DEF2VEC: Extensible Word Embeddings from Dictionary Definitions

Irene Morazzoni, **Vincenzo Scotti** and **Roberto Tedesco**
DEIB, Politecnico di Milano
Via Golgi 42, 20133, Milano (MI), Italy
irene.morazzoni@mail.polimi.it vincenzo.scotti@polimi.it
roberto.tedesco@polimi.it

Abstract

DEF2VEC introduces a novel paradigm for word embeddings, leveraging dictionary definitions to learn semantic representations. By constructing term-document matrices from definitions and applying *Latent Semantic Analysis* (LSA), DEF2VEC generates embeddings that offer both strong performance and extensibility. In evaluations encompassing *Part-of-Speech tagging*, *Named Entity Recognition*, *chunking*, and *semantic similarity*, DEF2VEC often matches or surpasses state-of-the-art models like WORD2VEC, GLOVE, and FASTTEXT. Our model's second factorised matrix resulting from LSA enables efficient embedding extension for out-of-vocabulary words. By effectively reconciling the advantages of dictionary definitions with LSA-based embeddings, DEF2VEC yields informative semantic representations, especially considering its reduced data requirements. This paper advances the understanding of word embedding generation by incorporating structured lexical information and efficient embedding extension.

1 Introduction

Nowadays, *semantic representations* are the core of Natural Language Processing (NLP), allowing machines to capture the intricate relationships between words and their meanings (Liu et al., 2020b). *Word embeddings* have emerged as a cornerstone of this representation, enabling the translation of textual data into numerical vectors that encapsulate semantic nuances (Jurafsky and Martin, 2023, Chapter 6). These embeddings facilitate a wide range of NLP tasks, from sentiment analysis to machine translation, by endowing algorithms with means to comprehend and manipulate language, (Raffel et al., 2020; Brown et al., 2020; Sanh et al., 2022).

Contemporary advances in NLP have witnessed a paradigm shift from traditional *static word embeddings* to *dynamic contextual embeddings*, enabled by *Transformer* network-based models (Vaswani et al., 2017). Contextual embeddings, such as those derived from BERT (Devlin et al., 2019), GPT (Radford et al., 2018, 2019; Brown et al., 2020; OpenAI, 2023), and their variants, capture not just the inherent meaning of a word, but also its significance within the surrounding context (Liu et al., 2020a; Wang et al., 2022). While these contextual embeddings have undeniably revolutionised NLP benchmarks, the utilisation of static word embeddings persists. These embeddings, often generated through methods like WORD2VEC (Mikolov et al., 2013a,b), GLOVE (Pennington et al., 2014), and FASTTEXT (Bojanowski et al., 2017), remain valuable for their simplicity, interpretability, and efficiency (Hosseini et al., 2023).

In this paper, we introduce DEF2VEC, an innovative approach to constructing word embeddings that marries traditional static embeddings' virtues with dictionary definitions' information-rich nature. Recognising the enduring utility of static embeddings alongside the dominance of contextual embeddings, we propose exploiting the structured knowledge encapsulated in dictionary definitions. This unique strategy involves building term-document matrices from the definitions of words, which are subsequently factorised using *Latent Semantic Analysis* (LSA) (Deerwester et al., 1989, 1990). Remarkably, the embeddings extracted from this factorisation exhibit competitive performance on various tasks, often matching or even outperforming state-of-the-art static embeddings.

Our primary contribution lies in the extensibility of DEF2VEC embeddings. With a twofold factorisation approach, DEF2VEC generates robust embeddings from existing definitions and offers a seamless mechanism for accommodating new words into the embedding space. This scalability addresses a longstanding challenge in word embeddings, where adding new words necessitates retraining the entire model. The presented empirical evaluations across multiple tasks underscore the

strengths of DEF2VEC embeddings, making it a valuable alternative for static embedding models.

We divide this paper into the following sections. In Section 2, we discuss related works in the domain of word embeddings. Section 3 elaborates on the DEF2VEC model, detailing the term-document matrix construction and the factorisation process. Section 4 presents the data set employed for experimentation. Section 5 outlines our evaluation methodology, encompassing benchmarks and metrics. Subsequently, Section 6 dissects these results, offering insights into the efficacy of DEF2VEC embeddings. Finally, Section 7 summarises our findings and proposes possible future extensions.

2 Related works

Nowadays, word and sentence embeddings are fundamental tools in NLP, capturing the essence of language in numerical representations. In this section, we provide a taxonomy of embedding models, classifying them based on the level of the linguistic unit (*words vs. sentences/documents*) and the nature of the representation (*static vs. contextual* for word embeddings, *parametric vs. non-parametric* for sentence/document embeddings).

2.1 Word Embeddings

Word embeddings are the cornerstone of NLP, encapsulating word meanings in vector spaces. These embeddings can be broadly categorised into two main classes: static and contextual.

Static (or *shallow*) word embeddings capture word meanings independently of context, representing words as fixed vectors. Examples of this category include WORD2VEC (Mikolov et al., 2013a,b), GLOVE (Pennington et al., 2014), and FASTTEXT (Bojanowski et al., 2017), which generate embeddings through methods like *skip-gram*, *Continuos-Bag-of-Words* (CBoW), global *co-occurrence statistics*, and *sub-word information*.

Contextual (or deep) embeddings, on the other hand, integrate *contextual information* to produce dynamic representations. Models like ELMO (Peters et al., 2018), BERT (Devlin et al., 2019), GPT (Radford et al., 2018), and their variants generate embeddings by considering the surrounding words or sentences (i.e., the context), resulting in nuanced and context-sensitive representations (Liu et al., 2020b). These models build on top of word embeddings, starting from static representations and using *Deep Neural Networks* (DNNs) for sequence processing such as *Recurrent* neural network (Elman, 1990; Hochreiter and Schmidhuber, 1997) or *Transformer* (Vaswani et al., 2017) neural networks.

2.2 Sentence and Document Embeddings

While word embeddings capture individual word meanings, sentence and document embeddings aim to capture the meaning of larger textual units. These embeddings can be classified into parametric and non-parametric based on their approach to representation and generation.

Parametric sentence embeddings are generated using neural network architectures trained to produce fixed-size vectors from input sequences (e.g., sentences). SKIP-THOUGHT vectors (Kiros et al., 2015), SENT2VEC (Pagliardini et al., 2018), and SENTENCE-BERT (Reimers and Gurevych, 2019, 2020; Thakur et al., 2021) are examples of such approaches. As for contextual embeddings, these models are often built using DNNs for sequence processing.

Non-parametric sentence embeddings rely on pre-trained models for word embeddings or statistical methods to generate representations. Examples include averaging word embeddings, Smooth Inverse Frequency (SIF) WEIGHTING (Arora et al., 2017), and DYNAMAX (Zhelezniak et al., 2019), SFBOW (Muffo et al., 2021, 2023).

3 Model

DEF2VEC presents a novel approach to constructing word embeddings that exploits the structured information contained within dictionary definitions. The underlying principle of DEF2VEC involves the generation of term-document matrices from dictionary definitions, followed by LSA to yield semantically informative and extendable embeddings. In this section, we explain how to build the term-document matrix and how LSA is applied to this matrix to extract the embeddings. Additionally, we explain how the model can be extended with new embeddings without requiring any re-training, and, to conclude, we summarise the model capabilities.

3.1 Building the Term-Document Matrix

Given a vocabulary $\mathcal{V}$ of $|\mathcal{V}|$ terms (either words of multi-word expressions), each term in $\mathcal{V}$ is associated with one or more definitions extracted from linguistic resources. DEF2VEC constructs a term-document matrix $\mathbf{D}$, where each row corresponds to the *Term Frequency-Inverse Document Frequency*

(TF-IDF) representation of the definitions associated with a term. In the case of terms with multiple definitions (e.g., polysemous words), the TF-IDF vectors of individual definitions are averaged.

Mathematically, given a term $w \in \mathcal{V}$ represented as a *one-hot vector* $\mathbf{x} \in \mathbb{1}^{|\mathcal{V}|}$ (where $\mathbb{1} \equiv \{0,1\}$ and $\|\mathbf{x}\| = 1$) and its corresponding TF-IDF definition vector $\mathbf{y} \in \mathbb{R}^{|\mathcal{V}|}$, we establish the relationship defined in Equation (1).

$$\mathbf{y} = \mathbf{x} \cdot \mathbf{D} \tag{1}$$

Where $\mathbf{D} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ is a *sparse matrix*, connecting terms to their definitions. $\mathbf{D}_i$, the i-th row of $\mathbf{D}$, is such that $\mathbf{D}_i = \mathbf{y}$ (supposing w is the i-th term in $\mathcal{V}$).

3.2 Latent Semantic Analysis

To distil semantic information and generate embeddings, DEF2VEC applies LSA to the term-document matrix $\mathbf{D}$. LSA is de facto reduced (or truncated) *Singular Value Decomposition* (SVD), a method for matrix factorisation.

The term-document matrix $\mathbf{D}$ is factorised as reported in Equation (2). The process is represented in Figure 1.

$$\mathbf{D} \simeq \mathbf{U} \cdot \mathbf{\Sigma} \cdot \mathbf{V}^{\top} \tag{2}$$

Here, $\mathbf{\Sigma} \in \mathbb{R}^{d \times d}$ is a diagonal matrix with the singular values and $\mathbf{U} \in \mathbb{R}^{|\mathcal{V}| \times d}$ and $\mathbf{V} \in \mathbb{R}^{|\mathcal{V}| \times d}$ are matrices containing the left and right singular vectors, respectively. d is the desired embedding dimensionality, a tunable hyperparameter of the DEF2VEC model.

3.3 Extensibility and Reconstruction

The significance of DEF2VEC lies in its extensibility. SVD decomposition yields embeddings as rows of the matrix $\mathbf{U}$. However, the right singular vectors in $\mathbf{V}$ and the singular values in $\mathbf{\Sigma}$ can be exploited to generate embeddings from the TF-IDF representation of a term's definition as presented by Equation (3):

$$\mathbf{x} \cdot \mathbf{U} = \mathbf{U}_i = \mathbf{u} \simeq \mathbf{y} \cdot \mathbf{V} \cdot \mathbf{\Sigma}^{-1} \tag{3}$$

Both processes of embedding fetching from $\mathbf{U}$ and the embedding reconstruction from $\mathbf{V}$ and $\mathbf{\Sigma}$ are visualised in Figure 2.

This approach makes the embeddings extensible, enabling adding new terms without retraining the entire model. The downside of the approach is the small reconstruction error the truncation introduces after the SVD process. However, neural networks, which operate based on these embeddings, are expected to remain robust to the slight variations introduced by the reconstruction process (as we show in our evaluation).

Table 1: Comparison of vocabulary size and data set size of the considered word embedding models.

Model	**Vocabulary size** $\times 10^9$	**No. training tokens** $\times 10^9$
DEF2VEC	0.76	0.05
WORD2VEC	3	100
GLOVE	2	840
FASTTEXT	2.19	600

3.4 Robustness and Quality of Representations

DEF2VEC's embeddings benefit from the semantic richness of dictionary definitions while preserving the efficiency of static embeddings. This model leverages LSA's decomposition to capture latent semantic relationships within definitions, yielding embeddings that demonstrate semantic coherence even in the presence of noise and variation inherent in textual definitions.

In summary, DEF2VEC introduces a novel methodology that combines the interpretability and extensibility of static embeddings with the rich semantic information present in dictionary definitions. We realised our implementation of DEF2VEC using *Scikit-Learn* (Pedregosa et al., 2012), which offer utilities for TF-IDF vectorisation and SVD.

4 Data

The foundation of the DEF2VEC model lies in the use of the WIKTIONARY[1] as a rich source of linguistic information. WIKTIONARY, a project by the *Wikipedia Foundation*, offers a comprehensive dictionary encompassing various languages, providing definitions, pronunciations, etymologies, and more for a wide array of terms. To construct the DEF2VEC data set, we "mined" the English-language instance of the WIKTIONARY, extracting definitions to form the basis of our semantic representations.

To the end of this work, we used the dump file of the English WIKTIONARY from September 2020[2].

[1]Website: https://en.wiktionary.org/wiki/Wiktionary:Main_Page.

[2]The latest dumps of the WIKTIONARY are available

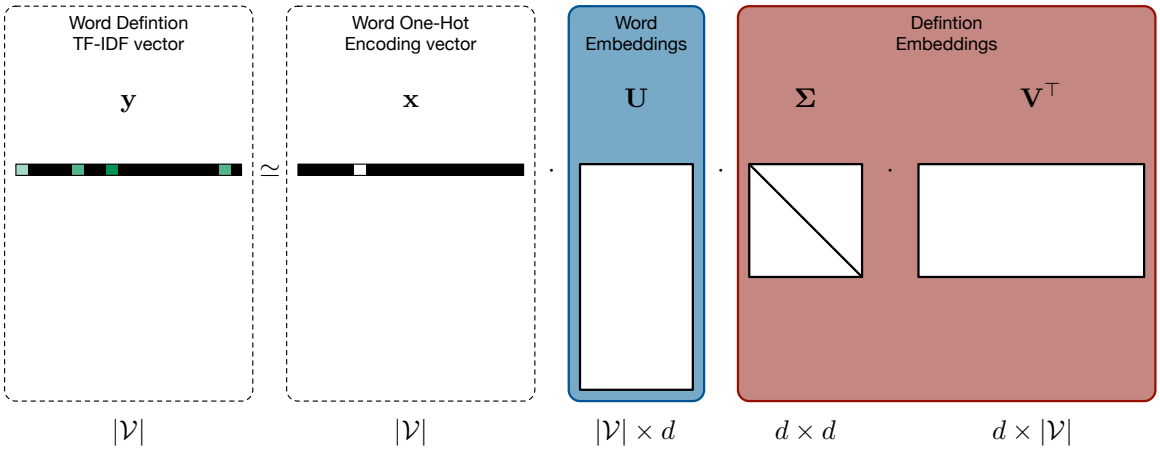

Figure 1: DEF2VEC term-document matrix decomposition.

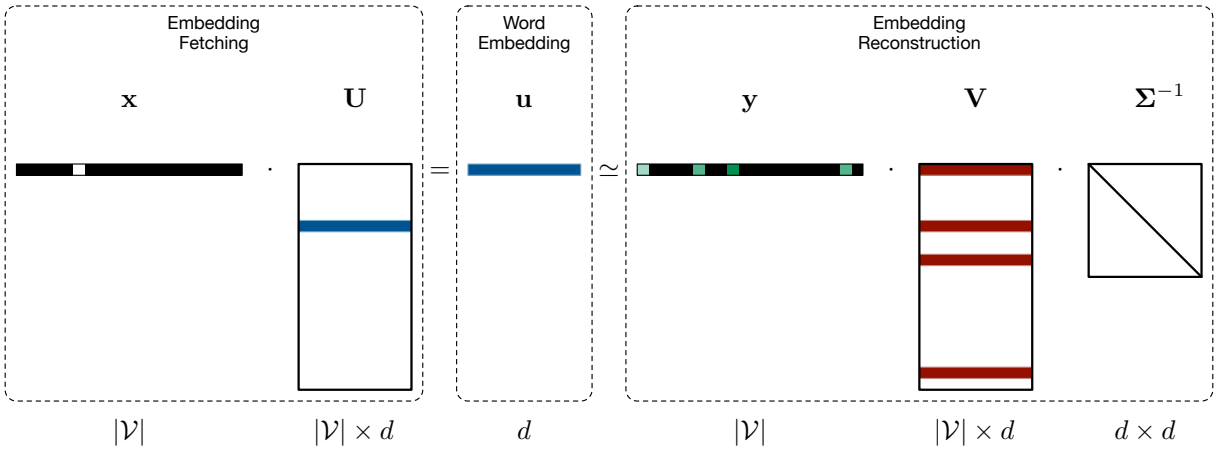

Figure 2: DEF2VEC embedding fetching and reconstruction processes.

The XML structure of the dump file consists of individual pages, each corresponding to a dictionary entry. A page includes a title, which is the term being defined, and a text section encapsulating the various elements of the entry, such as definitions, examples, synonyms, and more.

We cleaned and processed the data to generate a cohesive data set suitable for training the DEF2VEC model. We filtered out non-English entries and definitions associated with multiple languages, retaining only English ones. Additionally, we removed formatting tags, comments, and extraneous information, focusing solely on the textual content relevant to our work.

Each definition is preceded by the symbol `#`, which we removed during the parsing process. Our parser also excluded definitions marked with the label `rfdef`, indicating that the definitions did not exist on the corresponding WIKTIONARY web page.

at the following link: https://dumps.wikimedia.org/enwiktionary/.

We further addressed links to other WIKTIONARY pages, Wikipedia pages, or appendices, ensuring that only relevant words were retained.

We made distinctions between *locutions* (multi-word expressions) and *proper nouns*. While generic locutions were excluded, locutions related to proper nouns were retained. Proper nouns carry distinct semantic significance and enrich the contextual understanding of terms. The data set consists of approximately 764,595 tokens and 1,023,372 definitions. Additionally, it comprises 12,903 locutions. Notably, the data set includes 39,251 tokens containing punctuation, allowing the model to capture the nuances of language even in the presence of punctuated terms. Compared to other word embedding models, ours is less "data-hungry" as highlighted in Table 1.

By leveraging the structured information in WIKTIONARY, we constructed a comprehensive data set that serves as the foundation for training the DEF2VEC model. The subsequent sections delve into the architecture of DEF2VEC and its

Table 2: Main statistics of the benchmark data sets.

Bechmark	Split	No. samples	Avg. no. tokens
CoNLL-2003	**Train**	14,041	14.5
	Val.	3250	15.8
	Test	3453	13.5
	Total	20,744	14.5
STS	**Train**	5749	22.8
	Val.	1500	26.4
	Test	1379	22.6
	Total	8628	23.4

Table 3: Fraction of sentences containing tokens removed for the reconstruction evaluation.

Bechmark	Split	Faction of sentences [%]
CoNLL-2003	**Train**	40.7
	Val.	40.5
	Test	42.6
STS	**Train**	46.5
	Val.	50.0
	Test	60.2

performance across various tasks, illustrating its unique approach to word embeddings.

5 Evaluation

This comprehensive section presents our evaluation strategy for the DEF2VEC model. We describe the selected benchmarks, the evaluation approach, and the baselines we used for comparison.

5.1 Benchmarks

Our evaluation benchmarks encompass diverse linguistic tasks, providing a comprehensive understanding of DEF2VEC's performance.

We employed the CoNLL-2003 data set (Tjong Kim Sang and De Meulder, 2003) for sequence labelling tasks: namely *Part-of-Speech* (POS) tagging, *Named Entity Recognition* (NER) and *chunking* (CHUNK). The CoNLL-2003 data set was proposed as NER benchmark in the *CoNLL* conference (Daelemans and Osborne, 2003) and provides tags in *BIO* format for POS, NER, and CHUNK tasks. The data set, divided into training, validation, and test splits, facilitated an evaluation of DEF2VEC's capabilities in capturing linguistic structures and semantic nuances. Each sample is a pre-tokenised sentence (the input); each token of the sentence has its reference labels (the target output).

For sentence similarity, we turned to the Semantic Textual Similarity (STS) data set (Cer et al., 2017), evaluating DEF2VEC's ability to capture semantic relationships. The STS Benchmark comprises a selection of the English corpora used in organised in the context of the *SemEval* challenges between 2012 and 2016 (Agirre et al., 2012, 2013, 2014, 2015, 2016). The selection of corpora composing the data set includes text from image captions, news headlines, and user forums. Each sample in the corpus comprises a pair of sentences (the input) and their similarity score (the target output).

5.2 Approach

Our evaluation approach involves a two-pronged strategy: assessing embedding quality and reconstruction effectiveness. We trained Convolutional Neural Networks (CNNs) tailored to the tasks to evaluate embedding quality. These CNNs featured essential layers to ensure robust evaluations while mitigating overfitting risks:

- Dropout Layers: We introduced dropout layers with a 10% dropout probability, serving as regularisation mechanisms during training.
- Convolutional Layers: Our architecture included a convolutional layer with a kernel width of 5 embeddings, using the *Gaussian Error Linear Unit* (GELU) activation function for non-linearity.
- Additional Dropout: An extra dropout layer for further regularisation, with a 10% probability, followed the convolutional layers.

For sequence labelling tasks (POS, NER, CHUNK), our approach incorporated a linear layer to map input vectors to *logit* scores. Applying sequence-wise softmax operations yielded label probabilities. This enabled an in-depth evaluation of DEF2VEC's capacity to capture syntactic and semantic features across various linguistic tasks.

We utilised the *Semantic Textual Similarity* (STS) data set to train parametric sentence embedding models. We generated the sentence embeddings by employing a siamese network architecture with attention pooling. We computed the *cosine similarity* of these embeddings to quantify the

sentence similarity. This benchmark evaluated DEF2VEC's appropriateness to capture nuanced semantic relationships at sentence level.

The choices in the neural network architectures were mainly guided by the reference work of Collobert et al. (2011), where word embeddings were first used to improve results on different NLP tasks.

In addition to assessing embedding quality, we examined DEF2VEC's reconstruction capabilities. We pruned the WIKTIONARY data set, removing words with frequencies of 10 or fewer occurrences. Subsequently, we retrained DEF2VEC on the remaining 30,174 terms and their definitions.

We employed this reduced model to reconstruct embeddings for words in the benchmark samples lacking embeddings. We reported the statistics on the affected samples in Table 3. These reconstructed embeddings were generated from their Wiktionary definitions.

5.3 Baselines

Throughout all benchmark tasks, we conducted extensive comparisons between DEF2VEC (D2V) and established word embedding models: WORD2VEC (W2V), GLOVE (GV), and FASTTEXT (FT). These comparisons allowed us to gauge DEF2VEC's performance against widely recognised methods.

This comprehensive evaluation approach, supplemented by a thorough assessment of reconstruction capabilities, is the foundation for our analysis of DEF2VEC's performance in the subsequent results section. By investigating both embedding quality and reconstruction effectiveness, we aim to understand DEF2VEC's capabilities in capturing and representing semantic information within dictionary definitions.

6 Results

This section presents and discusses the results of our proposed DEF2VEC model across various linguistic tasks.

6.1 Sequence Labelling Tasks

Tables 4 to 6 showcase the classification results of DEF2VEC, along with comparisons to established embedding models, on the CONLL-2003 data set for the POS, NER, and CHUNK tasks, respectively. Across all tasks, DEF2VEC demonstrates competitive performances.

In the POS task, DEF2VEC achieves accuracy scores of 73.64% (Validation) and 72.42% (Test), showing its proficiency in capturing syntactic information. It clearly outperforms WORD2VEC, but is still distant from GLOVE and FASTTEXT.

For NER, DEF2VEC achieves 73.89% (Validation) and 71.98% (Test) accuracy. While GLOVE and FASTTEXT yields the highest accuracy, DEF2VEC remains competitive in precision, recall, F_1, and AUC. Results against WORD2VEC are still comparable on all metrics.

Table 4: Classification results on the POS task from CONLL-2003.

Model	Split	Metric [%]				
		Acc.	Prec.	Rec.	F_1	AUC
D2V	**Val.**	73.64	85.68	73.64	77.62	95.84
	Test	72.42	85.41	72.42	76.55	94.63
W2V	**Val.**	64.13	85.70	64.13	70.20	94.99
	Test	60.01	85.92	60.01	67.45	94.07
GV	**Val.**	82.53	90.49	82.53	85.43	97.94
	Test	82.38	90.79	82.38	85.51	97.86
FT	**Val.**	80.27	90.47	80.27	83.72	97.81
	Test	79.90	90.31	79.90	83.30	97.73

Table 5: Classification results on the NER task from CONLL-2003.

Model	Split	Metric [%]				
		Acc.	Prec.	Rec.	F_1	AUC
D2V	**Val.**	73.89	99.31	73.89	83.89	97.24
	Test	71.98	99.28	71.98	83.09	96.28
W2V	**Val.**	75.00	99.21	75.00	84.50	96.52
	Test	73.31	99.25	73.31	83.95	95.44
GV	**Val.**	91.80	99.58	91.80	95.34	99.29
	Test	90.52	99.47	90.52	94.60	99.21
FT	**Val.**	90.30	99.57	90.30	94.51	99.20
	Test	89.32	99.47	89.32	93.93	99.12

Table 6: Classification results on the CHUNK task from CONLL-2003.

Model	Split	Metric [%]				
		Acc.	Prec.	Rec.	F_1	AUC
D2V	**Val.**	77.79	86.81	77.79	81.34	94.37
	Test	77.69	86.56	77.69	81.45	93.07
W2V	**Val.**	66.12	82.97	66.12	71.35	90.28
	Test	64.94	82.19	64.94	71.00	87.91
GV	**Val.**	80.09	89.86	80.09	84.09	95.18
	Test	79.43	89.20	79.43	83.51	94.49
FT	**Val.**	82.38	90.21	82.38	85.60	95.03
	Test	82.28	89.63	82.28	85.39	94.55

Table 7: Spearman correlation score on the different subsets of the STS benchmark.

Model	Split	Spearman correlation [%]			
		Subset			Total
		Caption	Forum	News	
D2V	**Val.**	76.27	30.17	60.84	69.98
	Test	75.52	42.68	57.43	63.72
W2V	**Val.**	83.33	49.61	63.22	77.67
	Test	81.57	52.46	59.18	69.45
GV	**Val.**	83.45	55.96	66.60	78.37
	Test	80.17	53.49	63.16	69.00
FT	**Val.**	86.08	58.95	70.08	81.14
	Test	83.27	59.92	61.48	72.49

In the CHUNK task, DEF2VEC achieves an accuracy of 77.79% (Validation) and 77.69% (Test), consistently competing with established methods and again outperforming WORD2VEC.

Across these tasks, DEF2VEC demonstrates its proficiency in capturing syntactic and semantic features, effectively supporting sequence labelling tasks. DEF2VEC consistenstly performs better or similar to WORD2VEC embeddings. However, DEF2VEC only gets close, but never outperforms more sophisticated embeddings like GLOVE and FASTTEXT.

6.2 Sentence Similarity Benchmark

The Spearman correlation scores for the STS benchmark are shown in Table Table 7. Here, we can see that DEF2VEC's performance in capturing semantic relationships among sentence pairs is satisfactory.

For the validation subset, DEF2VEC achieves a Spearman correlation score of 69.98% (Total), with a string drop on the Forum subset (30.17%). All the other models share this drop, but it is not equally remarkable.

In the test subset, DEF2VEC obtains a correlation of 63.72% (Total). Differently from sequence labelling tasks, the gap with the validation results is more significant, but, again, it is a behaviour similar to that of all the baselines. Differently from the other models, the gap (absolute or relative) between validation and test is lower, hinting at higher robustness of the sentence embeddings.

While outperformed by the other model in all subsets, DEF2VEC maintains competitive performance and robustly captures semantic information, yielding overall Spearman correlations $> 60\%$.

Table 8: Classification results on the CONLL-2003 tasks of the reconstructed DEF2VEC embeddings.

Task	Split	Metric [%]				
		Acc.	Prec.	Rec.	F_1	AUC
POS	**Val.**	74.35	86.53	74.35	78.40	96.11
	Test	73.38	86.35	73.38	77.54	94.99
NER	**Val.**	74.19	99.32	74.19	84.09	97.23
	Test	72.28	99.29	72.28	83.30	96.37
CHUNK	**Val.**	77.84	86.97	77.84	81.47	94.44
	Test	77.84	86.61	77.84	81.56	93.08

Table 9: Differences between the classification scores on the CONLL-2003 tasks of the reconstructed DEF2VEC word embeddings and the original ones.

Task	Split	ΔMetric [%]				
		Acc.	Prec.	Rec.	F_1	AUC
POS	**Val.**	0.71	0.85	0.71	0.78	0.27
	Test	0.95	0.94	0.95	0.98	0.36
NER	**Val.**	0.29	0.01	0.29	0.20	−0.01
	Test	0.30	0.01	0.30	0.21	0.08
CHUNK	**Val.**	0.05	0.16	0.05	0.13	0.07
	Test	0.15	0.05	0.15	0.12	0.01

Table 10: Spearman correlation score on the STS benchmark of the reconstructed DEF2VEC embeddings.

Task	Split	Spearman correlation [%]			
		Subset			Total
		Caption	Forum	News	
STS	**Val.**	75.93	34.15	61.74	70.73
	Test	73.19	43.05	57.47	62.57

Table 11: Differences between the Spearman correlation scores on the STS benchmark of the DEF2VEC reconstructed word embeddings and the original ones.

Task	Split	ΔSpearman correlation [%]			
		Subset			Total
		Caption	Forum	News	
STS	**Val.**	−0.34	3.98	0.90	0.75
	Test	−2.32	0.36	0.05	−1.15

6.3 Reconstruction Capabilities

We evaluate DEF2VEC's reconstruction capabilities using the CONLL-2003 data set and the STS benchmark with reconstructed embeddings. Tables 8 and 10 depict the results of the models trained with the reconstructed embeddings, and

Tables 9 and 11 highlight the differences between the results obtained by the original DEF2VEC (trained on all the WIKITIONARY data) and the reconstructed embeddings.

For sequence labelling tasks (POS, NER, CHUNK), DEF2VEC's reconstructed embeddings exhibit slightly lower accuracy, precision, recall, and F_1 than original embeddings. However, the differences are generally marginal, showcasing the effectiveness of the reconstruction process.

Reconstructed embeddings exhibit varying performance across subsets in the STS benchmark. Some subsets show minor decreases in Spearman correlation scores, while others display improvements. Notably, the Forum subset's performance sees improvement in correlation scores, indicating the effectiveness of the reconstruction process in capturing specific nuances.

6.4 Model Discussion

DEF2VEC consistently showcases competitive performance across sequence labelling tasks and sentence similarity benchmarks. While its reconstructed embeddings exhibit slight variations in performance, the overall impact remains limited. This highlights DEF2VEC's robustness and potential to effectively capture and represent semantic information.

In conclusion, the DEF2VEC model presents a promising approach for learning word embeddings from dictionary definitions. Its semantic embedding quality and reconstruction capabilities demonstrate its utility in various linguistic tasks, making it a suitable alternative for advancing natural language understanding tasks in diverse applications.

7 Conclusion

In this study, we introduced DEF2VEC, a novel approach for learning word embeddings by leveraging dictionary definitions. DEF2VEC capitalises on the rich semantic information present in definitions to create embeddings that capture syntactic and semantic features. Through a comprehensive evaluation, we demonstrated the efficacy of DEF2VEC across various linguistic tasks, showcasing its ability to compete with established embedding models.

In the sequence labelling tasks of POS, NER, and CHUNK, DEF2VEC exhibited competitive accuracy, precision, recall, and F_1, illustrating its effectiveness in capturing linguistic nuances. Additionally, the model's performance on the STS benchmark reflected its capability to discern semantic relationships among sentence pairs, highlighting its utility in gauging semantic similarity across different contexts.

Moreover, we explored the dynamic extensibility of DEF2VEC, evaluating its ability to reconstruct embeddings of out-of-vocabulary words from their definitions. The results indicated that while the reconstructed embeddings displayed slight variations in performance, the overall impact remained limited, underscoring the robustness of the approach.

Our work opens for several future developments:

- Extending DEF2VEC to incorporate sub-word information, such as morphemes or character-level embeddings, could enhance its ability to capture finer linguistic nuances and improve its performance on tasks involving rare or out-of-vocabulary words.
- Adapting DEF2VEC to other languages can uncover cross-lingual variations in lexical semantics and offer insights into the universality of the approach. This could lead to the creation of embeddings that facilitate multilingual natural language processing tasks.
- Exploring different corpora than the Wiktionary could help assess the effect of the training data and identify better data sources.
- Conducting a more extensive evaluation of DEF2VEC on a broader array of linguistic tasks, such as syntactic parsing and semantic role labelling, could further validate its robustness and versatility.
- Utilising DEF2VEC embeddings as initialisation for training deep contextual models like BERT, GPT, or their successors could enhance language understanding and generation capabilities, potentially contributing to advancements in various natural language processing applications.

In conclusion, DEF2VEC introduces a novel perspective on word embedding learning that exploits dictionary definitions to produce embeddings with both syntactic and semantic information, which are also extensible. Its competitive performance across tasks and the potential for future extensions make it a promising candidate for enhancing the landscape of word embeddings and advancing natural language understanding.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. SemEval-2015 task 2: Semantic textual similarity, English, Spanish and pilot on interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 252–263, Denver, Colorado. Association for Computational Linguistics.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. SemEval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 81–91, Dublin, Ireland. Association for Computational Linguistics.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511, San Diego, California. Association for Computational Linguistics.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada. Association for Computational Linguistics.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43, Atlanta, Georgia, USA. Association for Computational Linguistics.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537.

Walter Daelemans and Miles Osborne, editors. 2003. *Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003*. ACL.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

Scott C Deerwester, Susan T Dumais, George W Furnas, Richard A Harshman, Thomas K Landauer, Karen E Lochbaum, and Lynn A Streeter. 1989. Computer information retrieval using latent semantic structure. US Patent 4,839,853.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jeffrey L. Elman. 1990. Finding structure in time. *Cogn. Sci.*, 14(2):179–211.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Marjan Hosseini, Alireza Javadian Sabet, Suining He, and Derek Aguiar. 2023. Interpretable fake news detection with topic and deep variational models. *Online Soc. Networks Media*, 36:100249.

Dan Jurafsky and James H. Martin. 2023. Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition (3rd edition). Draft.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 3294–3302.

Qi Liu, Matt J. Kusner, and Phil Blunsom. 2020a. A survey on contextual embeddings. *CoRR*, abs/2003.07278.

Zhiyuan Liu, Yankai Lin, and Maosong Sun. 2020b. *Representation Learning for Natural Language Processing*. Springer.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119.

Matteo Muffo, Roberto Tedesco, Licia Sbattella, and Vincenzo Scotti. 2021. Static fuzzy bag-of-words: a lightweight and fast sentence embedding algorithm. In *Proceedings of the Fourth International Conference on Natural Language and Speech Processing (ICNLSP 2021)*, pages 73–82, Trento, Italy. Association for Computational Linguistics.

Matteo Muffo, Roberto Tedesco, Licia Sbattella, and Vincenzo Scotti. 2023. *Static Fuzzy Bag-of-Words: Exploring Static Universe Matrices for Sentence Embeddings*, pages 191–211. Springer International Publishing, Cham.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2018. Unsupervised learning of sentence embeddings using compositional n-gram features. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 528–540, New Orleans, Louisiana. Association for Computational Linguistics.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2012. Scikit-learn: Machine learning in python. *CoRR*, abs/1201.0490.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *OpenAI blog*, 1(11):12.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Radim Řehůřek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *Proceedings of LREC 2010 workshop New Challenges for NLP Frameworks*, pages 45–50, University of Malta, Valletta, Malta. University of Malta.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon

Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 2021. Augmented SBERT: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 296–310, Online. Association for Computational Linguistics.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Thomas Wang, Adam Roberts, Daniel Hesslow, Teven Le Scao, Hyung Won Chung, Iz Beltagy, Julien Launay, and Colin Raffel. 2022. What language model architecture and pretraining objective works best for zero-shot generalization? In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 22964–22984. PMLR.

Vitalii Zhelezniak, Aleksandar Savkov, April Shen, Francesco Moramarco, Jack Flann, and Nils Y. Hammerla. 2019. Don't settle for average, go for the max: Fuzzy sets and max-pooled word vectors. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

A Pre-trained embeddings

The pre-trained embeddings for English used in the experiments are accessible in *Gensim*-compatible format (Řehůřek and Sojka, 2010). The embeddings can be downloaded from the *GitHub* repository[3].

[3] https : / / github.com / IreneMorazzoni / def_2_vec_irene