An Unsupervised Method for Learning Representations of Multi-word Expressions for Semantic Classification

Robert Vacareanu,[1,2] **Marco A. Valenzuela-Escaŕcega,**[1] **Rebecca Sharp,**[1] **and Mihai Surdeanu**[1]
[1] University of Arizona, Tucson, AZ, USA
[2] Technical University of Cluj-Napoca, Cluj-Napoca, Cluj, Romania
{rvacareanu,marcov,bsharp,msurdeanu}@email.arizona.edu

Abstract

This paper explores an unsupervised approach to learning a compositional representation function for multi-word expressions (MWEs), and evaluates it on the Tratz dataset, which associates two-word expressions with the semantic relation between the compound constituents (e.g. the label *employer* is associated with the noun compound *government agency*) (Tratz, 2011). The composition function is based on recurrent neural networks, and is trained using the Skip-Gram objective to predict the words in the context of MWEs. Thus our approach can naturally leverage large unlabeled text sources. Further, our method can make use of provided MWEs when available, but can also function as a completely unsupervised algorithm, using MWE boundaries predicted by a single, domain-agnostic part-of-speech pattern. With pre-defined MWE boundaries, our method outperforms the previous state-of-the-art performance on the coarse-grained evaluation of the Tratz dataset (Tratz, 2011), with an F1 score of 50.4%. The unsupervised version of our method approaches the performance of the supervised one, and even outperforms it in some configurations.

1 Introduction

Multi-word expressions (MWEs) are fundamental to language and, as such, having a robust semantic representation for MWEs is important for any natural language processing task that involves text understanding such as information extraction, or question answering (e.g., da Silva and Souza, 2012; Thurmair, 2018; Subramanian et al., 2018). While MWEs have received attention in recent years, leading to considerable progress in learning MWE representations (Mitchell and Lapata, 2010; Butnariu et al., 2010; Tratz, 2011; Hendrickx et al., 2013; Dima, 2016; Shwartz and Dagan, 2018; Shwartz, 2019), we argue that the proposed methods have limitations. First, some methods require concatenating words in specified MWEs, and treating the resulting MWE phrases as atomic units. Training a set of dedicated distributional embeddings for the new multi-word terms (Shwartz and Dagan, 2018; Dima, 2016) suffers from language sparsity. For example, the MWE "red flower" is two orders of magnitude less frequent in Google search results than the noun "flower," which is likely to affect the quality of its learned MWE representation. These methods additionally have no straightforward way of handling MWEs that are out of vocabulary. Second, other approaches require supervision for MWE boundaries (Yu and Dredze, 2015), which hinders scalability and portability to different languages. In all situations, the reliance on having determined your entities of interest ahead of time threatens to dramatically reduce the real-world utility of these approaches.

Here, we propose a method that addresses both limitations by combining recent advances in language modeling (Howard and Ruder, 2018; Merity et al., 2017) with the simplicity and proven capability of the Skip-Gram training objective (Mikolov et al., 2013a). Our approach is summarized in Figure 1. Intuitively, our method has two components. We use a bidirectional long short-term memory network (biLSTM) (Hochreiter and Schmidhuber, 1997) to encode each MWE. Then, we use this encoding to predict the words in the context of the MWE, similarly to the original Skip-Gram algorithm. Importantly,

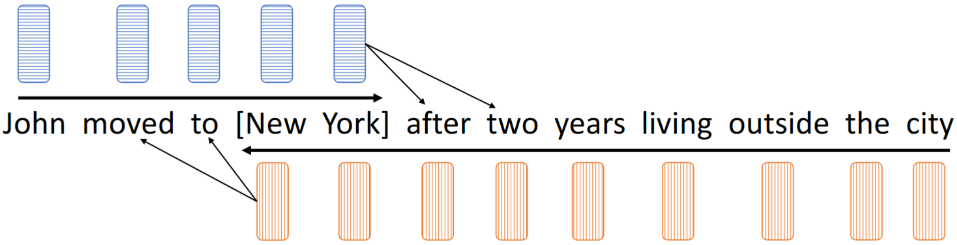
John moved to [New York] after two years living outside the city

Figure 1: Proposed architecture for learning MWE representations using a biLSTM. During training of the language model, for a given sentence that contains a MWE, e.g., *New York*, we use Skip-Gram to maximize the probability of the words in the right context, e.g., *after* and *two*, using the hidden state of *York* produced by the left-to-right (forward) LSTM, and of the words in the left context, e.g., *moved* and *to*, using the hidden state of *New* produced by the right-to-left (backward) LSTM. To generate a represention for the MWE for downstream tasks, we average the last biLSTM hidden states for the MWE, e.g., the hidden state for *York* from the left-to-right LSTM and the hidden state for *New* from the right-to-left LSTM.

our approach can use predicted MWE boundaries. In particular, here we use MWE boundaries predicted by a *single* domain-independent pattern over part-of-speech (POS) tags. This approach allows us to learn a context-aware MWE composition function that can be trained in an unsupervised way, maximizing its utility across domains and languages.

Specifically, the contributions of this work are:

(1) We introduce a straightforward approach for learning a contextualized composition function for MWEs that can make use of provided MWE boundaries when available, but which does not require them. Critically, our approach does not rely on training against previously learned distributional MWE embeddings in either setting. While here we apply our approach to English two-word expressions, we are optimistic that our method can generalize to multi-word expressions of arbitrary length. Similarly, the unsupervised form of our method has the potential to work with other languages, provided that there exists a part-of-speech tagger, and that an extraction pattern for MWEs based on part of speech tags is possible.

(2) We show that our approach that uses pre-defined MWE boundaries outperforms the previous state-of-the-art performance on the coarse-grained evaluation of the Tratz dataset (Tratz, 2011) with 50.40% F1 score. Our method is marginally behind one based on transformers (by 3% F1 points), but it is much faster, both during training and inference, and it has a much smaller memory footprint. Further, our unsupervised algorithm, which relies on MWE boundaries predicted by a single domain-independent part-of-speech pattern, has minimal performance loss as compared to the boundary-aware version, underperforming it by only 0.43 F1 on average on the fine-grained lexical evaluation, while gaining even more generality.

2 Related Work

Much like distributional similarity approaches for learning word representations (Mikolov et al., 2013a; Bojanowski et al., 2017; Pennington et al., 2014, *inter alia*), a semantic representation of MWEs can be trained using a distributional approach that treats MWEs as single tokens (Mikolov et al., 2013b). However, this approach cannot handle out of vocabulary (OOV) MWEs, and it is likely to suffer from sparsity (Shwartz, 2019), particularly as the MWEs grow in length.

On the other hand, compositional approaches address these issues by learning a function to compose the representations of the MWE constituent words. Dima (2016) proposes a compositional method which minimizes the L2 distance between the predicted, compositional embedding of the MWE and an observed, distributional one. However, while there are two methods for obtaining the distributional embedding, both suffer from an observer effect. In the first method, a single set of embeddings is trained and the corpus is modified such that sentences that contain MWEs are included in their original form as well as a copy with the MWE treated as a single token. This augmentation, however, alters the relative frequency of the words, which consequently affects the embeddings of the MWE constituents. In the second method, two distinct sets of embeddings are trained, one with no alteration and one with MWEs treated as single tokens. Thus the embeddings of the constituents are unaffected, but since the words in the two corpora have different relative frequencies, the resultant vector spaces are different. Our proposed approach does not suffer from this observer effect, as we learn our compositional function indirectly (through Skip-Gram), without relying on a distributionally learned embedding for MWEs for training.

Shwartz (2019) also avoid this reliance on the gold embedding of the multi-word, learning the function indirectly. The compositional function is used to encode the multiword and its paraphrase and is then trained to maximize the cosine similarity between the encodings. The paraphrases are generated either using backtranslation (Wieting et al., 2017; Wieting and Gimpel, 2018), or by treating frequent joint co-occurrences as paraphrases. However, this approach is outperformed by much cheaper unsupervised approaches, such as the average of the constituents' emebddings (Shwartz, 2019). Furthermore, the backtranslation approach depends on an external system, adding complexity to the model and restricting the languages and domains of application.

Alternatively, a compositional function can be trained directly with a language modeling objective, leveraging the vast amounts of available unlabeled data. Our proposed approach falls into this category, and is similar in nature with the work of Yu and Dredze (2015), as both approaches use an adapted Skip-Gram to learn a composition function. However, there are two key differences: they used a series of hand-crafted features, such as word clusters, while we rely only on word embeddings. Second, they assume that the word boundaries are given, while our system can work in a completely unsupervised setting, allowing it to be applied to domains, and, potentially, other languages, with no pre-determined set of entities of interest. Further, though they explore a recurrent neural network (RNN) variant, it relies on the availability of a constituency parser (which is unavailable for many domains and languages), is unable to scale (as noted by Yu and Dredze), and still requires knowing the MWE boundaries ahead of time. We address all of these limitations with our approach.

3 Approach

We propose an approach for extending the Skip-Gram method (Mikolov et al., 2013a) to handle multiword expressions (MWEs). While Mikolov et al. (2013b) proposed an extension that handles multiwords by treating them as *single tokens* during training, MWEs outside that training vocabulary have no representation. We propose an unsupervised method for learning a composition function capable of producing a representation of a MWE from the embeddings of its components, using bidirectional recurrent neural networks (RNNs).

3.1 Architecture

Our approach operates over the full sentence, and outputs a context-aware vector representation of the multi-word. We train our composition function using the standard Skip-Gram method, i.e., predicting the words in the context of the MWE. We obtain the MWE representation using a biLSTM over the sentence, such that to predict the right-contex of the MWE we use the forward LSTM hidden state of the right-most MWE constituent (*York* in the example sentence in Figure 1), and to predict the left-context we use the backward LSTM hidden state of the left-most constituent (*New*). In this way, the individual LSTMs haven't seen the context they are predicting. During inference, we average the last hidden states of the two LSTMs, at the boundaries of the MWE.[1]

To formalize, for a sentence S consisting of n words $[w_1, .., w_n]$, containing a multi-word expression of interest of length k, $[w_{e_1}, .., w_{e_k}]$, we train a RNN-based function f to output an embedding capable of predicting the context to the left of w_{e_1}, and the context to the right of w_{e_k}:

$$h^{(e_k)} = RNN_f([E(w_1), .., E(w_{e_k})]) \tag{1a}$$

$$h^{(e_1)} = RNN_b([E(w_n), .., E(w_{e_1})]) \tag{1b}$$

$$f(w_{e_1}, .., w_{e_k}; S) = (h^{(e_k)} + h^{(e_1)})/2 \tag{1c}$$

[1] Note that our proposed training technique is agnostic to the type of function used to generate the embedding of the MWE, and can support any function capable of mapping variable-length word input into a static-length vector.

where RNN_f is the forward LSTM, which traverses the text left-to-right, and RNN_b is the backward LSTM (traversing right-to-left). Each LSTM is trained to minimize the following Skip-Gram-like objective function:

$$log(\sigma(u_o^T\ m))+\sum_{i=1}^{w}\mathbb{E}_{w_i\ \ P_n(w)}[log(\sigma(-u_i^T\ m))]$$

where w represents the context window length, u_o represents the embedding of a word in the context, u_i represents the embedding of a randomly sampled negative example from the word distribution $P_n(w)$,[2] and m is the last/first hidden state of the forward/backward RNN for the corresponding MWE. The final loss is then the average of the losses of the forward and backward LSTMs for all MWEs identified in the training dataset.

We investigated two approaches for modeling the context in which the MWE appears at testing time. The first approach applies the learned composition function over the MWE *alone* during testing. This is necessary as the MWEs in this dataset appear in isolation, without any context. The second approach associates the MWE with context extracted automatically from a large unstructured corpus (Section 3.4).

Following Shwartz (2019), for our experiments we used FastText embeddings (Bojanowski et al., 2017), which, overall, performed the best on the Tratz dataset (Tratz, 2011). The embeddings were trained on the English Wikipedia dump from January 2018[3] to facilitate comparison between this work and that of Shwartz (2019).

3.2 Multi-word Expression Boundaries

Our approach can incorporate and use any set of MWE boundaries during training. One direction follows previous work, and uses the predefined MWEs in the provided Tratz dataset. The second trains with no supervised knowledge of the MWE boundaries, but rather for a fully unsupervised approach, we obtain silver MWE boundaries using a single pattern over part-of-speech tags: `(JJ | NN) NN`. That is, we require a sequence of either two nouns, or an adjective followed by a noun.[4] This method of obtaining silver MWEs for training allows our approach to work with other domains or languages, provided that there a part-of-speech tagger is available,[5] and that such an extraction pattern is reasonably straightforward to write (e.g., in Spanish an adjective often follows the noun rather than preceding it). Alternatively to the rule-based extraction, the system proposed in Boukobza and Rappoport (2009) for MWE identification may be used, provided that there is an initial set of MWEs available.

3.3 Training Variations

While we found empirically that running the mapping function over the whole sentence to produce a context-aware embedding of the MWE performs better overall, we also experimented with a variant that uses only the multi-word as input to an RNN. Formally, using the same notation as above, where we have a sentence S consisting of n words, which contains a MWE of length k, we take the last hidden state of a word-level RNN as the embedding of the entire expression:

$$\begin{aligned} h^{(e_k)} &= RNN([E(w_{e_1}),..,E(w_{e_k})]) \\ f(w_{e_1},..,w_{e_k}) &= h^{(e_k)} \end{aligned}$$

where RNN is a forward LSTM. Note that because the MWEs tend to be relatively short, we did not find it beneficial to use a bidirectional RNN in this setting.

3.4 Context during evaluation

We extract l sentences at random containing the MWE from this unstructured corpus, and apply the composition function over each, generating n candidate embeddings. The final embedding for the MWE is then the mean of these embeddings.

[2] We used unigram distribution raised to 3/4, same as Mikolov et al. (2013b).

[3] `https://dumps.wikimedia.org`

[4] We focus on MWEs of length 2 simply because the Tratz dataset used in this work contains MWEs of this length. This pattern can obviously be generalized to arbitrary lengths.

[5] We used CLU lab's processors software, available at `https://github.com/clulab/processors` to generate POS tags in this paper.

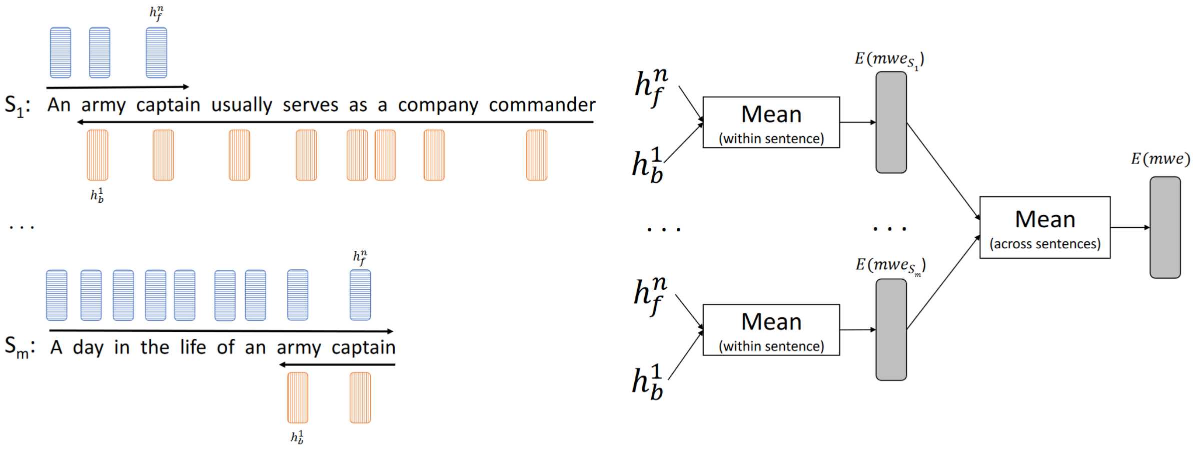

Figure 2: During the evaluation of our context-aware MWE approach, we extract contextual information from sentences from an unstructured corpus that contain the corresponding MWE. In each sentence, we construct the embedding of the MWE by averaging the last hidden state of the forward LSTM and the last hidden state of the backward LSTM. Lastly, to create a single embedding for the MWE, we aggregate all these embeddings using the average function.

Split	# MWE	# Labels
coarse-grained lexical	7102	12
fine-grained lexical	7179	37
coarse-grained random	18693	12
fine-grained random	19058	37

Table 1: The number of MWE (all of length two) for each split in the Tratz dataset Tratz (2011). Note that the fine-grained and coarse-grained splits are distinct datasets, but there is a certain level of overlap between each split. For example, the *coarse-grained lexical* split and the *fine-grained lexical* split have 37% of MWEs in common.

Since our proposed method was trained over complete sentences, we additionally propose variant that augments the MWEs with context at testing time. Specifically, for each MWE, we randomly sample sentences containing the MWE from the English Wikipedia dump.[6] As illustrated in Figure 2, we process each of these sentences independently, using the RNNs summarized in Equations (1) , and average the resulting embeddings from each.[7]

4 Experiments and Results

Similar to Shwartz (2019), we evaluate the quality of the learned MWE representations on the task and dataset introduced in Tratz (2011). This is a multi-class classification problem, where the task is to classify the semantic relation between the head of the MWE and the modifier. (e.g., *roof tile* is labeled as `location`, as *roof* provides the location of the *tile*). The Tratz dataset has two splits between train and test: *random* (where the MWEs are randomly split between train and test), and *lexical* (where they are assigned based on their head word such that there is no lexical overlap between train and test). Each of these splits can also be evaluated with coarse-grained or fine-grained labels. For example, *security fear* is labeled as *topical* in the coarse-grained lexical dataset, and as *topic_of_cognition&emotion* in the fine-grained dataset lexical. In Table 1 we list the total number of unique MWE of length two and labels for each split in the Tratz dataset (Tratz, 2011). Note that the fine-grained and coarse-grained splits are distinct datasets, with an overlap of only 2688 MWEs, or roughly 37%, between them. For the random split, the fine-grained split contains all the MWEs from the coarse-grained split.

We report results on both the random and lexical splits of the data for completeness. However, we argue that the random split does not produce a representative evaluation, because methods that rely on

[6] We sample at most 100 sentences for each MWE. We favor sentences in which the MWE appears in a hypernymy-hyponymy relation according to the hypernymy patterns introduced by Hearst (1992), to increase the likelihood that the semantic class of the MWE is discussed.

[7] Other aggregation functions are possible as well. We also tried a max pool function, but the mean performed better in our experiments.

	Gold embed	Gold boundaries	Coarse-grained Lexical	Fine-grained Lexical	Coarse-grained Random	Fine-grained Random
Unsupervised Baselines						
Majority Train Head	n	n	0.0	0.0	60.83 ± 0.0	54.72 ± 0.0
Majority Label	n	n	6.56	5.38	7.37	5.19
Random	n	n	9.18 ±0.77	3.62 ±0.6	9.22 ±1.14	3.5 ±0.68
Supervised Baselines						
FastText Max Pool	n	n	39.25	32.3	54.62	53.31
FastText Average	n	n	41.72	35.06	58.13	56.86
Supervised						
Minimize L2 (our implementation)	y	y	43.14 ±2.52	33.85 ±0.18	61.00 ±0.18	60.75 ±0.36
Minimize L2 ◆ Shwartz (2019)	y	y	47.5	38.1	66.2	63.9
Shwartz and Waterson (2018)	n	y	47.8	**42.9**	73.6	71.4
Dima (2016)	y	y	37.2	33.4	**77.5**	**72.5**
No supervision						
MWE-only RNN	n	n	41.42 ±2.09	37.18 ±0.65	60.91 ±0.50	59.70 ±0.89
Context-aware RNN (train only)	n	n	43.51 ±2.28	36.53 ±0.74	59.73 ±0.79	57.97 ±0.58
Context-aware RNN (train/test)	n	n	44.38 ±0.46	37.66 ±0.78	59.57 ±0.78	57.85 ±0.49
Supervised boundaries						
MWE-only RNN	n	y	47.95 ±0.88	36.90 ±0.75	62.46 ±0.23	60.44 ±0.36
Context-aware RNN (train only)	n	y	50.40 ±2.88	37.82 ±1.41	62.38 ±0.20	60.76 ±0.64
Context-aware RNN (train/test)	n	y	49.75 ±2.50	38.09 ±0.76	62.20 ±0.17	60.50 ±0.16
Transformers						
BERT base (frozen)	n	n	**53.69**	40.89	67.11	64.48

Table 2: Weighted F1 scores (Shwartz, 2019) of our proposed approach, various baselines, and other supervised methods, including the state-of-the-art methods of Dima (2016) and Shwartz and Waterson (2018), on the Tratz dataset. We differentiate between methods that need supervision for both distributional embeddings and boundaries of MWEs (*supervised*), only MWE boundaries (*supervised boundaries*), and those that do not need either of them (*no supervision*). We trained two methods, *MWE-only RNN* which operates only over the tokens of the MWE, and *Context-aware RNN* which is trained over full sentences. We evaluated the *Context-aware RNN* in two settings. In the first setting it has no access to context at evaluation time, i.e., the RNNs run only over the MWE constituents. In the second setting, the method has access to context in the form of *n* sentences that contain the MWE at evaluation time. In this setting, our method averages the embeddings produced from all these *n* sentences, as illustrated in Figure 2. (◆) Evaluation without *lexicalized*, *personal_title*, *personal_name* relations Shwartz (2019);

simple memorization perform artificially high due to the lexical overlap between train and test. For example, a simple baseline that predicts the label most commonly seen in training with the last word (typically the head word in a two-word MWE) achieves near state-of-the-art performance in the random split. For this reason, we focus most of the discussion in this paper on results measured on the lexical split of data, where this overlap is avoided.

We evaluate and compare different variations of our method against previous work and baselines. As each approach creates a vector representation of the MWE, in order to predict the discrete labels required by the task, we fit a linear classifier on top of this vector. Similar to Shwartz (2019), we select the best performing classifier on the development partition from 10 possible classifiers.[8]

We implemented five baselines, which fall in two categories: (a) baselines without a classifier, and (b) baselines that employ a linear classifier on the predicted MWE embedding (similar to our actual method). The first class of baselines includes: (i) *Random*, which predicts the label by randomly sampling from the label distribution observed in training; (ii) *Majority Label*, which produces the majority label in the dataset, as observed in training; (iii) *Majority Train Head*, which predicts the label most commonly seen with the last word of the MWE, which is typically its syntactic head. The second category contains (i) *FastText Average*, where for a given MWE we train a linear classifier on top of the average of the FastText embedding of each individual word, and (ii) *FastText Max Pool*, which is similar to Average, but uses a max pooling function rather than the mean.

[8] We used 5 logistic regressions and 5 support vector machines, each with different hyperparameters, similar to Shwartz (2019).

We performed our empirical analysis under three high-level settings: (a) *Supervised*, where both distributional embeddings and boundaries of MWEs are required during training, (b) *Supervised boundaries*, where only MWE boundaries are provided (from the Tratz dataset), and (c) *No supervision*, which does not require either of them (and uses our POS pattern to generate silver boundaries). In the latter setting, we sampled 75 thousand MWEs generated using our POS pattern, using the same sampling heuristic as in Mikolov et al. (2013b). We made sure that these training MWEs do not overlap lexically with the MWE in the testing partition.

Our methods fall under the last two categories: supervised boundaries or no supervision. In each category, we evaluate three variants of our method: (a) *MWE-only RNN*, which does not use context at all, i.e., it applies the RNNs solely on the MWE constituent words, (b) *Context-aware RNN (train only)*, which uses contextual information only during training but not in testing, and (c) *Context-aware RNN (train/test)*, which has access to context during both training and testing. For the latter method, we considered up to 100 sampled sentences at evaluation time.

Additionally, we compared our approach against a classifier built on top of BERT (Devlin et al., 2018), a transformer based (Vaswani et al., 2017) language model. We used the base variant, which has 12 layers and produces an embedding of size 768. To stay close to the previous work on learning representations for MWEs, which focuses on the signal encoded in the MWE representations rather than fine tuning on the indirect evaluation task, we froze the underlying transformer model and trained a linear classifier head on top of it.

Table 2 lists the results of all methods investigated. For our method, we averaged three different runs, and included the mean result and standard deviation. For a fair comparison, we used the same hyperparameters for all methods under our control. For the Skip-Gram algorithm, we used the same window size as it was used for the training of the underlying word embeddings. Similarly, we set the size of the LSTMs' hidden state to be the same as the size of the word embedding.

To mitigate the sparsity of context at evaluation time, for *Context-aware RNN (train/test)* we used context during testing only if the number of sentences available in a given context is larger than a threshold T. That is, for MWEs with more than T sentences in their context we compute their embedding as explained in Section 3.4 and illustrated in Figure 2, namely by averaging the candidate embeddings; for MWEs with fewer than T sentences in their context we completely disregard the context, computing the embedding only by running the model over the MWE constituents.[9] We motivate the need for this threshold in the next section.

We draw the following observations from the results in Table 2:

(1) First, we observe that the two random splits of the data yield unrealistic evaluations that are artificially easy due to the lexical overlap between the training and testing partitions (Dima, 2016; Levy et al., 2015). Our Majority Train Head baseline, which simply memorizes the most common labels associated with head words, performs well, e.g., at over 60% accuracy for the coarse-grained evaluation. On the other hand, the lexical splits, where lexical overlap is avoided, produce hard, realistic evaluations, in which most baselines perform poorly. For this reason, we focus our efforts and discussion mostly on the evaluations that rely on the lexical splits.

(2) Our methods that use supervised boundaries (second to last block in the table) improve the state-of-the-art on coarse-grained lexical evaluation. Our *Context-aware RNN (train only)* method obtains the highest performance to date (to our knowledge) on the coarse-grained task, despite the little supervision required (only MWE boundaries). Our supervised-boundary methods perform worse than the state-of-the-art on the fine-grained lexical evaluation, e.g., approximately 5% (absolute) lower than Shwartz and Waterson (2018), but they do outperform other supervised methods that require more complex supervision.

(3) Our unsupervised methods (third to last block in the table) approach the performance of the corresponding variants with supervised boundaries on the fine-grained lexical evaluation, even slightly out-

[9] We tuned this threshold for each dataset and evaluation measure. We used threshold values of 5, 5, 10, 10 for the unsupervised approach, and 10, 5, 25, 10 for the supervised one, respectively. The listed thresholds are for each dataset, in order: coarse-grained lexical, fine-grained lexical, coarse-grained random.

performing them in some configurations. This demonstrates the value of large data for this task. Because we made sure that the MWEs generated by our simple POS pattern, which formed the silver training data for the unsupervised methods, do not overlap with expressions in test, but we did not control for domain or topicality otherwise, we are optimistic that our approach will generalize to other domains and, possibly, languages. We leave this investigation for future work.

(4) Context usually helps. In most situations, our context-aware methods outperform the equivalent MWE-only approaches. This emphasizes the validity of the distributional hypothesis for this semantic task (Harris, 1968). However, context is not always beneficial: as mentioned before, the *Context-aware RNN (train/test)* method enables context at testing time only if the number of context sentences for a given MWE is larger than a threshold hyper parameter. This is likely because language sparsity (Zipf, 1949) impacts negatively the modeling of context, when not enough evidence is available.

(5) The classifier that relies on transformers performs better than our approach, but not by much. It outperforms our proposed method by approximately 3% on the lexical tasks. However, please note that the methods are not exactly comparable. For example, we trained our method on approximately 12 million sentences, considerably fewer than the training data used for BERT base. Moreover, our proposed system consists of only one LSTM layer, producing embeddings of size 200 with a total of approximately 650,000 parameters, while BERT is more powerful, consisting of 12 layers, produces embeddings of size 768, and has a total of approximately 110 million parameters. Further, from a run-time perspective, our method is faster. At training time, our method takes approximately 1 hour per epoch. At inference time, it generates the embeddings of the train partition for all the splits in the Tratz dataset in approximately 7.5 seconds, while running BERT over the same input with the same settings takes approximately 200 seconds.[10] All in all, our method provides competitive performance with a small memory footprint and fast training and inference times.

MWE	Gold	Predicted
computer whiz	topic_of_expert	topic
computer analyst	means	topic_of_expert
navy diver	employer	means
peacetime growth	time-of1	objective
apple storage	objective	purpose
Marsha Baker	personal_name	personal_title
company strategy	experience-of-experience	perform&engage_in

Table 3: Hand-picked examples of incorrect predictions of the *Context-aware RNN* in the development set of the fine-grained lexical spit. As can be seen, even though the prediction is incorrect according to the gold label, it is often a sensible prediction.

5 Discussion

5.1 Qualitative Analysis

To better understand our results, we performed a small qualitative analysis of the fine-grained lexical instances that we incorrectly predicted as well as a larger, frequency-based analysis of the overall predictions. In Table 3 we present some examples that demonstrate some of the errors made by our *Context-aware RNN (train only)* model on the fine-grained lexical dataset. In many of the instances we saw, the distinction between the gold and predicted labels was hard to define. This is in line with the findings of Tratz and Hovy (2010), who found that inter-annotator agreement for this task is fairly low.

We also analyzed the relation between the frequency of the MWE constituents and the subsequent performance on the Tratz classification task. The median frequency of the constituent words of MWEs that led to correct predictions is 25% larger than the frequency of those which yield incorrect predictions. Moreover, we found a small subset of labels for which our context-aware method always predicts

[10] We used an nVidia Tesla P4.

incorrectly.[11] We hypothesize that this is because the constituent words from these MWEs occur less frequently in the training set. For example, the median frequency of the constituent words from this set is more than two times smaller than the median frequency obtained when considering the full dataset. Furthermore, we applied the Kolmogorov-Smirnov test to compare the frequency distributions of the constituents for the cases in which we predict correctly and incorrectly, respectively. We found that the two distributions are different,[12] suggesting that our approach tends to make more mistakes when the frequency of the constituents is small.

This is related to the finding that a threshold is useful for our Context-aware RNN (train/test). MWEs whose constituents are less frequent will also have fewer available context sentences. Since here we show that our approach does not perform equally across all frequency ranges, it provides support for the utility of the threshold, i.e., using slightly different approaches for the different situations.

5.2 Data Issues

We noticed an overlap between the merged training partition originally built by (Shwartz, 2019) and each individual test partition of the Tratz dataset (Tratz, 2011). We suspect that this happened because there can be overlap between the train partition of a split and the test partition of another split. For example, entities that appeared in the training partition of the coarse-grained lexical split can also appear in the test partition of the fine-grained lexical split. This affected the models trained with our proposed auxiliary task but did not affect the baselines nor the Transformer-based models. To confirm that this did not invalidate our results, we retrained our *Context-aware RNN (train only)* model in both *no supervision* and *supervised boundaries* settings. The results remained similar to our initial results. We observed an average difference between our initial results and the results on the strict dataset (where no overlap is allowed) of *0.03%* in the supervised setting and of *-0.49%* in the unsupervised setting respectively.[13] Because these differences are small and to keep our results comparable with previous work, we used the original dataset of (Shwartz, 2019) for all experiments reported in Table 2.

6 Conclusion

We proposed an unsupervised method to learn representations of MWEs, which is capable of leveraging vast amounts of unlabeled data. Specifically, we indirectly learn a compositional function generated by a bidirectional RNN by training it using the Skip-Gram training objective. Our proposed approach achieves better performance on the coarse-grained lexical task of the Tratz dataset (Tratz, 2011) than the previous approaches, and it is only 3% behind a classifier that relies on BERT, despite its simplicity, and minimal hyperparameter tuning. Code is available at `https://github.com/clulab/releases/tree/master/coling2020-mwe`.

Acknowledgments

We would like to thank the reviewers for their helpful comments. Marco Valenzuela-Escárcega, Rebecca Sharp, and Mihai Surdeanu declare a financial interest in lum.ai. This interest has been properly disclosed to the University of Arizona Institutional Review Committee and is managed in accordance with its conflict of interest policies. This material is based upon work supported by the National Science Foundation under Grant No. 2006583.

[11] The subset is {whole+attribute&feature&quality_value_is_characteristic_of, amountof, variety&genus_of, user_recipient, topic_of_expert, experiencer-of-experience, justification, topic_of_cognition&emotion, means, time-of2, personal_name, partial_attribute_transfer, obtain&access&seek} which contains a total of 87 data points.

[12] With a p-value of 0.006, smaller than our chosen threshold of 0.05 for significance.

[13] The performance we obtained for coarse-grained lexical, fine-grained lexical, coarse-grained random and fine-grained random are: 44.38 ± 0.51, 37.07 ± 0.29, 59.75 ± 0.97, 58.52 ± 0.76 in the unsupervised setting, and 50.35 ± 1.5, 37.94 ± 0.53, 62.62 ± 0.57, 60.33 ± 0.48 in the supervised case.

References

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Boukobza, R. and Rappoport, A. (2009). Multi-word expression identification using sentence surface features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 468–477, Singapore. Association for Computational Linguistics.

Butnariu, C., Kim, S. N., Nakov, P., Ó Séaghdha, D., Szpakowicz, S., and Veale, T. (2010). SemEval-2 task 9: The interpretation of noun compounds using paraphrasing verbs and prepositions. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 39–44, Uppsala, Sweden. Association for Computational Linguistics.

da Silva, E. M. and Souza, R. R. (2012). Information retrieval system using multiwords expressions (mwe) as descriptors. *Jistem Journal of Information Systems and Technology Management*, 9:213–234.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Dima, C. (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, Berlin, Germany. Association for Computational Linguistics.

Harris, Z. (1968). Mathematical structures of language. *Interscience tracts in pure and applied mathematics*.

Hearst, M. (1992). Automatic acquisition of hyponyms from large text corpora. In *Proceedings of The 15th International Conference on Computational Linguistics*.

Hendrickx, I., Kozareva, Z., Nakov, P., Ó Séaghdha, D., Szpakowicz, S., and Veale, T. (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, Atlanta, Georgia, USA. Association for Computational Linguistics.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Levy, O., Remus, S., Biemann, C., and Dagan, I. (2015). Do supervised distributional methods really learn lexical inference relations? In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 970–976, Denver, Colorado. Association for Computational Linguistics.

Merity, S., Keskar, N. S., and Socher, R. (2017). Regularizing and optimizing lstm language models. *arXiv preprint arXiv:1708.02182*.

Mikolov, T., Chen, K., Corrado, G. S., and Dean, J. (2013a). Efficient estimation of word representations in vector space.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26.

Mitchell, J. and Lapata, M. (2010). Composition in distributional models of semantics. *Cognitive science*, 34:1388–429.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543.

Shwartz, V. (2019). A systematic comparison of English noun compound representations. In *Proceedings of the Joint Workshop on Multiword Expressions and WordNet (MWE-WN 2019)*, pages 92–103, Florence, Italy. Association for Computational Linguistics.

Shwartz, V. and Dagan, I. (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, Australia. Association for Computational Linguistics.

Shwartz, V. and Waterson, C. (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.

Subramanian, S., Wang, T., Yuan, X., Zhang, S., Trischler, A., and Bengio, Y. (2018). Neural models for key phrase extraction and question generation. In *Proceedings of the Workshop on Machine Reading for Question Answering*, pages 78–88, Melbourne, Australia. Association for Computational Linguistics.

Thurmair, G. (2018). Multiword expressions in multilingual information extraction. In *Multiword Units in Machine Translation and Translation Technology*, pages 104–123. John Benjamins.

Tratz, S. (2011). Semantically-enriched parsing for natural language understanding.

Tratz, S. and Hovy, E. (2010). A taxonomy, dataset, and classifier for automatic noun compound interpretation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 678–687, Uppsala, Sweden. Association for Computational Linguistics.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *CoRR*, abs/1706.03762.

Wieting, J. and Gimpel, K. (2018). ParaNMT-50M: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462, Melbourne, Australia. Association for Computational Linguistics.

Wieting, J., Mallinson, J., and Gimpel, K. (2017). Learning paraphrastic sentence embeddings from back-translated bitext. In Palmer, M., Hwa, R., and Riedel, S., editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 274–285. Association for Computational Linguistics.

Yu, M. and Dredze, M. (2015). Learning composition models for phrase embeddings. *Transactions of the Association for Computational Linguistics*, 3(0):227–242.

Zipf, G. (1949). Human behavior and the principle of least effort. *Addison-Wesley, Cambridge*.