Comparing Dependency-based Compositional Models with Contextualized Word Embeddings

Pablo Gamallo[a], Manuel de Prada Corral[b] and Marcos Garcia[c]
Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS),
Universidade de Santiago de Compostela, Galiza, Spain

Keywords: Compositional Distributional Models, Contextualized Word Embeddings, Transformers, Compositionality, Dependency-based Parsing.

Abstract: In this article, we compare two different strategies to contextualize the meaning of words in a sentence: both distributional models that make use of syntax-based methods following the Principle of Compositionality and Transformer technology such as BERT-like models. As the former methods require controlled syntactic structures, the two approaches are compared against datasets with syntactically fixed sentences, namely subject-predicate and subject-predicate-object expressions. The results show that syntax-based compositional approaches working with syntactic dependencies are competitive with neural-based Transformer models, and could have a greater potential when trained and developed using the same resources.

1 INTRODUCTION

A very important issue for the study of natural language semantics is to understand and formalize how the sense of a sentence is composed from the meaning of its constituent words. Compositionality, as defined in formal semantics, requires the notion of syntactic structure. More precisely, the Principle of Compositionality states that the meaning of a complex expression is a function of the meanings of the constituent words and of the way they are syntactically combined (Partee, 2007).

Many approaches dealing with compositional distributional semantics in the latest 10 years have made use of syntax-based models to build the meaning of complex expressions following the Principle of Compositionality (Baroni, 2013; Weir et al., 2016; Gamallo et al., 2019). In these approaches, there is an important interaction of meaning and context mediated through the syntactic structure. However, the most recent language models based on the Transformer architecture, such as BERT (Devlin et al., 2019) RoBERTa, (Liu et al., 2019), or DistilBERT (Sanh et al., 2020), do not make explicit use of syntactic information and, thereby, they do not follow the Principle of Compositionality as defined in formal semantics. Instead, they contextualize the sense of each word using distributional information extracted from the training corpus. There are very few works on syntax-augmented transformers incorporating dependency structure, but they raise some doubts regarding the viability of the use of syntax in basic Natural Language Processing applications and tasks such as information extraction (Sachan et al., 2020).

In recent years, the overwhelming use of Transformers and contextualization approaches to meaning construction has led to a decline in purely compositional models based on syntactic information and trained on parsed text. In fact, the main limitation of purely compositional approaches is that, in general, they are only able to work with controlled syntactic contexts: adjective-noun, subject-verb, subject-verb-object, etc. This is a strong limitation if we consider that most datasets built to measure the quality of detecting contextualized senses of words or complex meanings of sentences are not restricted to specific syntactic structures. They are constituted by free sentences or paragraphs with no specific syntactic structure, as the test sentences of the dataset provided by the *SemEval-2020 Shared Task 3 - Predicting the (Graded) Effect of Context in Word Similarity* (Armendariz et al., 2020), whose aim is to predict the degree of similarity of two words considering the context in which those words ap-

[a] https://orcid.org/0000-0002-5819-2469
[b] https://orcid.org/0000-0003-2731-079X
[c] https://orcid.org/0000-0002-6557-0210

Gamallo, P., Corral, M. and Garcia, M.
Comparing Dependency-based Compositional Models with Contextualized Word Embeddings.
DOI: 10.5220/0010391812581265
In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence (ICAART 2021) - Volume 2*, pages 1258-1265
ISBN: 978-989-758-484-8

pear. Another similar dataset is described in Pilehvar and Camacho-Collados (2019) for evaluating context-sensitive meaning representations.

To make a fair comparison between both syntax-based and transformer-based models, we will take advantage of syntactically controlled datasets containing subject-predicate and subject-predicate-object expressions. We will compare several settings of the following two main models: **(a) Transformers:** Contextualized vectors of constituent words generated by Transformer models (Devlin et al., 2019), and **(b) Compositional Approaches:** Compositional vectors generated by combining non contextual vectors following the syntactic restrictions that links the constituent words of a sentence (Gamallo, 2019).

There is a lot of context information encoded in the syntax that all these distributional-based Transformer models are ignoring. The objective of this paper is to highlight the true potential of syntax-driven compositional models and their competitiveness even in clear disadvantage with regard to the number of parameters required to train the models. Under these conditions, we will compare the performance of non-compositional Transformers with a compositional strategy based on syntactic dependencies.

The rest of the paper is organized as follows. The two strategies, in particular the compositional one, are introduced in Section 2. Experiments and comparative results are described and discussed in Section 3. Finally, conclusions are addressed in Section 4.

2 CONTEXTUALIZATION AND COMPOSITIONALITY

As mentioned above, contextualized embeddings based on Transformer architecture and compositional distributional models relying on syntactic dependencies are two different strategies to build the meaning of composite expressions and to deal with the representation of contextualized word senses. This section describes specific models of the two approaches.

2.1 Word Contextualization with Transformers

Transformers are the most popular implementation to build contextualized word embeddings. Transformer architecture is able to integrate word context thanks to only self-attention mechanism, dispensing with sequence-aligned recurrent or convolutional neural networks (Vaswani et al., 2017). The multi-headed attention mechanism is able to relate different word positions of a single sequence so as to compute the complex representation of the sequence.

Bidirectional Encoder Representations from Transformers, known as BERT (Devlin et al., 2019), is a bi-directional transformer-based language model learning information from left to right and from right to left. As any language model, it can be used to extract high quality language features from input text, but it can also be fine-tuned on specific NLP tasks such as entity recognition, classification, question answering, sentiment analysis, and so on. In the experiments described later, we will use BERT and family variations to extract both contextualized word embeddings and sentence embeddings from text in order to compute semantic similarity between complex expressions or sentences.

It is possible to generate context-sensitive vectors representing complex expressions or sentences using Transformers, even though recent research suggests that these representations do not capture high-level compositional information (Yu and Ettinger, 2020). In our experiments, two different Transformer techniques to generate the context-sensitive vector representing the meaning of a sentence will be explored:

Sentence Embeddings with Pooling Methods using SBERT (Reimers and Gurevych, 2019): it adds a pooling operation to the output of the Transformer to derive fixed sized sentence embeddings which are fine-tuned on sentence pairs from Natural Language Inference datasets. The default pooling strategy is to compute the mean of all output vectors. SBERT is the state-of-the-art strategy in several datasets requiring sentence similarity.

Contextualized Word Embeddings: Each transformer layer of 12-layer BERT-base model (or 24 in BERT-large and 6 in DistilBERT model) stands for a contextualized representation of a given word by putting the focus on different chunks of the input sequence. To elaborate the individual vectors of each word in context, we combine some of the 12 (6 or 24) layers of the deep neural network with the aim of finding the combination of layers that provides the best contextualization of each word in the sequence. By considering that the upper layers of contextualizing word models produce more context-specific representations (Ethayarajh, 2019) which are better suited to the purpose of the task at stake, we create contextualized vectors by combining the last four layers in two different ways: by summing or by concatenating. In our experiments, contextualized vectors of words will represent the meaning of the complex expression.

For those cases where the tokenizer separates a word into different sub-words (or affixes), we only consider the first one, which represent the lexical stem

of the full token.

It is worth pointing out that there is a controversy regarding the ability of Transformers to identify syntactic information. According to Rogers et al. (2020), syntactic structure is not directly encoded in weights generated by the self-attention mechanism. In fact, the predictions of BERT-like models are not altered even by the recurrent presence of syntactic problems in the input text, such as truncated sentences, shuffled word order, or removed subjects and objects (Ettinger, 2020). This suggests that the BERT's successful encoding of syntactic structure, as in Goldberg's work (Goldberg, 2019), does not indicate that it actually relies on that knowledge (Rogers et al., 2020), rather than in purely statistical coincidence.

2.2 Dependency-based Compositional Models

The first models to build the composite meaning of complex expressions were not compositional (Mitchell and Lapata, 2008, 2009, 2010), as they consisted of combining non contextual vectors of constituent words with arithmetic operations (addition or component-wise multiplication). By contrast, more recent distributional approaches directly rely on syntactic information and thereby follow the Principle of Compositionality. Some approaches develop sound compositional models of meaning where functional words are represented as high-dimensional tensors (Coecke et al., 2010; Baroni and Zamparelli, 2010; Grefenstette and Sadrzadeh, 2011b; Baroni, 2013). This idea is mostly based on Combinatory Categorial Grammar and typed functional application inspired by Montagovian semantics. However, there is an important issue concerning this strategy: it results in an information scalability problem, since tensor representations grow exponentially as the phrases grow longer (Turney, 2013).

Other compositional approaches, inspired by the work described in Erk and Padó (2008), take advantage of dependency analysis and the concept of selectional preferences. In these approaches, there is not a single meaning for a complex expression, but each constituent word is provided with a contextualized meaning built by considering its direct and indirect dependencies with the other constituents of the complex expression (Weir et al., 2016; Gamallo, 2017, 2019). A specific dependency-based strategy will be described in more detail in the following subsections. The dependency-based strategy will be compared with BERT-like models later in the experiments.

2.3 Compositional Operation

In our dependency-based approach, each syntactic dependency between two words is represented as a semantic compositional operation modeled by two specific functions, *head* and *dependent*, that take three arguments each:

$$head_{\uparrow}(r, \vec{x}, \vec{y}^{\circ}) \quad (1)$$

$$dep_{\downarrow}(r, \vec{x}^{\circ}, \vec{y}) \quad (2)$$

where $head_{\uparrow}$ and $dep_{\downarrow}$ represent the head and dependent functions, respectively, r is the name of the relation (nsubj, dobj, nmod, etc), and $\vec{x}$, $\vec{x}^{\circ}$, $\vec{y}$, and $\vec{y}^{\circ}$ stand for vector variables. On the one hand, $\vec{x}$ and $\vec{y}$ represent the denotation of the head and dependent words, respectively. They can be represented by means of standard word vectors derived from non-contextual embeddings. On the other hand, $\vec{x}^{\circ}$ represents the selectional preferences imposed by the head, while $\vec{y}^{\circ}$ stands for the selectional preferences imposed by the dependent word. Selectional preferences are dynamically constructed vectors and the way they are constructed is defined by using an specific example as follows:

Consider a specific dependency relation, nominal subject (*nsubj*), holding between lemma *cat*, the dependent, and *chase*, the head, which is the partial analysis of the composite *The cat chased*[1]. The application of the two functions consists of combining (either multiplying or adding) the non contextual vectors with the selectional preferences, by taking into account the *nsubj* relation:

$$head_{\uparrow}(nsubj, \vec{chase}, \vec{cat}^{\circ}) = \vec{chase} \odot \vec{cat}^{\circ} = \vec{chase}_{nsubj\uparrow} \quad (3)$$

$$dep_{\downarrow}(nsubj, \vec{chase}^{\circ}, \vec{cat}) = \vec{cat} \odot \vec{chase}^{\circ} = \vec{cat}_{nsubj\downarrow} \quad (4)$$

Each combinatorial operation (component-wise vector multiplication in Equation 4) results in a compositional vector which represents the contextualized sense of one of the two words (the head or the dependent). Here, $\vec{cat}^{\circ}$ and $\vec{chase}^{\circ}$ are selectional preferences resulting from the following vector additions:

$$\vec{cat}^{\circ} = \sum_{\vec{w} \in \boldsymbol{V}_{cat/nsubj}} \vec{w} \quad (5)$$

$$\vec{chase}^{\circ} = \sum_{\vec{w} \in \boldsymbol{N}_{nsubj/chase}} \vec{w} \quad (6)$$

where $\boldsymbol{V}_{cat/nsubj}$ is the vector set of those verbs having *cat* as subject. More precisely, given the linguistic context $< nsubj_{\downarrow}, cat >$, the dynamically constructed vector $\vec{cat}^{\circ}$ is obtained by adding the vectors

[1]Function words such as determiners and auxiliar verbs are not considered in this compositional approach. Only lexical words are taken into account.

$\{\vec{w}|\vec{w} \in \boldsymbol{V}_{cat/nsubj}\}$ of those verbs (*eat*, *jump*, etc) that are combined with the noun *cat* in that syntactic context. In more intuitive terms, $\vec{cat}^{\circ}$ stands for the inverse selectional preferences imposed by *cat* on any verb at the subject position.

On the other hand, $\boldsymbol{N}_{nsubj/chase}$ in equation 6 represents the vector set of nouns occurring as subjects of *chase*. Given the lexico-syntactic context $< nsubj_{\uparrow}, run >$, the vector $\vec{run}^{\circ}$ is obtained by adding the vectors $\{\vec{w}|\vec{w} \in \boldsymbol{N}_{nsubj/chase}\}$ of those nouns (e.g. *tiger*, *hunter*, etc) that might be at the subject position of the verb *chase*. The dinamically constructed vector $\vec{chase}^{\circ}$ stands for the selectional preferences imposed by the verb on any noun at the subject position.

2.4 Incremental Composition

Given the above definition of dependency-based composition, the iterative application of the syntactic dependencies found in a sentence or complex expression is modelled as the recursive and compositional process of constructing the contextualized sense of all the constituent words. This incremental and recursive process may go in two directions: from left-to-right and from right-to-left.

Let us take the expression *The cat chased a mouse*. The dependency-by-dependency functional application from left-to-right results in the following three contextualized word senses: $\vec{cat}_{nsubj\downarrow}$, $\vec{chase}_{nsubj\uparrow+dobj\uparrow}$ and $\vec{mouse}_{nsubj\downarrow+dobj\downarrow}$. They all together represent the meaning of the sentence in the left-to-right direction. Notice that $\vec{cat}_{nsubj\downarrow}$ is not a fully contextualized vector: it was only contextualized by the verb, but not by the direct object noun. In order to fully contextualize the subject, we need to initialize the composition process in the other way around: from right-to-left.

3 EXPERIMENTS

To compare the performance on models based on transformers (e.g. BERT-like models) with compositional approaches in the task of building the meaning of contextualized words, we are required to use datasets of expressions with controlled syntactic patterns. We need this type of datasets because syntax-based compositional approaches are not mature enough to deal with expressions of any type and size.

In the experiments, we used two versions of the dependency-based compositional approach:

comp_explicit: It relies on a count-based distributional model with context filtering. The model is provided with explicit dependencies extracted with DepPattern (Gamallo and Garcia, 2018) and only the more relevant contexts per word are considered. Compositional operation is implemented with component-wise multiplication.

comp_embed: The distributional model consists of word embeddings built with *word2vec*, configured with CBOW algorithm, window of 5 tokens, negative-sampling parameter of 15, and 300 dimensions (Mikolov et al., 2013). Compositional operation is implemented with component-wise vector addition. Preliminary experiments led us to the conclusion that vector addition works better than multiplication for this type of distributional model.

In both cases, the distributional models were built from the English Wikipedia (dump file of November 2019[2]), containing over 2,500M words.

Concerning the Transformers architecture, we made use of the *large* and *base* BERT variants and two BERT-based models:

bert-large with 24 layers, 335M parameters and trained on lower-cased English text.

bert-base with 12 layers, 110M parameters and trained on lower-cased English text.

roberta with 12 layers and 125M parameters (Liu et al., 2019).

distilbert with 6 layers and 66M parameters (Sanh et al., 2020).

All approaches were evaluated against two datasets: one with Noun-Verb (i.e. subject-predicate) expressions and the other with Noun-Verb-Noun (subject-predicate-object) expressions.

3.1 Subject-predicate Dataset: NV Expressions

The test dataset by Mitchell and Lapata (2008) comprises 120 different pairs of similar expressions evaluated by 30 humans, totalling 3,600 human similarity judgments. Each pair consists of an intransitive verb and a subject noun (NV expression), which is compared to another NV pair combining the same noun with a synonym of the verb. For instance, *"thought stray"* is related to *"thought roam"*, being *roam* a synonym of *stray*. To evaluate the results of the targeted systems, the harmonic mean[3] of two correlations (Spearman and Pearson) is computed between individual human similarity scores and the systems' predictions (cosine similarity) as in SemEval-2017

[2]https://dumps.wikimedia.org/enwiki/

[3]In general, harmonic mean is more robust to compute the average of the Spearman and Pearson correlations. However, if they are not both positive or negative, standard mean should be used instead, marked with an asterisk.

Table 1: Left: Mean of Spearman and Pearson correlations with intransitive expressions (NV) using the benchmark by Mitchell and Lapata (2008). Right: Correlation with transitive expressions (NVN) using the benchmark by Grefenstette and Sadrzadeh (2011). To allow comparison with previous approaches, we put Spearman values in brackets.

Models	ρ
nocomp_explicit - sentence	18.50
nocomp_emb - sentence	3.25
nocomp_explicit - head	8.37
nocomp_embed - head	21.29
comp_explicit - sentence	**32.22** (31.77)
comp_explicit - head	25.80
comp_explicit - dep	29.21
comp_embed - sentence	22.00
comp_embed - head	9.23
comp_embed - dep	5.32
Eck and Padò (2008)	(27)
Dinu et al. (2013)	(26)
Human	66

Models	ρ
nocomp_explicit - sentence	21.57
nocomp_emb - sentence	28.18
nocomp_explicit - head	8.37
nocomp_embed - head	35.49
comp_explicit - sentence (average)	44.80
comp_explicit_left-to-right - sentence	**46.79** (45.72)
comp_explicit_left-to-right - head	34.62
comp_explicit_left-to-right - dep	20.55
comp_explicit_right-to-left - sentence	36.17
comp_explicit_right-to-left - head	36.68
comp_explicit_right-to-left - dep	41.95
comp_emb - sentence (average)	37.59
comp_emb_left-to-right - sentence	34.78
comp_emb_left-to-right - head	29.98
comp_emb_left-to-right - dep	20.88
comp_emb_right-to-left - sentence	37.18
comp_emb_right-to-left - head	29.98
comp_emb_right-to-left - dep	36.06
Grefenstette and Sadrzadeh (2011)	(28)
Hashimoto and Tsuruoka (2014)	(43)
Polajnar et al. (2015)	(35)
Human	74

Table 2: Left: Mean of Spearman and Pearson correlations with intransitive expressions (NV) between the benchmark by Mitchell and Lapata (2008) and different BERT-based approaches. Right: Correlation with transitive expressions (NVN) between the Grefenstette and Sadrzadeh (2011) benchmark and different versions of BERT.

Models	ρ
bert-large - sentence	**32.12** (31.52)
bert-base - sentence	11.59
roberta - sentence	24.83
distilbert - sentence	2.72
bert-large - head (sum)	−11.03
bert-base - head (sum)	−7.75
roberta - head (sum)	10.40
distilbert - head (sum)	−11.23
bert-large - dep (sum)	14.44
bert-base - dep (sum)	7.23
roberta - dep (sum)	14.43
distilbert - dep (sum)	−5.52
bert-large - head (concat)	−11.68
bert-base - head (concat)	−7.78
roberta - head (concat)	9.94
distilbert - head (concat)	−11.45
bert-large - dep (concat)	14.50
bert-base - dep (concat)	6.04
roberta - dep (concat)	14.32
distilbert - dep (concat)	−4.60
Human	66

Models	ρ
bert-large - sentence	**56.46** (61.18)
bert-base - sentence	49.06
roberta - sentence	46.12
distilbert - sentence	38.38
bert-large - head (sum)	35.21
bert-base - head (sum)	31.05
roberta - head (sum)	11.51
distilbert - head (sum)	33.73
bert-large - dep-subj (sum)	9.29
bert-base - dep-subj (sum)	4.07
roberta - dep-subj (sum)	10.23
distilbert - dep-subj (sum)	−0.54*
bert-large - dep-obj (sum)	19.88
bert-base - dep-obj (sum)	4.24
roberta - dep-obj (sum)	3.85
distilbert - dep-obj (sum)	6.61
bert-large - head (concat)	34.48
bert-base - head (concat)	30.34
roberta - head (concat)	11.39
distilbert - head (concat)	32.81
bert-large - dep-subj (concat)	9.48
bert-base - dep-subj (concat)	4.89
roberta - dep-subj (concat)	10.15
distilbert - dep-subj (concat)	−0.09*
bert-large - dep-obj (concat)	20.36
bert-base - dep-obj (concat)	4.13
roberta - dep-obj (concat)	4.74
distilbert - dep-obj (concat)	8.08
Human	74

Task 2 (Camacho-Collados et al., 2017), by using the evaluation script provided in that shared task.

We compare three types of context-sensitive similarities (between pairs of NV sentences):

sentence: Each composite expression or sentence is associated with a single vector provided with a fixed size, and built from the contextualized vectors of its word constituents. Similarity is computed between the two vectors, one per sentence. In the compositional approach, this vector is just the average addition of its constituents. In the BERT-like approaches, we used SBERT to elaborate each sentence embedding (Reimers and Gurevych, 2019).

head: Similarity is computed between contextualized vectors, namely the head vectors of each expression. For instance, we compute the similarity between *eye flare vs eye flame* by comparing the verbs *flare* and *flame* after being contextualized by the subject noun. In the BERT-like approaches, contextualized vectors are built in two different ways: by adding the last 4 layers (sum) or by just concatenate them (concat).

dependent: Similarity is computed between the dependent vectors of each expression after having been contextualized by the corresponding verb. E.g., we compute the similarity between *eye flare vs eye flame* by comparing the noun *eye* in both contexts. As in the head-based similarity, BERT-like approaches are built with both addition and concatenation.

Table 1 (left side) shows the mean of Spearman and Pearson correlation values (ρ) for intransitive expressions (NV) using the benchmark by Mitchell and Lapata (2008). Non-compositional baselines are shown in the first rows. The sentence-based non-compositional strategy builds the meaning of each expression by adding the constituent vectors, while the head-based non-compositional approach computes similarity just on the basis of the head verb of each NV expression. Similarity between dependent words is not considered as the nouns of each NV pair are identical.

In the next rows, Table 1 shows the results obtained by the two configurations (explicit and embeddings) of our compositional strategy. Let us note that the best scores are achieved by averaging both head and dependent contextualized vectors with explicit vectors and embeddings: 32.22 and 22.00, respectively. In all system configurations, explicit count-based vectors outperform embeddings, which are predictive vector models. We put in brackets Spearman correlation values. The best system (*comp_explicit - sentence*) achieves 31.77 correlation, which outperforms the Spearman score reported in Dinu et al. (2013) using a corpus consisting of about 2.8 billion tokens merging Wikipedia, BNC and a ukWaC (Baroni et al., 2009). The highest score by *comp_explicit sentence* also improves all BERT-like configurations reported in Table 2, where the best system is a sentence-based configuration, namely *bert-large - sentence*: 32.12 correlation.

3.2 NVN Composite Expressions

The second experiment consists of the same evaluation task as in the previous subsection but performed on transitive sentences (NVN). The test dataset is described in Grefenstette and Sadrzadeh (2011a) and was built using the same guidelines as Mitchell and Lapata (2008). Given the dependency-based compositional strategy (*comp_explicit* and *comp_embed*), it is possible to compositionally build several vectors that somehow represent the compositional meaning of the whole NVN sentence. Take the expression *"the coach runs the team"*. If we follow the left-to-right strategy, at the end of the compositional process, we would obtain two fully contextualized senses:

left-to-right head. The sense of the verbal head *run*, as a result of being contextualized first by the preferences imposed by the subject and then by the preferences required by the direct object.

left-to-right dep. The sense of the direct object *team*, as a result of being contextualized by the preferences imposed by *run* previously combined with the subject *coach*. In the left-to-right direction the object is fully contextualized by the verb and the subject; by contrast, the subject is not contextualized by the object, so that this partially contextualized sense of the subject is not used to represent the sentence.

If we follow the right-to-left strategy, at the end of the compositional process, we also obtain two fully contextualized senses:

right-to-left head. The sense of the head *run* as a result of being contextualized first by the preferences imposed by the object and then by the subject.

right-to-left dep. The sense of the subject *coach*, as a result of being contextualized by the preferences imposed by *run* previously combined with the object *team*. Following this direction, the object is not contextualized by the subject.

Table 1 (right side) shows the results of the dependency-based compositional methods, both *comp_explicit* and *comp_embed*, as well as several non-compositional baseline strategies, namely only head vectors and non-compositional vector addition. The best configuration is the addition of the contextualized head and dependency words in the right-to-left strategy with explicit count-based vectors (*comp explicit left-to-right - sentence*), which reaches 46.79 correlation. To the best of our knowledge, this value is three points higher than the best compositional system on this dataset (Hashimoto et al., 2014). As in the previous experiment with intransitive expressions, explicit count-based vectors outperform predicted-based word embeddings. Let us note that the left-to-right strategy seems to build less reliable compositional vectors than the right-to-left counterpart in this specific dataset. This might be due to the weak semantic motivation of the selectional preferences involved in the subject dependency of transitive constructions in comparison to the direct object.

In addition to the fully contextualized words, we also build three global senses of the sentence, which are the addition of the head and dep left-to-right and right-to-left values, as well as the final average sum of these two additions. It is worth mentioning that the best fully contextualized word is the subject noun generated with the right-to-left algorithm (*right-to-left dep*: 41.95 in *comp_explicit*), which outperforms the two contextualized verb senses, both *left-to-right head* and *right-to-left head*. This result was not expected as the sense of the root verb should be better positioned to represent the core meaning of the sentence. However, the fact that the subject noun works so well is conceptually possible since any fully contextualized vector may represent the meaning of the whole sentence from a specific point of view.

The score value obtained by *right-to-left sentence* strategy outperforms other systems tested for this dataset: e.g., Grefenstette and Sadrzadeh (2011b) and Polajnar et al. (2015) (based on the categorical compositional distributional model of meaning of Coecke et al. (2010)), and also the neural network strategy described in Hashimoto and Tsuruoka (2015).

Concerning the BERT-like configurations, Table 2 (right side) includes not only the contextualized vector of the head and the subject (*dep_subject*), but also the contextualized vector of the direct object (*dep_object*) as all constituent words are fully contextualized in any Transformer architecture Table 2 shows that the sentence-based algorithm is again the best strategy to grasp the meaning of NVN expressions. As in the previous dataset, the best correlation is achieved with bert-large: 56.46 (and 61.18 Spearman correlation), which is by far, to the best of our knowledge, the highest correlation value reported on this dataset. Another relevant observation is the fact that there are no significant differences between adding or concatenating the last layers to build contextualized vectors. This is true for the values shown in both sides of Table 2.

An interesting further analysis is to compare the correlation scatter plot of the best Transformer and compositional-based configurations. Figure 1, where the similarity scores have been standardized, shows how Transformer models tend to overestimate similarity of the sentences (left side), producing greater errors when sentences are not similar. Conversely, compositional-based models have a more scattered and less biased error distribution. That dispersed er-

ror might be the result of limitations of the model and training resources, but the underlying semantics encoding seems to be powerful and less biased.[4]

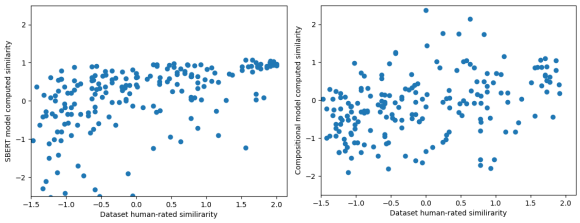

Figure 1: SBERT (left) and compositional (right) computed similarity scatter plots.

4 CONCLUSIONS

The fully compositional method based on transparent vectors and syntactic dependencies turns out to be competitive with regard to BERT-like configurations, even if the SBERT strategy using BERT-large as pre-trained model achieves the best correlation values among all configurations. It should be noted that the results of the compositional method have been obtained without requiring neural network architecture.

Another noteworthy characteristic of the compositional method is the fact that it is made up of transparent vectors. Transparency makes it possible to trace with some ease which syntactic contexts (and therefore linguistic features) are most relevant in the construction of the compositional vectors.

However, the main weakness of the compositional method is its dependence on syntactic parsing, which is an important source of errors. Likewise, another weakness of this method is the increasing difficulty to build compositional vectors of open sentences with multiple dependencies of different types. Finally, the compositional approach does not consider the difference between fully compositional expressions from non-compositional or even partially compositional, even though recent research suggests that neural-based representations are not able to correctly model semantic compositionality (Yu and Ettinger, 2020).

In future work, we will study different combinatorial mechanisms by distinguishing full compositionality from non-compositional expressions, and also by considering several degrees of partial compositionality. We will also design a strategy to build fully contextualized vectors for open sentences with whatever syntactic structure by dynamically interpreting words and their selectional restrictions. This will be done by analyzing and interpreting each sentence dependency-by-dependency in a bi-directional way: from left-to-right and from right-to-left.

[4]The software used to compare models is available at: https://github.com/manueldeprada/ComparingBERT/

ACKNOWLEDGEMENTS

This work has received financial support from DOMINO project (PGC2018-102041-B-I 00, MCIU/AEI/FEDER, UE), eRisk project (RTI2018-093336-B-C21), the Consellería de Cultura, Educación e Ordenación Universitaria (accreditation 2016-2019, ED431G/08, Groups of Reference: ED431C 2020/21, and ERDF 2014-2020: Call ED431G 2019/04) and the European Regional Development Fund (ERDF).

REFERENCES

Armendariz, C. S., Purver, M., Pollak, S., Ljubešić, N., Ulčar, M., Robnik-Šikonja, M., Vulić, I., and Pilehvar, M. T. (2020). SemEval-2020 task 3: Graded word similarity in context (GWSC). In *Proceedings of the 14th International Workshop on Semantic Evaluation*.

Baroni, M. (2013). Composition in distributional semantics. *Language and Linguistics Compass*, 7:511–522.

Baroni, M., Bernardini, S., Ferraresi, A., and Zanchetta, E. (2009). The wacky wide web: A collection of very large linguistically processed webcrawled corpora. *Language Resources and Evaluation*, 43(3):209–226.

Baroni, M. and Zamparelli, R. (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 (EMNLP'10)*, pages 1183–1193.

Camacho-Collados, J., Pilehvar, M., Collier, N., and Navigli, R. (2017). Semeval-2017 task 2: Multilingual and cross-lingual semantic word similarity. In *Proceedings of SemEval*, Vancouver, Canada.

Coecke, B., Sadrzadeh, M., and Clark, S. (2010). Mathematical foundations for a compositional distributional model of meaning. *Linguistic Analysis*, 36:345–384.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HTL 2019*, pages 4171–4186.

Dinu, G., Pham, N., and Baroni, M. (2013). General estimation and evaluation of compositional distributional semantic models. In *ACL 2013 Workshop on Continuous Vector Space Models and their Compositionality (CVSC 2013)*, pages 50–58, East Stroudsburg PA.

Erk, K. and Padó, S. (2008). A structured vector space model for word meaning in context. In *2008 Conference on Empirical Methods in Natural Language Processing (EMNLP-2008*, pages 897–906, Honolulu, HI.

Ethayarajh, K. (2019). How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65.

Ettinger, A. (2020). What bert is not: Lessons from a new suiteof psycholinguistic diagnostics for language models. *TACL*, 8:34–48.

Gamallo, P. (2017). The role of syntactic dependencies in compositional distributional semantics. *Corpus Linguistics and Linguistic Theory*, 13(2):261–289.

Gamallo, P. (2019). A dependency-based approach to word contextualization using compositional distributional semantics. *Language Modelling*, 7(1):53–92.

Gamallo, P. and Garcia, M. (2018). Dependency parsing with finite state transducers and compression rules. *Information Processing & Management*, 54(6):1244–1261.

Gamallo, P., Sotelo, S., Pichel, J. R., and Artetxe, M. (2019). Contextualized translations of phrasal verbs with distributional compositional semantics and monolingual corpora. *Computational Linguistics*, 45(3):395–421.

Goldberg, Y. (2019). Assessing bert's syntactic abilities. *CoRR*, abs/1901.05287.

Grefenstette, E. and Sadrzadeh, M. (2011a). Experimental support for a categorical compositional distributional model of meaning. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2011)*, pages 1394–1404.

Grefenstette, E. and Sadrzadeh, M. (2011b). Experimenting with transitive verbs in a discocat. In *Workshop on Geometrical Models of Natural Language Semantics (EMNLP 2011)*.

Hashimoto, K., Stenetorp, P., Miwa, M., and Tsuruoka, Y. (2014). Jointly learning word representations and composition functions using predicate-argument structures. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 1544–1555. ACL.

Hashimoto, K. and Tsuruoka, Y. (2015). Learning embeddings for transitive verb disambiguation by implicit tensor factorization. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 1–11, Beijing, China. ACL.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Mikolov, T., Yih, W.-t., and Zweig, G. (2013). Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2013)*, pages 746–751, Atlanta, Georgia.

Mitchell, J. and Lapata, M. (2008). Vector-based models of semantic composition. In *Proceedings of the Association for Computational Linguistics: Human Language Technologies (ACL 2008)*, pages 236–244, Columbus, Ohio.

Mitchell, J. and Lapata, M. (2009). Language models based on semantic composition. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP-2009)*, pages 430–439.

Mitchell, J. and Lapata, M. (2010). Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1439.

Partee, B. (2007). Private adjectives: Subsective plus coercion. In Bäuerle, R., Reyle, U., and Zimmermann, T. E., editors, *Presuppositions and Discourse*. Elsevier.

Pilehvar, M. T. and Camacho-Collados, J. (2019). WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2019)*, pages 1267–1273. ACL.

Polajnar, T., Rimell, L., and Clark, S. (2015). An exploration of discourse-based sentence spaces for compositional distributional semantics. In *Proceedings of the First Workshop on Linking Computational Models of Lexical, Sentential and Discourse-level Semantics*, pages 1–11. Association for Computational Linguistics.

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP)*.

Rogers, A., Kovaleva, O., and Rumshisky, A. (2020). A primer in bertology: What we know about how BERT works. *CoRR*, abs/2002.12327.

Sachan, D. S., Zhang, Y., Qi, P., and Hamilton, W. (2020). Do syntax trees help pre-trained transformers extract information? arXiv preprint: 2008.09084.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2020). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Turney, P. D. (2013). Domain and function: A dual-space model of semantic relations and compositions. *Journal of Artificial Intelligence Research (JAIR)*, 44:533–585.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc.

Weir, D. J., Weeds, J., Reffin, J., and Kober, T. (2016). Aligning packed dependency trees: A theory of composition for distributional semantics. *Computational Linguistics*, 42(4):727–761.

Yu, L. and Ettinger, A. (2020). Assessing phrasal representation and composition in transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4896–4907, Online. Association for Computational Linguistics.