DBnary2Vec: Preliminary Study on Lexical Embeddings for Downstream NLP Tasks

Nakanyseth Vuth, Gilles Serasset

DBnary2Vec: Preliminary Study on Lexical Embeddings for Downstream NLP Tasks

Nakanyseth VUTH and **Gilles SÉRASSET**
Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG
38000 Grenoble
France
first.last@univ-grenoble-alpes.fr

Abstract

In this preliminary study, we experiment with the use of DBnary, a big lexical knowledge graph, to create word embeddings that could be used in NLP downstream tasks. Our gamble is that word embeddings created from lexical data (instead of language corpora) may exhibit less biases while still being usable as the first layer of deep learning approaches to NLP tasks.

We tried very basic method of embedding creation from lexical graph and evaluate (1) the intrinsic performance of the created embeddings on word similarity and word analogy test sets and their extrinsic quality in POS tagging and NER downstream tasks, along with (2) the biases they may exhibit. Such embeddings show promising performances outperforming word2vec on few specific tasks, while still not on par on most others, but we confirm that they exhibit less bias overall.

1 Introduction

Most NLP tasks now use word or sub-word embeddings as their first ingredient. Such embeddings are created based on the proximity of words with others in a corpus. These embeddings have proven to be a valid approach in many practical systems, but they do suffer from biases, leading to research to de-bias through better selection of the training corpus or ad-hoc debiasing techniques on the embeddings themselves.

At the same time, there exist several huge lexical datasets that provide curated information on the words, word forms and senses of different natural languages. With growing size, such datasets are largely disregarded in current deep learning approaches to NLP tasks.

In this paper, we would like to know if training word embeddings from a lexical dataset could be an alternative to corpus based embeddings computation. This work is a preliminary attempt to answer 2 research questions: (1) *is it possible to create embeddings solely from a lexical graph that could be an alternative to corpus based embeddings for downstream tasks?* and (2) *do embeddings learned from lexical graphs suffer from the main biases identified in the corpus-based embeddings literature?*

For this first attempt, we will use the DBnary dataset that we present in section 2. Then, we discuss the evaluation of the adequacy of such embeddings in downstream tasks and of their potential biases in sections 4 and 5. Section 6 presents and discusses the experiments performed to address the research questions at hand.

2 DBnary, a multilingual lexical graph

DBnary (Sérasset, 2015)[1] is a lexical resource extracted from 23 language editions of Wiktionary.[2] This dataset is structured in RDF (Resource Description Framework), a W3C standard for modelling and exchanging metadata about web resources where information is given about resources using triples that consist of subject-predicate-object statements.[3]

DBnary data can be downloaded or queried online using the SPARQL language[4], accessed interactively through a faceted browser[5] or accessed by dereferencing any of the resource URI it defines,[6]

[1] See http://kaiko.getalp.org/about-dbnary/ for the current state of development of DBnary.

[2] See https://www.wiktionary.org/.

[3] See https://www.w3.org/TR/rdf11-primer/ for more details

[4] The *SPARQL Protocol and RDF Query Language* is the "standard query language and protocol for Linked Open Data on the web or for RDF triplestores", quoted from https://www.ontotext.com/knowledgehub/fundamentals/what-is-sparql/. The SPARQL endpoint of DBnary can be accessed at http://kaiko.getalp.org/sparql

[5] The browser can be accessed at http://kaiko.getalp.org/fct/

[6] Each DBnary resource has a URI that can be queried using any web browser or any programmable HTTP client.

making it fully compliant with the guidelines of Linguistic Linked Open Data (LLOD) framework (Declerck et al., 2020).[7]

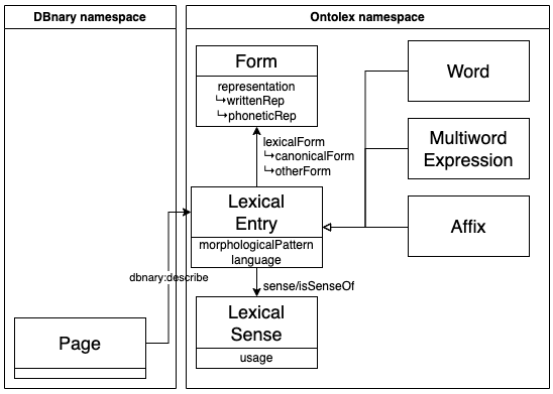

Figure 1: The OntoLex-Lemon core module excerpt (taken from https://www.w3.org/2016/05/ontolex/#core) that is used by DBnary, along with the additional dbnary:Page class that is used to represent a Wiktionary page describing several lexical entries.

The data consists of a huge multilingual graph were nodes (resources) are lexical objects (pages, lexical entries, forms, word senses, etc.), and edges (properties) are structural properties or lexical relations (translation, synonym, antonym, etc.). DBnary uses the core vocabulary of the OntoLex-Lemon model (McCrae et al., 2017) which was developed and which is further extended in the context of the W3C Community Group "Ontology Lexica".[8] As depicted in figure 1, an additional dbnary:Page class has been added to account for the fact that Wiktionary data is organized mainly as a set of pages, where each page describes several lexical entries (possibly in several languages). Other properties and classes are present in the dataset but are not currently used in this work.

The DBnary dataset has steadily grown since its first description (Sérasset, 2012, 2015) and, at the time of writing, contains more than 414M triples describing 6.7M lexical entries in 23 languages.

Figure 2 shows a (simplified) excerpt of the DBnary graph for dbnary:Page "cat". In this preliminary study, we only used the DBnary English subgraph.

E.g. http://kaiko.getalp.org/dbnary/bass_noun_1 represents one of the Lexical Entries described at page *bass* in the English edition of the Wiktionary project.

[7] See also http://www.linguistic-lod.org/.

[8] See https://www.w3.org/community/ontolex/ for more details.

3 Building embeddings from graphs

Current node embedding methods, which create embeddings for nodes in a graph, do not take into account most of the information available in the DBnary graph (namely, typing of the nodes or labelling of the relation). Hence, we have to create graphs suitable for embedding computation from DBnary.

For all our experiments, we use the same general modelling for graphs, but propose two graph topologies.

3.1 Graph Modeling

Formally, we model the graph as follows. Let $G = (V, E)$ denote the graph, where V denotes the set of nodes and E denotes the set of edges. In this graph, each node $x_w \in V$ represents a node in DBnary, such as a page, lexical entry, or word sense. Thus, we have:

$$V = \{x_w : w \in DBnary\} \quad (1)$$

and each edge $e_{u,v} \in E$ represents a relationship between two words u and v of weight $w_{x_u,x_v} \in \mathbb{R}$. The weight reflects the strength or relevance of the relationship between u and v. Graph G can be (un)directed or (un)weighted, depending on the type of graph being modeled.

3.1.1 DBnary topology

The first graph topology, we experiment with, directly uses the relational topology present in DBnary. We extracted all the pages, lexical entries, word sense, and their relations between them from the database and used this information to construct the graph. Each of them is represented as a node in the graph, while each relation between nodes is represented as an edge connecting the corresponding nodes.

Based on the topology, an edge e is formulated as:

$$e_{u,v} = \{(x_u, x_v, w(rel_{x_u,x_v})) : u, v \in V\} \quad (2)$$

For example, let's consider the node x_{cat} in Figure 2, which represents a page in DBnary. It is connected to another page node x_{kitty} through a $synonym_{x_{cat},x_{kitty}}$ relationship. Additionally, it has a $describes$ relationship with its lexical entries, namely $x_{cat\_Adjective\_1}$ and $x_{cat\_Noun\_1}$. Each of these lexical entries is also linked to its corresponding word sense. The weights of the edges are defined based on the relation property. For instance,

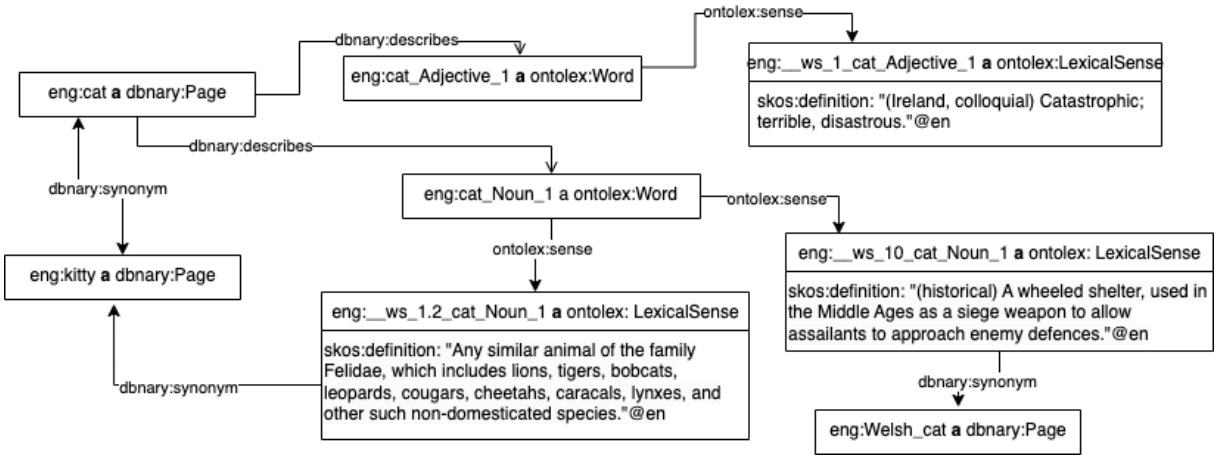

Figure 2: Excerpt of DBnary graph depicting page "cat", along with 2 of its lexical entries and some word senses, with their definition. DBnary graph also contains lexico-semantic relations (synonymy, antonymy, hyponymy...) between pages, lexical entries and/or word senses.

$synonym$ has a higher weight than $antonym$, and so on. This allows us to capture the strength of the relationship between different nodes. Furthermore, we can use the "nyms" relationships (e.g., synonym, antonym, hypernym, hyponym) to establish connections between lexical entries, word senses, and other nodes.

Using the DBnary topology, we construct the graph as a list of edges consisting of two nodes and a weight value based on their relationship. Specifically, an edge between nodes x_u and x_v with a weight of w_{x_u,x_v} is represented as:

$$< x_u >< x_v >< w_{x_u,x_v} > \quad (3)$$

For instance, the relationship between the nodes "cat" and "kitty" with a weight of 10 can be denoted as $< cat >< kitty >< 10 >$ in this format, where cat and $kitty$ correspond to the two nodes and the weight value of 10 indicates the strength of the edge. This format will be used in the graph embedding models, which will be described further in Section 3.2.

3.1.2 Text to Graph

The second graph topology involves utilizing the definitions of each word sense node to create a training corpus and representing the relationship between words in the corpus as edges in the graph. Specifically, we implemented a method that converts sentences into a graph by considering each word as a node and connecting them based on bigrams co-occurrence. The weight of each edge is based on the co-occurrence frequency of the bigram in the entire corpus.

$$w_{(t_i,t_{i+1})} = count\_occur(t_i, t_{i+1}) \quad (4)$$

where t_i and t_{i+1} are the two words in the bigram and $count\_occur$ is a function that returns the number of times the bi-gram appears in the corpus. The resulting edge can be represented as:

$$e = \{(v_{t_i}, v_{t_{i+1}}, w_{(t_i,t_{i+1})}) : t_i, t_{i+1} \in S\} \quad (5)$$

where S is the set of all unique words in the corpus, v_{t_i} and $v_{t_{i+1}}$ are the corresponding nodes in the graph, and $w_{(t_i,t_{i+1})}$ is the weight assigned to the edge between these nodes.

3.2 Embedding methods

In the context of our preliminary studies into graph embedding techniques, we have opted to examine three widely recognized algorithms for producing graph embeddings, namely DeepWalk (Perozzi et al., 2014), LINE (Tang et al., 2015), and node2vec (Grover and Leskovec, 2016). These techniques have been demonstrated to be effective in a variety of applications and have attained state-of-the-art performance in numerous benchmarks. In addition, we have incorporated the prevalent Skip-Gram technique (Mikolov et al., 2013a), word2vec, as a fundamental model for comparative analysis.

3.2.1 SGNS (word2vec)

The Skip-Gram with Negative Sampling is a well-known embedding method that aims to learn a dense, continuous vector representation for each

word in a given corpus. SNGS model predicts the surrounding context words given a center word. It focuses on maximizing probabilities of context words given a specific center word, which can be written as

$$P(w_{i-c}, w_{i-c+1}, .., w_{i-1}, w_{i+1}, .., w_{i+c-1}, w_{i+c}|w_i) \quad (6)$$

3.2.2 DeepWalk

DeepWalk is an unsupervised learning method for generating node embeddings by utilizing random walks on the graph. The objective of DeepWalk is to learn a representation for each node in the graph, which captures its structural context in the graph. The method starts by generating random walks on the graph, where each walk starts from a randomly selected node and traverses the graph by following its edges. The walks are then treated as sentences, and the Skip-gram model from word2vec is used to learn node embeddings by predicting the context nodes for each target node in the walk.

3.2.3 LINE

LINE on the other hand, aims to learn node embeddings by considering the global structure of the graph. The method uses a first-order proximity and a second-order proximity objective to capture the local and global structure of the graph, respectively. The first-order proximity objective is to maximize the probability of observing a context node given a target node in a random walk, similar to DeepWalk. The second-order proximity objective, on the other hand, is to maximize the probability of observing a node u being the second-order neighbor of node v.

3.2.4 node2vec

node2vec is another method for learning node embeddings by utilizing random walks on the graph. Similar to DeepWalk, the objective of node2vec is to learn a representation for each node in the graph that captures its structural context in the graph. node2vec improves upon DeepWalk by introducing a biased random walk strategy that allows for the generation of walks that balance the exploration and exploitation of the graph structure which in turn leads to representations obeying a spectrum of equivalences from homophily to structural equivalence. Specifically, node2vec uses a two-parameter family of random walks, where the parameters control the trade-off between depth-first and breadth-first search. It uses second-order biased random walks to generate sequences of nodes or "sentences" from a given graph. Once the sequences of nodes are generated, they are used as input to the SGNS model to learn embeddings for nodes.

4 Evaluating embeddings

As outlined in (Bakarov, 2018), the field of word embedding evaluation has developed two primary classes of methods for assessing the quality of embedding models: intrinsic and extrinsic evaluators. Intrinsic evaluators assess the quality of embedding models through specific tasks that are independent of downstream NLP applications. Extrinsic evaluators, on the other hand, use the vector representations of the embedding models in downstream NLP tasks, such as part-of-speech tagging and named entity recognition. These evaluations measure the effectiveness of embedding models in improving the performance of NLP tasks. It is important to note that both intrinsic and extrinsic evaluations have their limitations. Intrinsic evaluations may not necessarily correlate with the performance of embedding models in real-world NLP applications, while extrinsic evaluations may be affected by other factors such as the quality of the downstream NLP task. Therefore, it is better to use both intrinsic and extrinsic evaluations to get a comprehensive understanding of the quality of embedding models.

4.1 Intrinsic evaluator

Intrinsic evaluation is a method for assessing the quality of word embeddings by testing their ability to capture certain linguistic properties and relationships. The primary objective of intrinsic evaluation is to determine how well an embedding model captures semantic and syntactic information. This approach involves assessing the embedding quality through specific tasks that are independent of downstream NLP applications. Two commonly used intrinsic evaluation methods are word similarity and word analogy tasks. Intrinsic evaluation is an important step in assessing the quality of word embeddings, as it provides insight into the model's ability to capture linguistic properties and relationships.

4.1.1 Word similarity

Word similarity tasks are designed to measure the degree of similarity between pairs of words. These tasks typically involve a list of word pairs along with human judgments of the degree of similarity between the pairs. The model's performance

is evaluated based on its ability to produce similarity scores that match human judgments using cosine similarity. It measures the cosine of the angle between the two vectors and ranges from -1 to 1, where 1 represents identical vectors, 0 represents independent orthogonal vectors, and -1 represents opposite vectors. The cosine similarity between vectors a and b is calculated as follows:

$$cos(w_a, w_b) = \frac{w_a \cdot w_b}{\|w_a\|\|w_b\|} \tag{7}$$

where $\cdot$ represents the dot product of two vectors, and $\|w_a\|$ and $\|w_b\|$ denote the Euclidean norms of vectors w_a and w_b, respectively.

4.1.2 Word analogy

Word analogy tasks, on the other hand, assess the model's ability to capture the relationships between words, such as analogies. In these tasks, a set of word pairs is provided, and the model is required to complete an analogy by finding a fourth word that is related to the third word in the same way as the second word is related to the first word. For example, given the pair "man:woman," the model should find the word "queen" when presented with the pair "king:?". This task is calculated using the 3CosAdd method (Mikolov et al., 2013b). Given a pair of words a and a^* and a third word b, the analogy between a and a can be used to determine the word b^* that corresponds to b. It is mathematically expressed as:

$$a : a^* :: b : \_ \tag{8}$$

It solves for b^* using the following formula:

$$b^* = \underset{b'}{argmax}(cos(b', b + a^* - a)) \tag{9}$$

This method normalizes the vector length using cosine similarity. Alternatively, there is a refined method called 3CosMul (Levy and Goldberg, 2014) which is defined as:

$$b^* = \underset{b'}{argmax}\frac{cos(b', b)cos(b', a^*)}{cos(b', a^*) + \epsilon} \tag{10}$$

where ϵ = 0.001 is used for preventing zero division.

4.2 Extrinsic evaluator

Extrinsic evaluators are NLP downstream tasks that directly use embedding models to improve the performance of the task at hand. By using the embeddings as input features for these tasks, we can evaluate the effectiveness of the embedding model in contributing to the downstream task performance. In our preliminary study, we have chosen two specific tasks, Part-of-Speech (POS) tagging and Named Entity Recognition (NER), as extrinsic evaluators for our embedding models.

4.2.1 Part-of-speech tagging

Part-of-Speech (POS) tagging is a fundamental task in NLP that involves the identification of the grammatical category of words in a sentence. The goal of POS tagging is to automatically assign a specific part-of-speech tag (such as noun, verb, adjective, etc.) to each word in a sentence, based on its context and the grammatical rules of the language. POS tagging is an essential preprocessing step for many NLP applications, such as text classification, information retrieval, and machine translation. It is a challenging task, as words often have multiple possible tags, and the same word can have different meanings and functions in different contexts.

4.2.2 Named entity recognition

Named Entity Recognition (NER) is a task in NLP that involves identifying and extracting named entities from unstructured text. Named entities refer to specific objects, people, places, or concepts that have a unique name or identity. The goal of NER is to automatically identify and classify named entities in text, and assign them a pre-defined label such as PERSON, ORGANIZATION, LOCATION, etc. The task is crucial for a wide range of NLP applications, such as information extraction, document retrieval, and machine translation, and it is a challenging task due to the variability and complexity of named entities in text.

5 Biases in embeddings

Word embeddings have proven to be valuable tools for natural language processing tasks, but they are not immune to biases. Biases in embeddings arise from the underlying biases present in the training data, leading to certain groups or concepts being over-represented or under-represented in the embedding space (Garg et al., 2018). These biases can manifest in various forms, including gender, race, ethnicity, religion, and more. Recognizing and addressing these biases is crucial to ensure fairness, equity, and non-discrimination in NLP applications. Studies have highlighted the

presence of biases in word embeddings, revealing how societal biases can seep into the learned representations. For example, Bolukbasi et al. (Bolukbasi et al., 2016) demonstrated the existence of gender bias in word embeddings through the analogy "man:programmer::woman:homemaker", where the embedding model associated men with the profession of programmer and women with the role of homemaker. This finding illustrates how gender biases present in the training data can be reflected in the learned embeddings.

The consequences of biases in embeddings can be far-reaching and detrimental. Biased embeddings can perpetuate and reinforce harmful stereotypes, leading to discriminatory outcomes in downstream NLP applications. For instance, automated hiring systems that utilize biased embeddings may unfairly discriminate against certain demographic groups, resulting in an inequitable hiring process (Dastin, 2022). Search engines that rely on biased embeddings can produce biased search results, reinforcing existing societal biases and limiting access to diverse perspectives and information (Kay et al., 2015; Caliskan et al., 2017). Furthermore, automated hate speech detection models trained on biased corpora can inadvertently exhibit racial bias, potentially amplifying harm inflicted upon marginalized communities (Sap et al., 2019). Because of this, it is essential to gain an understanding of the biases that are present in word embeddings and to work to eliminate them in order to stop the negative effects they have on society.

6 Experiments

The following section presents the experimental setup used to evaluate the embedding models, as well as the evaluation results that highlight both performance and bias along with evaluation datasets used in this study.

6.1 Experimental Setup

We selected four models for our study, comprising three graph embedding models, DeepWalk, LINE, and node2vec, as well as a traditional word embedding model, SGNS. To obtain a comprehensive analysis, we trained the graph embedding models using two approaches described in Section 3.1, resulting in a total of six graph embedding models. We trained text-to-graph based models and SGNS using DBnary definition nodes that contained 945,525 definitions/sentences. Table 1 illustrates the properties of the graph used in the graph embedding models. For the node2vec approach, we used the official implementation[9]. We used Graphvite(Zhu et al., 2019) to train DeepWalk and LINE. Finally, we trained the SGNS model using Gensim (Rehurek and Sojka, 2011) word2vec library. To ensure consistency in our results, we used the same default settings for all the graph embedding models, including walk length $l = 40$, number of walks per node $r = 100$, and ($p = 1$, $q = 1$) specifically for node2vec method, and window size $w = 10$ for SGNS. We chose to use 256 dimensions for all the embedding models in our study.

Graph	# Edges	# Nodes	# Vocab
DBnary topology	2396346	3284911	1120225
Text to graph	1772040	276619	276617

Table 1: Graph's properties

6.1.1 Intrinsic

The embedding models were evaluated intrinsically through word similarity and word analogy tasks. In this study, we have selected a total of eight benchmark datasets for the purpose of evaluating word similarity. These datasets are presented in Table 2. The Google analogy test set (Mikolov et al., 2013a) and the Bigger Analogy test set (BATS) (Gladkova et al., 2016) were selected to serve as the benchmark datasets for the word analogy test. Both of these tasks were evaluated using GluonNLP [10]

6.1.2 Extrinsic

Extrinsic evaluation was performed using two different NLP downstream tasks: 1) part-of-speech tagging, and 2) named-entity recognition. We trained each task with the same architecture, which consisted of running a vanilla RNN on the Keras library (Chollet et al., 2015) for 25 epochs with 64 hidden dimensions, and a batch size of 128. The CoNLL-2000 (Tjong Kim Sang and Buchholz, 2000) from NLTK (Bird et al., 2009) and the CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003) from HuggingFace [11] were used for part-of-speech tagging and named-entity recognition tasks, respectively. The data split for both tasks is presented in Table 3.

[9] https://github.com/eliorc/node2vec

[10] https://nlp.gluon.ai/index.html

[11] https://huggingface.co/datasets/conll2003

Dataset	Pairs
WordSim-353 (Finkelstein et al., 2001)	353
WordSim-353-SIM (Agirre et al., 2009)	203
WordSim-353-REL (Agirre et al., 2009)	252
MEN (Bruni et al., 2014)	3000
RadinskyMTurk-287 (Radinsky et al., 2011)	287
RareWords (Luong et al., 2013)	2034
SimLex-999 (Hill et al., 2014)	999
SimVerb-3500 (Gerz et al., 2016)	3500
YangPowerVerb-130 (Yang and Powers, 2006)	130
SemEval17Task2(Camacho-Collados et al., 2017)	518

Table 2: Word Similarity benchmark datasets. The MEN dataset has been partitioned into a dev set consisting of 2000 pairs and a test set consisting of 1000 pairs. The SemEval17Task2 dataset is divided into two distinct subsets, comprising 18 pairs for the trial set and 500 pairs for the test set.

Table 3: Dataset splits for extrinsic tasks

Dataset	Train	Validation	Test
CoNLL-2000	7909	1396	1643
CoNLL-2003	14041	3250	3453

6.2 Bias experiment

To evaluate the presence of bias in our embedding models, we utilized the code [12] which replicates the paper of (Badilla et al., 2020). Following this paper, we used four metrics to measure biases: 1) the Word Embedding Association Test (WEAT) (Caliskan et al., 2017), 2) the WEAT effect size, 3) the Relative Norm Distance (RND) (Garg et al., 2018), and 4) the Relative Negative Sentiment Bias (RNSB) (Sweeney and Najafian, 2019). Details on the queries utilized in our study can be found in Table 4. Due to the size of the corpus used for training our text-to-graph models and SGNS model, we were only able to measure biases in Gender and Religion, as many of the embeddings for Ethnicity queries were not present in our models.

6.3 Embeddings evaluation and biases

This section presents the evaluation results of our embedding models in terms of their performance using intrinsic and extrinsic evaluators, as well as their biases.

Intrinsic - Word similarity results: We evaluated the performance of all our models on 13 different datasets, and the results are presented in Table 5. Our experimental findings reveal that the node2vec topology-based model outperforms the other models in capturing the similarity and relationship of word pairs, as evidenced by its superior performance in datasets such as SimLex999, SimVerb3500, and YangPowerVerb-130. These datasets were designed to focus more on measuring a range of semantic relationships. On the other hand, the SGNS model generally outperforms all other models in most datasets, except the ones that specifically focus on capturing semantic relationships. However, our node2vec text-to-graph model also shows promising results, coming in second after SGNS and outperforming the node2vec topology-based method in most cases. It is important to note that not all models were able to cover all pairs in the evaluation datasets, as shown by the percentage of out-of-vocabulary pairs in Table 6 word similarity.

Intrinsic - Word analogy results: The results obtained using 3CosAdd and 3CosMul methods for two datasets are presented in Table 7. We observe that the topology-based models perform the worst, with SGNS model achieving the highest scores in both datasets and methods. These findings suggest that while topology-based models may excel at capturing similarity and semantic relationships between word pairs, they do not perform as well in word analogy tasks. This could be attributed to the fact that topology-based models rely heavily on the graph structure, which may not always capture the full extent of the semantic relationships between words. Furthermore, the results also reveal some interesting insights into how the models perform on specific word analogy tasks. For instance, for the pair "man:king::women:?", our model predicted "face-sit" with a score of 0.70, and "queen" with a score of 0.68. This could be explained by the fact that in DBnary, the node "face-sit" shares an edge connection through a synonym relation to one of "queen"'s word senses, which leads to this result. Another example is the pair "Athens:Greece::Bangkok:?", where our model predicted "Krung_Thep" instead of "Thailand". This occurred because in DBnary, "Krung_Thep" is synonymous with "Bangkok" and the node "Bangkok" does not have an edge connecting to the node "Thailand" at all.

[12] https://github.com/dccuchile/wefe/blob/master/examples/WEFE_rankings.ipynb

Table 4: Bias experiment queries

	Target set	Attribute sets
Gender query	{Male terms, Female terms}	{Career, Family}, {Math, Arts}, {Science, Arts}, {Intelligence, Appearance}, {Intelligence, Sensitive},{Pleasant, Unpleasant}, {Positive words, Negative words},{Man Roles, Women Roles}
Religion query	{Christianity terms, Islam terms}	{Pleasant, Unpleasant}, {Conservative, Terrorism}, {Positive words, Negative Words}
	{Christianity terms, Judaism terms}	{Pleasant, Unpleasant}, {Conservative, Greed}, {Positive Words, Negative Words}
	{Islam terms, Judaism terms}	{Pleasant, Unpleasant}, {Terrorism, Greed}, {Positive Words, Negative Words}

Table 5: Word similarity evaluation results

	Word Similarity Datasets												
	WS-all	WS-sim	WS-rel	MEN-full	MEN-dev	MEN-test	MTurk	RW	SimLex	SimVerb	YP	SEval-trail	SEval-test
node2vec	0.3664	**0.6350**	0.1140	0.4284	0.4420	0.4022	0.2717	0.2289	**0.4630**	**0.4269**	**0.6672**	0.4757	0.4062
deepwalk	0.2900	0.4911	0.0955	0.2163	0.2128	0.2240	-0.0132	0.1238	0.2092	0.2378	0.3702	0.1889	0.2267
line	0.2501	0.4566	0.0103	0.2302	0.2325	0.2248	-0.0135	0.1163	0.1922	0.2476	0.3369	0.0918	0.2236
node2vec_t2g	0.5080	0.6174	0.4354	0.5745	0.5703	0.5839	**0.5099**	0.1778	0.2225	0.1393	0.1951	0.7523	0.3986
deepwalk_t2g	0.2877	0.4141	0.2368	0.4433	0.4545	0.4190	0.3322	0.1444	0.2031	0.1878	0.2695	0.6491	0.3390
line_t2g	0.2873	0.4132	0.2378	0.4417	0.4524	0.4180	0.3389	0.1459	0.2070	0.1858	0.2638	0.6347	0.3388
SGNS	**0.5511**	0.6278	**0.4555**	**0.6282**	**0.6283**	**0.6279**	0.4635	**0.3562**	0.3427	0.2661	0.3438	**0.7957**	**0.5268**

Table 6: Word similarity out-of-vocabulary percentage

	WS-all	WS-sim	WS-rel	MEN-full	MEN-dev	MEN-test	MTurk	RW	SimLex	SimVerb	YP	SEval-trail	SEval-test
Topology	0.00%	0.00%	0.00%	0.23%	0.35%	0.00%	18.82%	1.13%	0.00%	0.00%	0.00%	0.00%	5.00%
Text to graph	0.00%	0.00%	0.00%	0.43%	0.50%	0.30%	21.25%	20.94%	0.20%	0.06%	0.00%	0.00%	3.80%
SGNS	0.00%	0.00%	0.00%	0.43%	0.50%	0.30%	21.25%	20.85%	0.20%	0.06%	0.00%	0.00%	3.80%

Extrinsic Evaluation: Our embedding models were evaluated on two extrinsic tasks: part-of-speech tagging and named entity recognition using the F1 score as the performance metric. The experiment was run thrice, and the average F1 score was taken to obtain the final results, which are presented in Table 8. We observe that the text-to-graph based models outperform the topology-based and SGNS models in both tasks, with DeepWalk performing the best in named-entity recognition, and node2vec in part-of-speech tagging. This is an indication that our text-to-graph models have captured more contextual and semantic information and are able to better understand the relationship between words in a sentence.

Bias Evaluation: To evaluate the presence of bias in our experiment, we measured the similarity between the target sets (T1, T2) and attribute sets (A1, A2) for each bias query. For instance, in the case of Gender bias, we used Male Terms and Female Terms as target sets, and Intelligence and Appearance as attribute sets. Our bias evaluation results, presented in Table 9, demonstrate that the DeepWalk topology-based model exhibits the lowest bias in Gender queries, while the node2vec topology-based and SGNS models display the highest bias. Interestingly, for Religion bias, we found that the LINE topology-based model has the least bias, while the SGNS model shows the highest bias, with DeepWalk text-to-graph ranking second. We have also calculated the overall cumulative ranking for each model on both queries, and we present the results in Table 10. Our findings demonstrate that the traditional SGNS embedding method exhibits the most bias compared to the Lexical embedding methods.

Table 7: Word Analogy evaluation results

	Word Analogy Datasets					
	GoogleAnalogyTestSet			BiggerAnalogyTestSet		
	3CosAdd	3CosMul	% OOV pair	3CosAdd	3CosMul	% OOV pair
node2vec	0.0063	0.0073	0.00%	0.0161	0.0157	0.98%
deepwalk	0.0105	0.0092	0.00%	0.0135	0.0106	0.98%
line	0.0097	0.0086	0.00%	0.0131	0.0106	0.98%
node2vec_t2g	0.0578	0.0627	10.55%	0.0418	0.0422	9.65%
deepwalk_t2g	0.0495	0.0483	10.55%	0.0427	0.0373	9.65%
line_t2g	0.0511	0.0497	10.55%	0.0424	0.0378	9.65%
SGNS	**0.1425**	**0.1452**	10.55%	**0.0873**	**0.0851**	9.65%

Table 8: Extrinsic evaluation results

	POS		NER	
	Macro F1	Weighted F1	Macro F1	Weighted F1
node2vec	0.8089	0.8686	0.3729	0.9694
deepwalk	0.7831	0.8356	0.3651	0.9691
line	0.7809	0.8351	0.3520	0.9685
node2vec_t2g	**0.8345**	**0.9141**	0.4782	0.9786
deepwalk_t2g	0.8317	0.9115	**0.5002**	**0.9790**
line_t2g	0.8321	0.9121	0.4996	**0.9790**
SGNS	0.8274	0.9054	0.4767	0.9784

Model	Rank
line	1
deepwalk	2
line_t2g	3
node2vec_t2g	4
node2vec	5
deepwalk_t2g	6
word2vec	7

Table 10: Bias Ranking. Sorting by the best to the worst model.

7 Conclusion and future works

In our preliminary study, we proposed methods to create lexical embeddings for downstream NLP tasks using the DBnary Lexical Database. We conducted comprehensive evaluations and bias analysis of graph-based embeddings and compared them with the traditional SGNS corpus-based embedding model. Our results indicate that graph-based embeddings generated from the relational topology of the lexical graph outperform SGNS embeddings in capturing semantic relationships between words. However, further research is needed to explore methods for assigning edge weights automatically instead of relying on manual assignments. We observed that text-to-graph-based models perform better than topology-based models in most datasets except for those that focus on semantic relationships, where text-to-graph-based models rank second after SGNS. To improve the performance of text-to-graph-based models, better weight assignment methods need to be developed, for instance, using word probability. Moreover, the quality of the DBnary graph needs to be assessed to address missing and irrelevant nodes.

In addition to performance evaluations, we conducted a bias analysis of the embeddings. Our results demonstrated that SGNS embeddings exhibited higher levels of bias compared to lexical graph embeddings. This highlights the importance of considering bias in word embeddings and underlines the potential benefits of using lexical graphs to mitigate bias. However, a more comprehensive study is needed to gain a deeper understanding of the underlying factors contributing to bias, such as the characteristics of the training data and the embedding methods. Future research should also explore debiasing techniques to mitigate biases in the models. Furthermore, as our experiments utilized default parameters, future work will focus on hyperparameter tuning to optimize the performance of the lexical graph embedding models. Additionally, an interesting path for future exploration lies in leveraging the DBnary graph topology to employ Knowledge Graph Embedding methods for computing vector representations. By comparing the performance and characteristics of our baseline methods with a more specialized knowledge graph embedding technique we can gain insights into the advantages and limitations of different approaches.

Beyond improving current results, however, we acknowledge that this experiment is very preliminary and contains many limitations that should be

	Gender				Religion			
	WEAT	WEAT ES	RND	RNSB	WEAT	WEAT ES	RND	RNSB
node2vec	6 (0.117)	7 (0.263)	7 (0.116)	6 (0.061)	2 (0.027)	**1 (0.258)**	4 (0.101)	6 (0.074)
deepwalk	2 (0.057)	5 (0.206)	**1 (0.029)**	**1 (0.019)**	5 (0.043)	5 (0.439)	2 (0.078)	**1 (0.019)**
line	**1 (0.056)**	3 (0.182)	4 (0.049)	2 (0.02)	**1 (0.018)**	3 (0.387)	**1 (0.074)**	3 (0.02)
node2vec_t2g	4 (0.099)	4 (0.2)	5 (0.065)	3 (0.023)	4 (0.041)	4 (0.408)	3 (0.09)	2 (0.019)
deepwalk_t2g	5 (0.103)	2 (0.175)	3 (0.043)	5 (0.042)	6 (0.048)	6 (0.478)	6 (0.112)	5 (0.04)
line_t2g	3 (0.091)	**1 (0.138)**	2 (0.043)	4 (0.039)	3 (0.037)	2 (0.37)	5 (0.112)	4 (0.04)
word2vec	7 (0.181)	6 (0.245)	6 (0.092)	7 (0.16)	7 (0.062)	7 (0.878)	7 (0.3)	7 (0.109)

Table 9: Bias evaluation results for Gender and Religion queries. Lower scores indicate lower bias w.r.t to a metric.

handled if we want to provide alternatives to current first-layer initialization steps in deep learning-based models. We decided for the moment to focus on word embeddings as words represent a token granularity shared with lexical datasets, however, current approaches are now using so-called subwords as tokens bringing better results and handling of out-of-vocabulary terms. In the near future, we will address such approaches using lexical data. Moreover, many tokenizers/embedders are now multilingual, hence we will also experiment with other languages available in DBnary, either in a monolingual setting or in a multilingual setting.

References

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. A Study on Similarity and Relatedness Using Distributional and WordNet-based Approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27, Boulder, Colorado. Association for Computational Linguistics.

Pablo Badilla, Felipe Bravo-Marquez, and Jorge Pérez. 2020. WEFE: The Word Embeddings Fairness Evaluation Framework. In *Twenty-Ninth International Joint Conference on Artificial Intelligence*, volume 1, pages 430–436.

Amir Bakarov. 2018. A Survey of Word Embeddings Evaluation Methods.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 4356–4364, Red Hook, NY, USA. Curran Associates Inc.

E. Bruni, N. K. Tran, and M. Baroni. 2014. Multimodal Distributional Semantics. *Journal of Artificial Intelligence Research*, 49:1–47.

Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186.

Jose Camacho-Collados, Mohammad Taher Pilehvar, Nigel Collier, and Roberto Navigli. 2017. SemEval-2017 Task 2: Multilingual and Cross-lingual Semantic Word Similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 15–26, Vancouver, Canada. Association for Computational Linguistics.

François Chollet et al. 2015. Keras. `https://keras.io`.

Jeffry Dastin. 2022. *Amazon Scraps Secret AI Recruiting Tool that Showed Bias against Women* *.

Thierry Declerck, John Philip McCrae, Matthias Hartung, Jorge Gracia, Christian Chiarcos, Elena Montiel-Ponsoda, Philipp Cimiano, Artem Revenko, Roser Saurí, Deirdre Lee, Stefania Racioppa, Jamal Abdul Nasir, Matthias Orlikowsk, Marta Lanau-Coronas, Christian Fäth, Mariano Rico, Mohammad Fazleh Elahi, Maria Khvalchik, Meritxell Gonzalez, and Katharine Cooney. 2020. Recent developments for the linguistic linked open data infrastructure. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5660–5667, Marseille, France. European Language Resources Association.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: The concept revisited. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 406–414, New York, NY, USA. Association for Computing Machinery.

Nikhil Garg, Londa Schiebinger, Dan Jurafsky, and James Zou. 2018. Word embeddings quantify 100 years of gender and ethnic stereotypes. *Proceedings of the National Academy of Sciences*, 115(16):E3635–E3644.

Daniela Gerz, Ivan Vulić, Felix Hill, Roi Reichart, and Anna Korhonen. 2016. SimVerb-3500: A Large-Scale Evaluation Set of Verb Similarity. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2173–2182, Austin, Texas. Association for Computational Linguistics.

Anna Gladkova, Aleksandr Drozd, and Satoshi Matsuoka. 2016. Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. In *Proceedings of the NAACL Student Research Workshop*, pages 8–15, San Diego, California. Association for Computational Linguistics.

Aditya Grover and Jure Leskovec. 2016. Node2vec: Scalable Feature Learning for Networks.

Felix Hill, Roi Reichart, and Anna Korhonen. 2014. SimLex-999: Evaluating Semantic Models with (Genuine) Similarity Estimation.

Matthew Kay, Cynthia Matuszek, and Sean Munson. 2015. Unequal representation and gender stereotypes in image search results for occupations.

Omer Levy and Yoav Goldberg. 2014. Linguistic regularities in sparse and explicit word representations. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning (CoNLL-2014)*, pages 171–180.

Thang Luong, Richard Socher, and Christopher Manning. 2013. Better Word Representations with Recursive Neural Networks for Morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 104–113, Sofia, Bulgaria. Association for Computational Linguistics.

John P. McCrae, Paul Buitelaar, and Philipp Cimiano. 2017. The OntoLex-Lemon Model: development and applications. In *Proc. of the 5th Biennial Conference on Electronic Lexicography (eLex)*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient Estimation of Word Representations in Vector Space.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: Online Learning of Social Representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 701–710.

Kira Radinsky, Eugene Agichtein, Evgeniy Gabrilovich, and Shaul Markovitch. 2011. A Word at a Time: Computing Word Relatedness using Temporal Semantic Analysis. In *Proceedings of the 20th International Conference on World Wide Web, WWW 2011*, pages 337–346.

Radim Rehurek and Petr Sojka. 2011. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2).

Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A. Smith. 2019. The Risk of Racial Bias in Hate Speech Detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1668–1678, Florence, Italy. Association for Computational Linguistics.

Gilles Sérasset. 2012. Dbnary: Wiktionary as a LMF based Multilingual RDF network. In *Language Resources and Evaluation Conference, LREC 2012*, Istanbul, Turkey. Nicoletta Calzolari and Khalid Choukri and Thierry Declerck and Mehmet Uğur Doğan and Bente Maegaard and Joseph Mariani and Jan Odijk and Stelios Piperidis.

Gilles Sérasset. 2015. DBnary: Wiktionary as a lemon-based multilingual lexical resource in RDF. *Semantic Web*, 6(4):355–361.

Chris Sweeney and Maryam Najafian. 2019. A Transparent Framework for Evaluating Unintended Demographic Bias in Word Embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1662–1667, Florence, Italy. Association for Computational Linguistics.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. LINE: Large-scale Information Network Embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077.

Erik F. Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the CoNLL-2000 Shared Task Chunking. In *Fourth Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Dongqiang (东强) Yang and David Powers. 2006. Word similarity on the taxonomy of WordNet.

Zhaocheng Zhu, Shizhen Xu, Meng Qu, and Jian Tang. 2019. GraphVite: A High-Performance CPU-GPU Hybrid System for Node Embedding. In *The World Wide Web Conference*, pages 2494–2504.