Terms participating in cyclic paths are extracted and consolidated, and considered synonyms. However, instead of using fuzzy values, they proposed the idea of bidirectional consolidation.

2.3 Related Notions of Fuzziness

Different notions of fuzziness were proposed in the WordNet literature. Hossayni et al. (2020) and Alizadeh-Q et al. (2021) proposed to compute the frequency of each word-sense pair in a corpus that is annotated using a WSD algorithm. The frequency is then normalized and transformed into a "possibility" value between 0 and 1 reflecting the membership degree. In (Hossayni et al., 2020), the same notion is evaluated in an interval indicating minimum and maximum values by dividing the corpus into several categories. In both cases, These membership degrees depend on the number of times a word-sense pair appeared in a given corpus. We believe that this notion of fuzziness is valuable and complements our proposed work; however, it highly depends on the coverage of the used corpora and the accuracy of the WSD algorithm, which is typically not good enough (Maru et al., 2022). Another notion of fuzziness was used in (Oliveira and Santos, 2016), to compute how likely two words are synonyms based on much they share words in their dictionary definitions. This notion of fuzziness was used to extract a Portuguese synonym network from seven resources taking into account the number of times a relation between two given words exists across resources. This notion of fuzziness, similar to (Hossayni et al., 2020), depends on text mining rather than synonyms graphs. Additionally, it computes the fuzziness between two words rather than between a word and a given synset. Most importantly, as discussed in section 3.3, our fuzzy scores are designed to reflect meaningful values, i.e., semantic truth, rather than frequency of use.

2.4 Benchmarks

As far as benchmarking and evaluation are concerned, it is hard to compare previous works, given the lack of a common gold standard. Indeed, the above-reviewed approaches were evaluated using different ways and resources, as no evaluation benchmarks are available for synonymy extraction. More precisely, and to our knowledge, there are no datasets of synonyms with ranking or fuzzy values to indicate how much a term is likely to be a synonym with a given synset.

3 Dataset Construction

This section presents a benchmarking dataset annotated with fuzzy values[1]. The dataset can be used for training and evaluating (i.e., a baseline) synonym extraction algorithms. Additionally, the construction of this dataset can also be used as an experiment to measure how much linguists (dis/)agree on synonymy.

[1]The dataset and source code are publicly available at https://portal.sina.birzeit.edu/synonyms

Figure 1: Example of scoring candidate synonyms.

Score	Meaning
100	Same semantics, style, use
90	Same semantics, style, less used
80	Same semantics, style, rarely used
70	Same semantics, style, not used
60	Close semantics, weak style, uncommon
50	Close semantics, not exact purpose
40	Semantically related
30	Semantically related (somehow)
20	Semantically different
10	Semantically very different
0	Semantically unrelated

Table 1: The fuzzy scoring scale - synonymy strength

3.1 Data Selection

First, we selected 500 synsets from the 10K synsets in Arabic WordNet. For each synset, we extracted a set of Arabic candidate synonyms, which we collected using our algorithm presented in Section 4. The total number of candidate synonyms is 3K. The 500 synsets were selected proportionally to the WordNet's distribution: 350 noun synsets, 140 verb synsets, and 10 adjective synsets. These synsets were selected randomly but we also took into account synset length and selected 142, 207, and 151 synsets of 2, 4, and 6 words in each synset, respectively. The 3K candidate synonyms were then given to four linguists to give them scores.

3.2 Experimental Setup

The four linguists who participated in this experiment are top students, who graduated recently with high distinction from the department of linguistics and translation at Birzeit University. Three training workshops were organized to explain the experiment and to emphasize the notion of synonymy. To ensure that all linguists have the same understanding of the task, we gave each linguist a small quiz ($\sim$30 synonyms) to try alone, then we discussed the results jointly. After that, each linguist was given the 3K candidate synonyms in a separate file in Google Sheet. Figure 1 illustrates an example of a synset and four candidate synonyms as scored by one of the linguists. As shown in Figure 1, the scoring is based on the linguist's understanding of the given synset (both English and Arabic synonyms), the gloss, and the context example (if available), which we extracted from the Arabic WordNet.

3.3 Scoring Guidelines

Table 1 presents our scoring schema, which is a scale from 0 to 100 representing the strength of the synonymy relation. The main factor in the scoring is the semantics, which indicates *how much the truth of a sentence is altered if the candidate synonymy is substituted with one of the given synonyms*, as defined in Miller et al. (1990). The scoring schema should not be interpreted as absolute numbers, but rather, they are used as annotation methodology to maintain a degree of consistency among linguists' scores as will be discussed next. From a semantics viewpoint, the scoring schema is divided into three categories: same ($> 60\%$), close ($60\% - 50\%$), or related/different semantics ($< 50\%$). *Same semantics* means that a word can be substituted in a sentence without altering the truth of this sentence. The four different scores inside this range are used to capture the *use*; i.e., how much it is common that a word can be used in this context. For example, the word إئْتِلاف has the same semantics as the other synonyms in the synset that means "a union of political organizations", but this word is rarely used in this context. *Close Semantics* means that it is possible to use a word (e.g., جامِعَة) with this semantics, but with some doubts, for instance, the word has an uncommon meaning or is usually employed in different contexts/with different purposes. Scores less than 40% mean different, related, or unrelated semantics, which means that the word cannot be a candidate synonym in this context. It is worth noting that this fine-grained scoring schema emerged after different iterations of discussion with the linguists in order to create sound methodological guidelines to annotate the dataset with fuzzy values.

3.4 Linguists Agreement Evaluation

The scoring of the 3K synonyms spanned over three months and took about 100 working hours for each linguist. The results of the four linguists are aggregated, and an average of all scores was computed.

To measure the (dis)agreements between linguists, we computed the Root Mean Squared Error (RMSE) and the Mean Average Error (MAE) between their scores (see table 2). We also computed the RMSE and MAE between the scores of each

linguist with the average score for the four linguists. Later, we will use the same model (i.e., the average of answers) as a baseline to evaluate our algorithm, (see subsection 5.1). The RMSE might be more commonly used than MAE in measuring the differences between scores, but we provide both metrics in this paper. The MAE scores treat differences equally, while RMSE penalizes large variations (Wang and Lu, 2018).

As shown in table 2, linguists L_2 and L_3 have the closest RMSE to the average of all linguists. Linguists L_1 and L_4 have the highest RMSE distances if compared with the average scores. However, this does not indicate that they are more or less precise in their scores, it only shows that the scores of their answers deviate by the value stated by RMSE. Nevertheless, the RMSE of each linguist and the average ranges between 0.1 and 0.13. This indicates how much the scores of all linguists deviate from their average (i.e., which can be seen as an estimator of the standard deviation of errors between the linguist scores and the average of all linguists). It can be also noticed that the average deviation of the linguists and their average ranges between 0.31 to 0.39 from the algorithm. Though the algorithm deviates from the average score more than the individual linguists, the reported RMSE and MAE values are not considered high and further experiments are conducted to highlight if the difference between the scores is statistically significant.

To conform with this conclusion and to better understand the behavior of linguists in scoring these 3K synonyms, we perform a one-way ANOVA test (at $p < 0.05$). This test determines if the difference between the linguists' scores is generated at random or if their scores are different consistently (i.e., significantly different).

Post-hoc comparisons using the Tukey HSD test (using SPSS) indicated that the mean score for linguist L_1 (Mean = 0.4919, Standard Deviation = 0.34223) was significantly different than the other linguists (Mean = 0.4596, Standard Deviation = 0.31899). All included variables are following the normal distribution.

4 Algorithm Overview

The algorithm takes two inputs: a dictionary D, and a synset S. The output is a set of candidate synonyms C, each synonym c_i is assigned a fuzzy value f_i. The dictionary D itself is assumed to consist of set of synsets, $S_i \in D$. Each synset is a tuple $< t_1, .., t_n >$ of linguistic terms regardless of the language it belongs to. In this way, we can benefit from mono and multiple dictionaries and thesauri. In the first step, the algorithm extracts the candidate synonyms C, then it computes the fuzzy value f_i for each synonym c_i.

4.1 Candidate Synonym Extraction

For each term t_i in synset S, the algorithm finds all cyclic paths at level k, where k = 3, 4, 5,... n. That is, starting from t_i as a root, a graph is constructed using D, at level k, and all paths starting and ending with t_i are considered cyclic paths. If a term appears in any cyclic path, it is then considered a candidate synonym and is added to C.

Example: Figure 2 illustrates the synset {رَكِبَ, ride}, taken from the Arabic WordNet, and the generated graph at level 4 for each word in this synset. There are ten cyclic paths in this graph, highlighted as bold green lines, and shown below separately in Figure 3. The new terms participating in these ten cyclic paths are {إِمتَطى, sit}, which is the set C of candidate synonyms.

4.2 Candidate Synonym Selection

The intuition of our fuzzy model is that the more a candidate synonym appears in different cyclic paths and with different terms in S, the higher its fuzzy value, i.e., the stronger the synonymy. As such, to compute the fuzzy value f_i for each c_i in set C, we propose the following $Fuzzy$ function, which is based on two variables and two constant weights, as in the following formula:

$$Fuzzy(f_i) = \theta_1 \cdot P_i + \theta_2 \cdot Q_i$$

where P_i is the number of cyclic paths that c_i appears in, divided by the total number of cyclic paths, and Q_i is the number of root nodes t that appear in the cyclic paths of c_i, divided by the total number of terms in the synset S. θ_1 and θ_2 are two constant weights that we tuned using a 10-fold Cross-Validation (See section 4.3). The best values we found at level 3 and 4 are $(0.4, 0.6)$ and $(0.5, 0.5)$, respectively. As Figure 2 illustrates, the term (sit) appears six times among the ten cyclic paths found at the level 4, and appears in two root nodes among the two synonyms in the original synset; and similarly for (إِمتَطى). Therefore, their fuzzy values are:

$$Fuzzy(sit) = \tfrac{6}{10} \times 0.5 + \tfrac{2}{2} \times 0.5 = 0.8$$

$$Fuzzy(\text{إِمتَطى}) = \tfrac{6}{10} \times 0.5 + \tfrac{2}{2} \times 0.5 = 0.8$$

	L1		L2		L3		L4		Avg		Algorithm	
	RMSE	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**
L1			0.19	0.14	0.19	0.14	0.22	0.16	0.13	0.10	0.35	0.30
L2	0.19	0.14			0.16	0.12	0.20	0.15	0.10	0.11	0.31	0.26
L3	0.19	0.14	0.16	0.12			0.20	0.16	0.11	0.08	0.32	0.26
L4	0.22	0.16	0.20	0.15	0.20	0.15			0.13	0.08	0.39	0.34
Avg	0.13	0.10	0.10	0.08	0.11	0.08	0.13	0.11			**0.32**	**0.27**
Algorithm	0.35	0.30	0.31	0.26	0.32	0.26	0.39	0.34	**0.32**	**0.27**		

Table 2: The Root Mean Squared Error (RMSE) and the Mean Average Error (MAE) between the scores of each linguist, the average scores of all linguists, and the scores of the algorithm.

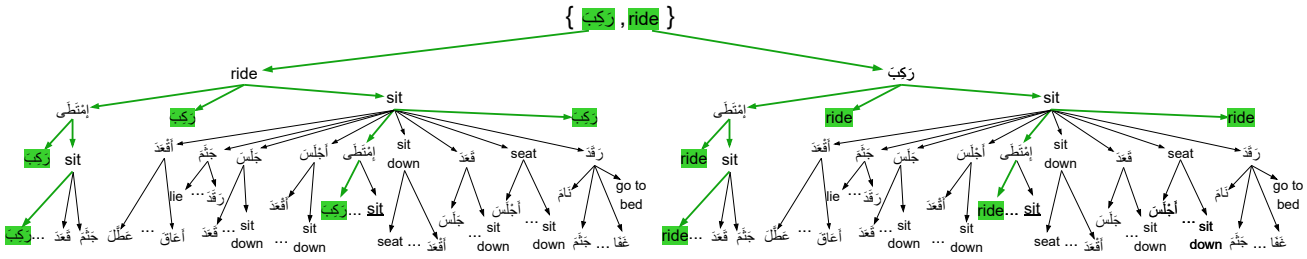

Figure 2: The cyclic paths for the {رَكِبَ , ride} synset from AWN

رَكِبَ→ride→sit→رَكِبَ
رَكِبَ→ride→sit→إِمْتَطَى→رَكِبَ
رَكِبَ→ride→رَكِبَ
رَكِبَ→ride→إِمْتَطَى→رَكِبَ
رَكِبَ→ride→إِمْتَطَى→sit→رَكِبَ
ride→رَكِبَ→sit→ride
ride→رَكِبَ→sit→إِمْتَطَى→ride
ride→رَكِبَ→ride
ride→رَكِبَ→إِمْتَطَى→ride
ride→رَكِبَ→إِمْتَطَى→sit→ride

Figure 3: The ten cyclic paths extracted from the graph generated in Figure 2.

4.3 Parameter Tuning

As our proposed $Fuzzy$ function depends on two constant weights (θ_1 and θ_2), our goal in this subsection is to find the best values of these θ_s. The best values are those that enable the $Fuzzy$ function to produce fuzzy values as close to linguists' scores as possible. Thus, we used our dataset, which contains 3K candidate synonyms, each with a fuzzy value (i.e., the average of the four linguists). To generate a model with the best results, we varied the values of the parameters θ_1 and θ_2 by selecting their values within the range of $[0.1, 0.9]$ with a step of 0.1 for each parameter. The total weight of both variables θ_1 and θ_2 should total to 1. This is because each of these variables is contributing to the score which ranges from 0 to 1.

Table 3 shows the average RMSE and the average MAE values using a 10-fold Cross-Validation of the algorithm run on all combinations of the variables. The results show that the best combination is 0.5 for θ_1 and 0.5 for θ_2 which resulted in the lowest RMSE value of 0.32, and the lowest MAE value of 0.27 at level 4. For level 3, the best combination is 0.4 for θ_1 and 0.6 for θ_2 with value of 0.35 for RMSE and 0.29 for MAE. Thus, we complete the RMSE and MAE calculations with level 4, as the RMSE and MAE values in level 4 are better than in level 3. These are the weights that are used in the algorithm evaluations in the next section.

5 Algorithm Evaluation

This section presents two experiments to evaluate the performance of our algorithm. The first experiment compares the results obtained by the algorithm with linguists' scores. The second experiment measures the accuracy of the algorithm.

5.1 Comparing the Algorithm with the Baseline

This experiment compares the results of our algorithm with the average of the linguists' scores (as a baseline) that we presented in section 3.4.

Table 2 shows 0.32 RMSE and 0.27 MAE scores

θ_1, θ_2		Level 4	Level 3
[0.1, 0.9]	RMSE	0.459	0.377
	MAE	0.375	0.319
[0.2, 0.8]	RMSE	0.408	0.362
	MAE	0.330	0.304
[0.3, 0.7]	RMSE	0.366	0.352
	MAE	0.299	0.296
[0.4, 0.6]	RMSE	0.336	**0.349**
	MAE	0.280	**0.293**
[0.5, 0.5]	RMSE	**0.321**	0.352
	MAE	**0.271**	0.296
[0.6, 0.4]	RMSE	0.323	0.363
	MAE	0.271	0.304
[0.7, 0.3]	RMSE	0.343	0.382
	MAE	0.272	0.316
[0.8, 0.2]	RMSE	0.378	0.407
	MAE	0.302	0.335
[0.9, 0.1]	RMSE	0.425	0.437
	MAE	0.335	0.357

Table 3: Average RMSE and MAE with various values of θ_1 and θ_2 obtained using 10-fold Cross-Validation

of the algorithm against the linguists' average. To understand the algorithm's 0.32 RMSE, one can notice that the RMSE difference between L_2 and L_4 is 0.20, and between L_1 and L_4 is 0.22. The RMSE difference between each pair of linguists ranges from 0.16 to 0.22. Now, the RMSE difference between the algorithm and the average of the linguists is 0.32. This means that the algorithm has only 0.10 more difference if compared with the RMSE variation between linguists.

Similarly, to understand the 0.27 MAE between the algorithm and the linguists, one can notice that the MAE between the four linguists themselves ranges from 0.12 to 0.16. Both RMSE and MAE, confirm the variation between the algorithm and the average of linguists. This illustrates that the algorithm's scores are close to the linguists' scores.

Nevertheless, as noted in section 3.4, the variations between linguists' scores, as well as the algorithm, do not tell us whether a linguist is better or more accurate than the others, which is because synonymy is a subjective notion. However, being close to the linguists' variations is a good indication that the algorithm scores are realistic. Next, we compare the behavior of the algorithm with the linguists' behavior in scoring synonyms, which provides an additional evaluation.

Testing the algorithm's *behavior*: to further understand the algorithm's behavior, we need to test whether the scores of the algorithm are statistically significant, i.e., the scores were consistent or resulted at random. In other words, we need to test whether the algorithm is consistently giving scores and behaving like a linguist - regardless of the differences in RMSE and MAE.

We performed a one-way ANOVA test (at $p < 0.05$) to check if there is a statistical difference between the algorithm and the other linguists. Before conducting this test, we first needed to check if all the linguists' and the algorithm's scores follow a normal distribution, or if there are no outliers, which are the main assumptions to conduct a one-way ANOVA test. Our result of the normality test (using SPSS) indicated that the scores of the algorithm are not normally distributed. Thus, we performed a univariate and multivariate outlier analysis. The results (using SPSS) indicated that there are no outliers, which means that the non-normality of the algorithm's scores are due to skewness in the data and not because of outliers. Therefore, the one-way ANOVA test can be applied, as explained by Tabachnick and Fidell (2001): *"it is assumed that the data has a normal distribution, however, note that violations of the normality assumption are not fatal and the result of the significant test is still reliable as long as non-normality is caused by skewness and not outliers"*.

The post-hoc comparisons (using the Tukey HSD test, in SPSS) indicated that the mean score for the algorithm (Mean = 0.4535, Standard Deviation = 0.16416) was significantly different only with linguist L_1 (Mean = 0.4919, Standard Deviation = 0.34223). This indeed confirms the findings shown in the previous section in which linguist L_1 has significantly different scores than the other linguists. In other words, the algorithm has shown to be not statistically different with the other linguists and their average (i.e., the baseline). Being not statistically different means that the algorithm's behavior in scoring synonyms is similar to the behavior of the linguists, except for linguist L_1.

To sum up, the variation between the scores of the algorithm and the linguists (using RMSE and MAE) are close to those between the linguists themselves. The one-way ANOVA test also confirms that the algorithm behaves as a linguist.

5.2 Accuracy Evaluation

We measure the accuracy of the algorithm in terms of retrieved words for each synset, by masking a synonym in a given synset, then try predicting it again. Masking is the process of removing a synonym from a synset, and then measure whether the masked term is retrieved back. The accuracy of the

algorithm is determined by the rank of the masked term. Ideally, if every masked term is retrieved with the highest (i.e., top) rank, it means the accuracy is 100%.

5.2.1 Experiment Setup

We used the 10K synsets in the Arabic WordNet (AWN), and we conducted four masking experiments. For every synset in the 10K AWN's synsets, we calculated the frequency of each synonym (Arabic and English), then selected the synonyms with (highest, lowest, average, and random) frequencies in each synset to conduct the experiment. The frequency of a term is the number of synsets in which this term appears. We considered synsets that contain more than two synonyms, regardless of the language. That is, the experiment was conducted on both Arabic and English terms. Terms with the frequency of 1 (i.e., appeared in one synset only) are not selected. The number of synsets that are longer than two terms, and with a term with a frequency more than 1 are $7,219$, while the number of synsets longer than two terms, and with a term with lowest frequency are only $1,085$. Similarly, we selected synsets with average and random term frequencies, $5,207$ and $4,153$, respectively. Table 4 shows the results of the masking experiments.

The algorithm was applied individually for each synset by eliminating (i.e., masking) a term, in this synset, and retrieving back the top-ranked term using the algorithm. That is, given a term c_1 in synset s_n, c_1 will be eliminated from s_n, then we compute the fuzzy value of c_1 using our algorithm and check if the algorithm was able to retrieve it with highest fuzzy value (i.e., top rank) among other possible candidate synonyms for s_n. In this way, the algorithm is applied on synsets by masking terms with highest, lowest, average, and random frequencies, at level 3; and repeated at level 4, as shown in Table 4.

5.2.2 Results

The accuracy of the algorithm was calculated as a ratio of the correctly retrieved synonyms (i.e., top rank) from all samples. For example, the algorithm was able to retrieve 7,157 (99.1%) of the masked terms with highest frequencies at level 3 with the top ranking (i.e., highest fuzzy values).

The results in Table 4 illustrate that the lower the frequency of a term in the lexicon the lower the accuracy, which is because the connectivity of less frequent terms yields less fuzzy values by the algorithm. This does not mean that the masked terms were not retrieved by the algorithm, but rather, they are not ranked as the top (highest fuzzy values). The accuracy at level 4 decreases because the synonymy graph at this level becomes larger, and thus it contains more candidate synonyms.

It is important to remark that the algorithm was able to obtain high accuracy in this experiment but the accuracy evaluation heavily depends on the structure of the used lexicon, which is AWN in our case. Changing the dictionary, by adding more synonymy/translation relations yields to constructing a different graph, thus different accuracy is expected.

Experiment	Sample Size	Accuracy at Level 3	Accuracy at Level 4
Exp.1 (Highest)	$7,219$	99.1%	95.2%
Exp.2 (Average)	$5,207$	98.7%	92.0%
Exp.3 (Lowest)	$1,085$	88.4%	62.0%
Exp.4 (Random)	$4,153$	98.1%	89.3%

Table 4: The accuracy of the algorithm using the masking experiment with the highest, average, lowest, and random frequencies within each synset.

6 Conclusion

We presented a benchmark dataset and an algorithm to extract synonyms and fuzzy values. The benchmark dataset consists of 3K candidate synonyms for 500 synsets, each candidate synonym was annotated with a fuzzy value by four linguists. The dataset is important for measuring how much linguists disagree on synonymy, which ranged between $0.16 - 0.22$ for RMSE and $0.12 - 0.16$ for MAE. These measures were also used as a baseline to evaluate our algorithm. The algorithm presented in this paper aims to enrich a given mono/multilingual synset with more synonyms. Our evaluation shows that our algorithm behaves as linguists in producing fuzzy values, and the fuzzy scores are also close to those of the linguists. The accuracy evaluation illustrates that it is highly accurate.

7 Limitations and Future Work

The current version of our algorithm neglects the effect of diacritics in the Arabic language (Jarrar et al., 2018), so that a word with different diacritics is considered as different, like كَتَبَ, كتَبَ, even if they are the same. Thus, we plan to enhance the algorithm to consider the characteristics of the Arabic language, and consider synonyms in MSA and

Arabic dialects as described in (Haff et al., 2022; Jarrar et al., 2017, 2022).

Acknowledgment

We acknowledge the support of the Research Committee at Birzeit University (No. 2021/49), and would like to thank Taymaa Hammouda and Muhannad Yaseen for the technical and statistical support, and all students who helped in the annotation process, especially Tamara Qaimari, Asala Hamed, Ahd Muhtasib, Doa Shwiki, Shaimaa Hamayel, Hiba Zayed, Rwaa Zaid, and others.

References

Moustafa Al-Hajj and Mustafa Jarrar. 2021. Arabglossbert: Fine-tuning bert on context-gloss pairs for wsd. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 40–48, Online. INCOMA Ltd.

Rawan N Al-Matham and Hend S Al-Khalifa. 2021. Synoextractor: a novel pipeline for arabic synonym extraction using word2vec word embeddings. *Complexity*, 2021.

Yousef Alizadeh-Q, Behrouz Minaei-Bidgoli, Sayyed-Ali Hossayni, Mohammad-R. Akbarzadeh-T., Diego Reforgiato Recupero, Mohammad Reza Rajati, and Aldo Gangemi. 2021. Interval probabilistic fuzzy wordnet. *CoRR*, abs/2104.10660.

Kareem Darwish, Nizar Habash, Mourad Abbas, Hend Al-Khalifa, Huseein T. Al-Natsheh, Houda Bouamor, Karim Bouzoubaa, Violetta Cavalli-Sforza, Samhaa R. El-Beltagy, Wassim El-Hajj, Mustafa Jarrar, and Hamdy Mubarak. 2021. A panoramic survey of natural language processing in the arab worlds. *Commun. ACM*, 64(4):72–81.

Sabry Elkateb, William Black, Piek Vossen, David Farwell, H Rodríguez, A Pease, and M Alkhalifa. 2006. Arabic wordnet and the challenges of arabic. In *Proceedings of Arabic NLP/MT Conference, London, UK*, pages 665–670.

Gonenc Ercan and Farid Haziyev. 2019. Synset expansion on translation graph for automatic wordnet construction. *Information Processing & Management*, 56(1):130–150.

Tiziano Flati and Roberto Navigli. 2012. The cqc algorithm: Cycling in graphs to semantically enrich and enhance a bilingual dictionary. *Journal of Artificial Intelligence Research*, 43:135–171.

Karim El Haff, Mustafa Jarrar, Tymaa Hammouda, and Fadi Zaraket. 2022. Curras + baladi: Towards a levantine corpus. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2022)*, Marseille, France.

Mamoun Abu Helou, Matteo Palmonari, and Mustafa Jarrar. 2016. Effectiveness of automatic translations for cross-lingual ontology mapping. *Journal of Artificial Intelligence Research*, 55(1):165–208.

Mamoun Abu Helou, Matteo Palmonari, Mustafa Jarrar, and Christiane Fellbaum. 2014. Towards building lexical ontology via cross-language matching. In *Proceedings of the 7th Conference on Global WordNet*, pages 346–354. Global WordNet Association.

Sayyed-Ali Hossayni, Mohammad-R. Akbarzadeh-T., Diego Reforgiato Recupero, Aldo Gangemi, Esteve del Acebo, and Josep Lluís de la Rosa i Esteva. 2020. An algorithm for fuzzification of wordnets, supported by a mathematical proof. *CoRR*, abs/2006.04042.

Mustafa Jarrar. 2005. *Towards Methodological Principles for Ontology Engineering*. Ph.D. thesis, Vrije Universiteit Brussel.

Mustafa Jarrar. 2011. Building a formal arabic ontology (invited paper). In *Proceedings of the Experts Meeting on Arabic Ontologies and Semantic Networks*. ALECSO, Arab League.

Mustafa Jarrar. 2020. *Digitization of Arabic Lexicons*, pages 214–217. UAE Ministry of Culture and Youth.

Mustafa Jarrar. 2021. The arabic ontology - an arabic wordnet with ontologically clean content. *Applied Ontology Journal*, 16(1):1–26.

Mustafa Jarrar and Hamzeh Amayreh. 2019. An arabic-multilingual database with a lexicographic search engine. In *The 24th International Conference on Applications of Natural Language to Information Systems (NLDB 2019)*, volume 11608 of *LNCS*, pages 234–246. Springer.

Mustafa Jarrar, Hamzeh Amayreh, and John P. McCrae. 2019. Representing arabic lexicons in lemon - a preliminary study. In *The 2nd Conference on Language, Data and Knowledge (LDK 2019)*, volume 2402, pages 29–33. CEUR Workshop Proceedings.

Mustafa Jarrar, Nizar Habash, Faeq Alrimawi, Diyam Akra, and Nasser Zalmout. 2017. Curras: An annotated corpus for the palestinian arabic dialect. *Journal Language Resources and Evaluation*, 51(3):745–775.

Mustafa Jarrar, Eman Karajah, Muhammad Khalifa, and Khaled Shaalan. 2021. Extracting synonyms from bilingual dictionaries. In *Proceedings of the 11th International Global Wordnet Conference (GWC2021)*, pages 215–222. Global Wordnet Association.

Mustafa Jarrar, Fadi Zaraket, Rami Asia, and Hamzeh Amayreh. 2018. Diacritic-based matching of arabic words. *ACM Asian and Low-Resource Language Information Processing*, 18(2):10:1–10:21.

Mustafa Jarrar, Fadi A Zaraket, Tymaa Hammouda, Daanish Masood Alavi, and Martin Waahlisch. 2022. Lisan: Yemeni, irqi, libyan, and sudanese arabic dialect copora with morphological annotations.

Mikhail Khodak, Andrej Risteski, Christiane Fellbaum, and Sanjeev Arora. 2017. Automated wordnet construction using word embeddings. In *Proceedings of the 1st Workshop on Sense, Concept and Entity Representations and their Applications*, pages 12–23.

Marco Maru, Simone Conia, Michele Bevilacqua, and Roberto Navigli. 2022. Nibbling at the hard core of Word Sense Disambiguation. In *Proceedings of the ACL2022 (Vol.1)*, pages 4724–4737, Dublin, Ireland. ACL.

George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. 1990. Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244.

Nora Mohammed. 2020. Extracting word synonyms from text using neural approaches. *International Arab Journal of Information Technology*, 17.

Eman Naser-Karajah, Nabil Arman, and Mustafa Jarrar. 2021. Current trends and approaches in synonyms extraction: Potential adaptation to arabic. In *Proceedings of the 2021 International Conference on Information Technology (ICIT)*, pages 428–434, Amman, Jordan. IEEE.

Hugo Gonçalo Oliveira and Paulo Gomes. 2014. Eco and onto.pt: A flexible approach for creating a portuguese wordnet automatically. *Language Resources and Evaluation*, 48.

Hugo Gonçalo Oliveira and Fábio Santos. 2016. Discovering fuzzy synsets from the redundancy in different lexical-semantic resources. In *Proceedings of LREC 2016*, Paris, France. ELRA.

Barbara G Tabachnick and LS Fidell. 2001. Using multivariate statistics. *Allyn &Bacon A Pearson Education Company: Boston*.

Feras Al Tarouti and Jugal Kalita. 2016. Enhancing automatic wordnet construction using word embeddings.

Daniel Torregrosa, Mihael Arcan, Sina Ahmadi, and John P McCrae. 2019. Tiad 2019 shared task: Leveraging knowledge graphs with neural machine translation for automatic multilingual dictionary generation. *Translation Inference Across Dictionaries*.

Marta Villegas, Maite Melero, Núria Bel, and Jorge Gracia. 2016. Leveraging rdf graphs for crossing multiple bilingual dictionaries. In *Proceedings of LREC2016*, pages 868–876.

Weijie Wang and Yanmin Lu. 2018. Analysis of the mean absolute error (mae) and the root mean square error (rmse) in assessing rounding model. In *IOP conference series: materials science and engineering*, volume 324. IOP Publishing.

Hua Wu and Ming Zhou. 2003. Optimizing synonym extraction using monolingual and bilingual resources. In *Proceedings of the second international workshop on Paraphrasing*, pages 72–79.