Hindawi
Complexity
Volume 2021, Article ID 6627434, 13 pages
https://doi.org/10.1155/2021/6627434

*Research Article*

# SynoExtractor: A Novel Pipeline for Arabic Synonym Extraction Using Word2Vec Word Embeddings

**Rawan N. Al-Matham and Hend S. Al-Khalifa**

*Department of Information Technology, College of Computer and Information Sciences, King Saud University, P.O. Box 12371, Riyadh, Saudi Arabia*

Correspondence should be addressed to Hend S. Al-Khalifa; hendk@ksu.edu.sa

Received 15 December 2020; Revised 21 January 2021; Accepted 27 January 2021; Published 17 February 2021

Academic Editor: M. Irfan Uddin

Automatic synonym extraction plays an important role in many natural language processing systems, such as those involving information retrieval and question answering. Recently, research has focused on extracting semantic relations from word embeddings since they capture relatedness and similarity between words. However, using word embeddings alone poses problems for synonym extraction because it cannot determine whether the relation between words is synonymy or some other semantic relation. In this paper, we present a novel solution for this problem by proposing the SynoExtractor pipeline, which can be used to filter similar word embeddings to retain synonyms based on specified linguistic rules. Our experiments were conducted using KSUCCA and Gigaword embeddings and trained with CBOW and SG models. We evaluated automatically extracted synonyms by comparing them with Alma'any Arabic synonym thesauri. We also arranged for a manual evaluation by two Arabic linguists. The results of experiments we conducted show that using the SynoExtractor pipeline enhances the precision of synonym extraction compared to using the cosine similarity measure alone. SynoExtractor obtained a 0.605 mean average precision (MAP) for the King Saud University Corpus of Classical Arabic with 21% improvement over the baseline and a 0.748 MAP for the Gigaword corpus with 25% improvement. SynoExtractor outperformed the Sketch Engine thesaurus for synonym extraction by 32% in terms of MAP. Our work shows promising results for synonym extraction suggesting that our method can also be used with other languages.

## 1. Introduction

Synonymy is one of the best known lexical semantic relations. Synonyms are words that have similar meanings or the same meaning but different forms. In contrast, the nouns "end" and "ending" have similar forms, but "end" is not considered to be a better synonym for "ending" than the noun "conclusion" [1]. Automatic extraction of synonyms can enhance numerous Natural Language Processing (NLP) applications, such as question answering and information retrieval [2, 3], automatic lexical database generation [4], automatic text summarization [5], lexical entailment acquisition [6], and language generation [7].

The WordNet thesaurus is the thesaurus most widely used for synonyms [1], and various NLP applications use it as the synonym source. However, the largest version of WordNet is available for English, but it is small or not available at all for other languages because constructing such resources manually is time consuming and expensive. The Alma'any dictionary [8] is an Arabic/Arabic dictionary that has a section on Modern Standard Arabic synonyms and antonyms. Although it is larger than the Arabic Wordnet, it does not cover a significant number of Arabic terms and is not updated frequently.

There have been many attempts to develop a methodology for automatic extraction and discovery of synonyms. In the early days, pattern matching was used to extract synonyms for dictionary and thesaurus building; for example, McCrae and Collier [9] have employed a novel algorithm for synonym set extraction from the biomedical literature using lexical pattern discovery. Similarly, Wang and Hirst [10] proposed three novel methods: one machine learning approach and two rule-based methods to extract synonyms from definition texts in a dictionary. In [43], a combination of link structure of various online encyclopedias such as Wikipedia is used in combination with machine learning techniques. Conditional Random Field (CRF) models were trained and used to find synonyms on the web.

On the contrary, machine learning based on semantic and dependency features were used to extract Turkish synonyms [11]. Also, graph models were used to extract synonyms, for example, Wei [12] used the synonym graph to refine synonym extraction by following two approaches. The first one splits each extraction result into two parts (synonyms and noise). The second approach ranks the extracted synonym words by computing their semantic distance in the synonym graph. In Hu et al.'s work [13], Sahin classified different relations which are hyponymy, holonymy, and antonymy pairs in Turkish using a set of machine learning classifiers. He examined the effect of using different features including lexico-syntactic patterns, cosine similarity of Word2Vec vectors, and WordNet similarity measures. The best result equals 84% of F1 which was obtained by the random forest classifier using lexico-syntactic pattern features.

In recent years, the research focus has been on extracting synonyms using word embeddings since they capture different types of similarities and relatedness between words. Word embeddings are represented as low-dimensional vectors. The dimensions of distributed word vectors are word features that represent different aspects of word meaning [14]. Word embeddings have been used widely to extract and detect synonyms in English [15–19]. The author in [19] uses cosine similarity, "a measure of similarity between two nonzero vectors of an inner product space that measures the cosine of the angle between them" [20]. However, the list of most similar words retrieved using cosine similarity contains words that share some relation with the seed word including not only synonymy but also other relations such as inflections and antonyms [19]. Thus, cosine similarity alone is not an effective measure for synonym extraction. Similarly, Mohammed [21] used word embeddings with a supervised neural network classifier to classify synonyms from other related words in an attempt to overcome the deficiency of the cosine similarity measure. However, using supervised approaches requires extensive human labor and is not efficient for many NLP tasks. Also, Zheng et al. [22] explored two models for semantic relation extraction: the CNN-based model and LSTM-based model.

This paper focuses on Arabic, a Semitic language. Almost 500 million people around the globe speak Arabic. It is the language officially used in many Arabic countries with different dialects. Formal written Arabic is Modern Standard Arabic (MSA). MSA is one form of classical Arabic, and the language used in the Qur'an, but currently contains a larger and modernized vocabulary. Because it is understood by almost everyone in the Arab world, MSA is used as the formal language in media and education. Arabic has spelling, grammar, and pronunciation features that distinguish it from other languages [23]. Arabic is one of the richest languages morphologically, and it is the sixth most-spoken language worldwide. Similarly to other languages, Arabic has semantic relations among its words that connect them to make sense of utterances [24].

To the best of our knowledge, the only study conducted on automatic Arabic synonym extraction involved constructing Quranic Arabic WordNet (QAWN) using the vector space model (VSM) as word representations and cosine similarity [23]. However, that study did not obtain adequate results because it clustered similar words to create a synset that was not validated as containing actual synonyms.

In this paper, we present an unsupervised and independent language methodology for automatic synonym extraction, using a two-phase approach. In the first phase, we trained our Arabic word embeddings using two very large corpora, the King Saud University Corpus of Classical Arabic (KSUCCA) [25] and Gigaword [26], with extensive experimentation to determine the best training settings for capturing the synonymy relations. Then, we used SynoExtractor, a novel pipeline that we developed to extract synonyms by filtering similar embeddings to address cosine similarity deficiencies. We used the Alma'any thesaurus as a gold standard and manual evaluation to evaluate our methodology. In addition, we compared our methodology with Sketch Engine, a text analysis tool that is used to explore text and find relations between words [27]. Consequently, this paper aims to answer the following research questions:

(1) Can using our proposed pipeline in word embeddings space extract Arabic synonyms?

(2) Does using the new measure of Relative Cosine Similarity (RCS) instead of cosine similarity in word embeddings space enhance Arabic synonym extraction?

(3) Is our approach for synonym extraction comparable to that of Sketch Engine?

The remainder of the paper is organized as follows. Section 2 discusses the related work on synonym extraction in Arabic and other languages. Section 3 describes our methodology. Section 4 presents our experimental setup. Section 5 reports on word embedding training. Section 6 contains our experimental results and discussion. Section 7 concludes the paper with discussion of our method's limitations and our future work.

## 2. Related Work

In this section, we review recent studies on automatic synonym extraction and detection using a variety of supervised and unsupervised approaches.

*2.1. Supervised Approaches.* Supervised approaches require annotated data to extract synonyms. In their work [11], Yıldız et al. investigated using a hybrid pattern-based approach with supervised machine learning to extract Turkish synonyms. They generated some lexico-syntactic patterns for a very large corpus based on specified grammatical relations between words. Then, they used those patterns as features for a logistic regression as a supervised machine learning classifier to detect synonyms, achieving an F-score of 80.3%.

Word embeddings capture some of the relations between words. However, they cannot detect the type of the relation or whether two words are similar or related. Thus, some researchers have attempted to use supervision from language thesauri to train sensitive word embeddings for semantic relations. One study by Ono et al. [15] proposed a word embeddings training model for detecting antonyms using distributional information for Wikipedia article raw text and thesauri information including WordNet [1] and Roget [28]. They modified the SkipGram (SG) objective function and used supervised synonym and antonym information from thesauri, with distributional information from large-scale unlabeled text data. In addition, they evaluated their model using a GRE antonym detection task and obtained an F-score of 89%.

Using a similar approach, Dou et al. [16] proposed a new word embeddings' training model, Word Embeddings Using Thesauri and SentiWordNet (WE-TSD). In this model, the researchers modified the objective function of the SG model and injected antonym and synonym information from a thesaurus into the embeddings. They evaluated their embeddings in three tasks, GRE antonym detection, word similarity, and semantic textual similarity. Their model obtained an F-score of 92% on the GRE antonym detection task.

Nguyen et al. [17] proposed a modification of the SG Word2Vec model by integrating distributional lexical contrast information as a supervision for word embeddings and modifying the SG objective function. They strengthened the dominant word similarity features based on the lexical contrast features using a thesaurus. Their embeddings achieved a precision between 0.66 and 0.76 for adjectives, nouns, and verbs in distinguishing synonyms and antonyms and outperformed the advanced models in guessing word similarities in SimLex-999.

*2.2. Unsupervised Approaches.* In contrast to supervised approaches, unsupervised approaches require no labelled data in training with a minimum of human supervision. Zhang et al. [18] used Word2Vec embeddings with spectral clustering for automatic synonym extraction. They trained the English Wikipedia corpus with a Word2Vec model, selected some keywords from the corpus, and then extracted the most similar words for each of them based on their cosine similarity. Next, a graph with the terms' adjacency matrix was constructed. Finally, they clustered similar words using spectral clustering. For the evaluation, they compared the use of spectral clustering to that of K-means clustering. Spectral clustering outperformed K-means clustering and achieved a precision of 80.8%, a recall of 74.4%, and an F-score of 77.5%.

Leeuwenberg et al. created an automatic approach for synonym extraction using word embeddings in two languages, English and German [19]. They used the NewsCrawl corpus, tagged it with part-of-speech (POS) tags, and trained it with different word embedding models, Word2Vec, SG with continuous bag of words (CBOW), and Glove. The researchers then evaluated the use of cosine similarity for synonym extraction from word embeddings and determined that cosine similarity is not a good measure for capturing synonyms. Consequently, they proposed RCS, a new measure that can be used to capture synonyms rather than inflections or related words. Then, they evaluated their approach automatically using the WordNet and GermaNet thesauri and conducted a human evaluation for 150 extracted pairs for each language. The evaluation results indicated that the use of POS tags and the relative cosine measure improved the precision of synonym extraction from word embeddings for the two languages. In addition, the best model used that captures synonym relations was found to be CBOW. Their model is language-independent and can be applied to other languages.

Based on the previous study, Mohammed [21] attempted to follow a similar methodology to extract synonyms from word embeddings trained with Word2Vec models. The researcher trained her own embeddings using the News-Crawl 2014 corpus. Then, she developed a supervised neural network classifier to classify synonyms from other related words. However, the supervision and annotation in her methodology were not suitable for our purpose, since we are aiming in this research to train a nonsupervised model.

The only study for Arabic synonym extraction was conducted by AlMaayah et al. [23]. They constructed Quranic Arabic WordNet (QAWN) using three resources, the Boundary Annotated Quran, some lexicon resources that were used to collect a set of derived words for Quranic words, and some traditional Arabic dictionaries. They represented the Quran using the VSM and extracted the Quran word meaning from the Arabic dictionaries. Then, they used cosine similarity to measure the similarity between the Quranic words and their extracted definitions, clustering similar words to create a synset. AlMaayah et al. obtained 6,918 synsets containing 8,400 word senses. They evaluated the effectiveness of the synsets in an information retrieval system and found that it increased the baseline performance from 7.01% to 34.13% in recall. However, their results were very low in terms of precision.

From the previous studies, we noticed that the supervised approach is the most accurate. However, this approach requires labelled data, and it was used to distinguish identified synonyms from other relations (i.e., labelled relations) and not for extracting relations. Embedding studies using this approach focus on modifying word embedding learning models to train sensitive word embeddings for specific relations. The modification requires supervision using large sets of relation examples from lexical thesauri, which are not available for all languages. In the unsupervised approach, any raw text corpus can be used for extraction with clustering techniques based on a distributional hypothesis. The use of word embeddings for unsupervised relation extraction is very promising because such embeddings are language-independent. Thus, it will be good if we can use it as a starting point since it captures similarity between words. However, it requires another layer to filter the synonyms from other similar relations. For these reasons, we developed the SynoExtractor pipeline to filter Arabic synonyms that are extracted from newly trained word embeddings using two Arabic corpora.

## 3. Methodology

In this section, we present our methodology, which involves two phases. In the first phase, we trained our Arabic embeddings with extensive experimentation to determine the best training settings for capturing synonymy relations. Then, we used SynoExtractor, a novel pipeline that we developed to extract synonyms by filtering the most similar words for a given word using cosine similarity.

*3.1. Word Embedding Training.* To obtain the word embeddings that are used for the synonym extraction process, we developed our own Arabic word embeddings' models from two Arabic corpora. Figure 1 shows the steps we followed to generate the final embeddings.

First, we used preprocessing on two corpora (KSUCCA and Gigaword) (more about these corpora can be found in the experimental setup section), including tokenization, diacritic removal, English letter and number removal, and normalization. In the normalization step, we removed tatweel (Elongation) (_) and replaced (ة) with ((ه and (أ،إ،آ) with (ا). Then, using the two Word2Vec models, CBOW and SG, we trained the corpora with different hyperparameters, including the number of vector dimensions, the contextual window size, and the number of training iterations. The goal of training with different hyperparameters was to fine tune the models and determine the best embeddings for synonym extraction. In addition, in this phase, we investigated the effect of adding POS tags to words before the training process. Finally, we selected the best models to use in the synonym extraction phase.

*3.2. SynoExtractor.* We treated synonym extraction as an unsupervised relation extraction task using trained word embeddings and cosine similarity. SynoExtractor is a language-independent pipeline that reads cosine similarities and filters the most similar words to yield a synonym list. Using filters is a novel approach that resulted from investigating language resources that describe the nature of synonym relations [29, 30]. The SynoExtractor pipeline is illustrated in Figure 2 and was created as follows:

(1) We found the most similar words using cosine similarity for a set of preselected words from our corpus. We used the cosine similarity measure because words that appear in the same context tend to have high scores. However, the list of most similar words retrieved using cosine similarity contains a list of words that share some relation with a seed word, including synonymy and other kinds of related words, such as inflections and antonyms. See Figure 3 for the results of the Sketch Engine embeddings' viewer [31] after searching for the word "قبيح/ugly." Consequently, we applied three filters to the extracted word lists in the next steps.

(2) Lemmatization filter: this filter is used to remove the inflections, discarding any two words that have the same lemma. Table 1 shows an example of finding the most similar words for "قبيح," which means "ugly." We noticed that the first similar words were inflected forms of "قبيح." The inflectional problem is observed in an inflectional language context [32] such as Arabic. However, synonymous words are words that have similar or the same meaning but different forms [29].

    We used FARASA (http://qatsdemo.cloudapp.net/farasa/) to create a lemma dictionary for our corpora. Each lemma dictionary has the corpus words with their lemmas. Table 2 shows a sample from our lemma dictionary.

(3) Collocation filter: this filter retains words that share a collocation with the seed word because previous research such as by Kaminski [30] has shown that two words with the same collocations are synonyms. Applying this filter requires the use of a collocation dictionary. We used the Natural Language Toolkit (NLTK) collocation python library (https://www.nltk.org/) to generate a collocation dictionary for our corpora. The collocations were generated as follows:

    (1) Using the Pointwise Mutual Information (PMI) measure for calculating collocation reliability, PMI is used to measure the collocation association score because good collocation pairs have high PMI scores [33]
    (2) Removing stop words
    (3) Removing collocations that appear fewer than five times because we tried different frequencies (1, 2, 3, 5, and 10), and five was the best threshold Table 3 shows a sample of our collocation dictionary.

(4) POS filter: this final filter retains words that have the same POS as the seed word because synonyms typically have the same POS. This gives us the final synonym list for each seed word.

## 4. Experimental Setup

In this section, we present the corpora used for word embeddings generation and synonym extraction, and we describe the methodology for generating the synonyms lexicon from the Alma'any thesaurus to be used in automatic evaluation. In addition, we report the measures used for experimental evaluation.

*4.1. Corpora Used.* To obtain high-quality word embeddings reflecting word relations, it is best to train very large corpora containing one million or more words [32]. We used two such corpora of Arabic text to train our embeddings,

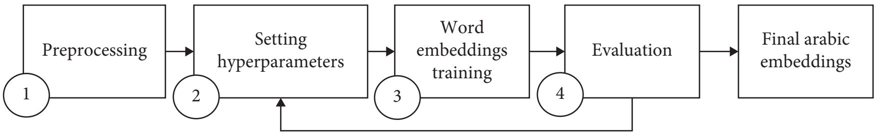


Figure 1: Steps for generating the final embeddings. (1) Preprocessing. (2) Setting hyperparameters. (3) Word embeddings' training. (4) Evaluation to choose the best embeddings that will be used in the synonyms extraction process.

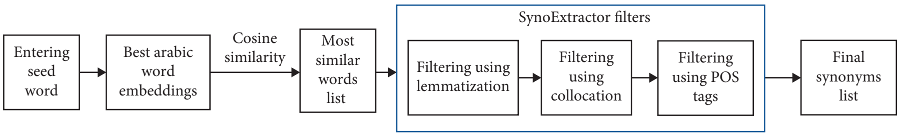


Figure 2: SynoExtractor pipeline: it starts with finding most similar words using cosine similarity for a set of seed words; then, the list of the most similar words are filtered using SynoExtractor filters to have the final synonyms list.

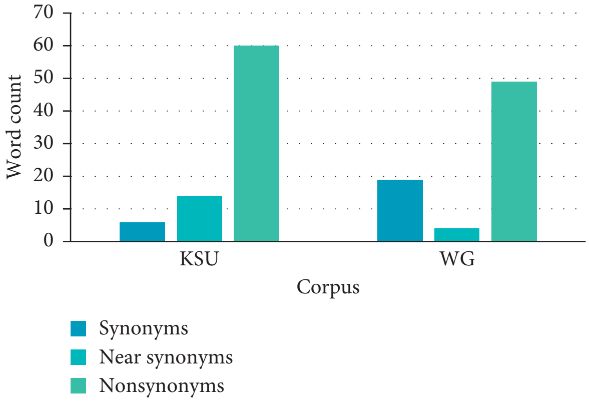


Figure 3: Number of words filtered by the collocation filter based on their evaluation.

Table 1: The most similar words for "قبيح/ugly" produced by Sketch Engine.

| Word | Cosine similarity score |
|---|---|
| قبح | 0.849 |
| قبيحا | 0.821 |
| القبيح | 0.819 |
| اقبح | 0.812 |
| يقبح | 0.798 |
| ذميم | 0.797 |
| تقبيح | 0.781 |

Table 2: Sample from the lemma dictionary.

| Word | Lemma |
|---|---|
| الرحيم (most merciful) | رحيم (most merciful) |
| المفلحون (the successful) | مفلح (the successful) |
| ربهم (their god) | رب (their god) |
| أنفسهم (themselves) | نفس (themselves) |

Table 3: Sample of the collocation dictionary.

| Word 1 | Word 2 | PMI score |
|---|---|---|
| قال/said | اعرابي/Arabian | 2.62 |
| اتى/came | الشفيع/intercessor | 6.43 |
| علاج/recover | فساد/corruption | 6.3 |

beginning with KSUCCA, which contains 50 million words [25]. This corpus has a rich content of classical raw Arabic text from many genres, including religion, linguistics, literature, science, sociology, and biography. We trained only one million words from KSUCCA using the Word2Vec models, CBOW and SG, since it was available at the time of conducting this experiment (October 2018). Table 4 shows the KSUCCA statistics [23].

The second corpus was Arabic Gigaword Third Edition, which contains 1,994,735 words of MSA Arabic text collected from six Arabic newspapers, Agence France Presse, Assabah, Al Hayat, An Nahar, Ummah Press, and Xinhua News Agency. Table 5 shows the statistics of this corpus [26].

*4.2. Synonyms' Lexicon.* To choose the seed words and evaluate the SynoExtractor pipeline, we developed an Arabic synonym lexicon list extracted from the Alma'any dictionary to be used as a benchmark for automatic evaluation of Arabic synonym extraction from the corpora. The criteria followed were suggested by a linguistic expert in Arabic, who also evaluated the final results.

The steps followed to extract the lexicons are as follows. First, we selected fifteen seed words from the Alma'any synonyms' dictionary based on the following criteria:

(1) The word is from a specific semantic field (in our case, Earth or News)
(2) The word is a name or verb
(3) Each word has at least four synonyms
(4) The selected word appears in the corpus

Second, we performed some filtration for the compiled synonyms' list for each word taking the following steps:

(1) Remove compound synonyms
(2) Remove synonyms that did not appear in the corpora
(3) Have the remainder of the words reviewed by an Arabic language specialist
(4) Apply the same preprocessing steps that were applied to the corpora

Table 4: Statistics of the KSUCCA corpus [23].

| Genre | Number of texts | Number of words | Percentage (%) |
|---|---|---|---|
| Religion | 150 | 23645087 | 46.73 |
| Linguistics | 56 | 7093966 | 14.02 |
| Literature | 104 | 7224504 | 14.28 |
| Science | 42 | 6429133 | 12.71 |
| Sociology | 32 | 2709774 | 5.36 |
| Biography | 26 | 3499948 | 6.92 |
| Total | 410 | 50602412 | 100 |

Table 5: Statistics of Arabic Gigaword Third Edition corpus.

| Source | Files | DOCs | Words |
|---|---|---|---|
| Agence France Presse | 152 | 147612 | 798436 |
| Assabah | 28 | 6587 | 15410 |
| Al Hayat | 142 | 171502 | 378353 |
| An Nahar | 134 | 193732 | 449340 |
| Ummah Press | 24 | 1201 | 4645 |
| Xinhua News Agency | 67 | 56165 | 348551 |
| Total | 547 | 576799 | 1994735 |

Table 6 shows a sample of the synonym lexicon for the raw corpora. Table 7 shows a sample of the synonym lexicon for the POS-tagged corpus.

### 4.3. Evaluation Measures.

*4.3. Evaluation Measures.* As evaluation measures for relation extraction, we chose Precision and Recall. Precision ($P$) is calculated as the proportion of correctly predicted synonym word pairs from all predictions. Since the synonyms were retrieved as ranked results based on cosine similarity, we calculated the precision at different ranks from one to ten. Recall (R) is calculated as the proportion of synonym pairs that were correctly predicted from all synonym pairs present in the Alma'any thesaurus. In addition, we calculated the mean average precision (MAP) and the mean average recall (MAR) for the extracted list to compare the models. The equations for the selected measures are as follows:

$$\text{precision} = \frac{TP}{TP + FP}, \tag{1}$$

$$\text{recall} = \frac{TP}{TP + FN}, \tag{2}$$

$$\text{MAP} = \frac{\sum_{i=1}^{|Q|} \text{Avg}\left(Pw_i\right)}{|W|}, \tag{3}$$

$$\text{MAR} = \frac{\sum_{i=1}^{|Q|} \text{Avg}\left(Rw_i\right)}{|W|}, \tag{4}$$

where $TP$ = True extracted relation, $FP$ = False extracted relation, $FN$ = False unextracted relation, $O = 0.5$, $W$ is the number of seed words, and $w_i$ is each word from the seed words.

## 5. Word Embeddings' Training

We opted to train our own word embeddings to provide new pretrained embeddings for the research community, to conduct a wide range of experimentation, and to explore the best training settings for generating high quality Arabic word embeddings capturing synonymy. The first experiment was conducted using unsupervised training on raw text to examine the effects of changing the hyperparameters for word embeddings' training. In the second, we examined the impact of tagging Arabic text with POS tags before word embeddings' training as a weak supervision on synonym extraction. Now, we present the result of word training experiments along with the best hyperparameters and the models chosen for synonym extraction experimentation.

*5.1. Unsupervised Training.* In this experiment, we trained the KSUCCA and Gigaword corpora on CBOW and SG models. An investigation was conducted on the training hyperparameters, including the following:

(1) Window size: context words are words surrounding the target word. We experimented with windows of sizes two, four, eight, and 16, as reported in [19], and we also attempted with a window of size five because it is the default setting for Word2Vec models [34].

(2) Dimensions: the number of vector dimensions. Our range in the experiment was 150, 300, and 600.

(3) Epochs: an epoch is one complete pass through the training data. It is typical to train a deep neural network for multiple epochs, and our range was 5, 10, 20, 30, 40, and 50.

Then, results were obtained through calculating the evaluation measures (p@(1–10), MAP, and MAR) by comparing the retrieved list with the synonym lists in the synonyms' lexicon (the lexicon extracted from the Alma'any [8] synonym thesaurus).

Tables 8 and 9 present the best results from both corpora, which we considered later as our baseline. The baseline for Gigaword was higher than that for KSUCCA, in part because of the size of the Gigaword corpus that the corpus contains more words.

Table 10 summarizes the best values for the hyperparameters: two for the window size, 150 for the dimensions, and 50 for the number of epochs for all models and corpora. It appears that the small window size was better for capturing synonymous words, perhaps because synonymous words tend to appear in similar contexts [35]. In addition, they mostly have the same collocation words [30] that can be captured in a window of size two. We experimented with different dimensions, 150, 300, and 600, with 150 showing the best results. Limited computational power resulted in the maximum epoch value being 50.

Table 6: Sample of the synonym lexicon.

| Word | Synonym list |
|---|---|
| جاء/came | أَتَى، أَقْبَل, زار , طَرَق , غَشِي , قَدِم , وافى , وَرَد , وَفَد /came, get in, reach, arrive, visit, hit, attain |
| علاج/medicine | بَلْسَم، تَجْرِبَة , تِرْيَاق , دَوَاء , طِبّ , عقّار , مُدَاوَاة , مُعَالَجَة /therapy, medical, diagnostic, pills, pharmacology |

Table 7: Sample of the synonym lexicon for the POS-tagged corpus.

| Word | Synonym list |
|---|---|
| جاء_ف came_verb | اتى_ف , اقبل_ف, زار_ف , طرق_ف , غشي_ف , قدم_ف , وافى_ف, ورد_ف, وفد_ا came_verb, get in_verb, reach_verb, arrive_verb, visit_verb, hit_verb, attain_verb |
| علاج_ا Medicine_noun | بلسم_ا, تجربه_ا , ترياق_ا, دواء_ا , طب_ا , عقار_ا , مداواه_ا , معالجه_ا therapy_noun, medical_noun, diagnostic_noun, pills_noun, pharmacology_noun |

Table 8: The best-obtained results from KSUCCA corpus for p@(1–10), MAP, and MAR after fine tuning CBOW and SG training parameters which were obtained with window size = 2, dimensions = 150, and epochs = 50 (baseline results for KSUCCA corpus).

| | P@1 | P@2 | P@3 | P@4 | P@5 | P@6 | P@7 | P@8 | P@9 | P@10 | MAP | MAR |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CBOW | 0.4 | 0.266 | 0.22 | 0.183 | 0.146 | 0.122 | 0.1 | 0.1 | 0.088 | 0.086 | 0.402 | 0.111 |
| SG | 0.266 | 0.133 | 0.155 | 0.116 | 0.106 | 0.088 | 0.083 | 0.083 | 0.074 | 0.066 | 0.0272 | 0.087 |

Table 9: The best-obtained results from the Gigaword corpus for p@(1–10), MAP, and MAR after fine tuning CBOW and SG training parameters which were obtained with window size = 2, dimensions = 150, and epochs = 50 (baseline results for the Gigaword corpus).

| | P@1 | P@2 | P@3 | P@4 | P@5 | P@6 | P@7 | P@8 | P@9 | P@10 | MAP | MAR |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CBOW | 0.533 | 0.4 | 0.31 | 0.25 | 0.200 | 0.177 | 0.158 | 0.158 | 0.140 | 0.14 | 0.535 | 0.164 |
| SG | 0.333 | 0.233 | 0.222 | 0.2 | 0.213 | 0.188 | 0.158 | 0.158 | 0.140 | 0.126 | 0.388 | 0.129 |

Table 10: Best hyperparameter values (window size, dimensions, and epochs) for word embeddings' training with KSUCCA and Gigaword corpora on CBOW and SG models.

| Corpus | Model | Window size | Dimensions | Epochs |
|---|---|---|---|---|
| KSUCCA | CBOW | 2 | 150 | 50 |
| | SG | 2 | 150 | 40 |
| Gigaword | CBOW | 2 | 150 | 50 |
| | SG | 2 | 150 | 40 |

*5.2. POS Weak-Supervised Training.* Leeuwenberg et al. [19] compared extracting synonyms from word embeddings' most similar words with and without POS tags. From their experiments, they concluded that using POS tags can help in three respects: (1) word senses can be separated with a slight effect; (2) words that are not quite similar in terms of grammar (e.g., plurals) can be filtered; (3) in cases in which there were few or no synonyms for words of a particular category (e.g., names), these words can be eliminated. In addition, relations typically appear between words that have the same POS [29]. Thus, in this experiment, we aimed to examine the effect of adding POS tags before word embedding training for Arabic text.

We tagged the KSUCCA corpus before training with POS tags (verb, noun, adjective, and adverb) using MADAMIRA [36]. We used (ا_) for nouns, e.g., (بيت_ا); (ف_) for verbs, e.g., (جاء_ف); (ص_) for adjectives, e.g., (جميل_ص/beautiful_adj); (ح_) for adverbs, e.g., (سعيدا_ح/happy_adv).

Table 11 shows the best results of synonym extraction from KSUCCA POS-tagged. The CBOW and SG performed better without POS tags. Adding POS tags degraded the performance of synonym extraction from Arabic text. This might be attributable to the quality of the POS tagger used, which labelled some words with incorrect tags. For example, the word تقهقر "lag behind," a verb synonymous with تأخر "was late," was tagged as a noun (i.e., " تقهقر_ا ) and did not appear in the synonym list for تأخر. Alternatively, it could be that KSUCCA text does not have diacritics, which are necessary for differentiating between Arabic words that have the same letters but different meanings (e.g., كَتَبَ "wrote" and كُتُب "books"). As a result, the POS word embeddings did not obtain high precision values. Additionally, the unsupervised models CBOW and SG performed better than training with weak supervision using POS tags.

As a conclusion, the CBOW was better than the SG for both corpora with the following settings: two for the window size, 150 for the number of dimensions, and 50 for the number of epochs. Consequently, we chose CBOW embeddings without POS tags for our synonym extraction experiments.

## 6. Results and Discussion

In this section, we describe the experiments for evaluating relations extracted using the SynoExtractor pipeline.

Table 12 shows each experiment's purpose and the corpora used. The first was conducted to examine the effectiveness of the SynoExtractor pipeline. The second was conducted to examine the effectiveness of using RCS as a

Table 11: Best results of synonym extraction from KSUCCA POS-tagged.

| | P@1 | P@2 | P@3 | P@4 | P@5 | P@6 | P@7 | P@8 | P@9 | P@10 | MAP | MAR |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CBOW | 0.4 | 0.266 | 0.22 | 0.183 | 0.146 | 0.122 | 0.1 | 0.1 | 0.088 | 0.086 | 0.402 | 0.111 |
| SG | 0.266 | 0.133 | 0.155 | 0.116 | 0.106 | 0.088 | 0.083 | 0.083 | 0.074 | 0.066 | 0.0272 | 0.087 |
| POS-CBOW | 0.266 | 0.2 | 0.177 | 0.133 | 0.119 | 0.122 | 0.091 | 0.091 | 0.081 | 0.08 | 0.292 | 0.109 |
| POS-SG | 0.2 | 0.166 | 0.133 | 0.116 | 0.093 | 0.077 | 0.058 | 0.058 | 0.051 | 0.046 | 0.27 | 0.068 |

Table 12: The Llist of synonym extraction experiments.

| Research question | Experiment | Purpose | Corpus |
|---|---|---|---|
| 1 | Synonym extraction | Examine the effectiveness of the SynoExtractor pipeline | KSUCCA CCA – Gigaword |
| 2 | RCS for synonym extraction | Examine the effectiveness of using RCS for synonym extraction | KSUCCA – Gigaword |
| 3 | SynoExtractor vs. Sketch Engine thesaurus | Compare the SynoExtractor pipeline with the Sketch Engine thesaurus for synonym extraction | KSUCCA |

measure for synonym extraction. RCS was introduced by Leeuwenberg et al. [19] for synonym extraction. The third and final experiment was conducted to compare our methodology's results with the Sketch Engine thesaurus for synonym extraction.

*6.1. Synonym Extraction.* This experiment was conducted to answer the first research question: can using our proposed pipeline in word embeddings space extract Arabic synonyms? We applied the SynoExtractor pipeline for synonym extraction on the best models for KSUCCA and Gigaword embeddings that were developed in the word embeddings' training phase. The models were trained on the CBOW with window size = 2, dimensions = 150, and epochs = 50.

In this experiment, we used automatic evaluation by comparing a synonym lexicon extracted from the Alma'any thesaurus with the synonyms extracted using SynoExtractor. We extracted the twenty most similar words for the selected seed words in the synonym's lexicon. We decided to retrieve twenty words after experimenting with the range of numbers 10, 20, and 30, thereby concluding that 20 was the number most suitable for capturing real synonyms because retrieving more only rarely found further synonyms.

Then, we used the SynoExtractor pipeline to filter the extracted similar words list and retain the synonyms. Table 13 shows the results of using the SynoExtractor pipeline for synonym extraction on the KSUCCA corpus.

We calculated *P*-ranks (1–10), MAP, and MAR after applying each filter. The lemmatization filter showed an overall improvement in terms of MAP by 6.2%, while the collocation filter increased the improvement to 12.4%. This indicates the benefit of applying lemmatization and collocation filters. However, the POS filter had no effect after the collocation filter. Further analysis of the results revealed that the extracted pairs had the same POS tags after the collocation filter (e.g., جاء - أتى, which means "came"). Therefore, the POS filter had no effect on the final results. Additionally, there was some enhancement in terms of the first six *P*-ranks. However, the performance has decreased compared to its listing in the Alma'any thesaurus in terms of MAR after the collocation filter. This indicates that it discarded some synonyms that can be found in the Alma'any thesaurus.

Table 14 shows the results of applying the SynoExtractor pipeline for synonym extraction on Gigaword embeddings. The lemmatization filter obtained a 0.607 MAP and showed similar behavior for both corpora because it increased the extraction precision. It shows a 13.5% overall improvement in terms of all *P*-ranks and MAP, while the collocation filter decreased the MAP to 0.479. This indicates it discarded some extracted synonyms from the Alma'any thesaurus, the opposite of what we found after the human evaluation. We expect that increasing the collocation quality will enhance the usefulness of the collocation filter. The POS filter showed similar behavior for both corpora.

Since the Alma'any has limited coverage, this evaluation might not show exact precision. Therefore, we used human evaluation on the synonyms extracted from the previous experiments. Two Arabic language specialists evaluated the synonyms extracted from the Gigaword and KSUCCA corpora by SynoExtractor. They classified the words into three classes, synonyms, near synonyms, and nonsynonyms. The kappa agreement between their evaluations reached 0.95, indicating that they agreed on most words' classes. Tables 15 and 16 show the *P*-ranks and MAP that we calculated after the human evaluation. However, MAR was not calculated because there is no reference that reports every synonym's list for each word in the corpus.

According to these results, the baseline of the human evaluation was higher than the baseline of the automatic evaluation for both corpora (25% for KSUCCA and 12% for Gigaword); the lemmatization filter shows overall improvement in terms of MAP equal to 10% for KSUCCA, while the collocation filter was improved by 21%. For the Gigaword corpus, the improvements were 22% for the lemmatization filter and 25% for all filters. This indicates that the collocation was useful with the Gigaword corpus contrary to what we saw in the automatic evaluation. The low performance in the automatic evaluation resulted from the misclassification of words that were correct synonyms but that are not covered in

Table 13: Automatic evaluation results after using the SynoExtractor pipeline on KSUCCA embeddings.

| | P@1 | P@2 | P@3 | P@4 | P@5 | P@6 | P@7 | P@8 | P@9 | P@10 | MAP | MAR |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Baseline | 0.4 | 0.266 | 0.22 | 0.183 | 0.146 | 0.122 | 0.1 | 0.1 | 0.088 | 0.086 | 0.402 | 0.111 |
| LM filter | 0.4 | 0.3 | 0.244 | 0.183 | 0.160 | 0.133 | 0.116 | 0.116 | 0.103 | 0.093 | 0.427 | 0.111 |
| Co filter | 0.4 | 0.3 | 0.244 | 0.183 | 0.160 | 0.133 | 0.1 | 0.1 | 0.088 | 0.086 | 0.452 | 0.102 |
| POS | 0.4 | 0.3 | 0.244 | 0.183 | 0.160 | 0.133 | 0.1 | 0.1 | 0.088 | 0.086 | 0.452 | 0.102 |

Table 14: Results of using the SynoExtractor pipeline on Gigaword embeddings.

| | P@1 | P@2 | P@3 | P@4 | P@5 | P@6 | P@7 | P@8 | P@9 | P@10 | MAP | MAR |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Baseline | 0.533 | 0.4 | 0.31 | 0.25 | 0.200 | 0.177 | 0.158 | 0.158 | 0.140 | 0.14 | 0.535 | 0.164 |
| LM filter | 0.6 | 0.433 | 0.355 | 0.283 | 0.240 | 0.211 | 0.175 | 0.175 | 0.162 | 0.166 | 0.607 | 0.164 |
| Co filter | 0.466 | 0.366 | 0.288 | 0.233 | 0.200 | 0.188 | 0.141 | 0.141 | 0.118 | 0.11 | 0.479 | 0.137 |
| POS | 0.466 | 0.366 | 0.288 | 0.233 | 0.200 | 0.188 | 0.141 | 0.141 | 0.118 | 0.11 | 0.479 | 0.137 |

Table 15: Human evaluation results of using the SynoExtractor pipeline on the KSUCCA corpus.

| | P@1 | P@2 | P@3 | P@4 | P@5 | P@6 | P@7 | P@8 | P@9 | P@10 | MAP |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Baseline | 0.733 | 0.533 | 0.466 | 0.366 | 0.320 | 0.322 | 0.291 | 0.291 | 0.266 | 0.266 | 0.501 |
| LM filter | 0.733 | 0.633 | 0.488 | 0.416 | 0.373 | 0.377 | 0.308 | 0.308 | 0.288 | 0.266 | 0.551 |
| Co filter | 0.733 | 0.633 | 0.533 | 0.45 | 0.413 | 0.411 | 0.3 | 0.3 | 0.281 | 0.266 | 0.605 |
| POS filter | 0.733 | 0.633 | 0.533 | 0.45 | 0.413 | 0.411 | 0.3 | 0.3 | 0.281 | 0.266 | 0.605 |

Table 16: The human evaluation results of applying the SynoExtractor pipeline on the Gigaword corpus.

| | P@1 | P@2 | P@3 | P@4 | P@5 | P@6 | P@7 | P@8 | P@9 | P@10 | MAP |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Baseline | 0.666 | 0.533 | 0.511 | 0.516 | 0.506 | 0.488 | 0.466 | 0.466 | 0.451 | 0.446 | 0.600 |
| LM filter | 0.8 | 0.666 | 0.666 | 0.683 | 0.64 | 0.633 | 0.541 | 0.541 | 0.548 | 0.526 | 0.731 |
| Co filter | 0.8 | 0.6 | 0.577 | 0.616 | 0.6 | 0.544 | 0.45 | 0.45 | 0.340 | 0.306 | 0.748 |
| POS filter | 0.8 | 0.6 | 0.577 | 0.616 | 0.6 | 0.544 | 0.45 | 0.45 | 0.340 | 0.306 | 0.748 |

the Alma'any thesaurus. As further evaluation of the collocation filter, we calculated the number of words filtered by the collocation filter for each category (synonyms, near synonyms, and nonsynonyms) from each corpus.

Figure 3 shows the number of filtered words from each category. This shows the effectiveness of the idea of the collocation filter because the number of nonsynonyms was ten times the number of synonyms in KSUCCA and 2.5 times the number of synonyms in Gigaword.

Figures 4(a) and 4(b) show a comparison between the automatic and human evaluations for the KSUCCA and Gigaword corpora. The results of the human evaluation were 83%, 100%, 112%, and 25% higher than the automatic evaluation for KSUCCA in terms of P@1, P@2, P@3, and MAP, respectively, and they were 71%, 64%, 100%, and 56% higher than the automatic evaluation for Gigaword in terms of P@1, P@2, P@3, and MAP, respectively. This highlights the limitations of the Alma'any thesaurus and shows that our methodology can be used to enhance such a thesaurus. In summary, our experiment shows that the SynoExtractor pipeline is capable of extracting a substantial number of Arabic synonyms.

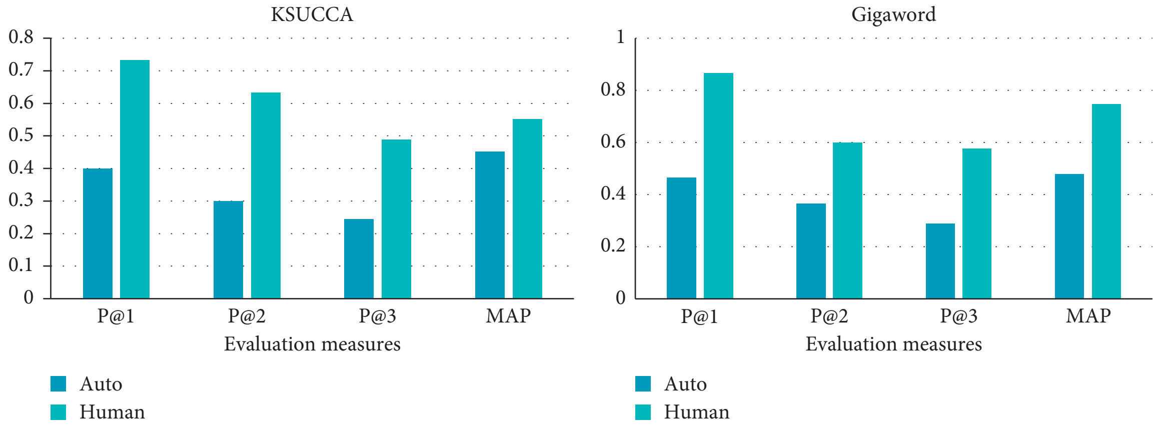
## 7. RCS for Synonym Extraction

Leeuwenberg et al. [19] introduced RCS, a measure that can be used for synonym extraction. The RCS between two words is equal to their cosine similarity divided by the summation of the cosine similarity for the $n$ most similar words. It is calculated based on the following:

$$rcs_n\left(w_i, w_j\right) = \frac{\text{cosine\_similarity}\left(w_i, w_j\right)}{\sum_{w_c \in TOP_c} \text{cosine\_similarity}\left(w_i, w_c\right)}, \quad (5)$$

where $n$ is the selected threshold for the most similar words, $w_i$ is the first word, $w_j$ is the second word, and $w_c$ is one of the $n$ most similar words. RCS assigns words that have high cosine similarity scores a higher value than other most similar words. The $n$ threshold is determined by experimentation. Two-word pairs are considered synonyms if their $rcs_n \geq n/100$.

This experiment was conducted to answer the second research question: does using RCS instead of cosine similarity in word embeddings' space enhance Arabic synonym extraction? We calculated the RCS values for the extracted synonyms from the Gigaword corpus experiment. We chose Gigaword because it had the best results for synonym extraction. The selected threshold value is $n = 10$. Leeuwenberg et al. stated that the extracted word will be a synonym if its RCS value is equal to or greater than $n/100 = 0.1$. Table 17 shows a sample of the synonyms extracted by the SynoExtractor pipeline along with their RCS values. They are classified into synonyms and nonsynonyms based on the human evaluation. For each word, there is a true extracted synonym with an RCS



(a)

(b)

Figure 4: Comparison between the automatic and human evaluations. (a) KSUCCA. (b) Gigaword in terms of $P$-ranks (1–3) and MAP.

Table 17: Sample of the synonyms extracted by the SynoExtractor pipeline along with their RCS values.

| | Seed word | Synonym | RCS | Nonsynonym | RCS |
|---|---|---|---|---|---|
| 1 | داء/disease | مرض/disease<br>وباء/pandemic | 0.117<br>0.099 | — | — |
| 2 | ذكاء/intelligence | دهاء/brilliant<br>نباهة/bright | 0.109<br>0.091 | غباء/stupidity | 0.106 |
| 3 | بدأ/start | باشر/start<br>استأنف/retreat | 0.104<br>0.099 | عاود/come back | 0.103 |
| 4 | انتهاء/end | انقضاء/finish<br>اكتمال/complete | 0.117<br>0.092 | بدء/start | 0.126 |
| 5 | فاز/win | تغلب/win<br>أحرز/achieve | 0.108<br>0.092 | احتفظ/keep | 0.100 |

value equal to or greater than 0.1. On the contrary, there is also a true extracted synonym with an RCS value less than 0.1. In addition, we noticed that the RCS value decreases when we go down in the most similar words' list similarly to the behavior of cosine similarity. Furthermore, there are nonsynonyms that have RCS values greater than 0.1. Therefore, using RCS did not improve Arabic synonym extraction. However, most of the nonsynonymous words are antonyms, and they are the words most similar to synonyms.

### 7.1. SynoExtractor vs. Sketch Engine.

*7.1. SynoExtractor vs. Sketch Engine.* Sketch Engine is a text analysis tool that is used to explore text and find relations between words [27]. It has a thesaurus feature used to extract synonyms from text, relying on word occurrence frequency in text. Two words are considered synonyms if they are collocated with the same word. When they share more collocations, they have a higher similarity score and are nominated as candidate synonyms.

This experiment was conducted to answer the third research question: is our approach for synonym extraction comparable to that of Sketch Engine? We used Sketch Engine to extract the synonyms' list for seed words taken from KSUCCA and applied automatic evaluation to the extracted synonyms. Table 18 shows a comparison between Sketch Engine and the SynoExtractor pipeline for synonym extraction.

The results show that SynoExtractor outperformed Sketch Engine in the first five precision ranks, MAP, and MAR. Additionally, the baseline for our pipeline using cosine similarity to retrieve similar words outperformed Sketch Engine for synonym extraction. This demonstrates the benefit of using the SynoExtractor pipeline on word embedding models.

## 8. Discussion

Filtration improved the precision of synonym extraction by 21% for KSUCCA and 25% for Gigaword in terms of MAP. The lemmatization filter demonstrated its effectiveness with both corpora and evaluation methodologies. However, there were some inflections for the seed words in the final synonyms' list. We expect that enhancing the lemmatization quality will have a positive impact.

Sketch Engine used the collocation sharing concept to extract synonyms. However, it depends on co-occurrence frequencies for the words in the corpus. In contrast, we used it as a filter for the most similar words using cosine similarity from word embeddings. Our approach showed its effectiveness compared to the Sketch Engine approach. However, more investigation is necessary regarding the measure used in collocation extraction to enhance the benefit of this filter.

Table 18: Comparison between Sketch Engine and the SynoExtractor pipeline for synonym extraction.

| | P@1 | P@2 | P@3 | P@4 | P@5 | P@6 | P@7 | P@8 | P@9 | P@10 | MAP | MAR |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SE | 0.266 | 0.166 | 0.177 | 0.166 | 0.146 | 0.144 | 0.108 | 0.108 | 0.096 | 0.086 | 0.343 | 0.097 |
| Baseline | 0.4 | 0.266 | 0.22 | 0.183 | 0.146 | 0.122 | 0.1 | 0.1 | 0.088 | 0.086 | 0.402 | 0.111 |
| SynoExtractor | 0.4 | 0.3 | 0.244 | 0.183 | 0.160 | 0.133 | 0.1 | 0.1 | 0.088 | 0.086 | 0.452 | 0.102 |

The most noticeable error category in the final synonyms' list after filtration was antonyms. Two studies aimed at extracting antonyms in Arabic reported similar problems, finding synonyms in the antonyms' list [37, 38]. In addition, there have been many studies on differentiating between synonyms and antonyms in English [15–17]. However, Aldhubayi [39] showed in her thesis that there is no available methodology for differentiating between synonyms and antonyms based on distributional features because those are similar. She used a pattern-based machine learning classifier, which can be used later as a filter on top of SynoExtractor.

The best results in terms of all measures were for the Gigaword corpus. That was expected because of the corpus size, which was larger than that of KSUCCA. To the best of our knowledge, this is the first study that was investigated using word embeddings for Arabic synonym extraction. We expect that the SynoExtractor pipeline will show similar performance with other words and corpora. It will be useful for many NLP applications that make use of synonyms, such as information retrieval, machine translation, and automatic thesaurus creation.

## 9. Conclusion, Limitations, and Future Work

In this paper, we aimed to develop an unsupervised methodology for synonym extraction. We used a two-phase approach to extract synonyms. In the first phase, we trained Arabic word embeddings using Gigaword and KSUCCA corpora. This was conducted through a large number of experiments to choose the best training hyperparameters for capturing more synonyms in the similar words' clusters. Then, we applied the SynoExtractor pipeline on the most similar words' list to filter synonyms from other relations. We evaluated the extracted words twice. Automatic evaluation using a synonyms' lexicon was constructed from the Alma'any thesaurus, and a manual evaluation was performed by two Arabic linguists. Additionally, we compared our results using the cosine similarity measure with those from the RCS measure for synonym extraction. Finally, we compared our approach to synonym extraction with that of Sketch Engine.

The methodology used for synonym extraction in this paper demonstrated its effectiveness and its potential usefulness for many other NLP fields, such as information retrieval for query expansion and machine translation. Moreover, it can be used to generate synonym thesauri and/or to enhance existing ones such as WordNet. It also can be used for writing tools such as Grammarly, which suggests synonyms to improve writing.

One of the principal limitations of this study is our limited computing power, since we trained a very large number of word embeddings' models. Therefore, we did not investigate training with the use of more epochs despite the fact that training with more epochs has shown to increase effectiveness.

We also expect that training on more data will produce better embeddings that can capture more synonyms. However, we used accessible resources not available as pretrained word embeddings in order to make them public for the research community.

In addition to the fact that automatic extraction demonstrated its effectiveness in facilitating and accelerating relation extraction, it also showed that lexicographers are no longer needed for synonym extraction, though they are still needed for validation.

The results presented in this paper can be extended and improved in a number of ways, including the following:

(i) Using more seed words to generalize our approach
(ii) Exploring the use of contextual word embedding models for synonym extraction, including ELMO[1] or transformer-based models such as BERT
(iii) Using different Arabic stemmers or lemmatizers in order to enhance the reliability of the lemmatization filter
(iv) Investigating the usage of the logDice measure that is used by Sketch Engine for collocation extraction to extract better collocations that can enhance collocation reliability
(v) Adding a pattern-based classifier on top of the SynoExtractor pipeline to remove the antonyms from the first ranked words in the final synonyms' list.
(vi) Training larger corpora can bring noticeable enhancement to synonym extraction.
(vii) Using the SynoExtractor pipeline for English synonym extraction to evaluate it with a different language.

## Data Availability

The code used for the experiments is available at https://github.com/RawanAlmatham/SynoExtractor/blob/main/README.md. The word embedding models for the KSUCCA corpus are available at https://github.com/RawanAlmatham/KSUaravec.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was funded by the Researchers Supporting Project (Grant no. RSP-2020/276) and King Saud University, Riyadh, Saudi Arabia.

## References

[1] G. A. Miller, “WordNet,” *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[2] M. Esposito, E. Damiano, A. Minutolo, G. De Pietro, and H. Fujita, “Hybrid query expansion using lexical resources and word embeddings for sentence retrieval in question answering,” *Information Sciences*, vol. 514, pp. 88–105, 2020.

[3] N. Lin, V. A. Kudinov, H. M. Zaw, and S. Naing, “Query expansion for Myanmar information retrieval used by wordnet,” in *Proceedings of the 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, pp. 395–399, Saint Petersburg, Russia, January 2020.

[4] I. A. Pisarev, “Method for automated thesaurus development in learning process support systems,” in *Proceedings of the 2015 XVIII International Conference on Soft Computing and Measurements (SCM)*, pp. 21–23, St. Petersburg, Russia, May 2015.

[5] J. Yadav and Y. K. Meena, ““Use of fuzzy logic and wordnet for improving performance of extractive automatic text summarization,” in *Proceedings of the 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 2071–2077, Jaipur, September 2016.

[6] S. Mirkin, I. Dagan, and M. Geffet, “Integrating pattern-based and distributional similarity methods for lexical entailment acquisition,” in *Proceedings of the COLING/ACL on Main Conference Poster Sessions*, pp. 579–586Accessed: Oct. 29, 2020. [Online], USA, July 2006.

[7] E. Manishina, B. Jabaian, S. Huet, and F. Lefèvre, “Automatic corpus extension for data-driven natural language generation,” in *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC)*, pp. 3624–3631, Portorož, Slovenia, May 2016.

[8] A. Team, “معجم المعاني المرادفة و المتضادة - مرادف
خار، عكس خار - مرادفات و أضداد اللغة العربية و
الانجليزية في قاموس و معجم المعاني الفوري,” accessed Oct. 29, 2020, https://www.almaany.com/ar/thes/ar-ar/.

[9] J. McCrae and N. Collier, “Synonym set extraction from the biomedical literature by lexical pattern discovery,” *BMC Bioinformatics*, vol. 9, no. 1, p. 159, 2008.

[10] T. Wang and G. Hirst, “Exploring patterns in dictionary definitions for synonym extraction,” *Natural Language Engineering*, vol. 18, no. 3, pp. 313–342, 2012.

[11] T. Yıldız, S. Yıldırım, and B. Diri, “An integrated approach to automatic synonym detection in Turkish corpus,” in *Advances in Natural Language Processing*pp. 116–127, Cham, Switzerland, 2014.

[12] W. Liu, “Automatically refining synonym extraction results: cleaning and ranking,” *Journal of Information Science*, vol. 45, no. 4, pp. 460–472, 2019.

[13] F. Hu, Z. Shao, and T. Ruan, “Self-supervised synonym extraction from the web,” *Journal of Information Science and Engineering*, vol. 31, no. 3, pp. 1133–1148, 2015.

[14] F. T. Asr, R. Zinkov, and M. Jones, ““Querying word embeddings for similarity and relatedness,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, pp. 675–684, New Orleans, LA, USA, June 2018.

[15] M. Ono, M. Miwa, and Y. Sasaki, ““Word embedding-based antonym detection using thesauri and distributional information,” in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 984–989, Denver, CO, USA, 2015.

[16] Z. Dou, W. Wei, and X. Wan, “Improving word embeddings for antonym detection using thesauri and sentiwordnet,” in *Proceedings of the CCF International Conference on Natural Language Processing and Chinese Computing*, pp. 67–79, Zhengzhou, China, October 2018.

[17] K. A. Nguyen, S. S. im Walde, and N. T. Vu, ““Integrating distributional lexical contrast into word embeddings for antonym–synonym distinction,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, p. 454, Berlin, Germany, August 2016.

[18] L. Zhang, J. Li, and C. Wang, “Automatic synonym extraction using Word2Vec and spectral clustering,” in *Proceedings of the Control Conference (CCC), 2017 36th Chinese*, pp. 5629–5632, Dalian, China, July 2017.

[19] A. Leeuwenberg, M. Vela, J. Dehdari, and J. van Genabith, “A minimally supervised approach for synonym extraction with word embeddings,” *The Prague Bulletin of Mathematical Linguistics*, vol. 105, no. 1, pp. 111–142, 2016.

[20] *Cosine Similarity - an Overview | ScienceDirect Topics*, https://www.sciencedirect.com/topics/computer-science/cosine-similarity accessed Oct. 29, 2020.

[21] N. Mohammed, “Extracting word synonyms from text using neural approaches,” *The International Arab Journal of Information Technology*, vol. 17, no. 1, pp. 45–51, 2020.

[22] S. Zheng, J. Xu, P. Zhou, H. Bao, Z. Qi, and B. Xu, “A neural network framework for relation extraction: learning entity semantic and relation pattern,” *Knowledge-Based Systems*, vol. 114, pp. 12–23, 2016.

[23] M. AlMaayah, M. Sawalha, and M. A. M. Abushariah, “Towards an automatic extraction of synonyms for Quranic Arabic WordNet,” *International Journal of Speech Technology*, vol. 19, no. 2, pp. 177–189, 2016.

[24] I. Sarhan, Y. El-Sonbaty, and M. A. El-Nasr, “Arabic relation extraction: a survey,” *International Journal of Computer and Information Technology*, vol. 5, no. 5, 2016.

[25] M. Alrabiah, A. Al-Salman, and E. S. Atwell, “The design and construction of the 50 million words KSUCCA,” in *Proceedings of WACL’2 Second Workshop on Arabic Corpus Linguistics*, pp. 5–8, Lancster University, UK, July 2013.

[26] *Arabic Gigaword Third Edition - Linguistic Data Consortium*, https://catalog.ldc.upenn.edu/LDC2007T40 accessed Dec. 24, 2018.

[27] *Sketch Engine | Language Corpus Management And Query System*, https://www.sketchengine.eu/ accessed Nov. 29, 2018).

[28] *ROGET’s Hyperlinked Thesaurus*, http://www.roget.org/ accessed Nov. 18, 2018).

[29] L. Murphy and M. Lynne, *Semantic Relations and the Lexicon: Antonymy, Synonymy and Other Paradigms*, Cambridge University Press, Cambridge UK, 2003.

[30] M. Kamiński, “Corpus-based extraction of collocations for near-synonym discrimination,” in *Proceedings of the Xvii Euralex International Congress*, pp. 367–374, Tbilisi, Georgia, September 2016.

[31] *Embedding Viewer*, accessed Dec. 19, 2018, https://embeddings.sketchengine.co.uk/.

[32] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” 2013, http://arxiv.org/abs/1301.3781.

[33] G. Bouma, “Normalized (pointwise) mutual information in collocation extraction,” *Proceedings of GSCL*, vol. 30, pp. 31–40, 2009.

[34] Q. Le and T. Mikolov, “Distributed representations of sentences and documents,” in *Proceedings of the International Conference on Machine Learning*, pp. 1188–1196, Washington, WA, USA, June 2014.

[35] M. Sahlgren, “The distributional hypothesis,” *Italian Journal of Disability Studies*, vol. 20, pp. 33–53, 2008.

[36] A. Abdelali, K. Darwish, N. Durrani, and H. Mubarak, “Farasa: a fast and furious segmenter for Arabic,” in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pp. 11–16, San Diego, CA, USA, 2016.

[37] M. Al-Yahya, S. Al-Malak, and L. Aldhubayi, “Ontological lexicon enrichment: the badea system for semi-automated extraction of antonymy relations from Arabic language corpora,” *Malaysian Journal of Computer Science*, vol. 29, no. 1, pp. 56–73, 2016.

[38] M. A. Batita and M. Zrigui, “The enrichment of Arabic wordnet antonym relations,” in *Proceedings of the International Conference on Computational Linguistics and Intelligent Text Processing*, pp. 342–353, Budapest, Hungary, April 2017.

[39] L. B. M. Aldhubayi, *Machine Learning of Antonyms in English and Arabic Corpora*, Phd Thesis, University of Leeds, Leeds, UK, 2019.