Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 20 October 2022 doi:10.20944/preprints202210.0294.v1

Article

Semantic Relationship-Based Embedding Models for Text Classification

Ana Laura Lezama-Sánchez [1,‡], Mireya Tovar Vidal [1,‡*] and José A. Reyes-Ortiz [2,‡*]

[1] Faculty of Computer Science, Benemerita Universidad Autonoma de Puebla, Puebla, Mexico; ana.lezama@alumno.buap.mx, mireya.tovarvidal@viep.com.mx
[2] System Department, Universidad Autonoma Metropolitana, Azcapotzalco, Ciudad de Mexico; jaro@azc.uam.mx
* Correspondence: mireya.tovarvidal@viep.com.mx (T.V.M.), jaro@azc.uam.mx (R.O.J.A.)
‡ These authors contributed equally to this work.

Abstract: Embedding representation models characterize each word as a vector of numbers with a fixed length. These models have been used in tasks involving text classification, such as recommendation and question-answer systems. Semantic relationships are words with a relationship between them providing a complete idea to a text. Therefore, it is hypothesized that an embedding model involving semantic relationships will provide better performance for tasks that use them. This paper presents three embedding models based on semantic relations extracted from Wikipedia to classify texts. The synonym, hyponym, and hyperonym semantic relationships were the ones considered in this work since previous experiments have shown that they are the ones that provide the most semantic knowledge. Lexical-syntactic patterns present in the literature were implemented and subsequently applied to the Wikipedia corpus to obtain the semantic relationships present in it. Several semantic relationships are used in different models: synonymy, hyponym-hyperonym, and a combination of the first two. A convolutional neural network was trained for text classification to evaluate the performance of each model. The results obtained were evaluated with the metrics of precision, accuracy, recall, and F_1-measure. The best values obtained with the second model were accuracy of 0.79 for the *20-Newsgroup* corpus. F_1-measure and recall of 0.87 respectively for the *Reuters* corpus.

Keywords: Deep Learning; Embedding models; Semantic Relationships; Lexical Syntactic Patterns; Convolucional Neural Networks

1. Introduction

The text classification as essays, research reports, medical diagnoses, opinions of a product or event, or news begins when a computer system needs to provide a user with the information he requests quickly and accurately [1]. A system that works with large amounts of documents requires appropriate methods or algorithms for the computer to understand and generate the desired results [2].

The study of the meaning of words and how they are related is a task of Natural Language Processing (*NLP*). The *NLP* has four levels of human language study, one of them is the semantic level. The objective is to discover associations between words that will allow us to define the implicit meaning of each sentence word by word and are used in the same context to give a complete and coherent idea. The associations between the meaning of each word are known as semantic relationships. The most used semantic relationships are synonymy, hyponymy, meronymy, and holonymy [2]. Semantic relations of the synonymy are those where there is a relation between two or more words that have the same or almost the same meaning [3]. Hyponymy is a relationship that includes the semantics of one term in another. Hyperonymy is the inverse relation to hyponymy. Therefore, hyperonymy is the relation of a term that encompasses others semantically [4]. Some existing methods in

 doi:10.20944/preprints202210.0294.v1

the literature for extracting synonymy are the extraction of keyphrases where the relevant words of each document are extracted. Then the relationship around them is identified [3].

On the other hand, the literature also uses Convolutional Neural Networks (CNN) that are trained with characteristics of the existing relationships between extracted keyphrases [3]. Nevertheless, lexical-syntactic patterns are generalized linguistic structures or schemes validated by humans that indicate semantic relationships between concepts. The patterns can be applied to identify formalized concepts and semantic relationships in natural language texts [4]. Some methods are capable of extracting hyponym-hyperonym semantic relationships from a text. The dictionary-based method is based on the use of lexical ontologies such as Wordnet [4]. Clustering methods are incorporated to extract this kind of relationship under the premise that similar words share similar contexts [4]. As in synonymy relationships, there are lexical-syntactic patterns validated by experts. Their function will be to strictly extract pairs of words where there is a hyponym-hyperonym relationship [4]. In [5], they use the relations contained in the WordNet lexical database, which has more than 120,000 related concepts. The existing semantic relationships are more than 25 between more than 155,000 words or lemmas, categorized as nouns, verbs, adjectives, and adverbs [6].

So [5] generated a relationship embedding model based on matrix factorization by extracting existing relationships from the WordNet lexical database. An embedding model is a valuable word representation capable of capturing lexical semantics and trained with natural language corpora. These are an improvement over traditional encodings like bag-of-words or the heavyweight *tf-idf*. In recent years they have been included in the use of algorithms developed in *NLP* [7]. They are reported in the literature as an essential tool in *NLP* tasks such as part-of-speech tagging, chunking, named entity recognition, semantic role tagging, and [7] parsing.

Natural language processing is responsible for generating algorithms so that a computer understands the task it has to perform, imitating human capacity. Some of the more popular embedding models are *word2Vec* [10], *Glove* [12], and *fastText* [11]. The concept of embedding or word embedding model came to fruition in 2013 when Tomas Mikolov and his team at Google developed the embedding model they named *word2vec*. The model has the sub-models continuous bag of words (*CBOW* [8]) and skip-gram [9]. *CBOW* receives a context, and predicts a target word [8]. On the other hand *skipgram* [9], where each word is represented as a bag of n-grams of [12] characters. The *Glove* embedding model was developed in 2014 by Jeffrey Pennington [10]. The *Glove* model combines the advantages of the two main family models in the literature: global matrix factorization and local context window methods. The model works with the non-zero elements in a word-word co-occurrence matrix rather than the entire sparse matrix or separate context windows in a large [10] corpus. However, in 2015 Facebook researchers created the embedding model called *fastText*, which has pre-trained models for 294 languages. The authors relied on the *skipgram* [11] model. The classification algorithms use word embedding models such as *Glove* or *fastText*, intending to improve the accuracy of the NLP algorithms.

The advancement of technology has made it possible to speed up processes, for example: searching for a specific document, generating a summary, and extracting keyphrases from a text. However, computational processes need to model knowledge to generate an accurate result as the human being would do [2].

Text classification is a task carried out by a neural network or an algorithm such as decision trees or nearest neighbors so that large amounts of unordered documents are ordered into classes according to the characteristics of each one [2].

Convolutional Neural Networks *(CNN)* have been adopted for text classification tasks, generating successful results. A CNN is a multilayer or hierarchical network. CNN is built by stacking multiple layers of features. One layer is made up of K linear filters, and an activation function [13]. A CNN is distinguished by the fact that the network weights are shared between different neurons in the [13] hidden layers. Each neuron in the network first computes a weighted linear combination of its inputs. This process can be visualized

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 20 October 2022 doi:10.20944/preprints202210.0294.v1

as evaluating a linear filter on the input values [14]. A CNN is the most effective learning to a set of filters. The same set of filters is used on the data set, forcing the network to learn a general encoding or representation of the data. The weights are restricted to be equal across different neurons on the CNN, allowing a better network generalization to perform normalization. What distinguishes a CNN is the presence of a subsampling or pooling layer. The latter allows optimizing the calculation processes to reduce the size of the data in learning new data, allowing recognition of different characteristics [13].

This research aims to develop three relationship embedding models. The creation of embedding models is conditional on the available semantic relations. Therefore, this paper focuses on extracting a corpus in English from Wikipedia with 5,881,000 documents as the first task. Subsequently, a repository of lexical-syntactic patterns was generated to extract synonymy, hyponymy, and hyperonymy from Wikipedia. For the development of the embedding model, the procedure based on matrix factorization proposed by [5] was used. The text classification was carried out to approximate the performance of the models proposed in this work and the one exposed in [5]. A CNN was used for this purpose, obtaining favorable results. The main contributions of this work are a) three models of embedding semantic relationships; b) a comparison of the performance of the embedding model based on semantic relationships present in WordNet and the three models proposed in this paper; c) three models that show that semantic relationships are a valuable resource for automatic text processing. It is observed that the results obtained are variable because each proposed embedding model has diverse semantic information.

The article is organized as follows: in Section 2, the work related to this research is exposed. Section 3 shows the methodology proposed in this research, while the results are presented in Section 8. The conclusions and future work are presented in the section 5. Finally, the references consulted in the development of this work are shown.

2. Related Works

This section presents related works in the same field. Most use word embedding models to mention a few Glove [12], fastText [11] and word2vec [10].

Authors such as [5] proposed developing an embedding model based on the WordNet semantic network. The relationships were taken into a relationship matrix, interpreting each relationship with different weights. Subsequently, they applied matrix factorization that included processes such as Pointwise Mutual Information (*PMI*), *L*2 norm, and Principal Component Analysis (*PCA*). The authors evaluated the performance of the resulting embeddings in a conventional semantic similarity task, obtaining results substantially superior to the performance of word embeddings based on huge data.

In [15] they expose a text classification method that uses the Bag-of-Words representation model with term frequency-inverse document frequency (*tf-idf*) to select the word(s) with the largest sum *tf-idf* as the most representative with similar signification. Also, the Glove word embedding model finds words with similar semantic meanings. The results were compared with methods such as Principal Component Analysis (*PCA*), Linear Discriminant Analysis (*LDA*), Latent Semantic Indexing (*LSI*), a hybrid approach based on *PCA+LDA* with the Naïve Bayes classifier. The data sets were BBC, Classic, and *20-Newsgroup*. The final results showed that the proposed algorithm provided better classification than the dimension reduction techniques. The authors defined a new metric to evaluate the classifier's performance on reduced features.

Random Multimodel Deep Learning (RMDL) for image, video, symbol, and text classification is proposed by [16]. RMDL aims to find a deep learning structure and architecture by improving robustness and accuracy. The data sets used were MNIST, CIFAR-10, WOS, IMDB, *Reuters* and *20-Newsgroup*. The text classification techniques used as a reference to evaluate the proposed model are Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN), and Deep Neural Networks (DNN). In addition, they incorporate the techniques of Support Vector Machine (SVM), Naïve Bayes Classification (NBC), and, finally, Hierarchical Deep Learning for Text Classification (HDLTex). Feature extraction from

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 20 October 2022 doi:10.20944/preprints202210.0294.v1

texts was performed with the Glove and word2vec embedding models. The evaluation metrics used were precision, recall, and F_1-measure.

The authors [17] expose an improved model based on Graph Neural Network (GNN) for document classification. The model builds different graphs for each text it receives and then classifies them, reducing memory consumption in a neural network. The data sets were from the *Reuters* and *20-Newsgroup*. The Glove embedding model was used with a Convolutional Neural Network and Long Short-Term Memory (LSTM). The metric used for model evaluation is accuracy. The results showed that the proposed model achieves higher accuracy levels than the existing literature models.

In [18], a study that compares the accuracy levels of the word2Vec, Glove, and fastText embedding models in text classification using a Convolutional Neural Network is carried out. The data sets used in the experiments comprised the UCI KDD file, which contains 19,977 news items and is grouped into 20 topics. The results showed that fastText performed better in the classification task. However, when comparing the results of Glove and word2Vec with those provided by fastText, the difference in accuracy is not crucially significant, so the authors conclude that their use depends on the data set used. The metric for the evaluation of the proposed model was accuracy.

In [19], a generative probabilistic model for text documents is exposed. The model combines word and knowledge graph embeddings to encode semantic information and related knowledge in a low-dimensional representation. The model encodes each document as points in the von Mises-Fisher distribution. The authors developed a variational Bayesian inference algorithm to learn unsupervised text embeddings. The results showed that the model is applied for text categorization and sentiment analysis. The data sets used were *Obsumed, 20-Newsgroup* and *Reuters*. The evaluation metrics used were $precision$, $recall$, $accuracy$, and F_1-measure.

The authors [20] present an approach to the problem of classifying texts from sets with few data and sets with data of different lengths. The proposed approach represents texts of any length with 138 features in a fixed-size linguistic vector. The authors addressed two classification tasks: text genres with or without adult content and sentiment analysis. The classification models used were Random Forests, RNN with BiLSTM layer, and the word2vec and BERT models. The evaluation metric used was accuracy.

In [21], the authors compare different strategies for aggregating contextualized word embeddings along lexical, syntactic, or grammatical dimensions. The purpose is to perform semantic retrieval for various natural language processing tasks. The authors defined a set of strategies for aggregating word embeddings along linguistic dimensions. The representations were applied to address tasks such as part-of-speech labeling, identifying relations and semantic frame induction, sequence and word-level labeling, named entity recognition, and word sense disambiguation. The experiments use the word2vec, ROBERTA embedding models, and the nearest neighbor classifier. The evaluation metric used was F_1-measure. The datasets used were those provided by Semeval 2007,2010,2018, CoNLL, SensEval, and TwitterAirline.

In [22], a methodology is presented for sentiment analysis with hybrid embeddings to improve the available pre-trained embedding functions. The authors applied Part of Speech (POS) tagging and the word2position vector over fastText to develop the hybrid embeddings. The metric used in the evaluation process was the accuracy with different deep learning ensemble models and standard sentiment datasets. The data set used was a movie review (MVR). The embedding models used were word2Vec, fastText, and Glove. The results demonstrate that the proposed methodology is effective for sentiment analysis and can incorporate techniques based on linguistic knowledge to improve the results further.

A text classification model with convolutional neural networks such as Graphical Neural Network (GCN) and Bidirectional Recursive Unit (Bi-GRU) is exposed in [23]. The model was designed to address the lack of ability of neural networks to capture contextual semantics. Furthermore, it extracts complex non-linear spatial features and semantic relationships. The word2vec embedding model is used during the experiments. The evaluation metrics were precision, recall, and F_1-measure. The dataset used in the experiments is THUCNews. The authors report that the proposed model can relate better to the context. Furthermore, by extracting information on spatial features and complex non-linear semantic relationships from the text, the model outperforms other models in terms of accuracy, recall, and F_1-measure.

Knowledge graphs as an additional modality for text classification is explored in [24]. Additionally, they explore the inclusion of domain-specific knowledge to deal with domain changes. The authors proved that combining textual embeddings and knowledge graphs achieved good results when applied to a BiLSTM network. The evaluation metrics used were precision, recall, and F_1-measure.

The authors in [25] present a study on the text classification task, investigating methods to augment the input to Deep Neural Networks (DNN) with semantic information. Word semantics are extracted from the Wordnet lexical database. A vector of semantic frequencies is formed using the weighted concept terms extracted from WordNet. They selected the concepts through various semantic disambiguation techniques, including a basic projection method, a POS-based method, and a semantic embedding method. In addition, they incorporated a weight propagation mechanism that exploits semantic relations and conveys a propagation activation component. The authors incorporated for semantic enrichment the word embedding word2vec, fastText, and Glove with the proposed semantic vector using concatenation or replacement, and the result was the input of a DNN classifier. The datasets used during the experiments were *20-Newsgroup* and *Reuters*. The evaluation metrics used for evaluation were F_1-measure and macro-F_1. Experimental results showed that the authors' proposed study increased classification performance.

The authors in [26] propose an investigation on applying a 3-layer CNN model in short and long text classification problems through experimentation and analysis. The model is trained using a word embedding model such as fastText. The datasets used are Ag News, Amazon Full and Polarity, Yahoo Question Answer, Yelp Full, and Polarity. In addition, they applied a pre-processing process to each dataset to remove missing, inconsistent and redundant values. Subsequently, each corpus was tokenized and converted into word vectors. The maximum sequence of a sentence was set to the maximum length of text in the dataset. The authors also applied classifiers such as random forest, logistic regression, extra tree classifier, gradient boosting machine, and stochastic gradient descent. The performance of each classifier was compared with that obtained with the model proposed by the authors. The results obtained showed that the proposed model outperforms traditional classifiers. The evaluation metrics used are precision, recall, accuracy, and F_1-measure.

In this paper, we propose to generate three word embedding models. The models will be based on matrix factorization proposed by [5]. In contrast to [5] the models proposed in this work will be formed by relations extracted with lexical, syntactic patterns from an English Wikipedia corpus. The only additional pre-processing applied over the corpus is to remove non-ASCII characters and convert them to lowercase. To evaluate the performance of the proposed models with the one provided by [5], classification of the corpus *20-Newsgroup* and *Reuters* will be carried out with a Convolutional Neural Network. The results evaluate the proposed models based on *precision*, *accuracy*, *recall* and F_1-measure metrics.

3. Proposed Method

This section presents the proposed method, which include the following steps: the extraction of semantic relationships from Wikipedia; the construction of embedding models based on semantic relationships; and, text classification with a Convolutional Neural Network (CNN).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 20 October 2022 doi:10.20944/preprints202210.0294.v1

3.1. Semantic relationships extraction from Wikipedia

This task is carried out as follows: Semantic relations are extracted from Wikipedia [29] in English. However, Wikipedia is a corpus that lacks labeled datasets with semantic relationships between concepts. Therefore lexical-syntactic patterns are proposed to extract word pairs and semantic relations between them from Wikipedia. The patterns were converted to regular expressions in the Python programming language. A previous preprocessing was applied to Wikipedia, including removing non *ascii* characters and converting them to lowercase. The implemented patterns are used to identify semantic relationships (synonymy, hyponymy, and hyperonymy) from Wikipedia.

The patterns were analyzed by each semantic relationship from the literature. In this way, pattern sets were obtained for synonymy from [3,27,27] and for hyponymy-hyperonymy from [4,28,30,31]. As an example, some obtained patterns applied in this paper are shown in Table 1 and Table 2.

Table 1. Lexical-syntactic patterns to extract synonymy relationships

Concept 1	Relation	Concept 2
X	*also called*	*Y*
X	*called as*	*Y*
X	*also known as*	*Y*
X	*usually called*	*Y*
X	*is called*	*Y*
X	*are called*	*Y*
X	*sometimes called*	*Y*
X	*know as*	*Y*
X	*also referred to as*	*Y*
X	*often described*	*Y*
X	*commonly known as*	*Y*
X	*also named as*	*Y*
X	*abbreviated as*	*Y*
X	*commonly called as*	*Y*
X	*is often referred to as*	*Y*
X	*is referred to as*	*Y*
X	*alias*	*Y*
X	*aka*	*Y*
X	*as known as*	*Y*
X	*frequently abbreviated as*	*Y*
X	*called as*	*Y*
X	*commonly known as*	*Y*
X	*anciently named as*	*Y*

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 20 October 2022 doi:10.20944/preprints202210.0294.v1

Table 2. Lexical-syntactic patterns to extract hyponymy and hyperonymy relationships

Concept 1	Relation	Concept 2
X	*such as*	Y
X	*include*	Y
X	*especially*	Y
X	*is/are*	Y
X	*is one of the*	Y
X	*like other*	Y
X	*usually*	Y
X	*one of these*	Y
X	*one of those*	Y
X	*be example of*	Y
X	*for example*	Y
X	*which be call*	Y
X	*which be name*	Y
X	*mainly*	Y
X	*mostly*	Y
X	*notably*	Y
X	*particularly*	Y
X	*principally*	Y
X	*in particular*	Y
X	*is a/and/the*	Y
X	*other than*	Y
X	*is the single*	Y
X	*including or/and*	Y
X	*except*	Y
X	*called*	Y
X	*including*	Y
X	*another*	Y
X	*called*	Y
X	*i.e.*	Y

The patterns are applied to Wikipedia texts to obtain sets of word pairs for each semantic relationship. Table 3 shows word pairs with the semantic relationship of hyponymy and hyperonymy. The words involved in semantic relationships are enumerated. In Table 3 column *id:Word* values 0 to 13 are assigned for each word. For example, in the relationships elephant-mammal, and cat-animal, the word elephant has the identifier 0; mammal 1, cat 2, and animal with the identifier 3. The id:Word column exposes the identifiers of the relationships without repetition. The assigned identifiers are used to fill a matrix as the representation model.

Table 3. Example of semantic relationship identifiers.

Word1	Word2	id:Word
0:elefante	1:mamífero	0:Elefante
2:gato	3:animal	1:Mamífero
4:dedos	5:mano	2:Gato
6:perro	7:animal	3:Animal
8:gato	9:mamífero	4:Dedos
10:brazo	11:mano	5:Mano
12:jirafa	13:animal	6:perro
		7:Brazo
		8:Jirafa

The number of relationships for synonym, hyponym-hyperonym extracted from Wikipedia are shown in Table 4.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 20 October 2022 doi:10.20944/preprints202210.0294.v1

Table 4. Semantic relationships extracted

Relationship	**Total**
Synonym	1,200,000
Hyponym-hyperonym	6,966,042

The sets of word pairs for each semantic relationship are used to represent them into embedding models.

3.2. Construction of Semantic Relationship-based Embedding models

The embedding models are based on the identifiers assigned in id:Word from Table 5. A matrix M is filled by adding a value of 1 to M. In Table 5 Word1 0:elephant and Word2 1:mammal are represented in the matrix 3 at position 0,1 add a 1. However Word1 8:cat and Word2 9:mammal are represented in position 2,1, because Word2 mammal already has an identifier.

Table 5. Example of filling a relationship matrix M

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
0	0	1	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	1
2	0	1	0	1	0	0	0	0	0
3	0	0	1	0	0	0	1	0	1
4	0	0	0	0	0	1	0	1	0
5	0	0	0	1	0	0	0	0	0
6	0	1	0	0	0	1	0	0	0
7	0	0	0	1	1	0	0	0	0
8	0	0	0	1	0	0	0	0	0

Three embedding models with semantic relationships are developed from matrix M. However, the models include the most frequent relationships from the vocabulary. It was achieved by weighing the type *tf-idf* and selecting the 40,000 relationships with the most frequent words.

For the first model, the semantic relationships extract $synonym_1$ and $synonym_2$ relationships. The synonym of both synonyms is of interest, adding the relation $synonym_2$ and $synonym_1$. Therefore, in the relationship matrix M, a one is assigned to represent the relation $synonym_1$ and $synonym_2$ and $synonym_2$ and $synonym_1$.

The second model represents the hyponymy and hyperonymy relationships and also represents the hyperonymy and hyponymy relationships at the same time.

Given the semantic contribution that synonymy, hyponymy, and hyperonymy generate, it is proposed to generate a model with the three semantic relations in a single model. A one is assigned in the M matrix for the three relationships. Therefore, the relationship matrix M is assigned to "1" value represent $synonym_1$ and $synonym_2$, $synonym_2$ and $synonym_1$, hyponymy and hyperonymy, and hyperonymy and hyponymy, respectively. The number of relationships used in this model was only 50% of those used in the model that only includes synonyms and 50% of those used in the model that only includes hyponymy and hyperonymy.

For each model, the M relationship matrix will be generated, i.e., the semantic relationships are represented with a 1. Subsequently, the following procedure is applied:

1. Enrichment of M to represent the strength of the semantic affinity of identified relations or nodes that are not directly connected by an edge, using the equation:

$$M_G = (I - \alpha M) \tag{1}$$

 doi:10.20944/preprints202210.0294.v1

Where

(a) I is the identity matrix.
(b) M is the array where each $M_{i,j}$ counts the number of paths of length n between nodes i and j.
(c) α decay factor that determines how shorter paths dominate.

2. M_G is subjected to the Pointwise Mutual Information (PMI) to reduce the possible bias introduced by the conversion to words with more senses.
3. For a correct conversion application: Each line in M_G is normalized using the $L2$ norm to correspond to a vector whose scores sum to 1, corresponding to a transition matrix.
4. The M_G matrix is transformed using Principal Component Analysis (PCA) to reduce the vectors' size and set the dimension of the encoded semantic space to 300.

3.3. Text classification using CNN

The text classification task is carried out with the three embedding models described above. The datasets *20-Newsgroup* and *Reuters* exposed in 4.1 are used to evaluate their performance. In addition, also the model proposed in [5], fasText and Glove are implemented to compare them.

The *20-Newsgroup* and *Reuters* sets are preprocessing prior to use in conjunction with embedding vectors in the Convolutional Neural Network. It includes the following steps:

1. Remove html tags
2. Remove punctuation symbols
3. Remove stop words
4. Convert to lowercase
5. Remove extra whitespace

The neural network used is composed of an input layer, an intermediate layer and an output layer. The middle layer is composed of:

1. Embedding layer: Embedding layer to incorporate a pre-trained embedding model.
2. Cov1D layer: Creates a kernel that convolves with the input of the layer over a single dimension to produce an output tensor
3. MaxPooling1D layer: Downsamples the input representation by taking the maximum value over a spatial window of size n.
4. Concatenate layer: takes a list of tensors as input, and returns a single tensor
5. Dropout layer: Prevents overfitting by giving each neuron a 50% probability of not activating during the training phase.
6. Flatten layer: Transforms the shape of the input to a one-dimensional vector. Dense layer: Fully connected layer with an output dimensionality of 512 and ReLu activation function.

Subsequently, the classification performance was evaluated with precision, accuracy, recall, and F_1-measure metrics.

4. Results and Discussion

The results obtained provided a view of the three proposed models. Based on them, it can be seen that they still do not outperform the Glove or fastText models. However, they are capable of outperforming models based on WordNet in some models and bodies. The following sections present the results obtained and evaluated with the metrics $precision, recall, accuracy$ and F_1-measure, as well as the data sets used in this work.

4.1. Dataset

An English corpus from Wikipedia was used to extract semantic relationships (synonymy, hyponymy, and hyperonymy). The extraction was performed using a repository of lexical-syntactic patterns previously taken from the literature for the three semantic relationships. Each pattern was converted to a regular expression. The extracted semantic

 doi:10.20944/preprints202210.0294.v1

relationships are what will form the embedding models. Table 6 exposes the number of documents and tokens of the Wikipedia corpora for the extraction of semantic relationships as well as *Reuters*[1] and *20-Newsgroup*[2] for the classification task.

Table 6. Dataset.

Corpus	Documents	Tokens
Wikipedia	5,881,000	3,380,578,354
20-Newsgroup	20,000	1,800,385
Reuters	18,456	3,435,808

Table 7 exposes the embedding models used in this work. The *Glove* and *fastText* models are the most popular in the literature and have been trained on large corpora. On the other hand, a model based on WordNet with 60,000 tokens used is exposed. The models proposed in this work are also exposed: synonymy and hyponymy-hyperonymy; and a combination of both. As can be seen, the relationships that form these three models contain fewer relationships than those shown in Table 4. The computer equipment used during the experiments has a memory supporting a low number of tokens.

Table 7. Embedding models.

Embedding models	Data	Vector size
Glove	6 billion tokens and have representations for 400 thousand words	300
fastText	1 million word vectors and 16 billion tokens	300
WordNet	60 thousand tokens	300
Synonyms	40,000 tokens	300
Hyponym-Hyperonym	40,000 tokens	300
Combination	40,000 tokens	300

4.2. Experimental Results

The experimental results obtained with the implementation of the proposed procedure are presented below. The results showed that the proposed relationship embedding model obtains better results than those proposed with relationships extracted from WordNet [5].

Table 8 shows the results obtained by classifying the corpora *20-Newsgroup* and *Reuters*. The precision metric is identified by the tag P, recall by R, accuracy by A, and F_1measure by the tag F. It is observed that the results obtained when applying the WordNet-based relationship embedding model do not exceed the results obtained with the *Glove* and *fastText* models.

The results for the corpus *20-Newsgroup* exceed the results obtained with *fastText* with a recall of 0.78 and an accuracy of 0.79 for the model that involves three relationships.

In addition, it outperforms WordNet, obtaining results of 0.75, 0.78, and 0.79 for the precision, recall, and accuracy metrics, respectively. The results when classifying the corpus *Reuters* outperforms *Glove* and *fastText* with an F_1 of 0.70 and a recall of 0.74 and only *Glove* with an accuracy of 0.84 for the model incorporating synonyms. For the same corpus, a performance of 0.80 is obtained for the precision metric and 0.87 for the recall and F_1-score metrics with the model incorporating three semantic relationships, improving WordNet.

In addition, the model that incorporates synonymy relationships obtains an accuracy of 0.84 in the classification of the corpus *Reuters* versus an accuracy of 0.68 reported by the WordNet model. It is estimated that the results exceeded those obtained with WordNet

1 https://trec.nist.gov/data/reuters/reuters.html
2 http://qwone.com/~jason/20Newsgroups/

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 20 October 2022 doi:10.20944/preprints202210.0294.v1

because the relationships included in each proposed model were the most frequent in the total number of relationships obtained.

In some cases the exposed models outperformed *Glove* and *fastText*. However, these results are still shallow, so it is expected that including a greater number of semantic relationships in each model will exceed both the model exposed by [5] but also *Glove* and *fastText*.

Table 8. Results obtained with the CNN and the proposed models

Embedding model	*20-Newsgroup*				*Reuters*			
	P	R	A	F_1	P	R	A	F_1
fastText	0.76	0.74	0.75	0.75	0.72	0.71	0.71	0.71
Glove	0.79	0.79	0.79	0.79	0.72	0.66	0.66	0.67
WordNet	0.66	0.64	0.64	0.64	0.71	0.68	0.68	0.68
Hyponym-hyperonym	0.75	0.78	0.79	0.66	0.72	0.67	0.67	0.68
Synonyms	0.66	0.64	0.64	0.64	0.70	0.74	0.84	0.70
Combination	0.67	0.59	0.59	0.60	0.80	0.87	0.77	0.87

5. Conclusions and future work

This paper has presented three embedding models based on semantic relationships for text classification with a Convolutional Neural Network.

Semantic relationships were extracted from Wikipedia using lexical-syntactic patterns. The models presented incorporate the semantic relationships of synonymy, and hyponymy-hyperonymy, combining them. Also, $synonym_1 - synonym_2$ and $synonym_2 - synonym_1$ are included. On the other hand, the inverse of the hyponym-hyperonym is also included. It generates three relationship embedding models: synonyms, and hyponyms-hyperonyms, and the three relationships.

On the other hand, the behavior of each model presented is evaluated through text classification. In addition, its performance is compared with the results obtained when evaluating the performance of the WordNet relationship embedding model proposed by [5]. The results showed that the proposed models outperform those obtained by the model based on the semantic relationships present in WordNet. The main contributions of this work are three models of embedding semantic relationships extracted from Wikipedia in English. The first model represents synonymy, the second for hyponymy-hyperonymy, and the third a combination of such three relationships. Also, a comparison of the embedding models based on semantic relationships is presented.

In this way, results showed the lack of a more significant number of tokens in each model. In addition, three models expose the importance of semantic relationships providing complete ideas in a text. They become a helpful resource in natural language to enrich a sentence.

Therefore, incorporating more semantic relationships will give each exposed model better results in the text classification task.

This work will be helpful for data analysts because semantic embedding models continue to be a tool that improves results when applied in a task that involves the treatment of textual information. It is observed that the results obtained are variable because each proposed embedding model has different semantic information.

As future work, different models of the lexical-syntactic patterns to extract semantic relationships could be incorporated. As well as adding another semantic relationship such as part-whole or causal and semantic roles, it is considered that it will improve the levels of performance obtained. In addition, an investigation addressing Spanish News and Wikipedia in Spanish will be relevant.

Author Contributions:

Funding:

 doi:10.20944/preprints202210.0294.v1

Institutional Review Board Statement:

Informed Consent Statement:

Data Availability Statement:

Acknowledgments:

Conflicts of Interest:

Sample Availability:

Abbreviations

1. K. Kowsari, K. J. Meimandi, M. Heidarysafa, and S. Mendu. Text Classification Algorithms: A Survey. *Journal Abbreviation* **2019**, *doi: 10.3390/info10040150*, 1–68.
2. M. Augusto, C. Vásquez, M. Hugo, V. Huerta, L. Jaime, and P. Quispe . Procesamiento de lenguaje natural. *Journal Abbreviation* **2009**, *vol. 6, no. 2*, 45–54.
3. Fanghuai Hu, Zhiqing Shao and Tong Ruan. Self-Supervised Synonym Extraction from the Web. *Journal of Informatio Science and Enginerring* **2013**, 1–18.
4. Hearst, M. A. Automatic Acquisition of hyponyms from large text corpora. *The 14th International Conference on Computational Linguistics* **1992**, *vol. 2* , 539-545.
5. C. Saedi, A. Branco, J. A. Rodrigues, and J. R. Silva. WordNet Embeddings. *In Proceedings of the third workshop on representation learning for NLP* **2018**, 122–131.
6. George A. Miller. WordNet: A Lexical Database for English. *Int. J. Inf. Manag. Data Insights* **1995**, vol. 38, no. 11, 39–41.
7. F. Almeida and G. Xexéo. Word Embeddings: A Survey. *The 14th International Conference on Computational Linguistics* **2019**, *arXiv preprint arXiv:1901.09069* , 1-10.
8. K. Orkphol and Wu Yang. Word Sense Disambiguation Using Cosine Similarity Collaborates with Word2vec and WordNet *Future Internet* **2019**, *vol. 11, no. 114 doi:10.3390/fi1105011*, 2-16.
9. Z. Zhou, B. Fu, and H. Qiu,Modeling medical texts for distributed representations based on Skip Gram model Modeling Medical Texts for Distributed Representations Based on Skip-Gram Model. In 2017 3rd International Conference on Information Management (ICIM) **2017**, doi: 10.1109/INFOMAN.2017.7950392, 279–283.
10. K. W. Church and Valia Kordoni. Emerging Trends *Natural Language Engineering* **2016**, *vol. 23, no. 1, doi: 10.1017/S1351324916000334.* , 155-162.
11. Athiwaratkun, B., Wilson, A. G., and Anandkumar, A.. Probabilistic fasttext for multi-sense word embeddings. **2019**, *arXiv preprint arXiv:1806.02901* , 1-11.
12. Pennington, J., Socher, R., Manning, C. D. Glove: Global vectors for word representation. *In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, **2014**, 1532-1543.
13. Antea Herrera Alonso Detección de texto utilizando Redes Neuronales Convolucionales. Tesis de licenciatura, Universitat Politècnica de Catalunya, 2015.
14. Simanovsky, A., and Ulanov, A. Mining text patterns for synonyms extraction.*In 2011 22nd International Workshop on Database and Expert Systems Applications, IEEE* ,473—477.
15. K. N. Singh, S. D. Devi, H. M. Devi, and A. K. Mahanta. A novel approach for dimension reduction using word embedding: An enhanced text classification approach. *Int. J. Inf. Manag. Data Insights* **2022**, vol. 2, no. 1, doi: 10.1016/j.jjimei.2022.100061, 1–10.
16. K. Kowsari, M. Heidarysafa, D. E. Brown, K. J. Meimandi, and L. E. Barnes, RMDL: Random multimodel deep learning for classification.*ACM Int. Conf. Proceeding Ser.*, **2018**,*doi: 10.1145/3206098.3206111*, 19–28.
17. Ai, B., Wang, Y., Ji, L., Yi, J., Wang, T., Liu, W., and Zhou, H. A graph neural network fused with multi-head attention for text classification. *In Journal of Physics: Conference Series*,**2021**,*vol. 2132, no. 1,IOP Publishing*,2–6.
18. E. M. Dharma, F. L. Gaol, H. L. H. S. Warnars, and B. Soewito, The Accuracy Comparison Among Word2Vec, Glove, and Fasttext Towards Convolution Neural Network (CNN) Text Classification. **2022** *J. Theor. Appl. Inf. Technol., vol. 100, no. 2*, 349–359.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 20 October 2022 doi:10.20944/preprints202210.0294.v1

19. H. Ennajari, N. Bouguila, and J. Bentahar. Knowledge-enhanced Spherical Representation Learning for Text Classification. *Proc. 2022 SIAM Int. Conf. Data Mining, SDM 2022*, **2022**, doi: 10.1137/1.9781611977172.72,639–647.
20. Remus, S., Wiedemann, G., Anwar, S., Petersen-Frey, F., Yimam, S. M., and Biemann, C.More Like This: Semantic Retrieval with Linguistic Information *In Proceedings of the 18th Conference on Natural Language Processing (KONVENS 2022)* **2022**,156–166.
21. K. N. Singh, S. D. Devi, H. M. Devi, and A. K. Mahanta. A novel approach for dimension reduction using word embedding: An enhanced text classification approach. *Int. J. Inf. Manag. Data Insights* **2022**, vol. 2, no. 1, doi: 10.1016/j.jjimei.2022.100061, 1–10.
22. Alotaibi, F. S., and Gupta, V. Sentiment analysis system using hybrid word embeddings with convolutional recurrent neural network. *Int. Arab J. Inf. Technol,vol. 19, no. 3*, 330–335.
23. Dong, Y., Yang, Z., and Cao, H. A Text Classification Model Based on GCN and BiGRU Fusion. *In Proceedings of the 8th International Conference on Computing and Artificial Intelligence*, 318–322.
24. Fabian Hoppe. A Improving Zero-Shot Text Classification with Graph-based Knowledge Representations. *In ISWC-DC 2022: Proceedings of the Doctoral Consortium at ISWC 2022; Proceedings of the Doctoral Consortium at ISWC 2022; co-located with 21st International Semantic Web Conference (ISWC 2022); Hangzhou, China* **2022**, vol. 2, no. 1, doi: 10.1016/j.jjimei.2022.100061, 1–7.
25. N. Pittaras, G. Giannakopoulos, G. Papadakis, and V. Karkaletsis. Text classification with semantically enriched word embeddings. *Natural Language Engineering.* **2021**, vol. 27, 10.1017/S1351324920000170 , 391–425.
26. Umer M, Imtiaz Z, Ahmad M, Nappi M, Medaglia C, Sang G. Impact of convolutional neural network and FastText embedding on text classification.. *Int. J. Inf. Manag. Data Insights* **2022**, vol. 2, no. 1, doi: 10.1016/j.jjimei.2022.100061, 1–10.
27. Simanovsky, A., Ulanov, A. Mining text patterns for synonyms extraction. **In 2011 22nd International Workshop on Database and Expert Systems Applications, IEEE**, 473–477).
28. Ai, B., Wang, Y., Ji, L., Yi, J., Wang, T., Liu, W., and Zhou, H. A graph neural network fused with multi-head attention for text classification. *In Journal of Physics: Conference Series IOP Publishing,vol. 2132, no. 1*,1—6.
29. Saorín, T. (18 de septiembre de 2022). Bienvenidos a Wikipedia. Nombre del sitio web. https://es.wikipedia.org/w/index.php?title=Wikipedia:Portadaoldid=123425818
30. Ortega-Mendoza, R. M., Villaseñor-Pineda, L., Montes-y-Gómez, M. Using lexical patterns for extracting hyponyms from the web. (2007) In Mexican International Conference on Artificial Intelligence (pp. 904-911). Springer, Berlin, Heidelberg.
31. Mititelu, V. B. Hyponymy patterns in romanian. Memoirs of the Scientific Sections of the Romanian Academy. (2011) 34, 31-40.