Lacking the Embedding of a Word? Look it up into a Traditional Dictionary

Elena Sofia Ruzzetti
University of Rome Tor Vergata, Italy

Leonardo Ranaldi
Guglielmo Marconi University, Italy

Michele Mastromattei
Campus Bio-Medico University, Italy
University of Rome Tor Vergata, Italy

Francesca Fallucchi
Guglielmo Marconi University, Italy

Noemi Scarpato
San Raffaele Roma Open University, Italy

Fabio Massimo Zanzotto
University of Rome Tor Vergata, Italy

Abstract

Word embeddings are powerful dictionaries, which may easily capture language variations. However, these dictionaries fail to give sense to rare words, which are surprisingly often covered by traditional dictionaries. In this paper, we propose to use definitions retrieved in traditional dictionaries to produce word embeddings for rare words. For this purpose, we introduce two methods: Definition Neural Network (DefiNNet) and Define BERT (DefBERT). In our experiments, DefiNNet and DefBERT significantly outperform state-of-the-art as well as baseline methods devised for producing embeddings of unknown words. In fact, DefiNNet significantly outperforms FastText, which implements a method for the same task-based on n-grams, and DefBERT significantly outperforms the BERT method for OOV words. Then, definitions in traditional dictionaries are useful to build word embeddings for rare words.

1 Introduction

Words without meaning are like compasses without needles: pointless. Indeed, meaningless words lead compositionally to meaningless sentences and, consequently, to meaningless texts and conversations. Second language learners may grasp grammatical structures of sentences, but, if they are unaware of the meaning of single words in these sentences, they may fail to understand the whole sentences, especially when there is an insufficient context for unfamiliar words. This is why a large body of natural language processing research is devoted to devising ways to capture word meaning.

As language is a living body, distributional methods (Turney and Pantel, 2010; Mikolov et al., 2013; Pennington et al., 2014) are seen as the panacea to capture word meaning as opposed to more static models based on dictionaries (Fellbaum, 1998) and other lexical resources (Baker et al., 1998; Kipper et al., 2000). Distributional methods may easily capture new meaning of existing words and, eventually, can easily assign meaning to emerging words. In fact, the different methods can scan corpora and derive the meaning of these new words by observing them in context (Harris, 1954; Firth, 1950; Wittgenstein, 1953). Words are then represented as vectors – now called word embeddings – which are then used to feed neural networks to produce meaning for sentences (Bengio et al., 2003; İrsoy and Cardie, 2014; Kalchbrenner et al., 2014; Tai et al., 2015) and meaning for whole texts (Joulin et al., 2017; Lai et al., 2015).

Distributional methods have a strong limitation: word meaning can be assigned only for words where sufficient contexts can be gathered. Rare words are not covered and become the classical out-of-vocabulary words, which may hinder the understanding of specific yet important sentences. To overcome this problem, n-grams based distributional models have emerged (Joulin et al., 2016) where word meaning is obtained by composing *"meaning"* of character n-grams forming a word. These n-grams act as proto-morphemes and, hence, meaning of unknown words can be obtained by composing meaning of proto-morphemes.

Traditional dictionaries can offer a solution to find meaning of rare words. They have been put aside since they cannot easily adapt to language evolution and they cannot easily provide distributed representations for neural networks.

In this paper, we propose to use definitions in dictionaries to compositionally produce distributional representations for out-of-vocabulary (OOV) words. Trying to reproduce in a distributional setting the compositional properties that hold between symbols is a debated task since compositional dis-

Findings of the Association for Computational Linguistics: ACL 2022, pages 2651 - 2662

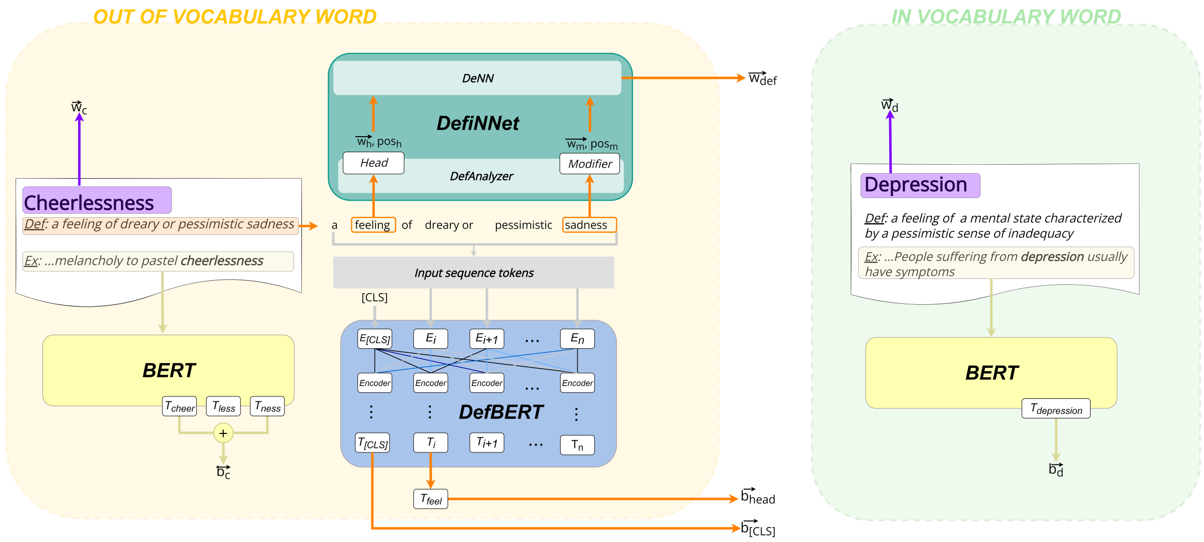


Figure 1: Exploiting definitions for out-of-vocabulary words: the DefiNNet and the DefBERT models.

tributional models were proposed (Mitchell and Lapata, 2008; Baroni and Zamparelli, 2010; Zanzotto and Dell'Arciprete, 2011; Paperno et al., 2014; Ferrone and Zanzotto, 2020). Definitions in dictionaries are intended to describe the meaning of a word to a human reader. Then, we propose two models to exploit definitions to derive the meaning of OOV words: (1) Definition Neural Network (DefiNNet), a simple neural network; (2) DefBERT, a model based on pre-trained BERT. We experimented with different tests and datasets derived from WordNet (Fellbaum, 1998). Firstly, we determined if DefiNNet and DefBERT can learn to derive word meaning from definitions. Secondly, we aimed to establish whether DefiNNet and DefBERT can cover OOV words, which are not covered by word2vec (Mikolov et al., 2013) or by the BERT pre-trained encoder, respectively. In our experiments, DefiNNet and DefBERT significantly outperform state-of-the-art as well as baseline methods devised for producing embeddings of unknown words. In fact, DefiNNet significantly outperforms FastText (Joulin et al., 2016), which implements a method for the same task-based on n-grams, and DefBERT significantly outperforms the BERT method for OOV words. Then, definitions in traditional dictionaries are useful to build word embeddings for rare words.

2 Background and Related Work

Out-of-vocabulary (OOV) words have been often a problem as these OOV words may hinder the applicability of many NLP systems. For example, if words are not included in a lexicon of a Probabilistic Context-Free Grammar, interpretations for sentences containing these words may have a null probability. Hence, solutions to this problem date back in time.

In the context of word embeddings, three families of solutions have been proposed: (1) context-based methods, (2) form-based methods, (3) combination of previous. The first family includes methods addressing the issue of learning new terms from tiny data either tuning existing models (Herbelot and Baroni, 2017) or performing a linear transformation on the average of all context word embedding (Khodak et al., 2018). In form-based methods, the most common solution is to use word n-grams (Joulin et al., 2016) or word pieces of variable length (Wu et al., 2016) as proxies to model morphemes. Embeddings are learned for 3-grams as well as for word pieces. In Joulin et al. (2016) these 3-grams are then combined to obtain the embedding for the entire word. For example, the word *cheerlessness*, which contains 3 morphemes (*cheer*, *less* and *ness*), is modeled by using embeddings for $\vec{che}$, $\vec{hee}$, ..., $\vec{ess}$ in the 3-gram approach and by using embeddings for $\vec{cheer}$ and $\vec{lessness}$ in the word pieces approach. These embeddings are possibly capturing information about the related morphemes. In this way, OOV word embeddings are correlated with meaningful bits of observed words. These models are our baselines. The last family includes methods taking into account both contextual and morphological information (Schick and Schütze, 2019; Hu et al., 2019; Schick and Schütze, 2020).

Deriving word embeddings for OOV words from dictionary definitions is an alternative approach. This approach has shown to be competitive in low resource scenarios in Bahdanau et al. (2017) where an LSTM model was fed with the definition. Dictionary definitions have been used in early attempts to train rudimentary compositional distributional semantic models (Zanzotto et al., 2010), which aimed to build embeddings for sequences of two words. In the word embedding field, several algorithms using definitions were proposed to build new embeddings matrices (Hill et al., 2016; Tissier et al., 2017; Bosc and Vincent, 2018). However, those methods are alternatives to the corpus-based distributional ones while our method is focused on tackling the OOV words problem, complementing existing word embedding spaces. Lexical resources have been also used exploiting their underlying semantic graph as an additional source of information (Pilehvar and Collier, 2017; Prokhorov et al., 2019). However, models based on those semantic graphs rely on a stronger assumption than models based on definitions only.

Universal sentence embedders (USEs) (Conneau et al., 2018) can play an important role in this novel approach. In fact, definitions are particular sentences aiming to describe meaning of words. Therefore, USEs should obtain an embedding representing the meaning of a word by composing embeddings of words in the definition.

Moreover, deriving word embeddings from definitions can be seen as a semantic stress test of universal sentence embedders. Generally, the ability of USEs (Devlin et al., 2019; Yang et al., 2020; Clark et al., 2020) to semantically model sentences is tested with end-to-end downstream tasks, for example, natural language inference (NLI) (Jiang and de Marneffe, 2019a; Raffel et al., 2020; He et al., 2021), question-answering (Zhang, 2019) as well as dialog systems (Wu et al., 2020). USEs such as BERT (Devlin et al., 2019) are encoding semantic features in hidden layers (Jawahar et al., 2019; Miaschi et al., 2020). However, USEs' success in downstream tasks may be due to superficial heuristics (as supposed in McCoy et al. (2019) and Ranaldi et al. (2022)) and not to deep modeling of semantic features. Therefore, our study can contribute to this debate. In fact, to the best of our knowledge, it is the first study aiming to investigate if USEs can model meaning by producing embedding for words starting from their definitions.

3 Model

This section introduces our proposals to use definitions in generating embeddings for out-of-vocabulary words: Definition Neural Network (DefiNNet) and BERT for Definitions (DefBERT). Section 3.1 describe the basic idea to process WordNet definitions. Section 3.2 describes the definition of the feed-forward neural network DefiNNet. Finally, Section 3.3 describes how we used the Universal Sentence Embedder BERT in producing embeddings for definitions.

3.1 Basic Idea

Our model stems from an observation: when someone steps into a rare unknown word while reading, definitions in traditional dictionaries are the natural resource used to understand the meaning of this rare, out-of-one's-personal-dictionary word. Then, as people rely on dictionaries in order to understand meanings for unknown words, learners of word embeddings could do the same.

Indeed, definitions in dictionaries are conceived to define compositionally the meaning of target words. Therefore, these are natural candidates for deriving a word embedding of an OOV word by composing the word embeddings of the words in the definition. The hunch is that universal sentence embedders can be used for this purpose.

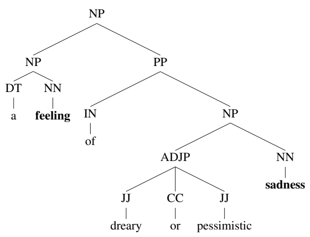
Moreover, these definitions have a recurrent structure, which can be definitely used to derive a simpler model. Definitions for words w are often organized as a particular sentence that contains the super-type of w and a modifier, which specializes the super-type (Amsler, 1980). For example (Fig. 1), *cheerlessness* is defined in WordNet as *a feeling*, which is the super-type, and *of dreary and pessimistic sadness*, which is the modifier. By using this structure, we propose a simpler model for composing meaning.

In the following sections, we propose two models: (1) DefiNNet, a model that exploits the structure of the definitions to focus on relevant words; and (2) DefBERT, a model that utilizes BERT as universal sentence embedder to embed the definition in a single vector.

3.2 DefiNNet: a feed-forward neural network to learn word embedding from definitions

The Definition Neural Network (DefiNNet) is our first model and has two main components (see Figure 1). The first component, *DefAnalyzer*, aims to spot the two important words of the definition: the super-type w_h and the main word w_m of the modifier of the super-type. The second component, *DeNN*, is a feed-forward neural network that takes in input the embeddings, $\vec{w}_h$ and $\vec{w}_m$, of the two selected words and produces the embedding for the target word $\vec{w}_{def}$.

To extract the two main words from a given definition, *DefAnalyzer* exploits the recurrent structure of definitions by using their syntactic interpretations. In our study, we use constituency parse trees and correlated rules to extract the super-type w_h and its closest modifier w_m. Basically, the simple algorithm is the following: given a definition s, parse the definition s and select the main constituent. If the main constituent contains a semantic head and a modifier, then those are the two target words. In the other case, select the semantic head of the main constituent as the super-type w_h and the semantic head of the first sub-constituent as the relevant modifier w_m. For example, the parse tree for the definition of $cherlessness$ in Fig. 1 is the following:



In this case, the main constituent is the first NP: the selected w_h is the word feeling which is semantic head of the first NP; w_m is noun sadness which is the semantic head of PP. The semantic heads are computed according to a slightly modified version of the semantic heads defined by Collins, 2003.

The second component is *DeNN* that, given the words embeddings $\vec{w}_h$ and $\vec{w}_m$ from the Word2Vec embedding space for respectively w_h and w_m from the definition, their POS tag p_h, p_m and the target's POS tag p_c as additional information, outputs the embedding $\vec{w}_c$ for the target word w_c. The input of DefiNNet is illustrated in Fig.1. The general equation for *DeNN* is:

$$\vec{w}_c = \mathbf{DeNN}(\vec{w}_h, \vec{w}_m, p_h, p_m, p_c)$$

The $\mathbf{DeNN}$ function can be described starting from three simpler subnets: (1) $\mathbf{FF}_w$ processes word embeddings $\vec{w}_h$ and $\vec{w}_m$; (2) $\mathbf{FF}_p$ embeds and processes p_h, p_m and p_c; finally, (3) $\mathbf{FF}$ processes the joint information from the previous steps.

The equation describing the subnet $\mathbf{FF}_w$ that takes as input $\vec{w}_h$ and $\vec{w}_m$ is the following:

$$\vec{s} = \mathbf{FF}_w(\vec{w}_h, \vec{w}_m) = \sigma(\mathbf{W}_s\sigma(\mathbf{W}_h\vec{w}_h + \mathbf{W}_m\vec{w}_m)) \tag{1}$$

where $\mathbf{W}_h$, $\mathbf{W}_m$ and $\mathbf{W}_s$ are dense layers and σ is the LeakyReLU activation function.

The subnet $\mathbf{FF}_p$ processes POS tags: p_h, p_m, p_c. Each p_i for $i \in \{h, m, c\}$ is firstly fed into an embedding layer ϵ which weights are learned from scratch. The resulting embedding $\epsilon(p_i)$ is then fed into a dense layer $\mathbf{W}_i$. Hence $\vec{p}_i$ is defined as follows:

$$\vec{p}_i = \mathbf{W}_i\epsilon(p_i)$$

The resulting $\vec{p}_h$, $\vec{p}_m$, $\vec{p}_c$ are then concatenated ($\oplus$) and fed into a dense layer $\mathbf{W}_p$. The following equation describes the subnet $\mathbf{FF}_p$:

$$\vec{p} = \mathbf{FF}_p(p_h, p_m, p_c) = \sigma(\mathbf{W}_p(\vec{p}_h \oplus \vec{p}_m \oplus \vec{p}_c) \tag{2}$$

The $\vec{s}$ resulting from Equation 1 and the $\vec{p}$ from Equation 2 are then concatenated ($\oplus$):

$$\vec{h} = \vec{s} \oplus \vec{p}$$

As final step $\vec{h}$ is fed into a feed-forward subnet $\mathbf{FF}$ composed of the dense layers $\mathbf{W}_1$, $\mathbf{W}_2$ and $\mathbf{W}_3$ as follows:

$$\mathbf{FF}(\vec{h}) = \mathbf{W}_3\sigma(\mathbf{W}_2(\sigma(\mathbf{W}_1\vec{h}))) \tag{3}$$

Hence the following:

$$\vec{w}_c = \mathbf{FF}(\mathbf{FF_w}(\vec{w}_h, \vec{w}_m), \mathbf{FF_p}(p_h, p_m, p_c))$$

describes how *DeNN* computes the embedding $\vec{w}_c$ for an OOV word having as input $\vec{w}_h$, $\vec{w}_m$, p_h, p_m from *DefAnalyzer* and p_c.

For comparative purposes, we defined two additional baseline models: an hypernym model ($Head$) and an additive model ($Additive$) (Mitchell and Lapata, 2008). The $Head$ model derives the embedding for the OOV word c by using the embedding for its hypernym h in WordNet, that is, $\vec{w}_c = \vec{w}_h$. The $Additive$ model instead adds the embeddings of the two words in the definition used by DefiNNet, that is, $\vec{w}_c = \vec{w}_h + \vec{w}_m$.

3.3 DefBERT: Transforming definitions in word embeddings

DefBERT aims to use BERT's ability to process sentences to use directly the definition for w_c in order to produce its embedding $\vec{w}_c$. DefBERT$_{[CLS]}$

and DefBERT$_{Head}$ are the approaches followed in exploiting the definition.

DefBERT$_{[CLS]}$ is the first of these approaches: in this case, the definition of w_c is given in input to a pretrained BERT-base model and, as shown in Figure 1, $\vec{b}_{[CLS]}$, the embedding for the [CLS] token, is taken as sentence embedding in the USE acceptation of BERT.

DefBERT$_{Head}$ is the second approach and in this case is selected $\vec{b}_{head}$, which is contextual embedding of $\vec{w_h}$ from the definition. Since BERT's embedding are contextual, $\vec{b}_{head}$ could benefit from the definition being the input sentence. A BERT pretrained model as USE in DefBERT$_{[CLS]}$ and its ability in producing contextualized word embeddings in DefBERT$_{Head}$ definition can hence be exploited in producing embeddings for OOV.

For comparative purposes, we also define BERT$_{wordpieces}$ and BERT$_{Head-Example}$. BERT$_{wordpieces}$ is used to see if our model outperforms the classical behavior of BERT when it encounters OOV words. In this case, BERT is fed with a sample sentence containing the target OOV word, for example *"... melancholy to pastel cheerlessness"* for the target OOV *"cheerlessness"* (see Figure 1). Then, the word is divided into word pieces. To obtain the embedding for the target word, we sum up vectors of these word pieces. BERT$_{Head-Example}$ instead is used to determine if definitions are really useful for modeling meaning of the head word. BERT$_{Head-Example}$ is similar to DefBERT$_{Head}$ but the input is different. BERT$_{Head-Example}$ has a random sentence that contains the head word. Hence, comparing DefBERT$_{Head}$ with BERT$_{Head-Example}$ gives intuition if the head in definition really absorbs its meaning.

4 Experiments

Experiments aim to investigate three issues: (1) if DefiNNet and DefBERT word embeddings are reasonably better than baseline models for indirectly generating embeddings; (2) the highly debated question whether similarity measures over WordNet are correlated with word embeddings (Lastra-Díaz et al., 2019); (3) finally, if DefiNNet and DefBERT word embeddings for out-of-vocabulary words obtained are good word representations in terms of their correlation with similarity measures on WordNet. Clearly, issue (2) is necessary to investigate issue (3).

The rest of the section is organized as follows. Section 4.1 introduces the general settings of our experiments. Section 4.2 presents results and it is organized in four subsections, which address the above three issues. If needed, these subsections introduce additional settings for the experiments.

4.1 Experimental set-up

Our experiments are defined around WordNet (Fellbaum, 1998) and around the two word embedding spaces of Word2Vec (Mikolov et al., 2013) (W_{w2v}) and of BERT (Devlin et al., 2019) (W_{BERT}). WordNet (Fellbaum, 1998) is the source of word definitions, it is used to collect testing sets of pairs of similar and dissimilar words and similarity measures over WordNet are used to rank them.

Then, IV_{w2v} and IV_{BERT} are WordNet words in the target embedding matrices W_{w2v} and W_{BERT}, respectively, and OOV_{w2v} and OOV_{BERT} are WordNet words outside these matrices.

Additionally, IV_{BERT} and OOV_{BERT} are restricted to words with usage example in WordNet as these examples are needed for applying DefBERT. The datasets derived from those sets are described in Table 1.

Word2Vec (Mikolov et al., 2013) and BERT (Devlin et al., 2019) offer instead large pre-trained word embedding spaces. Indeed, Word2Vec's embedding space (Mikolov et al., 2013) is pre-trained on part of Google News dataset (about 100 billion words) and the BERT's word embedding space (Devlin et al., 2019) is pre-trained on lower-cased English text from BooksCorpus (800M words) (Zhu et al., 2015) and English Wikipedia (2,500M words) as described by Devlin et al. (2019).

Dataset	Subset of	Size	
$Train_{w2v}$	IV_{w2v}	31,471 (train) 7,867 (val)	
$Test_{w2v}$	IV_{w2v}	9,931	
$Test_{BERT}$	IV_{BERT}	3,218	
Dataset	Subset of	Size	# Sublists
$Pairs_{IV_{w2v}}$	$IV_{w2v} \times IV_{w2v}$	14,000	2,000
$Pairs_{IV_{BERT}}$	$IV_{BERT} \times IV_{BERT}$	560	80
$Pairs_{IV_{fasttext}}$	$IV_{fasttext} \times IV_{fasttext}$	14,000	2,000
$Pairs_{w2v}$	$OOV_{w2v} \times IV_{w2v}$	4,500	600
$Pairs_{BERT}$	$OOV_{BERT} \times IV_{BERT}$	3,500	450
$Pairs_{w2v \cap BERT}$	$Pairs_{w2v} \cap Pairs_{BERT}$	450	60

Table 1: Datasets defined over WordNet

To investigate the first issue described at the beginning of this section, we introduced $Train_{w2v}$, $Test_{w2v}$, and $Test_{BERT}$. $Train_{w2v}$ is DefiNNet training set: this dataset contains definition for IV_{w2v} words since they are needed as target of

DefiNNet. $Test_{w2v}$ is a test dataset and it is completely analogous to $Train_{w2v}$ (Sec, 4.2.1). Since DefBERT$_{[CLS]}$ is not trained, $Test_{BERT}$ is the dataset prepared. Benchmarks on similarity and relatedness are also introduced in Sec 4.2.2

DefiNNet and DefBERT are also tested to assess their ability to produce embeddings for OOV that may replicate some similarity measure between words in pairs. The investigated pairs consist of WordNet "sister terms": two words are sister if they are both immediate hyponyms of the same node. In WordNet sister terms are definitely positive examples of similar words as well as negative example pairs can be generated by selecting pairs of words uniformly at random. $Pairs$ datasets are composed of positive or negative examples of sister terms. To address the second issue presented in Sec 4, $Pairs_{IV_{w2v}}$, $Pairs_{IV_{BERT}}$, $Pairs_{IV_{fasttext}}$ datasets are generated. In this datasets both w_1 and w_2 are IV words. Then, we collected two sets of pairs of words $Pairs_{w2v}$ and $Pairs_{BERT}$: those datasets are used to test if the correlation with similarity measures holds with OOV word embedding derived from DefiNNet or DefBERT. To capture different degrees of similarity among pairs of words in WordNet, we selected three similarity measures defined over WordNet: $path$ (Rada et al., 1989), wup (Wu and Palmer, 1994) and res (Resnik, 1995). To correctly apply Spearman's correlation between our systems and the expected rank on the list of pairs induced by a similarity measure, we divided $Pairs$ datasets into lists of 7 pairs. Pairs in the list are selected to have 7 clearly different values of the selected similarity ($path$, wup and res) between the two words. The final Spearman's correlation is a distribution of correlation over these lists.

To comparatively investigate our DefiNNet and DefBERT, we used FastText (Bojanowski et al., 2016) as realized in Grave et al. (2018) along with: (1) Additive and Head defined in Section 3.2; (2) BERT$_{wordpieces}$ and BERT$_{Head-Example}$ defined in Section 3.3. FastText defines embeddings unknown words c by combining embeddings of 3-grams, for example, the embedding for the OOV word *cheerlessness* is represented as the vector $\vec{f_c} = \vec{che} + \vec{hee} + ... + \vec{ess}$.

As final experimental setting, definitions are parsed using Stanford's CoreNLP probabilistic context-free grammar parser (Manning et al., 2014). NLTK (Loper and Bird, 2002) is used to access WordNet and compute similarity measures over it.

4.2 Results and discussion

For clarity, this section is organized around the three issues we aim to investigate: the ability of proposed methods to build embeddings of words starting from dictionary definitions (Sec. 4.2.1, Sec. 4.2.2); the debated relation between similarity over word embeddings and similarity in WordNet (Sec. 4.2.3); and, finally, the ability of the proposed methods to produce embeddings for OOV words (Sec. 4.2.4).

4.2.1 Word Embeddings from Dictionary Definitions

The first issue to investigate is whether our methods produce word embeddings from dictionary definitions that are similar with respect to word embeddings directly discovered. We then studied the cosine similarity between the two kinds of embeddings, for example, between the embedding of *cheerlessness* and the embedding of the definition *a feeling of .... sadness*. For the diffent methods, the comparison is on their own space, that is, $sim(\vec{w}_c, \vec{w}_{def})$ for DefiNNet and $sim(\vec{b}_c, \vec{b}_{[CLS]})$ or $sim(\vec{b}_c, \vec{b}_{head})$ for DefBERT$_{[CLS]}$ and DefBERT$_{Head}$, respectively (see Fig. 1). Experiments are conducted on In-Vocabulary words for both spaces by using the $Test_{w2v}$, $Test_{BERT}$ and $Test_{w2v \cap BERT}$ datasets.

Dataset	*Model*	**nouns** *sim*	**verbs** *sim*
$Test_{w2v}$	Additive	$0.25(\pm0.17)^{\circ}$	$0.29(\pm0.19)^{\circ}$
	Head	$0.26(\pm0.21)^{\star}$	$0.29(\pm0.25)^{\star}$
	DefiNNet	$\mathbf{0.39}(\pm\mathbf{0.18})^{\circ\star}$	$\mathbf{0.46}(\pm\mathbf{0.14})^{\circ\star}$
$Test_{BERT}$	DefBERT$_{Head}$	$\mathbf{0.46}(\pm\mathbf{0.13})^{\dagger\ddagger}$	$\mathbf{0.41}(\pm\mathbf{0.14})^{\dagger\ddagger}$
	DefBERT$_{[CLS]}$	$0.32(\pm0.08)^{\dagger}$	$0.30(\pm0.09)^{\dagger}$
	BERT$_{Head-Example}$	$0.41(\pm0.12)^{\ddagger}$	$0.39(\pm0.12)^{\ddagger}$
$Test_{w2v \cap BERT}$	DefBERT$_{Head}$	$\mathbf{0.47}(\pm\mathbf{0.13})^{\dagger\triangle}$	$0.42(\pm0.15)^{\dagger\triangle}$
	DefBERT$_{[CLS]}$	$0.28(\pm0.09)^{\dagger\diamond}$	$0.30(\pm0.09)^{\dagger\diamond}$
	DefiNNet	$0.33(\pm0.13)^{\triangle\diamond}$	$\mathbf{0.47}(\pm\mathbf{0.13})^{\triangle\diamond}$

Table 2: Cosine similarity between word embeddings and embeddings of their definitions. The marking signs $\star$, $\circ$, $\dagger$, $\ddagger$ and $\diamond$ indicate pairs of models results for which the higher result is statistically significant better than the other (with a 95% confidence level) according to the one-sided Wilcoxon signed-rank test.

Definitions seem to be better sources of word embeddings instead of baseline methods and other solutions. In fact, both DefiNNet and DefBERT$_{Head}$ outperform different methods in their respective tests for both nouns and verbs (see Table 2). For nouns, DefiNNet has an average cosine similarity of $0.39(\pm0.18)$, which is well above that of Additive $(0.25(\pm17))$ and Head $(0.26(\pm21))$.

In the same syntactic category, DefBERT$_{Head}$ outperforms BERT$_{Head-Example}$, $0.46(\pm0.13)$ vs. $0.41(\pm0.12)$. For verbs, DefiNNet has an average cosine similarity of $0.46(\pm0.14)$, which is well above the Additive and the Head. In the same category, DefBERT$_{Head}$ slightly outperforms BERT$_{Head-Example}$. Finally, in the common test, that is, $Test_{w2v \cap BERT}$, definition-based models outperform simpler models. DefBERT$_{Head}$ has a better similarity for nouns and DefiNNet has a better similarity for verbs.

For BERT, the embedding related to the token [CLS] does not seem to represent the good token where to take semantics of the sentence in terms of a real composition of the meaning of component words. DefBERT$_{[CLS]}$ performs poorly with respect to DefBERT$_{Head}$ and also with respect to BERT$_{Head-Example}$ in both syntactic categories for $Test_{BERT}$ (see Table 2). This is confirmed in the restricted set $Test_{w2v \cap BERT}$. Therefore, even if the embedding in token [CLS] is often used as universal sentence embedding for classification purposes (Devlin et al., 2019; Adhikari et al., 2019; Jiang and de Marneffe, 2019b), it may not contain packed meaning whereas it may contain other kinds of information regarding the sentence.

4.2.2 Standard Relatedness and Similarity Tests

In this section, DefiNNet embeddings are evaluated by measuring their ability to capture similarity and relatedness of words pairs. The used benchmarks contain words pairs and a score of similarity for each pair assigned by human assessors. If the similarity among embeddings correlates with the assigned similarity score, then the embeddings are considered capable of capturing similarity and relatedness. In this scenario, the first word's embedding of each pair is computed according to the examined method, the second embedding comes from the Word2Vec embedding space. The obtained Spearman's coefficients are presented in Table 3. Head and Additive baseline models are also tested.

DefiNNet achieves better correlation with all the tested relatedness benchmarks: `MEN` (Bruni et al., 2014), `MTurk-287` (Radinsky et al., 2011) and `MTurk-771` (Halawi et al., 2012). Among the similarity benchmarks, DefiNNet outperforms the Additive and Head baseline in different tasks. With `RareWords` (Luong et al., 2013), composed of words with low occurrences, DefiNNet significantly outperforms both baselines. The correlation coefficients calculated with `SimLex999` (Hill et al., 2015) are instead closer and relatively lower. Head achieves the best results with the smaller `RG-65` (Rubenstein and Goodenough, 1965) and its subset `MC-30` (Miller and Charles, 1991). DefiNNet achieves a higher Spearman's coefficient in `SimVerb-3500` (Gerz et al., 2016), `Verb-143` (Baker et al., 2014) and `YP-130` (Yang and Powers, 2006) which assess similarity on verbs pair.

Benchmark	DefiNNet	Head	Additive
`MEN`	$\mathbf{0.48}(\pm\mathbf{0.01})^{\diamond\dagger}$	$0.37^{\diamond}$	$0.39^{\dagger}$
`MTurk-287`	$\mathbf{0.46}(\pm\mathbf{0.02})^{\diamond\dagger}$	$0.39^{\diamond}$	$0.39^{\dagger}$
`MTurk-771`	$\mathbf{0.37}(\pm\mathbf{0.01})^{\diamond\dagger}$	$0.33^{\diamond}$	$0.33^{\dagger}$
`RareWords`	$\mathbf{0.32}(\pm\mathbf{0.01})^{\diamond\dagger}$	$0.20^{\diamond}$	$0.02^{\dagger}$
`SimLex999`	$0.18(\pm0.01)^{\diamond\dagger}$	$0.15^{\diamond}$	$\mathbf{0.19}^{\dagger}$
`RG-65`	$0.43(\pm0.04)^{\diamond}$	$\mathbf{0.63}^{\diamond}$	0.41
`MC-30`	$0.27(\pm0.07)^{\diamond\dagger}$	$\mathbf{0.71}^{\diamond}$	$0.33^{\dagger}$
`SimVerb-3500`	$\mathbf{0.27}(\pm\mathbf{0.01})^{\diamond\dagger}$	$0.22^{\diamond}$	$0.22^{\dagger}$
`Verb-143`	$\mathbf{0.41}(\pm\mathbf{0.02})^{\diamond\dagger}$	$0.25^{\diamond}$	$0.26^{\dagger}$
`YP-130`	$\mathbf{0.43}(\pm\mathbf{0.02})^{\diamond\dagger}$	$0.27^{\diamond}$	$0.27^{\dagger}$

Table 3: Spearman's correlation coefficients on similarity and relatedness benchmarks. Mean and standard deviation results in DefiNNet are obtained from 10 runs. The symbols $\diamond$ and $\dagger$ indicate a statistically significant difference between two results (with a 95% confidence level) according to the one-sided Wilcoxon signed-rank test.

4.2.3 Word Embedding Spaces and WordNet

WordNet and its correlated similarly metrics can be an interesting opportunity to extract testsets for assessing whether our methods can be used to derive embeddings of OOV words. However, it is a strongly debated question whether similarities in WordNet are correlated with similarities over word embeddings (Lastra-Díaz et al., 2019).

Model	*Dataset*	*Measure*	*Spearman*
Word2Vec	$Pairs_{IV w2v}$	*path*	$0.25(\pm0.39)$
		wup	$0.25(\pm0.38)$
		res	$0.50(\pm0.31)$
FastText	$Pairs_{IV fasttext}$	*path*	$0.31(\pm0.38)$
		wup	$0.40(\pm0.35)$
		res	$0.52(\pm0.29)$
BERT	$Pairs_{IV BERT}$	*path*	$0.09(\pm0.41)$
		wup	$0.30(\pm0.39)$
		res	$0.28(\pm0.38)$

Table 4: Average Spearman's coefficient measuring correlation on cosine similarity among embedding and similarity over WordNet taxonomy.

The aim of this section is to select WordNet

Dataset	*Model*	*Corr(path)*	*Corr(wup)*	*Corr(res)*
$Pairs_{w2v}$	Additive	$\mathbf{0.24(\pm 0.40)}^{\circ}$	$0.46(\pm 0.32)^{\circ}$	$0.44(\pm 0.34)^{\circ}$
	Head	$0.23(\pm 0.37)^{\star}$	$0.49(\pm 0.30)$	$0.49(\pm 0.31)^{\star}$
	FastText	$0.07(\pm 0.40)$	$0.43(\pm 0.36)^{\diamond}$	$0.41(\pm 0.35)^{\diamond}$
	DefiNNet	$0.03(\pm 0.42)^{\circ\star}$	$\mathbf{0.50(\pm 0.31)}^{\circ\diamond}$	$\mathbf{0.51(\pm 0.31)}^{\circ\star\diamond}$
$Pairs_{BERT}$	DefBERT$_{Head}$	$\mathbf{0.27(\pm 0.36)}^{\ddagger\bullet}$	$\mathbf{0.33(\pm 0.37)}^{\dagger\ddagger\bullet}$	$\mathbf{0.31(\pm 0.36)}^{\dagger\ddagger\bullet}$
	DefBERT$_{[CLS]}$	$0.26(\pm 0.36)$	$0.17(\pm 0.37)\dagger$	$0.11(\pm 0.39)^{\dagger}$
	BERT$_{Head-Example}$	$0.15(\pm 0.41)^{\ddagger}$	$0.25(\pm 0.38)^{\ddagger}$	$0.19(\pm 0.40)^{\ddagger}$
	BERT$_{wordpieces}$	$0.09(\pm 0.37)^{\bullet}$	$0.19(\pm 0.37)^{\bullet}$	$0.23(\pm 0.38)^{\bullet}$
$Pairs_{w2v \cap BERT}$	DefBERT$_{Head}$	$0.12(\pm 0.44)^{\diamond}$	$0.33(\pm 0.36)^{\bullet}$	$0.27(\pm 0.39)^{\bullet}$
	DefiNNet	$\mathbf{0.31(\pm 0.37)}^{\diamond\triangle}$	$\mathbf{0.39(\pm 0.33)}^{\triangle}$	$\mathbf{0.35(\pm 0.36)}^{\triangle}$
	FastText	$0.19(\pm 0.42)$	$0.35(\pm 0.36)$	$0.32(\pm 0.37)$
	BERT$_{wordpieces}$	$0.11(\pm 0.37)^{\triangle}$	$0.14(\pm 0.42)^{\bullet\triangle}$	$0.18(\pm 0.34)^{\bullet\triangle}$

Table 5: Average Spearman's coefficient from the sister terms investigation. The marking signs $\star$, $\circ$, $\bullet$, $\dagger$, $\ddagger$, $\triangle$ and $\diamond$ indicate pairs of models results for which the higher result is statistically significant better than the other (with a 95% confidence level) according to the one-sided Wilcoxon signed-rank test.

similarity measures that can be used to investigate the quality of embeddings generated for OOV words. For this experimental session, we used the $Pairs_{IV_{w2v}}$, $Pairs_{IV_{BERT}}$ and $Pairs_{IV_{fasttext}}$ datasets defined in Section 4.1, which are composed of sister terms in WordNet.

Sister terms may be very similar or less similar. For example, *cheerlessness* and *depression* (see Figure 1) are sister terms and are definitely similar. On the contrary, *house* and *architecture* are sister terms but are less similar with respect to the previous pair of words. In WordNet, this difference in similarity is captured by using many different metrics.

We investigated three different WordNet similarity measures: $path$ (Rada et al., 1989), wup (Wu and Palmer, 1994) and res (Resnik, 1995). The measure $path$ uses the length of the path connecting two synsets over the WordNet taxonomy. The measure wup is still based on the length of path between the synsets related to the two words and takes into account the number of edges from synsets to their Least Common Subsumer (LCS) and the number of links from the LCS up to the root of the taxonomy. Finally, the measure res belongs to another family of measures as it is based on the Information Content. In res, the similarity between synsets of the related words is a function of the Information Content of their LCS. In this case, a more informative LCS (a rare as well as a specific concept) indicates that the hyponym concepts are more similar.

The best correlated WordNet measure is res. In fact, it is highly correlated for two spaces out of three, Word2Vec and FastText, and it is on par with wup in the BERT space (see 4). The average Spearman's correlation between the word embedding spaces of Word2Vec and res is $0.50(\pm 0.31)$, which is well above $path$ and wup. The same happens for the space FastText where the correlation is $0.52(\pm 0.29)$.

As a final consideration, for our purposes, word embedding spaces are correlated and the best measure that captures this correlation is res.

4.2.4 Testing over out-of-vocabulary words

The final analysis is on real OOV words for Word2Vec and for BERT. These last experiments are carried out by considering the positive relation between WordNet similarity measures and the word embedding spaces.

Using definitions for deriving word embeddings for OOV words seems to be the good solution compared to alternative available approaches.

In its space, DefiNNet achieves very important results for the correlation with the two WordNet similarity measures wup and res (see Table 5). In both cases, it outperforms FastText, which is a standard approach for deriving word embeddings for OOV words (0.51 ± 0.31 vs. 0.41 ± 0.35 for res and 0.50 ± 0.30 vs. 0.43 ± 0.36 for wup). Moreover, DefiNNet outperforms Head, a baseline method based on WordNet, and Additive, the simplest model to use WordNet definitions.

The same happens for DefBERT$_{Head}$ in its space (see Table 5). DefBERT$_{Head}$ significantly outperforms BERT$_{wordpieces}$, showing that DefBERT$_{Head}$ is a better model to treat OOV with respect to that already included in BERT. Results on DefBERT$_{Head}$ confirm that the output related to the token representing the head carries better information than the output related to the token [CLS]. Moreover, the definition has is a positive effect on shaping the word embedding of the head word towards the defined word. In fact, DefBERT$_{Head}$ and BERT$_{Head-Example}$ are applied on the same head word and DefBERT$_{Head}$ transforms better the meaning than BERT$_{Head-Example}$, which is applied to a random sentence containing the head word. Indeed, also for BERT, definitions are important in determining embeddings of OOV words.

The final comparison is between DefiNNet and DefBERT$_{Head}$ and it is done on the small dataset $Pairs_{w2n \cap BERT}$. DefiNNet achieves better results than DefBERT$_{Head}$ for all the three WordNet measures (see Table 5) but statistical significance between them cannot be asserted with the fixed p-value (0.05).

5 Conclusions and Future Work

Building word embedding for rare out-of-vocabulary words is essential in natural language processing systems based on neural networks. In this paper, we proposed to use definitions in dictionaries to solve this problem. Our results show that this can be a viable solution to retrieve word embedding for OOV rare words, which work better than existing methods and baseline systems.

Moreover, the use of dictionary definitions in word embedding may open also another possible line of research: a different semantic probe for universal sentence embedders (USEs). Indeed, definitions offer a definitely interesting equivalence between sentences and words. Hence, unlike existing semantic probes, this approach can unveil if USEs are really changing compositionally the meaning of sentences or are just aggregating pieces of sentences in a single representation.

Finally, this paper promotes responsible Artificial Intelligence as intended in Human-in-the-Loop Artificial Intelligence (Zanzotto, 2019). In fact, it gives the possibility to track how human knowledge is used by learning algorithms.

Acknowledgments

This work is partially funded by the 2019 BRIC INAIL ID32 SfidaNow project. We would like to thank Giorgio Gambosi, Arianna Patrizi and Aria Nourbakhsh for the useful discussions.

References

Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin. 2019. Docbert: Bert for document classification.

Robert Amsler. 1980. The structure of the merriam-webster pocket dictionary /.

Dzmitry Bahdanau, Tom Bosc, Stanislaw Jastrzebski, Edward Grefenstette, Pascal Vincent, and Yoshua Bengio. 2017. Learning to compute word embeddings on the fly. *CoRR*, abs/1706.00286.

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*.

Simon Baker, Roi Reichart, and Anna Korhonen. 2014. An unsupervised model for instance level subcategorization acquisition. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 278–289, Doha, Qatar. Association for Computational Linguistics.

Marco Baroni and Roberto Zamparelli. 2010. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1183–1193, Cambridge, MA. Association for Computational Linguistics.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3(null):1137–1155.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Tom Bosc and Pascal Vincent. 2018. Auto-encoding dictionary definitions into consistent word embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1522–1532, Brussels, Belgium. Association for Computational Linguistics.

Elia Bruni, Nam Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

Michael Collins. 2003. Head-driven statistical models for natural language parsing. *Computational Linguistics*, 29(4):589–637.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 1:2126–2136.

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.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

Lorenzo Ferrone and Fabio Massimo Zanzotto. 2020. Symbolic, distributed, and distributional representations for natural language processing in the era of deep learning: A survey. *Frontiers in Robotics and AI*, 6.

J. R. Firth. 1950. Personality and language in society. *The Sociological Review*, a42(1):37–52.

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.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Guy Halawi, Gideon Dror, Evgeniy Gabrilovich, and Yehuda Koren. 2012. Large-scale learning of word relatedness with constraints. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, page 1406–1414, New York, NY, USA. Association for Computing Machinery.

Zellig S. Harris. 1954. Distributional structure. *<i>WORD</i>*, 10(2-3):146–162.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention.

Aurélie Herbelot and Marco Baroni. 2017. High-risk learning: acquiring new word vectors from tiny data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 304–309, Copenhagen, Denmark. Association for Computational Linguistics.

Felix Hill, Kyunghyun Cho, Anna Korhonen, and Yoshua Bengio. 2016. Learning to understand phrases by embedding the dictionary. *Transactions of the Association for Computational Linguistics*, 4:17–30.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. SimLex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Ziniu Hu, Ting Chen, Kai-Wei Chang, and Yizhou Sun. 2019. Few-shot representation learning for out-of-vocabulary words. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4102–4112, Florence, Italy. Association for Computational Linguistics.

Ozan İrsoy and Claire Cardie. 2014. Opinion mining with deep recurrent neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 720–728, Doha, Qatar. Association for Computational Linguistics.

Ganesh Jawahar, , Benoît Sagot, , and Djamé Seddah. 2019. What Does BERT Learn about the Structure of Language? In *Proceedings of the Conference of the Association for Computational Linguistics*, pages 3651–3657. Association for Computational Linguistics (ACL).

Nanjiang Jiang and Marie-Catherine de Marneffe. 2019a. Evaluating BERT for natural language inference: A case study on the CommitmentBank. 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 6086–6091, Hong Kong, China. Association for Computational Linguistics.

Nanjiang Jiang and Marie-Catherine de Marneffe. 2019b. Evaluating BERT for natural language inference: A case study on the CommitmentBank. 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 6086–6091, Hong Kong, China. Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665, Baltimore, Maryland. Association for Computational Linguistics.

Mikhail Khodak, Nikunj Saunshi, Yingyu Liang, Tengyu Ma, Brandon Stewart, and Sanjeev Arora. 2018. A la carte embedding: Cheap but effective induction of semantic feature vectors. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12–22, Melbourne, Australia. Association for Computational Linguistics.

Karin Kipper, Hoa Trang Dang, Martha Palmer, et al. 2000. Class-based construction of a verb lexicon. *AAAI/IAAI*, 691:696.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 2267–2273. AAAI Press.

Juan J. Lastra-Díaz, Josu Goikoetxea, Mohamed Ali Hadj Taieb, Ana García-Serrano, Mohamed Ben Aouicha, and Eneko Agirre. 2019. A reproducible survey on word embeddings and ontology-based methods for word similarity: Linear combinations outperform the state of the art. *Engineering Applications of Artificial Intelligence*, 85:645–665.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. *CoRR*, cs.CL/0205028.

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.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

R. Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference.

Alessio Miaschi, Dominique Brunato, Felice Dell'Orletta, and Giulia Venturi. 2020. Linguistic profiling of a neural language model.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

George A Miller and Walter G Charles. 1991. Contextual correlates of semantic similarity. *Language and cognitive processes*, 6(1):1–28.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of ACL-08: HLT*, pages 236–244, Columbus, Ohio. Association for Computational Linguistics.

Denis Paperno, Nghia The Pham, and Marco Baroni. 2014. A practical and linguistically-motivated approach to compositional distributional semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 90–99, Baltimore, Maryland. Association for Computational Linguistics.

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.

Mohammad Taher Pilehvar and Nigel Collier. 2017. Inducing embeddings for rare and unseen words by leveraging lexical resources. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 388–393, Valencia, Spain. Association for Computational Linguistics.

Victor Prokhorov, Mohammad Taher Pilehvar, Dimitri Kartsaklis, Pietro Lio, and Nigel Collier. 2019. Unseen word representation by aligning heterogeneous lexical semantic spaces. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6900–6907.

Roy Rada, Hafedh Mili, Ellen Bicknell, and Maria Blettner. 1989. Development and application of a metric on semantic nets. *Systems, Man and Cybernetics, IEEE Transactions on*, 19:17 – 30.

Kira Radinsky, Eugene Agichtein, Evgeniy Gabrilovich, and Shaul Markovitch. 2011. A word at a time: Computing word relatedness using temporal semantic analysis. pages 337–346.

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.

Leonardo Ranaldi, Francesca Fallucchi, and Fabio Massimo Zanzotto. 2022. Dis-cover ai minds to preserve human knowledge. *Future Internet*, 14(1).

Philip Resnik. 1995. Using information content to evaluate semantic similarity in a taxonomy. *arXiv preprint cmp-lg/9511007*.

Herbert Rubenstein and John B Goodenough. 1965. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633.

Timo Schick and Hinrich Schütze. 2019. Attentive mimicking: Better word embeddings by attending to informative contexts. 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 489–494, Minneapolis, Minnesota. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2020. BERTRAM: Improved word embeddings have big impact on contextualized model performance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3996–4007, Online. Association for Computational Linguistics.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566, Beijing, China. Association for Computational Linguistics.

Julien Tissier, Christophe Gravier, and Amaury Habrard. 2017. Dict2vec : Learning word embeddings using lexical dictionaries. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 254–263, Copenhagen, Denmark. Association for Computational Linguistics.

Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *J. Artif. Int. Res.*, 37(1):141–188.

Ludwig Wittgenstein. 1953. *Philosophical Investigations*. Wiley-Blackwell.

Chien-Sheng Wu, Steven Hoi, Richard Socher, and Caiming Xiong. 2020. Tod-bert: Pre-trained natural language understanding for task-oriented dialogue.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Zhibiao Wu and Martha Palmer. 1994. Verb semantics and lexical selection. In *32nd Annual Meeting of the Association for Computational Linguistics*, pages 133–138, Las Cruces, New Mexico, USA. Association for Computational Linguistics.

Dongqiang Yang and David Powers. 2006. Verb similarity on the taxonomy of wordnet. *Proceedings of the 3rd International WordNet Conference (GWC)*, pages 121–128.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2020. Xlnet: Generalized autoregressive pretraining for language understanding.

Fabio Massimo Zanzotto. 2019. Viewpoint: Human-in-the-loop Artificial Intelligence. *J. Artif. Intell. Res.*, 64:243–252.

Fabio Massimo Zanzotto and Lorenzo Dell'Arciprete. 2011. Distributed structures and distributional meaning. In *Proceedings of the Workshop on Distributional Semantics and Compositionality*, pages 10–15, Portland, Oregon, USA. Association for Computational Linguistics.

Fabio Massimo Zanzotto, Ioannis Korkontzelos, Francesca Fallucchi, and Suresh Manandhar. 2010. Estimating linear models for compositional distributional semantics. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1263–1271, Beijing, China. Coling 2010 Organizing Committee.

Y. Zhang. 2019. Bert for question answering on squad 2 . 0.

Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. *CoRR*, abs/1506.06724.