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

Vered Shwartz **Ido Dagan**
Computer Science Department, Bar-Ilan University, Ramat-Gan, Israel
vered1986@gmail.com 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, consisting of six tasks related to lexical composition effects, can serve future research aiming to improve 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 constituent 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 an *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 has been a line of 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 these 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., 2019). 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 six tasks related to the above two phenomena, as shown in Figure 1, and develop generic models that rely on pre-trained representations. We test six 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., 2019). Our contributions are as follows:

1. We created a unified framework that tests the capacity of representations to address lexical composition via classification tasks, focusing on detecting meaning shift and recovering

Transactions of the Association for Computational Linguistics, vol. 7, pp. 403–419, 2019. Action Editor: Marco Baroni. https://doi.org/10.1162/tacl_a_00277.
Submission batch: 4/2019; Published 7/2019.

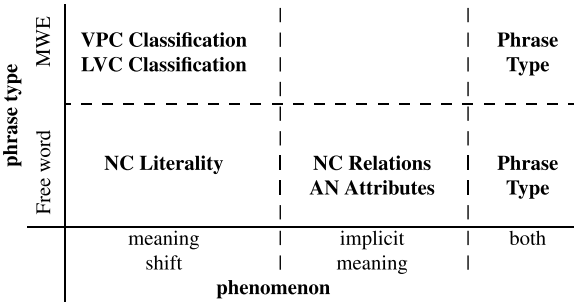

Figure 1: A map of our tasks according to type of phrase (MWE/free word combination) and the phenomenon they test (meaning shift/implicit meaning).

implicit meaning. We test six representations and provide an in depth analysis of the results.

2. We relied on existing annotated data sets used in various tasks, and recast them to fit our classification framework. We additionally annotated a sample from each test set to confirm the data validity and estimate the human performance on each task.

3. We provide the classification framework, including data and code (available at `https://github.com/vered1986/lexcomp`), which would allow testing future models for their capacity to address lexical composition.

Our results confirm that the contextualized word embeddings perform better than the static ones. In particular, we show that modeling context indeed contributes to recognizing meaning shift: On such tasks, contextualized models performed on par with humans.

Conversely, despite hopes of filling missing information with world knowledge provided by the contextualized 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. We expect that improving the ability of such representations to reveal implicit meaning would require more than a language model training objective. In particular, one future direction is a richer training objective that simultaneously models multiple co-occurrences of the constituent words across different texts, as is commonly done in noun compound interpretation (e.g., Ó Séaghdha and Copestake, 2009; Shwartz and Waterson, 2018; Shwartz and Dagan, 2018).

2 Composition Tasks

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

We rely on existing tasks and data sets, but make substantial changes and augmentations in order to create a uniform framework. First, we cast all tasks as classification tasks. Second, we add sentential contexts where the original data sets annotate the phrases out-of-context, by extracting average-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 (Section 6). Finally, we split each data set 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 data set of Tu and Roth (2012), which consists of 1,348 sentences from the British National Corpus (BNC), each containing a verb V and a preposition P annotated to whether it is a VPC or not. The data set 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 data set lexically by verb—that is, the train, test, and validation sets contain distinct verbs in their V and P combinations.

2.2 Recognizing Light Verb Constructions

The meaning of a light verb construction (LVC, e.g., *make a decision*) is mainly derived from its noun object (*decision*), whereas the meaning of its main verb (*make*) is ''light'' (Jespersen, 1965). As a rule of thumb, an LVC can be replaced by the verb usage of its direct object noun (*decide*) without changing the meaning of the sentence.

Task	Data Source	Train/val/test Size	Input	Output	Context
VPC Classification	Tu and Roth (2012)	919/209/220	sentence s VP = w_1 w_2	is VP a VPC?	O
LVC Classification	Tu and Roth (2011)	1521/258/383	sentence s span = w_1 ... w_k	is the span an LVC?	O
NC Literality	Reddy et al. (2011) Tratz (2011)	2529/323/138	sentence s NC = w_1 w_2 target w $\in \{w_1, w_2\}$	is w literal in NC?	A
NC Relations	SemEval 2013 Task 4 (Hendrickx et al., 2013)	1274/162/130	sentence s NC = w_1 w_2 paraphrase p	does p explicate NC?	A
AN Attributes	HeiPLAS (Hartung, 2015)	837/108/106	sentence s AN = w_1 w_2 paraphrase p	does p describe the attribute in AN?	A
Phrase Type	STREUSLE (Schneider and Smith, 2015)	3017/372/376	sentence s	label per token	O

Table 1: A summary of the composition tasks included in our evaluation suite. In the context column, O means the context is part of the original data set, and A is used for data sets in which the context was added in this work.

Task Definition. Given a sentence s that includes a potential light verb construction (*make an easy decision*), the goal is to determine whether it is an LVC or not.

Data. We use the data set of Tu and Roth (2011), which contains 2,162 sentences from BNC in which a potential LVC was found (with the same six common verbs as in Section 2.1), annotated to whether it is an LVC in a given context or not. We split the data set lexically by the verb.

2.3 Noun Compound Literality

Task Definition. Given a noun compound NC = w_1 w_2 in a sentence s, and a target word w $\in \{w_1, w_2\}$, 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 data set 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 to 5, 0 being non-literal and 5 being literal. For each noun compound and each of its constituents we consider examples with a score ≥ 4 as literal, and ≤ 2 as non-literal, ignoring the middle range. We obtain 72 literal and 72 non-literal examples.

To increase the data set size we augment it with literal examples from the Tratz (2011) data set of noun compound classification. Compounds in this data set 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 regard both words as literal. This method produces additional 3,061 literal examples.

Task Adaptation. We add sentential contexts from Wikipedia, keeping up to 10 sentences per example. We downsample from the literal examples to balance the data set, allowing for a ratio of up to four literal to non-literal examples. We split the data set 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).[1]

2.4 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 data set

[1] We chose to split by head rather than by modifier based on the majority baseline that achieved better performance.

consists of 356 noun compounds annotated with 12,446 human-proposed free text paraphrases.

Task Adaptation. The goal of the SemEval task was to generate a list of free-form paraphrases for a given noun compound, which explicate the implicit semantic relation between its constituent. To match with our 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.

The positive examples for an NC w_1 w_2 are trivially derived from the original data by sampling up to five of its paraphrases and creating a (NC, p, TRUE) example for each paraphrase p. The same number of negative examples is then created using negative sampling from the paraphrase templates of other noun compounds w'_1 w_2 and w_1 w'_2 in the data set that share a constituent word with NC. 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 that 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 one sentence per example, and split the data set lexically by head.

2.5 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 1,598 adjective-noun compositions annotated with their implicit attribute meaning. The data were extracted from WordNet and manually filtered. The label set consists of attribute synsets in WordNet that are linked to adjective synsets.

Task Adaptation. Because the data set is small and the number of labels is large (254), 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 three 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 chance 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 (0.4).

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

2.6 Identifying Phrase Type

The last task consists of multiple phrase types and addresses detecting both meaning shift and implicit meaning.

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 the 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 texts 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 phrase.[2]

Task Adaptation. We are interested in a simpler version of the annotations. Specifically, we exclude the discontinuous spans (e.g., a span like

[2]Sorted by frequency: noun phrase, weak (compositional) MWE, verb-particle construction, verbal idioms, prepositional phrase, auxiliary, adposition, discourse / pragmatic expression, inherently adpositional verb, adjective, determiner, adverb, light verb construction, non-possessive pronoun, full verb or copula, conjunction.

	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.

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

Word Embeddings. Word embedding models provide a fixed d-dimensional vector for each word in the vocabulary. Their training is based on the distributional hypothesis, according to which semantically similar words tend to appear in the same contexts (Harris, 1954). **word2vec** (Mikolov et al., 2013) can be trained with one of two objectives. We use the embeddings trained with the Skip-Gram objective, which predicts the context words given the target word. **GloVe** (Pennington et al., 2014) learns word vectors with the objective of estimating the log-probability of a word pair co-occurrence. **fastText** (Bojanowski et al., 2017) extends word2vec by adding information about subwords (bag of character n-grams). This is especially helpful in morphologically rich languages, but can also help handling rare or misspelled words in English.

Contextualized Word Embeddings are 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 with the other model parameters 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 6).

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, utilizing bytepair encoding. Finally, **BERT** (Bidirectional Encoder Representations from Transformers; Devlin et al., 2019) is also based on the transformer, but it is bidirectional as opposed to left-to-right 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 somewhat 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. The rationale behind the model design

was to keep them uniform for easy comparison between the representations, and make them simple so that the model's success can be attributed directly to the input representations.

Embed. We use the embedding model to embed each word in the sentence $s = w_1 \dots w_n$, obtaining

$$\vec{v}_1, \dots, \vec{v}_n = \text{Embed}(s) \quad (1)$$

where $\vec{v}_i$ stands for the word embedding of word w_i, which may be computed as a function of the entire sentence (in the case of contextualized word embeddings).

Depending on the specific task, we may have another input $w'_1, \dots, w'_l$ 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:

$$\vec{v'}_1, \dots, \vec{v'}_l = \text{Embed}(w'_1, \dots, w'_l) \quad (2)$$

Encode. We encode the embedded sequences $\vec{v}_1, \dots, \vec{v}_n$ and $\vec{v'}_1, \dots, \vec{v'}_l$ using one of the following three encode variants. As opposed to the pre-trained embeddings, the encoder parameters are updated during training to fit the specific task.

- `biLM`: Encoding the embedded sequence using a biLSTM with a hidden dimension d, where d is the input embedding dimension:

$$\vec{u}_1, \dots, \vec{u}_n = \text{biLSTM}(\vec{v}_1, \dots, \vec{v}_n) \quad (3)$$

- `Att`: Encoding the embedded sequence using self-attention. Each word is represented as the concatenation of its embedding and a weighted average over other words in the sentence:

$$\vec{u}_i = [\vec{v}_i; \sum_{j=1}^{n} a_{i,j} \cdot \vec{v}_j] \quad (4)$$

  The weights a_i are computed by applying dot-product between $\vec{v}_i$ and every other word, and normalizing the scores using softmax:

$$\vec{a}_i = \text{softmax}(\vec{v}_i^T \cdot \vec{v}) \quad (5)$$

- `None`: In which we don't encode the embedded text, but simply define:

$$\vec{u}_1, \dots, \vec{u}_n = \vec{v}_1, \dots, \vec{v}_n \quad (6)$$

For all encoder variants, $\vec{u}_i$ stands for the vector representing w_i, which is used as input to the classifier.

Predict. We represent a span by concatenating its end-point vectors, for example, $\vec{u}_{i\dots i+k} = [\vec{u}_i; \vec{u}_{i+k}]$ is the target span vector of $w_i, \dots, w_{i+k}$. In tasks which require a second span, we similarly compute $\vec{u'}_{1\dots l}$, representing the encoded span $w'_1, \dots, w'_l$ (e.g., the paraphrase in NC relations). The input to the classifier is the concatenation of $\vec{u}_{i\dots i+k}$, and, when applicable, the additional span vector $\vec{u'}_{1\dots l}$. In the general case, the input to the classifier is:

$$\vec{x} = [\vec{u}_i; \vec{u}_{i+k}; \vec{u'}_1; \vec{u'}_l] \quad (7)$$

where each of $\vec{u}_{i+k}$, $\vec{u'}_1$, and $\vec{u'}_l$ can be empty vectors in the cases of single word spans or no additional inputs.

The classifier output is defined as:

$$\vec{o} = \text{softmax}(W \cdot \text{ReLU}(\text{Dropout}(h(\vec{x})))) \quad (8)$$

where h is a 300-dimensional hidden layer, the dropout probability is 0.2, $W \in \mathcal{R}^{k \times 300}$, and k is the number of class labels 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 Baselines

5.1 Human Performance

The human performance on each task can be used as a performance upper bound that shows both the inherent ambiguity in the task and the limitations of the particular data set. 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 Amazon Mechanical Turk. We asked three workers to

Task	Agreement	Example Question
VPC Classification	84.17%	*I feel there are others far more suited to* ***take on*** *the responsibility.* What is the verb in the highlighted span? (take/take on)
LVC Classification	83.78%	*Jamie* ***made a decision*** *to drop out of college.* Mark all that apply to the highlighted span in the given context: 1. It describes an action of ''*making something*'', in the common meaning of ''*make*''. 2. The essence of the action is described by ''*decision*''. 3. The span could be rephrased without ''*make*'' but with a verb like ''*decide*'', without changing the meaning of the sentence.
NC Literality	80.81%	*He is driving down memory* ***lane*** *and reminiscing about his first love.* Is ''lane'' used literally or non-literally? (literal/non literal)
NC Relations	86.21%	*Strawberry shortcakes were held as celebrations of the* ***summer fruit*** *harvest.* Can ''summer fruit'' be described by ''fruit that is ripe in the summer''? (yes/no)
AN Attributes	86.42%	*Send my* ***warm*** *regards to your parents.* Does ''warm'' refer to temperature? (yes/no)

Table 3: The worker agreement (%) and the question(s) displayed to the workers in each annotation task.

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 human intelligence tasks, and had them pass a qualification test.

In each annotation task, we showed the workers the context sentence with the target span highlighted, and asked them questions regarding the target span, as exemplified in Table 3. In addition to the possible answers given in the table, annotators were always given the choice of ''I can't tell'' or ''the sentence does not make sense''.

Of all the annotation tasks, the LVC classification task was more challenging and required careful examination of the different criteria for LVCs. In the example given in Table 3 with the candidate LVC *make a decision*, we considered a worker's answer as positive (LVC) either if: 1) the worker indicated that *make a decision* does not describe an action of *making something* AND that the essence of *make a decision* is in the word *decision*; or 2) if the worker indicated that *make a decision* can be replaced in the given sentence with *decide* without changing the meaning of the sentence. The second criterion was given in the original guidelines of Tu and Roth (2011). The replacement verb *decide* was selected as it is linked to *decision* in WordNet in the derivationally related relation.

We didn't compute the estimated human performance on the phrase type task, which is more complicated and requires expert annotation.

5.2 Majority Baselines

We implemented three majority baselines:

- Majority$_{ALL}$ is computed by assigning the most common label in the training set to all the test items. Note that the label distribution may be different between the train and test sets, resulting in accuracy $< 50\%$ even for binary classification tasks.
- Majority$_1$ assigns for each test item the most common label in the training set for items with the same first constituent. For example, in the VPC classification task, it classifies *get through* as positive in all its contexts because the verb *get* appears in more positive than negative examples.
- Majority$_2$ symmetrically assigns the label according to the last (typically second) constituent.

6 Results

Table 4 displays the best performance of each model family on the various tasks.

Representations. The general trend across tasks is that the performance improves from the majority baselines through word embeddings and to the contextualized word representations, with a large gap in some of the tasks. Among the contextualized word embeddings, BERT performed best on four out of six tasks, with no consistent preference between ELMo and the OpenAI GPT. The best

Model Family	VPC Classification	LVC Classification	NC Literality	NC Relations	AN Attributes	Phrase Type
	Acc	*Acc*	*Acc*	*Acc*	*Acc*	F_1
Majority Baselines	23.6	43.7	72.5	50.0	50.0	26.6
Word Embeddings	60.5	74.6	80.4	51.2	53.8	44.0
Contextualized	**90.0**	**82.5**	**91.3**	**54.3**	**65.1**	**64.8**
Human	93.8	83.8	91.0	77.8	86.4	-

Table 4: Summary of the best performance of each family of representations on the various tasks. The evaluation metric is accuracy except for the phrase type task in which we report span-based F_1 score, excluding O tags.

Model	VPC Classification		LVC Classification		NC Literality		NC Relations		AN Attributes		Phrase Type	
	Layer	Encoding	Layer	Encoding	Layer	Encoding	Layer	Encoding	Layer	Encoding	Layer	Encoding
ELMo	All	Att	All	biLM	All	Att/None	**Top**	**biLM**	All	None	All	biLM
OpenAI GPT	All	None	**Top**	**Att/None**	Top	None	All	biLM	Top	None	All	biLM
BERT	**All**	**Att**	All	biLM	**All**	**Att**	All	None	**All**	**None**	**All**	**biLM**

Table 5: The best setting (layer and encoding) for each contextualized word embedding model on the various tasks. **Bold** entries are the best performers on each task.

Model	VPC Classification	LVC Classification	NC Literality	NC Relations	AN Attributes	Phrase Type
word2vec	biLM	Att	biLM/Att	None	-	None/biLM
GloVe	**biLM**	**Att**	**Att**	biLM	-	**biLM**
fastText	Att	biLM	biLM	**biLM**	**Att**	biLM

Table 6: The best encoding for each word embedding model on the various tasks. **Bold** entries are the best performers on each task. Dash marks no preference.

word embedding representations were GloVe (4 of 6) followed by fastText (2 of 6).

Phenomena. The gap between the best model performance (achieved by the contextualized representations) and the estimated human performance varies considerably across tasks. The best performance in NC Literality is on par with human performance, and only a few points short of that in VPC Classification and LVC Classification. This is evidence for the utility of contextualized word embeddings in detecting meaning shift, which has positive implications for the yet unsolved problem of detecting MWEs.

Conversely, the gap between the best model and the human performance is as high as 23.5 and 21.3 points in NC Relations and AN Attributes, respectively, suggesting that tasks requiring revealing implicit meaning are more challenging to the existing representations.

Layer. Table 5 elaborates on the best setting for each representation on the various tasks. In most cases, there was a preference to learning a scalar mix of the layers rather than using only the top layer. We extracted the learned layer weights for each of the All models, and found that the model usually learned a balanced mix of the top and bottom layers.

Encoding. We did not find one encoding setting that performed best across tasks and contextualized word embedding models. Instead, it seems that tasks related to meaning shift typically prefer Att or no encoding, whereas tasks related to implicit meaning performed better with either biLM or no encoding.

When it comes to word embedding models, Table 6 shows that biLM was preferred more often. This is not surprising. A contextualized word embedding of a certain word is, by definition, already aware of the surrounding words, obviating the need for a second layer of order-aware encoding. A word embedding based model, on the other hand, must rely on a biLSTM to learn the same.

Finally, the best settings on the Phrase Type task use `biLM` across representations. It may suggest that predicting a label for each word can benefit from a more structured modelling of word order. Looking into the errors made by the best model (BERT+All+biLM) reveals that most of the errors were predicting `O`, that is, 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. Ten percent were missed (classified as `O`) 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 `O`) and the latter only 8% (62% were classified as `O`). Overall, it seems that the model relied mostly on syntactic cues, failing to recognize semantic subtleties such as idiomatic meaning and level of compositionality.

7 Analysis

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

7.1 Meaning Shift

Does the representation capture VPCs? The best performer on the VPC Classification task was the BERT+All+Att. To get a better idea of the signal that BERT contains for VPCs, we chose several ambiguous verb-preposition pairs in the data set. We define a verb-preposition pair as ambiguous if it appeared in at least eight examples as a VPC and at least eight examples as a non-VPC. For a given pair we computed the BERT representations of the sentences in which it appears in the data set, and, similarly to the model, we represented the pair as the concatenation of its word vectors. In each vector we averaged the layers using the weights learned by the model. Finally, we projected the computed vectors into 2D space using t-SNE (Maaten and Hinton, 2008). Figure 2 demonstrates four example pairs. The other pairs we plotted had similar t-SNE plots, confirming that the signal for separating different verb usages comes directly from BERT.

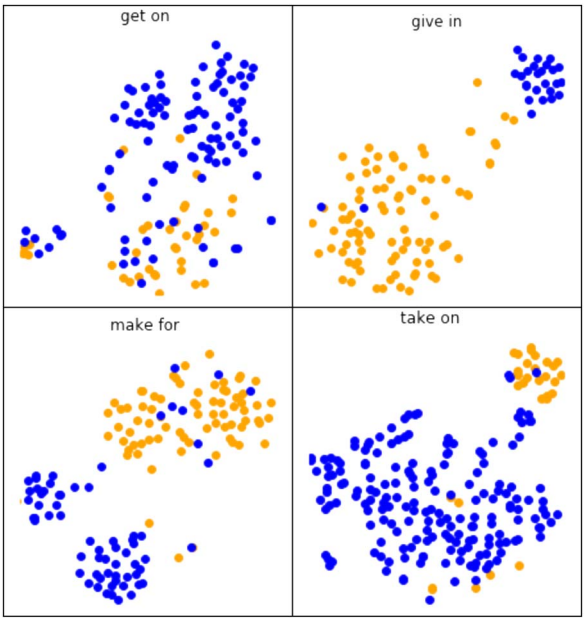

Figure 2: t-SNE projection of BERT representations of verb-preposition candidates for VPC. Blue (dark) points are positive examples and orange (light) points are negative.

Non-literality as a rare sense. Nunberg et al. (1994) considered some non-literal compounds as ''idiosyncratically decomposable'', that is, 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 could possibly use the NC literality data for word sense induction, in which recent work has shown that contextualized word representations are successful (Stanovsky and Hopkins, 2018; Amrami and Goldberg, 2018). We are interested in testing not only whether the contextualized models are capable of detecting rare senses induced by non-literal usage, which we have confirmed in Section 6, but whether they can also model these senses. To that end, 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 7 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.

ELMo		OpenAI GPT		BERT		ELMo		OpenAI GPT		BERT	
The Queen and her husband were on a train [trip]$_L$ from Sydney to Orange.						Creating a guilt [trip]$_N$ in another person may be considered to be psychological manipulation in the form of punishment for a perceived transgression.					
ride	1.24%	to	0.02%	**travelling**	19.51%	tolerance	0.44%	that	0.03%	**reaction**	8.32%
carriage	1.02%	**headed**	0.01%	**running**	8.20%	**fest**	0.23%	so	0.02%	**feeling**	8.17%
journey	0.73%	**heading**	0.01%	**journey**	7.57%	avoidance	0.16%	**trip**	0.01%	attachment	8.12%
heading	0.72%	that	0.009%	**going**	6.56%	onus	0.15%	he	0.01%	**sensation**	4.73%
carrying	0.39%	and	0.005%	**headed**	5.75%	association	0.14%	she	0.01%	note	3.31%
Richard Cromwell so impressed the king with his valour, that he was given a [diamond]$_L$ ring from the king's own finger.						She became the first British monarch to celebrate a [diamond]$_N$ wedding anniversary in November 2007.					
diamond	0.23%	and	0.01%	**silver**	15.99%	customary	0.20%	new	0.11%	royal	1.58%
wedding	0.19%	of	0.01%	**gold**	14.93%	royal	0.17%	british	0.02%	1912	1.23%
pearl	0.18%	to	0.01%	**diamond**	13.18%	sacrifice	0.15%	victory	0.01%	recent	1.10%
knighthood	0.16%	ring	0.01%	**golden**	12.79%	400th	0.13%	french	0.01%	1937	1.08%
hollow	0.15%	in	0.01%	new	4.61%	**10th**	0.13%	royal	0.01%	1902	1.08%
China is attempting to secure its future [oil]$_L$ share and establish deals with other countries.						Put bluntly, I believe you are a snake [oil]$_N$ salesman, a narcissist that would say anything to draw attention to himself.					
beyond	0.44%	in	0.01%	**market**	98.60%	auto	0.52%	in	0.05%	**oil**	32.5%
engagement	0.44%	and	0.01%	**export**	0.45%	egg	0.42%	and	0.01%	pit	2.94%
market	0.34%	for	0.01%	**trade**	0.14%	hunter	0.42%	that	0.01%	bite	2.65%
nuclear	0.33%	**government**	0.01%	**trading**	0.09%	consummate	0.39%	of	0.01%	skin	2.36%
majority	0.29%	**supply**	0.01%	**production**	0.04%	rogue	0.37%	charmer	0.008%	jar	2.23%

Table 7: Top substitutes for a target word in literal (left) and non-literal (right) contexts, along with model scores. **Bold** words are words judged reasonable (not necessarily meaning preserving) in the given context, and underlined words are suitable substitutes for the entire noun compound, but not for a single constituent.

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. Although 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, although contextualized representations excel at detecting shifts from the common meanings of words, their ability to obtain meaningful representations for such rare senses is much more limited.

7.2 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? 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 to 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 (-X signifies the ablation of the X feature):

1. **-Phrase**: where we mask the phrase in its context sentence, for example, replacing *Today, the house has become a wine bar*

	NC Relations	AN Attributes
Majority	50.0	50.0
-Phrase	50.0	55.66
-Context	45.06	63.21
-(Context+Phrase)	45.06	59.43
Full Model	54.3	65.1

Table 8: Accuracy scores of ablations of the phrase, context sentence, and both features from the best models in the NC Relations and AN Attributes tasks (ELMo+Top+biLM and BERT+All+None, respectively).

or bistro called Barokk with *Today, the house has become a* ***something*** *or bistro called Barokk*. Success in this setting may indicate that the implicit information is given explicitly in some of the context sentences.[3]

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 8 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, whereas the context is secondary.

8 Related Work

Probing Tasks. One way to test whether dense representations capture a certain linguistic property is to design a probing task for this property, and build a model that takes the tested 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 on specific linguistic properties such as agreement (Giulianelli et al., 2018), and tense (Bacon and Regier, 2018).

More recently, Tenney et al. (2019) and Liu et al. (2019) each 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. Tenney et al. (2019) found that all the models produced strong representations for syntactic phenomena, but gained smaller performance improvements upon the baselines in the more semantic tasks. Liu et al. (2019) found that some tasks (e.g., identifying the tokens that comprise the conjuncts in a coordination construction) required fine-grained linguistic knowledge which was not available in the representations unless they were fine-tuned for the task. 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), verb-particle constructions (e.g., Baldwin, 2005; Pichotta and DeNero, 2013), and light verb constructions (e.g., Tu and Roth, 2011; Chen et al., 2015).

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

[3] A 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.

compositions (Hartung et al., 2017) and in noun compounds (Surtani and Paul, 2015; Dima, 2016; Shwartz and Waterson, 2018; Shwartz and Dagan, 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 yet been used for tasks related to lexical composition.

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 that it operates on phrases with a predefined number of words. Moreover, we can expect the resulting compositional vectors to capture properties inherited from the constituent words, but it is unclear whether they also capture new properties introduced by 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). Although they don't have constraints on the phrase length, these methods still suffer from the two other drawbacks above: 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.

9 Discussion and Conclusion

We have shown that contextualized word representations perform generally better than static word embeddings on tasks related to lexical composition. However, although they are on par with human performance in recognizing meaning shift, they are still far from that in revealing implicit meaning. This gap may suggest a limit on the information that distributional models currently provide about the meanings of phrases.

Going beyond 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 used 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 level of context modeling seems to be beyond the scope of current text representations.

We expect that improving the ability of representations to reveal implicit meaning will require training them to handle this specific phenomenon. Our evaluation suite, the data, and code will be

made available. It is easily extensible, and may be used in the future to evaluate new representations for their ability to address lexical composition.

Acknowledgments

This work was supported in part by an Intel ICRI-CI grant, the Israel Science Foundation (grant 1951/17), the German Research Foundation, and the German-Israeli Project Cooperation (DIP, grant DA 1600/1-1). Vered is also supported by the Clore Scholars Programme (2017).

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer 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 biLM and symmetric patterns. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4860–4867, Brussels.

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.

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.

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.

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.

Wei-Te Chen, Claire Bonial, and Martha Palmer. 2015. English light verb construction identification using lexical knowledge. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* 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.

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.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, MN.

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.

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 natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne.

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.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme 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 Jebbara, 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.

Iris Hendrickx, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó 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.

Otto Jespersen. 1965. *A Modern English Grammar: On Historical Principles*, George Allen & Unwin Limited.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew Peters, and Noah A. Smith. 2019. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, MN.

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, Beijing.

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 Ó Séaghdha, 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.

Diarmuid Ó Séaghdha and Ann Copestake. 2009. Using lexical and relational similarity to classify semantic relations. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 621–629, Athens.

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.

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, LA.

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, WA.

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: Volume 2, Short Papers*, pages 503–508, Valencia.

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.

Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword expressions: A pain in the neck for NLP. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 1–15.

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, CO.

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, CO.

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 Computation in 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.

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.

Vered Shwartz and Ido Dagan. 2018. Paraphrase to Explicate: Revealing Implicit Noun-Compound Relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1200–1211, Melbourne.

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, LA.

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.

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.

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. 2011. Learning English light verb constructions: Contextual or Statistical. In *Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World*, pages 31–39, Portland, OR.

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.

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.

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.

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.

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.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial data set for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels.