Computational Linguistics in the Netherlands Journal 11 (2021) 105-124 Submitted 10/2021; Published 12/2021

WN-BERT: Integrating WordNet and BERT for Lexical Semantics in Natural Language Understanding

Mohamed Barbouch M.BARBOUCH@UMAIL.LEIDENUNIV.NL
Suzan Verberne S.VERBERNE@LIACS.LEIDENUNIV.NL
Tessa Verhoef T.VERHOEF@LIACS.LEIDENUNIV.NL

Leiden Institute of Advanced Computer Sciences (LIACS), Niels Bohrweg 1, 2333 CA Leiden, Netherlands

Abstract

We propose an integration of BERT and WordNet to supplement BERT with explicit semantic knowledge for natural language understanding (NLU). Although BERT has shown its superiority in several NLU tasks, its performance seems to be relatively limited for higher level tasks involving abstraction and inference. We argue that the model's implicit learning in context is not sufficient to infer required relationships at this level. We represent the semantic knowledge from WordNet as embeddings using *path2vec* and *wnet2vec*, and integrate this with BERT both, externally, using a top multi-layer perceptron, and internally, building on VGCN-BERT. We evaluate the performance on four GLUE tasks. We find that the combined model gives competitive results on sentiment analysis (SST-2) and linguistic acceptability (CoLA), while it does not outperform the BERT-only model on sentence similarity (STS-B) and natural language inference (RTE). Our analysis of self-attention values shows a substantial degree of attention from WordNet embeddings to relevant words for the task.

1. Introduction

BERT (Bidirectional Encoder Representations from Transformers, Devlin et al. (2019)) has become the most popular paradigm for natural language modeling in recent years. This is mainly due to its superior performance in Natural Language Understanding (NLU), and Natural Language Processing (NLP) in general. Most of the research that has been done to uncover the secrets behind BERT's success have addressed syntax (Rogers et al., 2020). For example, the model is good at representing Parts-of-Speech (PoS), syntactic chunks and roles. However, (from the limited work addressing semantic characteristics) there are indications that the model performs relatively less well on semantic aspects and common knowledge, such as coreference (Balasubramanian et al., 2020), abstraction (Da and Kasai, 2019), reasoning (Forbes et al., 2019), and pragmatic inference (Ettinger, 2020).

We hypothesize that this has to do with the implicit nature of BERT's training process, with the use of self-supervision, while it lacks explicit – semantic – knowledge. During pre-training and fine-tuning, the model captures patterns from text itself by updating its internal structure weights, without exposure to any – explicit – rules about the language. Although it is a major advantage that the model learns largely independently of human supervision, in this way the model becomes dependent on input text, while inference is required for relationships that are not explicitly mentioned.

This shortcoming of Deep Learning models, and attention-(Transformer-)based models (Vaswani et al., 2017) in particular, is addressed by Lu et al. (2020) as a gap between *local* information, which is captured from text a model is trained on, and *global* information, which is general language knowledge that connects words to concepts. In places where literal words have an implicit higher abstracted meaning, a model like BERT may fail to capture this if it is trained on local context only and does not have access to higher concepts behind mentioned words. On the other hand, relying solely on global knowledge would not suffice for specific tasks, in which word order matters as well as content.

Our proposal is to enrich the local knowledge, which these types of models are good at, with global explicit language knowledge. In prior work, Lu et al. (2020) constructed a Vocabulary Graph Convolutional Network (VGCN) of word occurrences built from the downstream dataset and combined it with BERT for application to classification tasks, considering the VGCN part as 'global information' provision. However, in this method, the information extraction is still task-dependent. In our method, we introduce global information that exists externally, independent of the task itself, and that is more generic to have broad abstraction of information on the one hand, and provides coverage for the task when linked to its content on the other hand. Lu et al. suggested WordNet as future work for considering other types of vocabulary graphs. In fact, WordNet fits our criteria for 'global information' that is external, explicit and semantic in nature. WordNet is a lexico-semantic database (Miller et al., 1990) in which words are connected to each other hierarchically from more generic to more specific level, and vice versa. (More about this in Section 2.1.)

Thus, in this work we extend BERT's implicitly captured knowledge with *explicit* knowledge extracted from WordNet[1] (WN). We compare different models for representing WordNet information as embeddings. We then investigate how WordNet embeddings can be best combined with the BERT architecture, resulting in our proposed WN-BERT model. We evaluate the effectiveness of the combination on four NLU tasks. In addition, we carry out a more detailed analysis on the attention-heads of the combined model in order to analyze the contributions of the WordNet information to the BERT representations.

Our contributions are:

- We trained both path2vec (Kutuzov et al., 2019) and wnet2vec (Saedi et al., 2018) on the entire WordNet and make these models available for the NLP community.[2]
- We integrate explicit semantic knowledge from *WordNet* (Miller et al., 1990) with *BERT* (Devlin et al., 2019), building, in particular, on the suggested future work of Lu et al. (2020);
- We evaluate the created WordNet-BERT models on four GLUE tasks (Wang et al., 2018).

In the remainder of this paper, we discuss related work in Section 2. In Section 3 we describe our methods, consisting of the WordNet to embeddings conversion, integration with BERT, as well as mitigation of encountered limitations. We present our experimental results in Section 4, followed by a discussion of points that require additional attention and directions for future work in Section 5. Finally, we draw the conclusions from our findings in Section 6.

2. Related Work

BERT models have been highly successful on various NLU tasks since the initial publication in 2018.[3] The base-model turned out to be especially good at capturing syntactic knowledge, while having a bit more difficulty with word semantics and higher abstraction (Rogers et al., 2020; Clark et al., 2019; Ettinger, 2020). These language aspects require common knowledge that is not necessarily mentioned in text. We hypothesize that a supplement of explicit semantic knowledge would be an improvement for NLU.

[1] https://wordnet.princeton.edu/

[2] The models are available at: https://github.com/mbarbouch/WN-BERT.

[3] https://gluebenchmark.com/leaderboard

Since then, incorporating explicit information from external knowledge bases (KB) in large pre-trained language models has been receiving increasing attention. Verga et al. (2021) proposed Facts-as-Experts (FaE) building on Entity-as-Expert (EaE) (Févry et al., 2020), in which factual information from a symbolic KB is injected into BERT at inference time. They outperform EaE on FreebaseQA and WebQuestionsSP Q&A datasets by nearly 10 points. Yu et al. (2020) presented JAKET for joint pre-training of knowledge graphs (KGs) and language understanding. They improved the accuracy on the FewRel relation extraction dataset by about 2% points. Zhang et al. (2020) have achieved good improvements on classification and inference tasks by incorporating explicit contextual semantics from pre-trained semantic role labeling.

BERT has therefore widely extended after its introduction and has been offered in different combinations for different types of tasks. However, most research has focused on the syntactic model aspects, while the semantic aspects have not been looked at as broadly (Rogers et al., 2020). In addition, given the indications that the model does not always perform as well semantically, our focus in this study is on the semantic supplementation of BERT. As external resource to get the supplement from, we involve WordNet[1] (Miller, 1995) lexico-semantic database. This database had a prominent role in NLP research in the '90s and early 2000s. However, due to its limitations using 'traditional statistical methods' (Section 2.1), it was not developed further. With the latest successes of Deep Learning in NLP (especially of Transformers-based models like BERT, GPT-3 and T5), we aim to further explore WN's potential by injecting its explicitly constructed knowledge into BERT. To the best of our knowledge, WN has not yet been integrated in BERT to study its quality on semantically-based tasks.

2.1 WordNet

WordNet represents a hierarchical network of words with their semantic relationships by *synonymy*, *hypernymy*, *hyponymy*, *meronymy* and *antonymy*. The network includes *nouns*, *verbs*, *adjectives* and *adverbs* as parts-of-speech. These are categorized in sets of word senses and synonyms called *synsets*. Semantic similarities and relatedness between words can be determined by, for example, calculating the tree distance or gloss overlap between associated synsets (Pedersen et al., 2004). However, these methods are limited by word pairs computations, method constraints (e.g. assuming the 'is a' relationship in the Lowest Common Hypernyms (LCH) metric (Budanitsky and Hirst, 2001)), and same pair similarity when path length is equal Meng et al. (2013).

In this paper we propose to use WordNet as secondary external knowledge base, to provide BERT with explicit global semantic knowledge that BERT would lack in downstream tasks. As BERT encodes words as word embeddings, we first need to convert the WordNet representation to a similar vectorized format. In this, we rely on two existing methods: *path2vec* (Kutuzov et al., 2019) and *wnet2vec* (Saedi et al., 2018).

Path2vec re-encodes WordNet synsets as node embeddings, where the embeddings are learned as dense vectors based on pairwise synset similarities. A dot product is taken between pairs of corresponding synset node vectors, such that the value is close to a given ground truth similarity. These ground truths come from four graph distance measures, as implemented for WordNet in NLTK, i.e. Leacock-Chodorow (LCH), Jiang-Conrath calculated over the SemCor corpus (JCNS), Wu-Palmer (WuP), and Shortest path (ShP), in addition to a (fifth) user-defined similarity, i.e. the SimLex-999 gold standard. The evaluation was done by taking Spearman correlation between the estimated score and the ground truth. The best representations were found for ShP, reaching correlations up to 0.952 and 0.512 for WordNet similarities and SimLex-999, respectively. This model has also outperformed other methods like Node2Vec and Deepwalks. For application of the model to larger sequences of words with meaningful statements (e.g. sentences), where for a given word multiple synsets could be found, the model was further evaluated on SensEval and

SemEval Word Sense Disambiguation (WSD) tasks, achieving F1 scores between 0.50 to 0.55. Nonetheless, path2vec, in its initial form, is limited to support noun synsets only.

Wnet2vec builds on the intuition that the more edges between two synset nodes exist and the shorter the edges are, the stronger the semantic similarity between two words is. The model constructs an adjacency matrix of words where the cells are set to 1 if there is an edge between two word nodes, and 0 otherwise. Then the matrix vectors for each word are iteratively adapted until convergence to an inverted matrix, using Positive Point-wise Mutual Information (PMI) matrix transformation, in addition to $L2$ normalization, and Principal Component Analysis (PCA) for dimensionality reduction. The evaluation is done using six testsets, three of which for semantic similarity, i.e. SimLex-999, RG1965, and WordSim-353-Similarity; and three for semantic relatedness, i.e. WordSim-353-Relatedness, MEN, and MTurk-771. In contrast to path2vec, wnet2vec covers all the four parts-of-speech from WordNet, i.e. nouns, verbs, adjectives and adverbs. However, this method does not provide multiple embeddings for a word with multiple synsets (or meanings); instead, it expresses all synsets related to a given word in one single embeding vector. For this reason we call wnet2vec embedings, *WordNet word embeddings* – whereas we stick to *synset embeddings* for path2vec.

2.2 Embedding-based BERT ensembles

To combine our WordNet embeddings with BERT, we build on the work of Ostendorff et al. (2019) and Lu et al. (2020).

Ostendorff et al. (2019) enrich BERT in their approach with external author embeddings by adding a 2-layer multilayer perceptron (MLP) top-classifier. On a classification task of German books, they achieved up to 4.21 % points higher micro-F1 score than a BERT-German baseline.

VGCN-BERT, on the other hand, follows a completely different approach. To provide global information, Lu et al. (2020) construct a vocabulary graph (VG) of word co-occurrences and use a convolutional network to represent the information by embeddings. They concatenate these embeddings with BERT embeddings and feed them into BERT model starting from the first layer. This way local and global information would interact with each other through all layers. The authors presented slightly better F1-scores ($<$ 1% point) for sentiment analysis, binary single-sentence classification and hate speech compared to the baselines.

3. Method

In this section we describe our methodology. First, we describe the techniques used to transform WordNet to a suitable format for integration with BERT. Second, we determine WordNet coverage for datasets that will be used later on in the process for evaluation. Third, we go into the method for WordNet-BERT integration. In this, we distinguish between two approaches; one used externally, and the other internally.

3.1 From WordNet to Word Embeddings

BERT represents its textual knowledge by distributed representations of dense vectors, called *word embeddings.* Consistently, we convert WordNet information also to a dense embeddings format for the purpose of compatibility between the two models. In this work we rely on two prior methods for representing WordNet as embeddings, i.e.: path2vec (Kutuzov et al., 2019) and wnet2vec Saedi et al. (2018).

We use the path2vec model based on shortest paths as this yielded the best results in the original paper. The published model[4] contains embeddings of noun synsets only. These synsets cover both monosemous and polysemous words, meaning that multiple embeddings are available for words with multiple meanings. For selecting synset nodes related to our input words, we adjust the code used by path2vec for evaluation on SensEval and SemEval WSD tasks[5], such that any input text is accepted and all parts-of-speech are supported – instead of being dependent on the evaluated tasks and having support for nouns only.

In contrast to path2vec, the published pre-trained model of wnet2vec[6] covers all parts-of-speech supported by WordNet (i.e. nouns, verbs, adverbs and adjectives). However, due to memory limitations in the original paper, the model was trained to contain only about 60k word embeddings. In addition, this model compresses all synsets connected to unique words to single embeddings, resulting in the incapability of dealing with lexical ambiguity. For quick access, we first convert the output file of wnet2vec to a dictionary with the terms as keys and the embeddings as values. Initially, after tokenization of the input text, we retrieve the embeddings for each input token by looking for a match with terms in the dictionary.

3.2 WordNet Coverage

WordNet 3.0 covers with 117k synsets around 155k words.[7] This is a number far below the total number of English words, which is approximated between 600k (Oxford English Dictionary, 1989) and 1M (Michel et al., 2011). Using the initial pre-trained models of path2vec and wnet2vec, covering each 82k and 60k words respectively, we find that the word count coverage on the GLUE datasets is on average 27.6% in path2vec and 24.5% in wnet2vec (Table 1 **(a)**). Although path2vec only covered nouns, while wnet2vec also included verbs, adverbs and adjectives, path2vec found a little more synsets. There are two differences that affect the search results: the model size, and the way each model is queried. Path2vec finds the words through lemmas with NLTK's WordNet implementation[8], while we initially query wnet2vec by looking for exact word matches in the published embeddings file.[6] The reason for this is that wnet2vec is not prepared – by design – to support querying the output file by using NLTK's WordNet API, like path2vec does.

In order to increase the coverage, we retrain both WN models to cover all 155k words from WordNet. As adverbs and adjectives are disconnected, path2vec's vocabulary size only increased to 88k, comprising 75k noun[9] and 13k verb synsets. For wnet2vec, we also take lemmatization into account. If a word is not found directly in the provided file, we fall back to using lemma embeddings. We first perform a search through the `synsets()` function of NLTK's WordNet and, if any result is found, we retry to get the embeddings from wnet2vec.

After retraining both models and adding support for lemmas in wnet2vec, the total coverage increased on average to 42.2% in path2vec and 56.1% in wnet2vec, with a respective PoS coverage of 67.4% and 88.0% (Table 1 **(b)**).

3.3 WordNet–BERT

For integrating WordNet with BERT, we distinguish two types of ensembles: *'external combination'* and *'internal inclusion'*. The external approach combines the outputs of two independent models

[4] https://github.com/uhh-lt/path2vec/#pre-trained-models-and-datasets

[5] Starting from *def sentence_wsd(ids_list, sentences, poses)*: https://github.com/uhh-lt/path2vec/blob/master/wsd/graph_wsd_test_v2.py#L66

[6] https://github.com/nlx-group/WordNetEmbeddings

[7] https://wordnet.princeton.edu/documentation/wnstats7wn

[8] See *def synsets()* in: https://www.nltk.org/_modules/nltk/corpus/reader/wordnet.html

[9] The difference of 7k with the initial version is for disconnected noun synsets that are included with initial weights.

Table 1: WordNet coverage in % for used datasets.

		SST-2	CoLA	STS-B	RTE	avg.
		Initial WN models **(a)**				
p2vec	All	23.2	24.1	33.8	29.1	27.6
	PoS	37.2	40.3	52.0	46.0	43.9
wn2vec	Exact	26.3	24.5	24.1	23.2	24.5
	Lemma	38.2	43.2	38.8	39.7	40.0
	PoS	61.2	72.4	55.1	62.8	62.9
		After retraining the models **(b)**				
p2vec	All	**35.3**	**44.3**	**47.4**	**41.6**	**42.2**
	PoS	**56.6**	**74.1**	**72.9**	**65.8**	**67.4**
wn2vec	Exact	**45.7**	**38.9**	**38.7**	**37.5**	**40.2**
	Lemma	**56.3**	**55.1**	**57.2**	**55.7**	**56.1**
	PoS	**90.2**	**92.1**	**81.4**	**88.1**	**88.0**

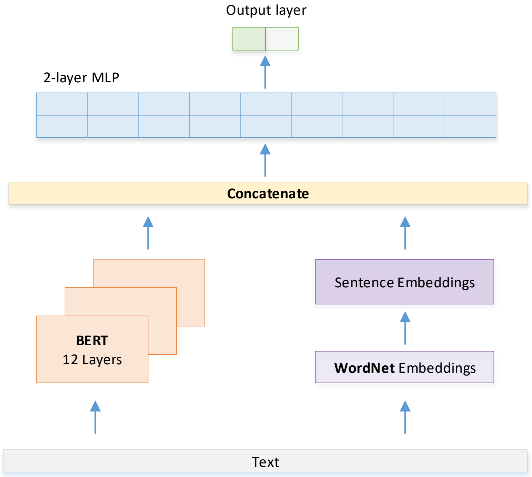


Figure 1: The model with external learning of embeddings combination using a 2-layer MLP top classifier.

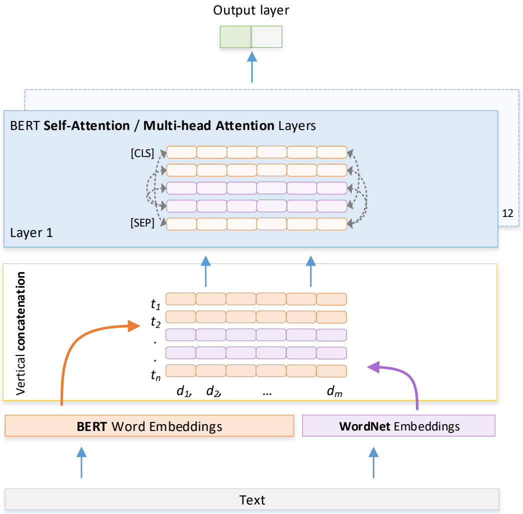


Figure 2: The model of internal inclusion integrating WordNet embeddings into BERT, based on the approach of VGCN-BERT (Lu et al., 2020).

in an additional 2nd level classifier, while internal inclusion incorporates the representation produced by one model into the internal architecture of the other one. We use the uncased $\text{BERT}_{\text{Base}}$ model with 12 self-attention layers.

3.3.1 External Combination

To combine embedding vectors on the outside, we follow the approach of Ostendorff et al. (2019). We concatenate BERT and WN embeddings to provide them as one input chain to the MLP.

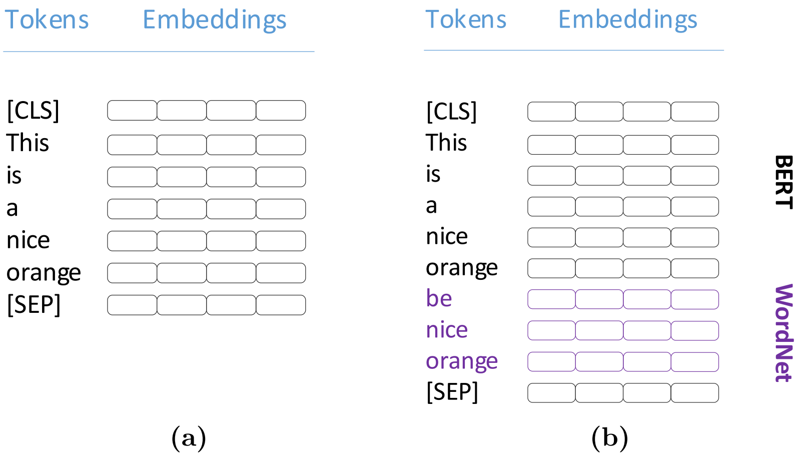


Figure 3: Vertical concatenation of BERT word embeddings **(a)** and WordNet embeddings **(b)** used in the internal ensemble. The simple input sentence "This is a nice orange." is used as example for illustration.

For this, we adjust the input layer, such that in addition to text input, also embeddings from WordNet are accepted. This is illustrated in Figure 1.

Sentence embedding representation. Both path2vec and wnet2vec provide the embeddings for individual words. However, keeping the embeddings in their raw format will lead to a horizontal dimensional explosion. We therefore suggest to reduce the dimensionality by converting the words to sentence embeddings.

Obtaining sentence embeddings can be accomplished in several ways, e.g., by concatenation, averaging (Coates and Bollegala, 2018) or meta-learning (Yin and Schütze, 2015). We combine averaging with meta-learning of the MLP, as the first does not suffer from dimensionality explosion, and the second learns the combination with BERT embeddings, making them, in contrast to horizontal concatenation at word level, a more efficient method.

Before concatenation, we first fine-tune the BERT embeddings in 3 epochs, as we found this to yield better performance in experimental runs. We then take the vectors from the [CLS] tag in the output layer, corresponding with all tokens produced by the WordPiece tokenizer, and average them to represent 'BERT sentence embeddings'. The [CLS] tag is used by BERT for classification tasks at the sentence level.

3.3.2 Internal Inclusion

In the internal approach, we include WN embeddings in BERT, similar to VGCN-BERT (Lu et al., 2020). The benefit this model has over external combination is the utilization of BERT's self-attention. This allows the included embeddings to influence the attention scores, which in turn also helps with interpretation of the importance of the input embeddings.

To include WN embeddings, we adjust the base model architecture, see Figure 2. First, we feed the input text with n tokens and get initial word embeddings of m dimensions from BERT and WordNet.[10] Then, we combine both and send the new representation through the network of BERT, starting from 'Embedding' (first) layer. Internally, we combine WN embeddings at token level, while utilizing the attention mechanism. For efficiency, we concatenate the embeddings

[10] BERT uses its own tokenizer; for WordNet we use NLTK's TreebankWordTokenizer.

vertically. Then, the whole combined chain of embeddings is fine-tuned at once. (See section 4.2 for experimental setup, e.g., of batch size and the number of epochs.)

For example, consider the sentence *"This is a nice orange"* for which we want to combine the embeddings for a classification task. BERT calculates the embeddings for the seven individual tokens {*[CLS], this, is, a, nice, orange, [SEP]*}, see Figure 3 **(a)**. Suppose that for the same sentence WordNet embeddings were found for only the words *'is'* (lemmatized as *'be'*), *'nice'* and *'orange'* (Fig. 3 **(b)**). The three vector embeddings can be simply added at the tail of word embeddings matrix −1. BERT will then treat these WordNet embeddings as part of the sentence.

In cases where an input consists of two sentences, the enrichment is implemented analogously as {*[CLS] [BERT_tokens_1st_sentence], [WN_tokens_1st_sentence] [SEP] [BERT_tokens_2nd_sentence], [WN_ tokens_2nd_sentence] [SEP]*}.

4. Evaluation

Given that we aim to complement BERT with external semantic knowledge from WordNet, for evaluation we consistently opt for tasks that are strongly semantically driven. We approach this from 'understanding' point of view, where meaning is crucial, which in turn relies heavily on semantics. One type of tasks that are suitable for this are Natural Language Understanding (NLU) tasks, on which BERT itself has also been evaluated (Devlin et al., 2019).

4.1 Datasets

We evaluate our model on the SST-2, CoLA, STS-B and RTE datasets from the General Language Understanding Evaluation (GLUE) (Wang et al., 2018) benchmark. These concern Sentiment Analysis, Linguistic Acceptability, Sentence Similarity and Natural Language Inference tasks. We consider the four datasets as relevant for evaluation of our methods, since sentiment is strongly dependent on adjectives, acceptability and inference on semantic relationships, and similarity on synonyms and semantic distance between words. All these aspects are covered by WordNet. Hence, BERT gets the potential to benefit from any additional semantic (global) knowledge that is missing locally.

SST-2. The Stanford Sentiment Treebank 2 (Socher et al., 2013) for binary sentiment classification is constructed on partial phrases of positive and negative movie reviews, labeled using Amazon Mechanical Turk. The original dataset contains 9,613 examples in total; 6,920 for training, 872 for dev, and 1,821 for test. The GLUE benchmark provides a much bigger version of the dataset, where the training-set is extended to 67,349 instances.[11] We will refer to the original SST-2 as 'SST-2' and to the GLUE version as 'SST-2 (GLUE)'. SST-2 is evaluated using weighted average F1-Score, SST-2 (GLUE) by accuracy.

CoLA. The Corpus of Linguistic Acceptability (Warstadt et al., 2019) consists of 9.5k text phrases that can be linguistically either correct or incorrect. The dataset is subdivided in 8,553 train, 277 dev, and 1,063 test phrases. However, as the labels for the testset are not publicly available, we use the dev set as testset, and optimize our training on 5% of the training set. The output is evaluated using Matthews correlation.

STS-B. The Textual Similarity Benchmark (Cer et al., 2017) provides a collection with 8,628 sentence pairs extracted from different text sources, divided into train, test and dev sets of sizes 5,749, 1,500 and 1,379, respectively. The corresponding task is to express the similarity

[11] After inspection we found (additional) partial phrases split from original sentences.

between two sentences on a scale of $[0, 5]$. The models are evaluated using Pearson and Spearman correlations with human scores.

RTE. The Recognizing Textual Entailment concerns textual entailment of sentence pairs collected from different NLP sources. GLUE combines RTE1, RTE2, RTE3 and RTE5 from 2006, 2007 and 2009 in one single dataset, where a pair is labelled as entailing or contradicting. The dataset contains 5.5k pairs in total – 2,494 for training, 277 for dev and 3,000 for test. Like CoLA, as test labels are not published, we use 5% of the training set for optimization, and the dev set for final evaluation. Model quality is evaluated by accuracy.

4.2 Experimental Setup

For the experimental setup we follow largely the same hyperparameter and training settings as used in prior work, i.e., Lu et al. (2020) for *internal inclusion* and Ostendorff et al. (2019) for *external combination*. Since BERT is not deterministic in each run, we perform 5 runs for each experiment and report averages. In the external approach, we first fine-tune BERT in 3 epochs before combining the embeddings. On average, 5 epochs were sufficient to optimize the training on the dev set. In the *internal inclusion* model, we use 3 epochs for SST-2 (GLUE), CoLA and STS-B, the same number as used by BERT (Devlin et al., 2019). For RTE we achieved better results with 9 epochs (the default of VGCN-BERT). The sentence pairs in STS-B and RTE led to a high dimensionality, making the 11GB GPUs that were used run out of memory. For this we reduced the batch size to 12. Furthermore, we set up the task of STS-B as regression with 1 output node, using ReLU activation function with MSE loss.

4.3 Results

The results are shown in Table 2. We take BERT and VGCN-BERT as baselines and compare our own combined models to their results. Besides using results as reported in reference papers (indicated in the table with ‘ref.’), we run both models ourselves with the same experimental setup for a fair comparison (‘ours’). STS-B and RTE tasks are excluded for the external combined model, as its architecture is not suitable for sentence-pair tasks.

The best scores of the integrated models are achieved with WN2V-BERT on sentiment analysis (SST-2 (GLUE)), improved by 0.29% point, and with P2V-BERT on linguistic acceptability (CoLA), improved by 3.36 correlation points. On sentence-pair tasks of sentence similarity (STS-B) and language inference (RTE), BERT remained better.

The scores of our implementation of the baselines (except CoLA) are on average slightly lower than reported in the respective papers. This could be a result of the random behavior of BERT and the average we take over multiple runs. On CoLA and RTE the scores are not comparable to prior work because we used the dev set as testset instead (see Section 4.1). For the BERT-only baseline model we did the experiments with the same architecture used for VGCN-BERT, but we only use BERT embeddings. This architecture, with the experimental setup from Section 4.2, could have influenced the lower score, but it has also made the comparison with our WordNet-BERT models more fair.

It is notably that our best two relative improvements, on CoLA and SST-2, were achieved with models following two different approaches, namely the one of external combination P2V-BERT and the one of internal inclusion WN2V-BERT, respectively. A possible explanation for this is that in the case of CoLA the complete sentence construction is more determinative for linguistic acceptability, while in case of SST-2, sentiment usually depends on individual adjectives (such as ‘good’, ‘bad’, ‘happy’, etc). In the external model we learn the combination at sentence level and internally at word level.

Table 2: Results for the four GLUE tasks. For the baseline, both the published results and the average result of our 5 runs are included. The results of the integrated models with WordNet – following *internal inclusion* and *external combination* strategies – follow below. (Standard deviation between brackets.). * As reported by Devlin et al. (2019); and ** by Lu et al. (2020).

	Model	SST-2	SST-2 (GLUE)	CoLA	STS-B	RTE
	Metric	F1	acc.	Matt. corr.	P/S corr.	acc.
Baseline	BERT (ref.)*	-	93.50	52.1	- / 85.80	66.4
	VGCN-BERT (ref.)**	91.93	-	-	-	-
	BERT (ours)	**91.56** (0.13)	92.94 (0.35)	58.35 (1.59)	**83.66** (0.22) / **82.55** (0.28)	**61.81** (2.14)
	VGCN-BERT (ours)	91.33 (0.15)	92.99 (0.19)	58.01 (1.61)	83.53 (0.24) / 82.32 (0.23)	51.70 (2.02)
Internal	P2V-BERT	91.22 (0.32)	92.92 (0.17)	56.68 (2.18)	82.98 (0.62) / 81.70 (0.69)	59.42 (1.67)
	P2V-VGCN-BERT	91.51 (0.29)	92.98 (0.20)	55.56 (1.09)	83.19 (0.36) / 81.98 (0.39)	58.34 (1.13)
	WN2V-BERT	91.36 (0.45)	**93.23** (0.37)	58.44 (1.43)	82.91 (0.29) / 81.65 (0.40)	60.36 (3.05)
	WN2V-VGCN-BERT	91.42 (0.29)	93.10 (0.11)	57.47 (1.64)	83.12 (0.33) / 81.95 (0.39)	59.42 (2.74)
Ext.	P2V-BERT	90.41 (0.29)	92.53 (0.19)	**61.71** (0.64)	-	-
	WN2V-BERT	90.34 (0.10)	92.51 (0.10)	61.01 (0.49)	-	-

4.4 Analysis

One of the challenges of (deep) neural networks nowadays is the explainability of the results obtained by the internal structure of the models, while they are a black-box for their workings. However, one approach that helps trace clues that lead to a given output is the inspection of the attention weights – given that BERT is built on the Transformers architecture, which is in its core attention-driven. Vig (2019) has already shown that different lexical patterns can be found in attention heads throughout the network layers of BERT (and GPT-2), e.g., patterns related to co-reference, subject-verb pairs, and dependency relations. In our case we inspect the attention weights of the SST-2 task, since this has shown the best result by the internally combined model. We use BertViz[12] for the visualization of the heads.

4.4.1 Local token attention

We examine the attention contribution of all tokens to the [CLS] token in the last layer. The embeddings of this token are used by BERT for final classification. We are especially interested in the differences in the [CLS] token with the addition of WordNet embeddings. The assumption is that any difference between the BERT-only model and the *BERT-WordNet* integrated model is due to the addition of WN embeddings. We first present examples where WN embeddings had positively influenced the final prediction, i.e., giving the expected label while BERT-only

[12] BertViz GitHub repository: https://github.com/jessevig/bertviz.

Table 3: Top 5 sentences with positive contribution from WordNet. In Label column you can see the expected sentiment, either positive = 1, or negative = 0, and in the columns to the right you can see the number of times each model got the prediction right.

Sentence	Label	BERT	WN2V-BERT	P2V-BERT
director rob marshall went out gunning to make a great one.	1	1/5	5/5	4/5
-lrb- howard -rrb- so good as leon barlow ... that he hardly seems to be acting.	1	2/5	5/5	5/5
a teasing drama whose relentless good-deed/bad-deed reversals are just interesting enough to make a sinner like me pray for an even more interesting, less symmetrical, less obviously cross-shaped creation.	1	2/5	4/5	4/5
not every animated film from disney will become a classic, but forgive me if i've come to expect more from this studio than some 79-minute after-school "cartoon".	0	1/5	5/5	1/5
home alone goes hollywood, a funny premise until the kids start pulling off stunts not even steven spielberg would know how to do.	0	2/5	5/5	4/5

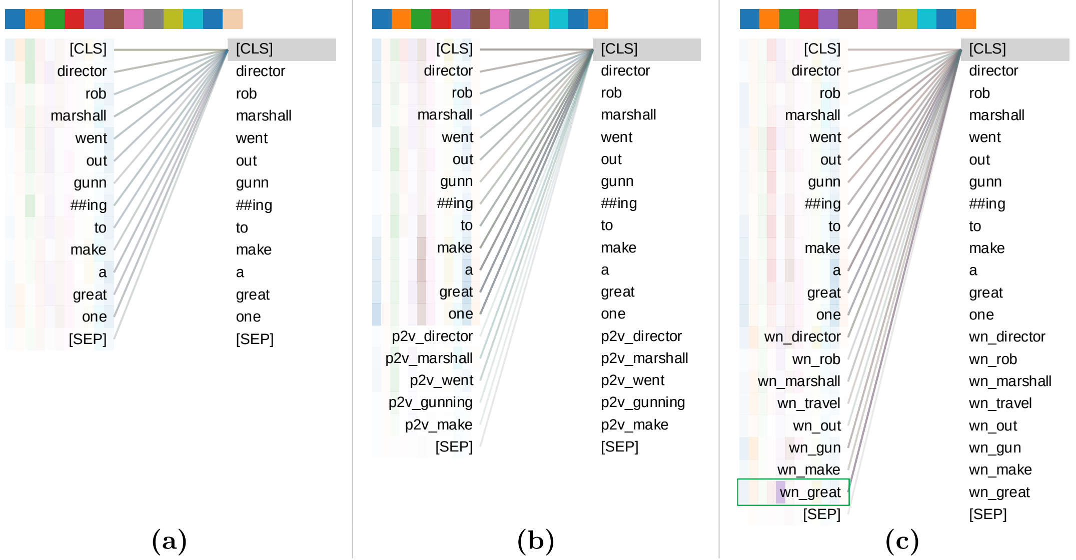


Figure 4: Token attention to the output [CLS] token for the first sentence from Table 3 that has correctly received a positive label by WN2V-BERT **(c)** and P2V-BERT **(b)** and was mislabeled by BERT-only **(a)**.

label was incorrect. Second, we show opposite cases where WN-BERT was deteriorating the BERT-only output. Running the model 5 times each, the models all agreed on the output in 91.7% of the cases, and differently disagreed on the rest. For the analysis we specifically look at the top disagreements, i.e. where there is the most deviation in the given results.

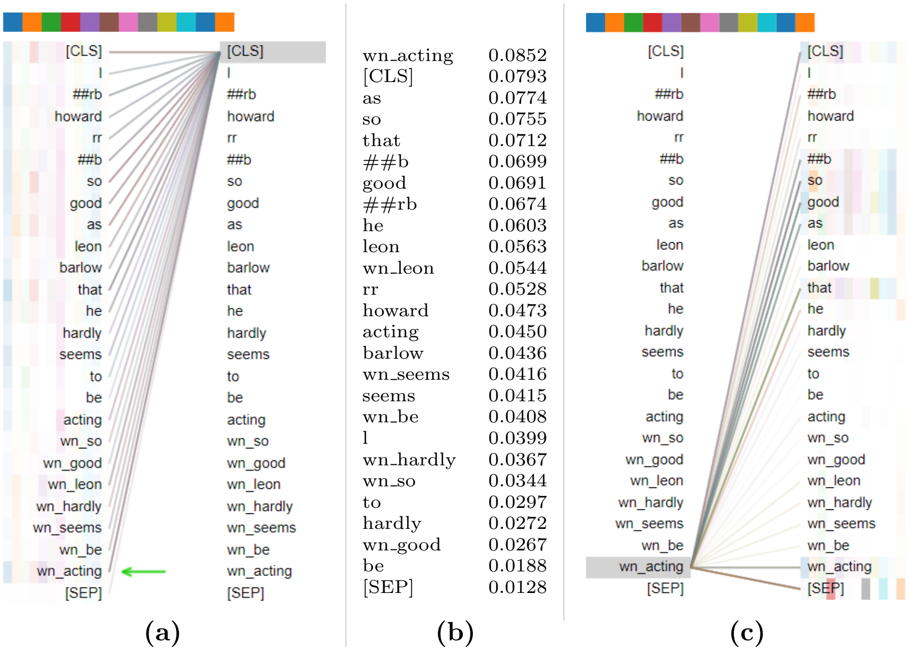
Positive Contribution Table 3 shows the top 5 sentences where there was a positive contribution to the final output from the WN embeddings. In the first example "*director rob marshall went out gunning to make a great one*", a *positive* sentiment label is expected. The BERT-only model classified this sentence as *negative* in 4 out of 5 cases, while WN2V-BERT and P2V-BERT were respectively in 5/5 and in 4/5 cases correct. The attention to [CLS] is visualized in Figure 4. This gives us an indication of the tokens importance for the [CLS] token. In the case of WN2V-BERT (c), we see that, in addition to BERT tokens, WN embeddings also give a reasonable color density towards [CLS]. This means that they also affect the output.

However, based on the coloring only, it is hard to tell which tokens provide more attention than others. For that reason, we quantify the attention weights for the sentence. We sum over all attention weights (of the 12 heads in the last layer) per token, normalize by the number of heads, and rank them by highest value. The resulting ranking is shown in Table 4. For BERT, [CLS] in itself is the most important token in this sentence. P2V-BERT only shows a different arrangement of BERT tokens. Interestingly, the wnet2vec token 'wn_great' turned out to provide the most attention in WN2V-BERT model. The adjective is very much in line with the expected positivity. From this point of view, therefore, we consider 'wn_great' to be the most influential token embedding to determine the positive sentiment for the given sentence.

In the second sentence *"-lrb- howard -rrb- so good as leon barlow ... that he hardly seems to be acting."* from Table 3 also a positive sentiment is expected. Here, the inference of 'good acting' is determinative. Again, WN2V-BERT (and P2V-BERT) succeeded to predict the correct label in each run, whereas BERT failed to pick up the right pattern in 3 runs. Looking at the visualization of the attentions in WN2V-BERT, in Figure 5 we can see a balanced contribution from the WN tokens **(a)**. While in BERT more density seems to come from the tokens [CLS], 'so', 'as', and 'that' – where the last three are quite neutral –, in the WN model, 'wn_acting'

Table 4: Ranked aggregated token attentions for the example from Fig. 4.

R	BERT		P2V-BERT		WN2V-BERT	
1	[CLS]	0.0678	one	0.0917	**wn_great**	**0.0866**
2	director	0.0548	[CLS]	0.0912	a	0.0767
3	one	0.0546	a	0.0911	wn_director	0.0665
4	marshall	0.0522	great	0.0894	great	0.0645
5	a	0.0496	make	0.0783	one	0.0642
6	great	0.0478	to	0.0579	to	0.0641
7	went	0.0452	director	0.0570	went	0.0598
8	rob	0.0435	rob	0.0543	out	0.0574
9	##ing	0.0435	out	0.0524	make	0.0573
10	out	0.0435	went	0.0505	wn_gun	0.0558
11	to	0.0421	##ing	0.0457	##ing	0.0521
12	make	0.0393	marshall	0.0455	[CLS]	0.0503
13	gunn	0.0317	gunn	0.0411	rob	0.0470
14	[SEP]	0.0267	p2v_marshall	0.0304	gunn	0.0457
15			p2v_went	0.0296	marshall	0.0453
16			p2v_gunning	0.0144	director	0.0440
17			[SEP]	0.0129	wn_marshall	0.0389
18			p2v_director	0.0126	wn_make	0.0353
19			p2v_make	'0.0117	wn_travel	0.0294
20					wn_rob	0.0262
21					wn_out	0.0213
22					[SEP]	0.0138



wn_acting	0.0852
[CLS]	0.0793
as	0.0774
so	0.0755
that	0.0712
##b	0.0699
good	0.0691
##rb	0.0674
he	0.0603
leon	0.0563
wn_leon	0.0544
rr	0.0528
howard	0.0473
acting	0.0450
barlow	0.0436
wn_seems	0.0416
seems	0.0415
wn_be	0.0408
l	0.0399
wn_hardly	0.0367
wn_so	0.0344
to	0.0297
hardly	0.0272
wn_good	0.0267
be	0.0188
[SEP]	0.0128

Figure 5: Attention to the output [CLS] token **(a)** for the second sentence from Table 3 using WN2V-BERT model. In the middle **(b)** the tokens are ranked by attention weights in descending order. On the right **(c)** the visual attention from the top-ranked token 'wn_acting' is shown.

provided the most attention **(b)**. Zooming in on the 'wn_acting' token to see which other tokens are receiving attention from it **(c)**, we see that it now focuses very much on the part 'so good as', where the other main term 'good' for determining the sentiment is also covered.

Negative Contribution The contribution of path2vec and wnet2vec embeddings is not always positive. In Table 5 we show 5 sentences from the top where WordNet is negatively influencing the output, i.e. providing a wrong sentiment when BERT is relatively often correct.

For example, as shown in Figure 6, P2V-BERT and WN2V-BERT assign, in contrast to the BERT-only model, a positive label, while a negative sentiment is expected. This example is tricky, as it consists of a positive first part and a negative second one. Path2vec with a very low attention contribution only seems to confuse the model here. Wnet2vec on the other hand

Table 5: Top 5 sentences with negative contribution from WordNet.

Sentence	Label	BERT	WN2V-BERT	P2V-BERT
and if the hours wins 'best picture' i just might.	1	5/5	0/5	2/5
rather quickly, the film falls into a soothing formula of brotherly conflict and reconciliation.	1	5/5	2/5	0/5
i kept thinking over and over again, 'i should be enjoying this.'	0	4/5	2/5	0/5
if melville is creatively a great whale, this film is canned tuna.	0	2/5	0/5	0/5
the whole damn thing is ripe for the jerry springer crowd.	0	2/5	0/5	0/5

gives the most attention to 'wn_tuna' (see Table 6) – in itself a neutral word –, for which it can be inferred from context that this combined with 'canned' gives a sarcastic negative sentiment. The negativity is determined by the word combination of "*canned tuna*". In both models this combination is not highlighted, while in BERT-only the words are among the top 5 tokens.

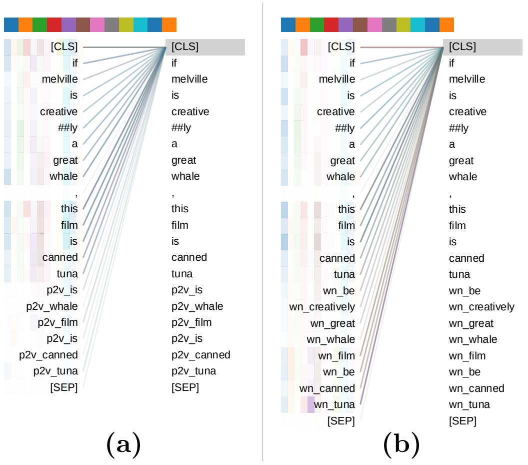


Figure 6: P2V-BERT and WN2V-BERT [CLS] token attention when a wrong output is given.

Table 6: Top tokens for the example from Fig. 6.

R	P2V-BERT		WN2V-BERT	
1	[CLS]	0.1141	wn_tuna	0.0865
2	this	0.0953	is	0.0834
3	film	0.0874	this	0.0799
4	is	0.0815	[CLS]	0.0762
5	whale	0.0723	wn_film	0.0648
6	if	0.0659	film	0.0608
7	canned	0.0596	wn_be	0.0503
8	tuna	0.0571	tuna	0.0498
9	great	0.0544	whale	0.0490
10	a	0.0504	canned	0.0487
11	##ly	0.0483	wn_canned	0.0474
12	creative	0.0417	##ly	0.0463
13	is	0.0409	if	0.0445
14	melville	0.0365	a	0.0396
15	p2v_film	0.0205	great	0.0389
16	p2v_is	0.0201	wn_be	0.0380
17	p2v_tuna	0.0156	is	0.0380
18	p2v_is	0.0141	melville	0.0294
19	[SEP]	0.0093	creative	0.0282
20	,	0.0083	wn_creatively	0.0281
21	p2v_canned	0.0069	wn_great	0.0253
22	p2v_whale	0.0065	wn_whale	0.0164
23			[SEP]	0.0075
24			,	0.0071

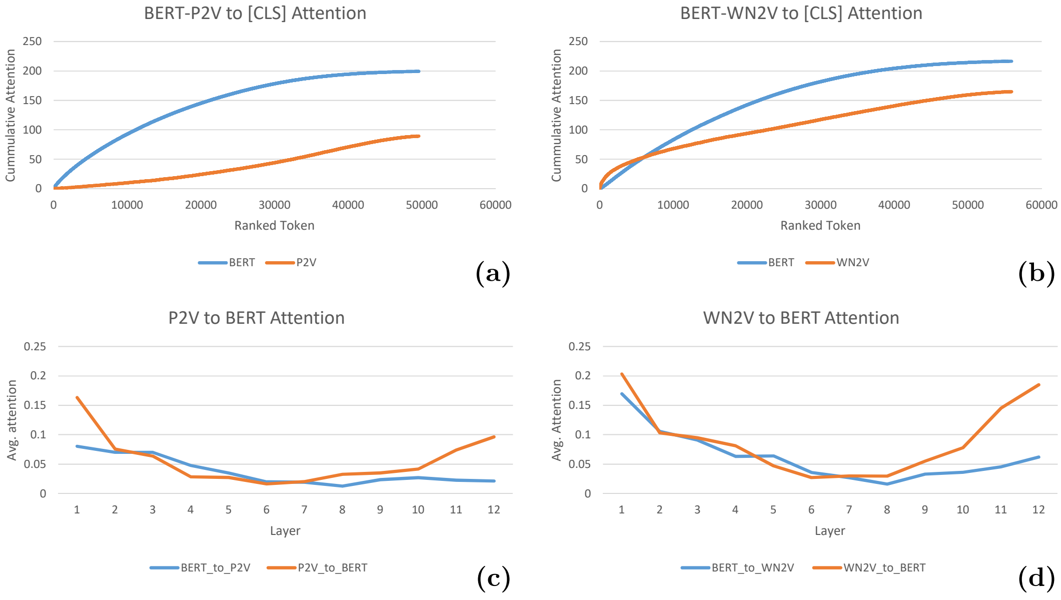


Figure 7: P2V- and WN2-BERT global cumulative token attention to [CLS] in the last layer for the SST-2 dataset ((**a**), (**b**)), in addition to mutual token attention through all layers ((**c**), (**d**)).

Table 7: Top ranked tokens in SST-2, using P2V-BERT and WN2V-BERT.

R	path2vec		wnet2vec	
	Unique	Avg.	Unique	Avg.
1	ridiculous	propelled	proves	wn.caricature
2	point	dreadful	wn.movie	wn.manhattan
3	extremely	confusing	wn.ridiculous	wn.table
4	a	brit	wn.caricature	wn.infomercial
5	extremely	extremely	wn.idea	wn.elsewhere
6	extremely	moderately	wn_boring	wn_Nash
7	confusing	model	wn_forgive	wn_security
8	dumb	##pha	wn_actress	wn_aberration
9	proves	substitutes	wn_more	wn_unmolested
10	good	##zard	wn_make	wn_bothersome
11	boring	guided	wn_confusing	substitutes
12	and	tremendous	wn_dumb	wn_evil
13	is	succeeds	wn_status	wn_drizzle
14	good	wildly	wn_clue	wn_unfaithful
15	movie	rates	wn_deeply	wn_frenetic
16	fails	diane	ridiculous	propelled
17	waste	##id	no	wn_egg
18	,	bears	wn_enjoy	wn_Pluto
19	,	quick	wn_manhattan	wn_journal
20	but	suffer	wn_better	wn_moot

4.4.2 Global token attention

To determine the overall attention contribution of path2vec and wnet2vec vs. BERT, we calculate this for all tokens in the testset and rank the values in descending order. The cumulative attention to [CLS] in the last layer is visualized in Figure 7 **(a)**, **(b)**. We can clearly see that the first 500 top tokens of wnet2vec provide more attention to the output than BERT, while path2vec lags far behind.

For the attention development across all layers, we reduce the attention weight per type of model to a single value, normalizing over all tokens, number of attention heads and model token ratio. The attention from and to WordNet and BERT embeddings is visualized in Figure 7 **(c)**, **(d)**. Here we see that both path2vec and wnet2vec give more attention to BERT than vice versa. With wnet2vec's highest attention in the last layers, its embeddings contribute more to the final output than path2vec's.

In Table 7 we show the top tokens for Fig. 7 **(a)**, **(b)** using *P2V-BERT* and *WN2V-BERT* models. We do this both for all tokens uniquely, and for tokens with weights averaged over all occurrences, when there are multiple occurrences found of the same token. Here the finding that wnet2vec tokens give much more attention than the path2vec tokens is confirmed. Some of the top words are sentiment words ('ridiculous, 'caricature'); other ('table', 'elsewhere') might be artefacts of small numbers of occurrences.

5. Discussion

While we have established a model integration where WordNet affects the functioning of BERT, the question remains why the presented results in Section 4.3 are overall (except for CoLA) not superior to BERT. There could be several reasons for this; we identified the following limitations.

(1) Incomplete WN coverage allowing gaps of unfound synsets to change meaning of sentences. With a WordNet coverage of 56% (including lemmas), we keep missing a lot of terms from the input text. Although the assumption is that the terms found are complementary, there are cases where the gaps can lead to completely different meanings, such as missing word negations. **(2) A limited F1 score for synset selection** in *path2vec* when dealing with Word Sense Disambiguation (WSD) for ambiguous words, e.g., for head, character, nature, etc. On average, in 43.7% of the cases more than 1 synset are found for a given token in all datasets (with standard deviation of 2.1%). Given an F1 score of path2vec of .555 for word sense disambiguation (WSD) (Kutuzov et al., 2019), there is great room for selecting synsets that could change the message in a piece of text. In addition, although the pruning thresholds set for neighborhood and similarity speed up the training process of path2vec enormously, they would exclude synsets that do not fall in the thresholds' range. However, Kutuzov et al. have shown that the overall impact is very small. A more concerning aspect is the disconnected synsets, which applies to all adjectives and adverbs and partly to nouns and verbs. **(3) Wnet2vec expresses all synsets related to a word in one single embedding.** This both avoids WSD as well as ensures that a searched word is in any case represented in the embeddings. However, irrelevant synsets are also included. Yet, the model appears to perform slightly better than path2vec.

In relation to limitation 1 and 2, relevant related work is Loureiro and Jorge (2019) in which results on state-of-the-art cross-domain tasks are improved using contextualized embeddings models: Based on sense-annotated corpora, a pre-trained language model (e.g. BERT) is used to get sense contextualized embeddings. Using a sense-annotated dataset like SemCor, tokens in text sequences are annotated with the senses from WordNet. During evaluation, for an input token embedding the closest neighbor is selected from the sense embeddings collection using k-NN. With the best model an F1 score of 77.0 is achieved on SemEval2015, improving 5.8 points on the best baseline score. In extended versions of the model, coverage from WordNet is gradually increased to full by including, not only senses, but also synsets, hypernyms, up to lexnames, respectively. For dealing with the limitations of path2vec this will help us to cover adjectives and adverbs as well, while having much better WSD. In relation to limitation 2 (WSD) in particular, for a given sentence with words to disambiguate, Huang et al. (2019) leveraged possible sense glosses[13] from WordNet for each ambiguous word and trained BERT (as GlossBERT) on multiple sentence-gloss pairs, with one pair containing the gloss mapping to the right sense label. On some evaluation sets, such as Senseval2 and SemEval2007, the results are even improved, reaching F1-scores up to 0.78. Moreover, Yap et al. (2020) presented yet a more interesting addition for integrating BERT with WordNet. Their approach is similar to GlossBERT, but instead of fine-tuning on sentence-gloss pairs as binary classification, in this model a relevance score is computed for each pair, using a single neuron linear output layer. Additionally, SemCor sense-annotated data is extended with training sentences extracted from WordNet. On SemEval-15 a new state-of-the-art F1 result of 84.4 is achieved. A potential solution direction addressing limitation 3 is that, when the gaps between input words and found synsets change the meaning of the input sequence, one could decide to leave out the supplement from WordNet for the given input, preventing negative influence on the output (as shown in Section 4.4.1).

A more general limitation of our approach comes from the challenge that sentiment words are strongly context-based. A word can be positive in one context and negative in another. By design BERT uses context and WordNet does not. However, the idea was that with the *internal inclusion* approach, WN embeddings would become contextualized as well. Given the attention contribution shown, this is very likely, but we only include WN embeddings during fine-tuning,

[13] Gloss is the formal term used by WordNet for a definition corresponding with one of its concepts, in this case with a sense.

while BERT is pre-trained with BERT tokens only. In a complete scenario, WN embeddings should also be included during pre-training.

Finally, in our experiments with the VGCN-BERT model, on average we got a relatively lower score for SST-2 (Table 2) than with BERT. This is in contrast to the result presented in the paper of Lu et al. (2020), where the model scores slightly higher than BERT. The difference we see is that we average over 5 runs, while the referenced work reports only one run. Based on our experiments, the VGCN results confirm that it allows for interaction of external embeddings with BERT word embeddings through the attention mechanism, but not that it necessarily leads to superior results. The latter can be tested again when the aforementioned limitations have been tackled.

6. Conclusions

In this paper, we have integrated BERT with WordNet for exploiting explicit lexical semantics and understanding their role in natural language understanding. We have represented semantic relationships in WordNet as synset embeddings using path2vec (Kutuzov et al., 2019), and as word embeddings using wnet2vec (Saedi et al., 2018). We have retrained these models on all WordNet synsets, increasing their coverage from 27.6% to 42.2% and from 24.5% to 56.1% respectively.

We find that *internal inclusion* of explicit semantic knowledge from WordNet in BERT gives competitive results on sentiment analysis (SST-2), and *external combination* gives better results on linguistic acceptability (CoLA). However, the model was not found to be outperforming on the sentence-pair tasks STS-B and RTE. The architecture should be revised for better support. Interestingly, analysing multi-head self-attentions of BERT has shown a substantial degree of attention from WordNet embeddings to BERT. Wnet2vec made the strongest contribution. The cases in which WordNet contributed positively are mainly positive or negative words that determine the sentiment in a sentence.

In conclusion, this work is a first result showing the possibilities and limitations of the combination of BERT and WordNet. The analysis of attention weights for the WordNet tokens shows that the learned representations are promising as a method for injecting external knowledge and influencing BERT's attention inside sentences. Nonetheless, our findings require subsequent research on increasing WN coverage and improving WSD.

References

Balasubramanian, Sriram, Naman Jain, Gaurav Jindal, Abhijeet Awasthi, and Sunita Sarawagi (2020), What's in a Name? Are BERT Named Entity Representations just as Good for any other Name?, *Proceedings of the 5th Workshop on Representation Learning for NLP*, Association for Computational Linguistics, Online, pp. 205–214. https://aclanthology.org/2020.repl4nlp-1.24.

Budanitsky, Alexander and Graeme Hirst (2001), Semantic Distance in Wordnet: An Experimental, Application-Oriented Evaluation of Five Measures, *Workshop on WordNet and other lexical resources*, Vol. 2, pp. 2–2.

Cer, Daniel, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia (2017), SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation, *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, Association for Computational Linguistics, Vancouver, Canada, pp. 1–14. https://www.aclweb.org/anthology/S17-2001.

Clark, Kevin, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning (2019), What Does BERT Look At? An Analysis of BERT's Attention, *BlackBoxNLP@ACL*.

Coates, Joshua and Danushka Bollegala (2018), Frustratingly Easy Meta-Embedding – Computing Meta-Embeddings by Averaging Source Word Embeddings, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, Association for Computational Linguistics, New Orleans, Louisiana, pp. 194–198. https://www.aclweb.org/anthology/N18-2031.

Da, Jeff and Jungo Kasai (2019), Cracking the Contextual Commonsense Code: Understanding Commonsense Reasoning Aptitude of Deep Contextual Representations, *Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing*, Association for Computational Linguistics, Hong Kong, China, pp. 1–12. https://aclanthology.org/D19-6001.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019), BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, *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)*, Association for Computational Linguistics, Minneapolis, Minnesota, pp. 4171–4186. https://www.aclweb.org/anthology/N19-1423.

Ettinger, Allyson (2020), What BERT Is Not: Lessons from a New Suite of Psycholinguistic Diagnostics for Language Models, *Transactions of the Association for Computational Linguistics* **8**, pp. 34–48, MIT Press, Cambridge, MA. https://aclanthology.org/2020.tacl-1.3.

Févry, Thibault, Livio Baldini Soares, Nicholas FitzGerald, Eunsol Choi, and Tom Kwiatkowski (2020), Entities as Experts: Sparse Memory Access with Entity Supervision, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, Online, pp. 4937–4951. https://aclanthology.org/2020.emnlp-main.400.

Forbes, Maxwell, Ari Holtzman, and Yejin Choi (2019), Do Neural Language Representations Learn Physical Commonsense?, *arXiv preprint arXiv:1908.02899*.

Huang, Luyao, Chi Sun, Xipeng Qiu, and Xuanjing Huang (2019), GlossBERT: BERT for Word Sense Disambiguation with Gloss Knowledge, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Association for Computational Linguistics, Hong Kong, China, pp. 3509–3514. https://aclanthology.org/D19-1355.

Kutuzov, Andrey, Mohammad Dorgham, Oleksiy Oliynyk, Chris Biemann, and Alexander Panchenko (2019), Learning Graph Embeddings from WordNet-based Similarity Measures, *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, Association for Computational Linguistics, Minneapolis, Minnesota, pp. 125–135. https://www.aclweb.org/anthology/S19-1014.

Loureiro, Daniel and Alípio Jorge (2019), Language Modelling Makes Sense: Propagating Representations through WordNet for Full-Coverage Word Sense Disambiguation, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Florence, Italy, pp. 5682–5691. https://www.aclweb.org/anthology/P19-1569.

Lu, Zhibin, Pan Du, and Jian-Yun Nie (2020), VGCN-BERT: Augmenting BERT with Graph Embedding for Text Classification, *European Conference on Information Retrieval*, Springer, pp. 369–382.

Meng, Lingling, Runqing Huang, and Junzhong Gu (2013), A Review of Semantic Similarity Measures in WordNet, *International Journal of Hybrid Information Technology* **6** (1), pp. 1–12.

Michel, Jean-Baptiste, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K. Gray, Joseph P. Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, Jon Orwant, Steven Pinker, Martin A. Nowak, and Erez Lieberman Aiden (2011), Quantitative Analysis of Culture Using Millions of Digitized Books, *Science* **331** (6014), pp. 176–182, American Association for the Advancement of Science. https://science.sciencemag.org/content/331/6014/176.

Miller, George A (1995), WordNet: a Lexical Database for English, *Communications of the ACM* **38** (11), pp. 39–41, ACM New York, NY, USA.

Miller, George A, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller (1990), Introduction to WordNet: An On-line Lexical Database, *International journal of lexicography* **3** (4), pp. 235–244, Oxford University Press.

Ostendorff, Malte, Peter Bourgonje, Maria Berger, Julian Moreno-Schneider, and Georg Rehm (2019), Enriching BERT with Knowledge Graph Embedding for Document Classification, *Proceedings of the GermEval 2019 Workshop*, Erlangen, Germany.

Oxford English Dictionary (1989), Oxford english dictionary, *Simpson, JA & Weiner, ESC*. https://thereaderwiki.com/en/Oxford_English_dictionary.

Pedersen, Ted, Siddharth Patwardhan, Jason Michelizzi, et al. (2004), WordNet:: Similarity-Measuring the Relatedness of Concepts., *AAAI*, Vol. 4, pp. 25–29.

Rogers, Anna, Olga Kovaleva, and Anna Rumshisky (2020), A Primer in BERTology: What We Know About How BERT Works, *Transactions of the Association for Computational Linguistics* **8**, pp. 842–866, MIT Press, Cambridge, MA. https://aclanthology.org/2020.tacl-1.54.

Saedi, Chakaveh, António Branco, João Rodrigues, and Joao Silva (2018), Wordnet Embeddings, *Proceedings of the third workshop on representation learning for NLP*, pp. 122–131.

Socher, Richard, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts (2013), Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank, *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Seattle, Washington, USA, pp. 1631–1642. https://aclanthology.org/D13-1170.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017), Attention is All you Need, *Advances in neural information processing systems*, pp. 5998–6008.

Verga, Pat, Haitian Sun, Livio Baldini Soares, and William Cohen (2021), Adaptable and Interpretable Neural MemoryOver Symbolic Knowledge, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, Online, pp. 3678–3691. https://aclanthology.org/2021.naacl-main.288.

Vig, Jesse (2019), A Multiscale Visualization of Attention in the Transformer Model, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Association for Computational Linguistics, Florence, Italy, pp. 37–42. https://www.aclweb.org/anthology/P19-3007.

Wang, Alex, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman (2018), GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding, *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Association for Computational Linguistics, Brussels, Belgium, pp. 353–355. https://www.aclweb.org/anthology/W18-5446.

Warstadt, Alex, Amanpreet Singh, and Samuel R. Bowman (2019), Neural Network Acceptability Judgments, *Transactions of the Association for Computational Linguistics* **7**, pp. 625–641. https://aclanthology.org/Q19-1040.

Yap, Boon Peng, Andrew Koh, and Eng Siong Chng (2020), Adapting BERT for Word Sense Disambiguation with Gloss Selection Objective and Example Sentences, *Findings of the Association for Computational Linguistics: EMNLP 2020*, Association for Computational Linguistics, Online, pp. 41–46. https://aclanthology.org/2020.findings-emnlp.4.

Yin, Wenpeng and Hinrich Schütze (2015), Learning Meta-Embeddings by Using Ensembles of Embedding Sets, *arXiv preprint arXiv:1508.04257*.

Yu, Donghan, Chenguang Zhu, Yiming Yang, and Michael Zeng (2020), JAKET: Joint Pre-training of Knowledge Graph and Language Understanding, *arXiv preprint arXiv:2010.00796*.

Zhang, Zhuosheng, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou (2020), Semantics-aware BERT for Language Understanding, *the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-2020)*.