Still a Pain in the Neck: Evaluating Text Representations on Lexical Composition

Vered Shwartz
Computer Science Department, Bar-Ilan University, Ramat-Gan, Israel
vered1986@gmail.com

Ido Dagan
Computer Science Department, Bar-Ilan University, Ramat-Gan, Israel
dagan@cs.biu.ac.il

Abstract

Building meaningful phrase representations is challenging because phrase meanings are not simply the sum of their constituent meanings. Lexical composition can shift the meanings of the constituent words and introduce implicit information. We tested a broad range of textual representations for their capacity to address these issues. We found that as expected, contextualized word representations perform better than static word embeddings, more so on detecting meaning shift than in recovering implicit information, in which their performance is still far from that of humans. Our evaluation suite, including 5 tasks related to lexical composition effects, can serve future research aiming to improve such representations.

1 Introduction

Modeling the meaning of phrases involves addressing semantic phenomena that pose non-trivial challenges for common text representations, which derive a phrase representation from those of its constituents words. One such phenomenon is meaning shift, which happens when the meaning of the phrase departs from the meanings of its constituent words. This is especially common among verb-particle constructions (carry on), idiomatic noun compounds (guilt trip) and other multi-word expressions (MWE, lexical units that form a distinct concept), making them “a pain in the neck” for NLP applications (Sag et al., 2002).

A second phenomenon is common for both MWEs and free word combinations such as noun compounds and adjective-noun compositions. It happens when the composition introduces implicit meaning that often requires world knowledge to uncover. For example, that hot refers to the temperature of tea but to the manner of debate (Hartung, 2015), or that olive oil is made of olives while baby oil is made for babies (Shwartz and Waterson, 2018).

There have been attempts to learn compositional phrase representations (e.g. Mitchell and Lapata, 2010; Baroni and Zamparelli, 2010; Wieting et al., 2017; Poliak et al., 2017), but many of them are tailored to a specific type of phrase or to a fixed number of constituent words, and they all disregard the surrounding context. Recently, contextualized word representations boasted dramatic performance improvements on a range of NLP tasks (Peters et al., 2018; Radford et al., 2018; Devlin et al., 2018). Such models serve as a function for computing word representations in a given context, making them potentially more capable to address meaning shift. These models were shown to capture some world knowledge (e.g Zellers et al., 2018), which may potentially help with uncovering implicit information.

In this paper we test how well various text representations address these composition-related phenomena. Methodologically, we follow recent work that applied “black-box” testing to assess various capacities of distributed representations (e.g. Adi et al., 2017; Conneau et al., 2018). We construct an evaluation suite with 5 tasks related to the two phenomena, as shown in Figure 1, and develop generic models that rely on pre-trained representations. We test 6 representations, including static word embeddings (Mikolov et al., 2013; Pennington et al., 2014; Bojanowski et al., 2017) and contextualized word embeddings (Peters et al., 2018; Radford et al., 2018; Devlin et al., 2018).

Our results confirm that indeed the contextualized word embeddings perform better than the static ones. In particular, we show that modeling context contributes to recognizing meaning shift...
in non-literal word usages. Despite hopes of filling missing information with world knowledge provided by the representations, the signal they yield for recovering implicit information is much weaker, and the gap between the best performing model and the human performance on such tasks remains substantial.

Our evaluation suite is available at https://github.com/vered1986/lexcomp. It is easily extensible, and may be used in the future to evaluate new representations for their ability to address lexical composition.

2 Composition Tasks

We experimented with 5 tasks that address the meaning shift and implicit meaning phenomena, summarized in Table 1 and detailed below.

To create a uniform framework, we cast all tasks as classification tasks. We add sentential contexts where the original datasets annotate the phrases out-of-context, by extracting averaged-length sentences (15-20 words) from English Wikipedia (January 2018 dump) in which the target phrase appears. We assume that the annotation does not depend on the context, an assumption that holds in most cases, judging by the human performance scores in Section 5.2. We split each dataset to roughly 80% train and 10% for each of the validation and test sets, under lexical constraints as detailed for each task.

2.1 Recognizing Verb-Particle Constructions

A verb-particle construction (VPC) consists of a head verb and a particle, typically in the form of an intransitive preposition, which changes the verb’s meaning (e.g. carry on vs. carry).

Task Definition. Given a sentence s that includes a verb V followed by a preposition P, the goal is to determine whether it is a VPC or not.

Data. We use the dataset of Tu and Roth (2012) which consists of 1,348 sentences from the British National Corpus, each containing a V P and annotated to whether it is a VPC or a verb-preposition combination. The dataset is focused on 23 different phrasal verbs derived from six of the most frequently used verbs (take, make, have, get, do, give), and their combination with common prepositions or particles. To reduce label bias, we split the dataset lexically by verb, i.e. if V P is in one set, there are no V P’ examples in the other sets.

2.2 Noun Compound Literality

Task Definition. Given a noun compound NC = w_1 w_2 in a sentence s, and a target word w ∈ {w_1, w_2}, the goal is to determine whether the meaning of w in NC is literal. For instance, market has a literal meaning in flea market but flea does not.

Data. We use the dataset of Reddy et al. (2011) which consists of 90 noun compounds along with human judgments about the literality of each constituent word. Scores are given in a scale of 0-5, 0 being non-literal and 5 being literal. We consider examples with a score ≥ 4 as literal, and ≤ 2 as non-literal, ignoring the middle range.

To increase the dataset size we augment it with literal examples from the Tratz (2011) dataset of noun compound classification. Compounds in this dataset are annotated to the semantic relation that holds between w_1 and w_2. Most relations (except for lexicalized, which we ignore), define the meaning of NC as some trivial combination of w_1 and w_2, allowing us to treat both words as literal.

Task Adaptation. We add sentential contexts from Wikipedia, keeping up to 10 sentences per example. We split the dataset lexically by head, i.e. if w_1 w_2 is in one set, there are no w'_1 w_2 NCs in the other sets.

2.3 Noun Compound Relations

Task Definition. Given a noun compound NC = w_1 w_2 in a sentential context s, and a paraphrase p, the goal is to determine whether p describes the semantic relation between w_1 and w_2 or not. For example, “part that makes up body” is a valid paraphrase for body part, but “replacement part bought for body” is not.
Data. We use the data from SemEval 2013 Task 4 (Hendrickx et al., 2013). The dataset consists of noun compounds and their paraphrases suggested by human annotators.

Task Adaptation. The original task is to generate a list of free-form paraphrases for a given noun compound. To match with the other tasks, we cast the task as a binary classification problem where the input is a noun compound NC and a paraphrase p, and the goal is to predict whether p is a correct description of NC.

We create negative examples for \(w_1 \ w_2\) from the paraphrase templates of other noun compounds \(w' \ w_2\) and \(w_1 \ w'_2\) in the dataset that share a constituent word with it. For example, “replacement part bought for body” is a negative example constructed from the paraphrase template “replacement [w_2] bought for [w_1]” which appeared for car part. We require one shared constituent in order to form more fluent paraphrases (which would otherwise be easily classifiable as negative). To reduce the chances of creating negative examples which are in fact valid paraphrases, we only consider negative paraphrases whose verbs never occurred in the positive paraphrase set for the given NC.

We add sentential contexts from Wikipedia, randomly selecting a sentence per example, and split the dataset lexically by head.

2.4 Adjective-Noun Attributes

Task Definition. Given an adjective-noun composition AN in a sentence s, and an attribute AT, the goal is to determine whether AT is implicitly conveyed in AN. For example, the attribute temperature is conveyed in hot water, but not in hot argument (emotionality).

Data. We use the HeiPLAS data set (Hartung, 2015), which contains adjective-noun compositions annotated with their implicit attribute meaning. The data was extracted from WordNet and manually filtered.

Task Adaptation. Since the dataset is small and the number of labels is large (> 200), we recast the task as a binary classification task. The input to the task is an AN and a paraphrase created from the template “[A] refers to the [AT] of [N]” (e.g. “loud refers to the volume of thunder”). The goal is to predict whether the paraphrase is correct or not with respect to the given AN.

We create up to 3 negative instances for each positive instance by replacing AT in another attribute that appeared with either A or N. For example, (hot argument, temperature, False). To reduce the chances that the negative attribute is in fact a valid attribute for AN, we compute the Wu-Palmer similarity (Wu and Palmer, 1994) between the original and negative attribute, taking only attributes whose similarity to the original attribute is below a threshold.

Similarly to the previous task, we attach a context sentence from Wikipedia to each example. Finally, we split the dataset lexically by adjective, i.e. if A N is in one set, there are no A N’ examples in the other sets.

2.5 Identifying Phrase Type

The last task consists of multiple phrase types and addresses both phenomena.

Task Definition. The task is defined as sequence labeling to BIO tags. Given a sentence, each word is classified to whether it is part of a phrase, and the specific type of phrase.

Data. We use the STREUSLE corpus (Supersense-Tagged Repository of English with a Unified Semantics for Lexical Expressions, Schneider and Smith, 2015). The corpus contains
text from the web reviews portion of the English Web Treebank, along with various semantic annotations, from which we use the BIO annotations. Each token is labeled with a tag, a B–X tag marks the beginning of a span of type X, I occurs inside a span, and O outside of it. B labels mark specific types of MWEs (noun compounds, idioms, light verb constructions, verb-particle constructions, etc.).

**Task Adaptation.** We are interested in a simpler version of the annotations. Specifically, we exclude the discontinuous spans (e.g. a span like “turn the [TV] off” would not be considered as part of a phrase). The corpus distinguishes between “strong” MWEs (fixed or idiomatic phrases) and “weak” MWEs (ad hoc compositional phrases). The weak MWEs are untyped, hence we label them as COMP (compositional).

### 3 Representations

We experimented with 6 common word representations from two different families detailed below. Table 2 summarizes the differences between the pretrained models used in this paper.

|            | Training objective                      | Corpus (#words)            | Output Dimension | Basic Unit |
|------------|-----------------------------------------|-----------------------------|------------------|------------|
| **Word Embeddings**                                                                 |
| WORD2VEC   | Predicting surrounding words            | Google News (100B)          | 300              | word       |
| GloVe      | Predicting co-occurrence probability    | Wikipedia + Gigaword 5 (6B) | 300              | word       |
| fastText   | Predicting surrounding words            | Wikipedia + UMBC + statmt.org (16B) | 300              | subword    |

|            | Contextualized Word Embeddings           |
|------------|-----------------------------------------|-----------------------------|------------------|------------|
| ELMo       | Language model                          | 1B Word Benchmark (1B)      | 1024              | character  |
| OpenAI GPT | Language model                          | BooksCorpus (800M)         | 768              | subword    |
| BERT       | Masked language model (Cloze)           | BooksCorpus + Wikipedia (3.3B) | 768              | subword    |

Table 2: Architectural differences of the specific pre-trained representations used in this paper.

**Contextualized Word Embeddings.** Functions computing dynamic word embeddings for words given their context sentence, largely addressing polysemy. They are pre-trained as general purpose language models using a large-scale unannotated corpus, and can later be used as a representation layer in downstream tasks (either fine-tuned to the task or fixed). The representations used in this paper have multiple output layers. We either use only the last layer, which was shown to capture semantic information (Peters et al., 2018), or learn a task-specific scalar mix of the layers (see Section 5).

**ELMo** (Embeddings from Language Models, Peters et al., 2018) are obtained by learning a character-based language model using a deep biLSTM (Graves and Schmidhuber, 2005). Working at the character-level allows using morphological clues to form robust representations for out-of-vocabulary words, unseen in training. The **OpenAI GPT** (Generative Pre-Training, Radford et al., 2018) has a similar training objective, but the underlying encoder is a transformer (Vaswani et al., 2017). It uses subwords as the basic unit, employing bytepair encoding. Finally, **BERT** (Bidirectional Encoder Representations from Transformers, Devlin et al., 2018) is also based on the transformer, but it is bidirectional as opposed to right-to-left as in the OpenAI GPT, and the directions are dependent as opposed to ELMo’s independently trained left-to-right and right-to-left LSTMs. It also introduces a slightly different objective called “masked language model”: during training, some tokens are
randomly masked, and the objective is to restore them from the context.

4 Classification Models

We implemented minimal “Embed-Encode-Predict” models that use the representations from Section 3 as inputs, keeping them fixed during training.

**Embed.** We use the embedding model to embed each word in the sentence \( s = w_1...w_n \), obtaining:

\[
\vec{v}_1, ..., \vec{v}_n = \text{Embed}(s)
\]  

Depending on the specific task, we may have another input \( w'_1, ..., w'_l \) to embed separately from the sentence: the paraphrases in the NC Relations and AN Attributes tasks, and the target word in the NC Literality task (to obtain an out-of-context representation of the target word). We embed this extra input as follows:

\[
\vec{v}'_1, ..., \vec{v}'_l = \text{Embed}(w'_1, ..., w'_l)
\]

**Encode.** We implemented two variants for each model. In our encoded variant (biLM in Table 3) we encode the embedded sequence using a biLSTM with a hidden dimension \( d \), where \( d \) is the input embedding dimension:

\[
\vec{u}_1, ..., \vec{u}_n = \text{biLSTM}(\vec{v}_1, ..., \vec{v}_n)
\]

As opposed to the pre-trained embeddings, the biLSTM parameters are updated during training. In the non-encoded variant (None), we simply define:

\[
\vec{u}_1, ..., \vec{u}_n = \vec{v}_1, ..., \vec{v}_n
\]

**Predict.** The input to the classifier is a concatenation of a vector representing the target span from Equation 3 (e.g. the noun compound in NC relations) and, when applicable, the additional span from Equation 2 (e.g. the paraphrase in NC relations). In both cases, to represent a target span, we concatenate its end vectors. In the general case, the input to the classifier is:

\[
\vec{x} = [\vec{u}_{i...i+k}; \vec{u}'_{1...l}] = [\vec{u}_i; \vec{u}_{i+k}; \vec{u}'_{1}; \vec{u}'_{l}]
\]

where each of \( u_{i+k} \), \( u'_{1} \), and \( u'_{l} \) can be empty vectors in the cases of single word spans or no additional inputs.

The classifier output is defined as:

\[
\tilde{o} = \text{softmax}(W \cdot \text{ReLU}(\text{Dropout}(h(\vec{x}))))
\]

where \( h \) is a 300-dimensional hidden layer, the dropout probability is 0.2, \( W \in \mathbb{R}^{k \times 300} \), and \( k \) is the number of relations for the specific task.

**Implementation Details.** We implemented the models using the AllenNLP library (Gardner et al., 2018) which is based on the PyTorch framework (Paszke et al., 2017). We train them for up to 500 epochs, stopping early if the validation performance doesn’t improve in 20 epochs.

The phrase type model is a sequence tagging model that predicts a label for each embedded (potentially encoded) word \( w_{i} \). During decoding, we enforce a single constraint that requires that a B-X tag must precede I tag(s).

5 Experiments

5.1 Baselines

**Majority Baselines.** We implemented two majority baselines: Majority-ALL is computed by assigning the most common label in the training set to all the items, while majority-WORD assigns the most common label per constituent word. For example, in the VPC classification task, it classifies get through as positive in all its contexts because both the verb get and the preposition through appear in more positive than negative examples.

**Human Performance.** We estimated the human performance on each task, by sampling and re-annotating 100 examples from each test set. The annotation was carried out in Mechanical Turk. We asked 3 workers to annotate each example, taking the majority label as the final prediction. To control the quality of the annotations, we required that workers must have an acceptance rate of at least 98% on at least 500 prior HITs, and had them pass a qualification test. We didn’t compute the estimated human performance on the phrase type task, which is more complicated and requires expert annotation.\(^1\)

5.2 Results

Table 3 displays the performance scores (usually accuracy) of different models on the various tasks.

\(^1\)The agreement between the workers was: VPC - 84.17%, NC Literality - 80.81%, NC Relations - 86.21%, and AN Attributes - 86.42%.
The general trend across tasks is that the performance improves from the word embeddings to contextualized word representations, with a large gap in some of the tasks. Among the latter, BERT performs best on 4/5 tasks. There is no consistent preference among ELMo and the OpenAI GPT.

Encoding the embedded input with a biLSTM improves the performance of models using word embeddings, but less consistently when the contextualized word embeddings are used. We did not find a clear preference to either using only the top layer or learning a scalar mix of the layers in the contextualized embeddings. We extracted the learned layer weights for each of the All models, and found that in most cases, the model learned a balanced mix of the top and bottom layers.

The gap between the majority baselines and the best performance ranges from 4.1 points in NC Relations to 62.3 in VPC Classification. The gap between the best performance and the estimated human performance is as low as 1.9 and 5.1 in NC Literality and VPC Classification, respectively, and as high as 23.5 and 21.3 in NC Relations and AN Attributes. Unsurprisingly, the gap from human performance is larger among the tasks that require revealing implicit meaning than in those that need to recognize meaning shift.

In the Phrase Type task, looking into the errors made by the best model (BERT+All+biLM) reveals that most of the errors were predicting 0, i.e. missing the occurrence of a phrase. With respect to specific phrase types, near perfect performance was achieved among the more syntactic categories. Specifically, auxiliary (“Did they think we were [going to] feel lucky to get any reservation at all?”), adverbs (“any longer”), and determiners (“a bit”). In accordance to the VPC Classification task, the VPC label achieved 85% accuracy. 10% were missed (classified as 0) and 5% were confused with a “weak” MWE. Two of the more difficult types were “weak” MWEs (which are judged as more compositional and less idiomatic) and idiomatic verbs. The former achieved accuracy of 22% (68% were classified as 0) and the latter only 8% (62% were classified as 0). Overall it seems that the model relied mostly on syntactic cues, failing to recognize semantic subtleties such as idiomatic meaning and level of compositionality.

### 6 Analysis

We focus on the contextualized word embeddings, and look into the representations they provide.

#### 6.1 Meaning Shift

**Does the representation capture VPCs?** The best performer on the VPC Classification task was the BERT+All+None. To get a better idea of the

| Model | Layer | Encoding | VPC Classification | NC Literality | NC Relations | AN Attributes | Phrase Type |
|-------|-------|----------|--------------------|---------------|--------------|---------------|-------------|
|       |       |          | Acc | Acc | Acc | Acc | F1 |
| Majority | All | None | 23.0 | 66.7 | 50.0 | 50.0 | 0.0 |
| Human | Word | None | 23.0 | 67.2 | 50.0 | 50.0 | 0.0 |
|       |       |          | 93.8 | 91.0 | 77.8 | 86.4 | 0.0 |
| word2vec | None | biLM | 23.0 | 50.0 | 50.0 | 32.4 | 0.0 |
| GloVe | None | biLM | 23.0 | 50.0 | 50.0 | 32.4 | 0.0 |
| fastText | None | biLM | 23.0 | 48.8 | 50.0 | 0.0 | 0.0 |
| ELMo | Top | None | 80.0 | 78.3 | 48.8 | 32.3 | 0.0 |
|       | All | None | 79.5 | 79.7 | 50.0 | 32.3 | 0.0 |
| OpenAI GPT | Top | None | 65.0 | 87.0 | 46.9 | 53.1 | 29.0 |
|       | All | None | 78.1 | 75.4 | 50.0 | 54.1 | 59.0 |
| BERT | Top | None | 78.6 | 89.1 | 47.5 | 64.2 | 44.0 |
|       | All | None | 85.9 | 88.4 | 51.2 | 65.1 | 58.0 |

Table 3: Performance on the various tasks. The evaluation metric is accuracy except for the phrase type task in which we report span-based F1 score, excluding O tags.
signal that BERT contains for VPCs, we chose several ambiguous verb-preposition pairs in the dataset, each appearing in at least 8 examples as a VPC and 8 examples as a non-VPC. We computed their BERT vectors, averaging the layers using the weights learned by the model, and concatenated the verb and preposition vectors. We projected them into 2D space using t-SNE (Maaten and Hinton, 2008), as demonstrated in Figure 2. The example VPCs behave similarly to the other VPCs, showing that BERT contains a signal for separating different verb usages.

Non-literality as a rare sense. Nunberg et al. (1994) considered some non-literal compounds as “idiosyncratically decomposable”, i.e. which can be decomposed to possibly rare senses of their constituents, as in considering bee to have a sense of “competition” in spelling bee and crocodile to stand for “manipulative” in crocodile tears. Using this definition, we can cast our NC literality task as a coarse-grained word sense disambiguation task. Recent work has shown that contextualized word representations are successful in word sense induction (Stanovsky and Hopkins, 2018; Amrami and Goldberg, 2018). To test whether they can also capture these rare senses, we sample target words that appear in both literal and non-literal examples, and use each contextualized word embedding model as a language model to predict the best substitutes of the target word in each context. Table 4 exemplifies some of these predictions.

Bold words are words judged reasonable in the given context, even if they don’t have the exact same meaning as the target word. It is apparent that there are more reasonable substitutes for the literal examples, across models (left part of the table), but BERT performs better than the others. The OpenAI GPT shows a clear disadvantage of being uni-directional, often choosing a substitute that doesn’t go well with the next word (”a train to from”).

The success is only partial among non-literal examples. While some correct substitutes are predicted for (guilt) trip, the predictions are much worse for the other examples. The meaning of diamond in diamond wedding is “60th”, and ELMo makes the closest prediction, 10th (which would make it “tin wedding”). 400th is a borderline prediction, because it is also an ordinal number, but an unreasonable one in the context of years of marriage.

Finally, the last example snake oil is unsurprisingly a difficult one, possibly “non-decomposable” (Nunberg et al., 1994), as both constituents are non-literal. Some predicted substitutes, rogue and charmer, are valid replacements for the entire noun compound (e.g. “you are a rogue salesman”). Others go well with the literal meaning of snake creating phrases denoting concepts which can indeed be sold (snake egg, snake skin). Overall, modeling context seems to help build meaningful representations for rare senses, yet the literal senses affect the representations.

6.2 Implicit Meaning

The performance of the various models on the tasks that involve revealing implicit meaning are substantially worse than on the other tasks. In NC Relations, ELMo performs best with the biLM-encoded model using only the top layer of the representation, surpassing the majority baseline by only 4.3 points in accuracy. The best performer in AN Attributes is BERT, with no encoding and using all the layers, achieving accuracy of 65.1%, well above the majority baseline (50%).

We are interested in finding out where the knowledge of the implicit meaning originates. Is it encoded in the phrase representation itself, or does it appear explicitly in the context sentences?
We trained the following models: NC Relations and BERT+All+None for AN Attributes setting for each (ELMo+Top+biLM for context tests for each of the tasks, using the best performance of the models learning to distinguish between abstract idioms from non-idioms, but their ablation study showed the phrase representation contains this implicit information.

Finally, could it be that the performance gap from the majority baseline is due to the models learning to recognize which paraphrases are more probable than others, regardless of the phrase itself?

To try answer this question, we performed ablation tests for each of the tasks, using the best performing setting for each (ELMo+Top+biLM for NC Relations and BERT+All+None for AN Attributes). We trained the following models:

1. **Phrase:** where we mask the phrase in its context sentence, e.g. replacing “Today, the house has become a wine bar / bistro called Barokk” with “Today, the house has become a something / bistro called Barokk”. Success in this setting may indicate that the implicit information is given explicitly in some of the context sentences.²

2. **Context:** the out-of-context version of the original task, in which we replace the context sentence by the phrase itself, as in setting it to “wine bar”. Success in this setting may indicate that the phrase representation contains this implicit information.

3. **(Context+Phrase):** in which we omit the context sentence altogether, and provide only the paraphrase, as in “bar where people buy and drink wine”. Success in this setting may indicate that negative sampled paraphrases form sentences which are less probable in English.

Table 5 shows the results of this experiment. A first observation is that the full model performs best on both tasks, suggesting that the model captures implicit meaning from various sources. In the NC Relations, all variants perform on par or worse than the majority baseline, achieving a few points less than the full model. In the AN Attributes task it is easier to see that the phrase (AN) is important for the classification, while the context is secondary.

2A somewhat similar phenomenon was recently reported by Senaldi et al. (2019). Their model managed to distinguish idioms from non-idioms, but their ablation study showed the model was in fact learning to distinguish between abstract contexts (in which idioms tend to appear) and concrete ones.
7 Related Work

Probing Tasks. One way to test whether dense representations capture a certain linguistic property is to design a task for this property, and build a model that takes the representation as an input. This kind of “black box” testing has become popular recently. Adi et al. (2017) studied whether sentence embeddings capture properties such as sentence length and word order. Conneau et al. (2018) extended their work with a large number of sentence embeddings, and tested various properties at the surface, syntactic, and semantic levels. Others focused on intermediate representations in neural machine translation systems (e.g. Shi et al., 2016; Belinkov et al., 2017; Dalvi et al., 2017; Sennrich, 2017), or in specific linguistic properties such as agreement (Giulianelli et al., 2018), and tense (Bacon and Regier, 2018).

More recently, Tenney et al. (2019) designed a suite of tasks to test contextualized word embeddings on a broad range of sub-sentence tasks, including part-of-speech tagging, syntactic constituent labeling, dependency parsing, named entity recognition, semantic role labeling, coreference resolution, semantic proto-role, and relation classification. All the representations that were tested (McCann et al., 2017; Peters et al., 2018; Radford et al., 2018; Devlin et al., 2018) produced strong representations for syntactic phenomena, but gained smaller performance improvements upon the baselines in the more semantic tasks. To the best of our knowledge, we are the first to provide an evaluation suite consisting of tasks related to lexical composition.

Lexical Composition. There is a vast literature on multi-word expressions in general (e.g. Sag et al., 2002; Vincze et al., 2011), and research focusing on noun compounds (e.g. Nakov, 2013; Nastase et al., 2013), adjective-noun compositions (e.g. Baroni and Zamparelli, 2010; Boleda et al., 2013), and verb-particle constructions (e.g. Baldwin, 2005; Pichotta and DeNero, 2013).

In recent years, word embeddings have been used to predict the compositionality of phrases (Salehi et al., 2015; Cordeiro et al., 2016), and to identify the implicit relation in adjective-noun compositions (Hartung et al., 2017) and in noun compounds (Surtani and Paul, 2015; Dima, 2016; Schwartz and Waterson, 2018).

Pavlick and Callison-Burch (2016) created a simpler variant of the recognizing textual entailment task (RTE, Dagan et al., 2013) that tests whether an adjective-noun composition entails the noun alone and vice versa in a given context. They tested various standard models for RTE and found that the models performed poorly with respect to this phenomenon. To the best of our knowledge, contextualized word embeddings haven’t been employed for tasks related to lexical composition yet.

Phrase Representations. With respect to obtaining meaningful phrase representations, there is a prominent line of work in learning a composition function over pairs of words. Mitchell and Lapata (2010) suggested simple composition via vector arithmetics. Baroni and Zamparelli (2010) and later Maillard and Clark (2015) treated adjectival modifiers as functions that operate on nouns and change their meanings, and represented them as matrices. Zanzotto et al. (2010) and Dinu et al. (2013) extended this approach and composed any two words by multiplying each word vector by a composition matrix. These models start by computing the phrases’ distributional representation (i.e. treating it as a single token) and then learning a composition function that approximates it.

The main drawbacks of this approach are that it assumes compositionality and operates on phrases with a pre-defined number of words. Moreover, we can expect the resulting compositional vectors to capture properties inherited from the constituent words, but it is unclear if they also capture new properties of the phrase. For example, the compositional representation of olive oil may capture properties like green (from olive) and fat (from oil), but would it also capture properties like expensive (a result of the extraction process)?

Alternatively, other approaches were suggested for learning general phrase embeddings, either using direct supervision for paraphrase similarity (Wieting et al., 2016), indirectly from an extrinsic task (Socher et al., 2012), or in an unsupervised manner by extending the word2vec objective (Poliak et al., 2017). While they don’t have constrains on the phrase length, these methods still suffer from the two other drawbacks: they assume that the meaning of the phrase can always be composed from its constituent meanings, and it is unclear whether they can incorporate implicit information and new properties of the phrase. We expected that contextualized word embeddings, which assign a
different vector for a word in each given context, would address at least the first issue by producing completely different vectors to literal vs. non-literal word occurrences.

8 Discussion and Conclusion

We showed that contextualized word representations perform generally better than static word embeddings on tasks related to lexical composition. However, on some tasks they are still far from human performance. This gap may suggest a limit on the information that distributional models can provide about the meanings of phrases, when those exhibit phenomena such as meaning shift and introducing implicit meaning.

Moving away from the distributional models, an approach to build meaningful phrase representations can get some inspiration from the way that humans process phrases. A study on how L2 learners process idioms found that the most common and successful strategies were inferring from the context (57% success) and relying on the literal meanings of the constituent words (22% success) (Cooper, 1999).

As opposed to distributional models that aim to learn from a large number of (possibly noisy and uninformative) contexts, the sentential contexts in this experiment were manually selected, and a follow up study found that extended contexts (stories) help the interpretation further (Asl, 2013). The participants didn’t simply rely on adjacent words or phrases, but also employed reasoning. For example, in the sentence “Robert knew that he was robbing the cradle by dating a sixteen-year-old girl”, the participants inferred that 16 is too young to date, combined it with the knowledge that cradle is where a baby sleeps, and concluded that rob the cradle means dating a very young person. This context modeling seems to be beyond the scope of the current text representations.

We hope that future representations would be designed to address the issues presented in this paper. Our evaluation suite, which is easily extensible, would facilitate testing future text representations on issues related to lexical composition.

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofir Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In International Conference on Learning Representations (ICLR).

Asaf Amrami and Yoav Goldberg. 2018. Word sense induction with neural biLSTM and symmetric patterns. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 4860–4867, Brussels, Belgium. Association for Computational Linguistics.

Fatemeh Mohamadi Asl. 2013. The impact of context on learning idioms in EFL classes. TESOL Journal, 37(1):2.

Geoff Bacon and Terry Regier. 2018. Probing sentence embeddings for structure-dependent tense. In Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, pages 334–336. Association for Computational Linguistics.

Timothy Baldwin. 2005. Deep lexical acquisition of verb–particle constructions. Computer Speech & Language, 19(4):398–414.

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

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural machine translation models learn about morphology? In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 861–872, Vancouver, Canada. Association for Computational Linguistics.

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

Gemma Boleda, Marco Baroni, The Nghia Pham, and Louise McNally. 2013. Intensionality was only alleged: On Adjective-noun composition in distributional semantics. In Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Long Papers, pages 35–46, Potsdam, Germany. Association for Computational Linguistics.

Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $\phi$ vector: Probing sentence embeddings for linguistic properties. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.
Thomas C Cooper. 1999. Processing of idioms by L2 learners of English. *TESOL quarterly*, 33(2):233–262.

Silvio Cordeiro, Carlos Ramisch, Marco Idiart, and Aline Villavicencio. 2016. Predicting the compositionality of nominal compounds: Giving word embeddings a hard time. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1986–1997, Berlin, Germany. Association for Computational Linguistics.

Ido Dagan, Dan Roth, Mark Sammons, and Fabio Zanzotto. 2013. Recognizing Textual Entailment. Morgan & Claypool Publishers.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, and Stephan Vogel. 2017. Understanding and improving morphological learning in the neural machine translation decoder. In Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 142–151, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805v1*.

Corina Dima. 2016. On the compositionality and semantic interpretation of English noun compounds. In Proceedings of the 1st Workshop on Representation Learning for NLP, pages 27–39.

Georgiana Dinu, Nghia The Pham, and Marco Baroni. 2013. General estimation and evaluation of compositional distributional semantic models. In Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality, pages 50–58, Sofia, Bulgaria. Association for Computational Linguistics.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic language processing platform. In Proceedings of Workshop for NLP Open Source Software (NLP-OSS), pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

Mario Giulianelli, Jack Harding, Florian Mohnert, Dieuwke Hupkes, and Willem Zuidema. 2018. Under the hood: Using diagnostic classifiers to investigate and improve how language models track agreement information. In Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, pages 240–248. Association for Computational Linguistics.

Alex Graves and Jürgen Schmidhuber. 2005. Framework pheme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5-6):602–610.

Z Harris. 1954. Distributional Hypothesis. *Word*, 10(23):146–162.

Matthias Hartung. 2015. *Distributional Semantic Models of Attribute Meaning in Adjectives and Nouns*. Ph.D. thesis, Heidelberg University.

Matthias Hartung, Fabian Kaupmann, Soufian Jебbара, and Philipp Cimiano. 2017. Learning compositionality functions on word embeddings for modelling attribute meaning in Adjective-noun phrases. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 54–64, Valencia, Spain. Association for Computational Linguistics.

Iris Hendrickx, Zornitsa Kozareva, Preslav Nakov, Diarmuid O Séaghdha, Stan Szpakowicz, and Tony Veale. 2013. SemEval-2013 task 4: Free paraphrases of noun compounds. In Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), pages 138–143. Association for Computational Linguistics.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov):2579–2605.

Jean Maillard and Stephen Clark. 2015. Learning adjective meanings with a tensor-based Skip-Gram model. In Proceedings of the Nineteenth Conference on Computational Natural Language Learning, pages 327–331, Beijjing, China. Association for Computational Linguistics.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6294–6305.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations (ICLR)*.

Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive science*, 34(8):1388–1429.

Preslav Nakov. 2013. On the interpretation of noun compounds: Syntax, semantics, and entailment. *Natural Language Engineering*, 19(3):291–330.

Vivi Nastase, Preslav Nakov, Diarmuid O Seaghdha, and Stan Szpakowicz. 2013. Semantic Relations between Nominals. *Synthesis lectures on human language technologies*, 6(1):1–119.

Geoffrey Nunberg, Ivan A Sag, and Thomas Wasow. 1994. Idioms. *Language*, 70(3):491–538.
Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. In Autodiff Workshop, NIPS 2017.

Ellie Pavlick and Chris Callison-Burch. 2016. Most "babies" are "little" and most "problems" are "huge": Compositional Entailment in Adjective-nouns. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2164–2173, Berlin, Germany. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for word representation. In Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543.

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

Karl Pichotta and John DeNero. 2013. Identifying phrasal verbs using many bilingual corpora. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 636–646, Seattle, Washington, USA. Association for Computational Linguistics.

Adam Poliak, Pushpendre Rastogi, M. Patrick Martin, and Benjamin Van Durme. 2017. Efficient, compositional, order-sensitive N-gram embeddings. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 503–508, Valencia, Spain. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. URL https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf.

Siva Reddy, Diana McCarthy, and Suresh Manandhar. 2011. An empirical study on compositionality in compound nouns. In Proceedings of 5th International Joint Conference on Natural Language Processing, pages 210–218, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Bahar Salehi, Paul Cook, and Timothy Baldwin. 2015. A word embedding approach to predicting the compositionality of multiword expressions. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 977–983, Denver, Colorado. Association for Computational Linguistics.

Nathan Schneider and Noah A. Smith. 2015. A corpus and model integrating multiword expressions and supersenses. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1537–1547, Denver, Colorado. Association for Computational Linguistics.

Marco Silvio Giuseppe Senaldi, Yuri Bizzoni, and Alessandro Lenci. 2019. What do neural networks actually learn, when they learn to identify idioms? Proceedings of the Society for Computational Linguistics, 2(1):310–313.

Rico Sennrich. 2017. How grammatical is character-level neural machine translation? assessing MT quality with contrastive translation pairs. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 376–382, Valencia, Spain. Association for Computational Linguistics.

Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does string-based neural MT learn source syntax? In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 1526–1534. Association for Computational Linguistics.

Vered Shwartz and Chris Waterson. 2018. Olive oil is made of olives, baby oil is made for babies: Interpreting noun compounds using paraphrases in a neural model. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 218–224, New Orleans, Louisiana. Association for Computational Linguistics.

Richard Socher, Brody Huval, D. Christopher Manning, and Y. Andrew Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 1201–1211. Association for Computational Linguistics.

Gabriel Stanovsky and Mark Hopkins. 2018. Spot the odd man out: Exploring the associative power of lexical resources. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 1533–1542, Brussels, Belgium. Association for Computational Linguistics.
Nitesh Surtani and Soma Paul. 2015. A VSM-based statistical model for the semantic relation interpretation of noun-modifier pairs. In Proceedings of the International Conference Recent Advances in Natural Language Processing, pages 636–645.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. In International Conference on Learning Representations.

Stephen Tratz. 2011. Semantically-enriched Parsing for Natural Language Understanding. University of Southern California.

Yuancheng Tu and Dan Roth. 2012. Sorting out the most confusing English phrasal verbs. In *SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012), pages 65–69, Montréal, Canada. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in Neural Information Processing Systems, pages 5998–6008.

Veronika Vincze, István Nagy T., and Gábor Berend. 2011. Multiword expressions and named entities in the Wiki50 corpus. In Proceedings of the International Conference Recent Advances in Natural Language Processing 2011, pages 289–295. Association for Computational Linguistics.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Towards universal paraphrastic sentence embeddings. In International Conference on Learning Representations (ICLR).

John Wieting, Jonathan Mallinson, and Kevin Gimpel. 2017. Learning paraphrastic sentence embeddings from back-translated bitext. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 274–285, Copenhagen, Denmark. Association for Computational Linguistics.

Zhibiao Wu and Martha Palmer. 1994. Verbs semantics and lexical selection. In Proceedings of the 32nd annual meeting on Association for Computational Linguistics, pages 133–138. Association for Computational Linguistics.

Fabio Massimo Zanzotto, Ioannis Korkontzelos, Francesca Fallucchi, and Suresh Manandhar. 2010. Estimating linear models for compositional distributional semantics. In Proceedings of the 23rd International Conference on Computational Linguistics, pages 1263–1271. Association for Computational Linguistics.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. Swag: A large-scale adversarial dataset for grounded commonsense inference. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 93–104, Brussels, Belgium. Association for Computational Linguistics.