A Systematic Comparison of English Noun Compound Representations

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

Abstract

Building meaningful representations of noun compounds is not trivial since many of them scarcely appear in the corpus. To that end, composition functions approximate the distributional representation of a noun compound by combining its constituent distributional vectors. In the more general case, phrase embeddings have been trained by minimizing the distance between the vectors representing paraphrases. We compare various types of noun compound representations, including distributional, compositional, and paraphrase-based representations, through a series of tasks and analyses, and with an extensive number of underlying word embeddings. We find that indeed, in most cases, composition functions produce higher quality representations than distributional ones, and they improve with computational power. No single function performs best in all scenarios, suggesting that a joint training objective may produce improved representations.

1 Introduction

The simplest way to obtain a vector representation for a multiword term is to treat it as a single token, e.g. by replacing spaces with underscores, and train a standard word embedding algorithm. This is typically done for common n-grams, which often include named entities (e.g. New York), but in theory can also be based on syntactic criteria, for instance in order to learn noun compound vectors. The main issue with this approach is that word embedding algorithms require sufficient term frequency to obtain meaningful representations, and many noun compounds rarely occur in text corpora (Kim and Baldwin, 2006).

To overcome the sparsity issue, it is common to learn a composition function which computes a noun compound vector from its constituents' distributional representations, e.g. vec(*cost estimate*) = f(vec(*cost*), vec(*estimate*)). Various functions have been proposed in the literature, typically based on vector arithmetics (e.g. Mitchell and Lapata, 2010; Zanzotto et al., 2010; Dinu et al., 2013). Such functions are learned with the objective of minimizing the distance between the observed (distributional) vector and the composed vector of each noun compound, and most functions are limited to binary noun compounds.

A parallel line of work computes phrase embeddings for variable-length phrases, by adapting the word embedding training objective (Poliak et al., 2017) or by minimizing the distance between the representations of paraphrases (Wieting et al., 2016; Wieting and Gimpel, 2017; Wieting et al., 2017). Paraphrase-based phrase embeddings require a large number of paraphrases as training instances. Such paraphrases are often generated by translating an English phrase into a foreign language and back to English, considering variations in translation as paraphrases. This technique is referred to as "bilingual pivoting" or "backtranslation" (Barzilay and McKeown, 2001; Bannard and Callison-Burch, 2005; Ganitkevitch et al., 2013; Mallinson et al., 2017).

In this work we test the quality of noun compound representations produced by different methods, including distributional representations, composition functions, and paraphrase-based phrase embeddings. We extend the work of Dima (2016), who evaluated various composition functions on the noun compound relation classification task, in several aspects. First, we test a broader range of representations, which may differ both in their architectures and in their training objectives. Second, we train each representation with a wide variety of underlying word embeddings, and analyze the representation's behaviour across the different word embeddings. Finally, we use several tasks to

Proceedings of the Joint Workshop on Multiword Expressions and WordNet (MWE-WN 2019), pages 92–103
Florence, Italy, August 2, 2019.

evaluate the representation quality: relation classification (what is the relationship between the constituents?), property classification (is a *cheese wheel* round?), as well as a qualitative and quantitative analysis of the nearest neighbours. The results confirm that the distributional representations of rare noun compounds are indeed of low quality. Across representations, the nearest neighbours of a target noun compound vector typically include many trivial similarities such as other noun compounds with a shared constituent.

Among the composition functions, functions with more computational power and parameters generally produced higher quality representations. The paraphrase-based functions outperformed the others in the property prediction task, while the compositional functions performed better on relation classification. The results suggest that learning a composition function with a combined training objective is a promising research direction that may result in improved noun compound representations.[1]

2 Representations

We trained 315 distributional semantic models (DSMs) that differ by their training objective (Section 2.1) and the underlying embeddings used for the constituent nouns (Section 2.2).

2.1 Training Objective

Distributional. This approach simply treats a noun compound as a single token $w_1\_w_2$, and learns standard word embeddings for the words and noun compounds in the corpus.

Compositional. We learn a function $f(\cdot,\cdot) : \mathcal{R}^d \times \mathcal{R}^d \rightarrow \mathcal{R}^d$ which, for a given noun compound, operates on the word embeddings of its constituent nouns, and returns a vector representing the compound. Following Dima (2016) and earlier work, the training objective is to minimize the distance between the observed distributional embedding $\vec{v}_{w_1\_w_2}$ and the composed vector $f(\vec{v}_{w_1}, \vec{v}_{w_2})$.

We train the following composition functions:

- **Add** (Mitchell and Lapata, 2010): $f(\vec{v}_{w_1}, \vec{v}_{w_2}) = \alpha\vec{v}_{w_1} + \beta\vec{v}_{w_2}$, α, β are scalars.
- **FullAdd** (Zanzotto et al., 2010; Dinu et al., 2013): $f(\vec{v}_{w_1}, \vec{v}_{w_2}) = W_1\vec{v}_{w_1} + W_2\vec{v}_{w_2}$, where $W_1, W_2 \in \mathcal{R}^{d\times d}$ are matrices.
- **Matrix** (Dima, 2016): $f(\vec{v}_{w_1}, \vec{v}_{w_2}) = tanh(W \cdot [\vec{v}_{w_1}; \vec{v}_{w_2}])$, where $W \in \mathcal{R}^{2d\times d}$. This is the application of the recursive matrix-vector method of Socher et al. (2012) to binary phrases.[2]
- **LSTM**: encoding the compound with a long short-term memory network (LSTM; Hochreiter and Schmidhuber, 1997): $f(\vec{v}_{w_1}, \vec{v}_{w_2}) = LSTM(\vec{v}_{w_1}, \vec{v}_{w_2})$.

Paraphrase-based. In this approach we follow the literature of paraphrase-based phrase embeddings (e.g. Wieting et al., 2016, 2017). We generate paraphrases for each noun compound, and train the function with the objective of producing similar vectors to the noun compound and its paraphrase.

To obtain the representation of a phrase (either a noun compound or its variable-length paraphrase), we encode it with an LSTM. For a given noun compound NC = w_1 w_2 and its paraphrase p, we set the loss to:

$$max(0, \lambda - cos(v_{NC}, v_p) + cos(v_{NC}, v_{p'}))$$

where $v_x = LSTM(x)$ is the encoding of phrase x, p' is a negative-sampled paraphrase, and λ was set to 0.6 based on its value in Wieting et al. (2016). The following approaches were used to obtain the paraphrases:

- **Backtranslation**: We translate each noun compound to foreign language(s) and back to English, as in Wieting et al. (2017). Specifically, we use the DeepL Translator web interface,[3] performing translation from English to 4 different foreign languages (French, Italian, Spanish, and Romanian) and back to English. We focused on Romance languages because they translate English noun compounds to noun phrases with prepositions (Girju, 2007), and we were hoping that this would drive the backtranslation to be more explicit. For example, *baby oil* is translated in French to *huile pour bébé*, which literally means *oil for baby*. In

[1] The code and data is available at https://github.com/vered1986/NC_Embeddings.

[2] Originally, this method was trained with an extrinsic training objective of sentiment analysis.

[3] https://www.deepl.com

practice, translating back to English mostly generates paraphrases which are other noun compounds (synonyms or related terms), rather than prepositional paraphrases.

We use all the suggested translations to generate a large list of paraphrases for each noun compound, but we apply two filters. First, we trivially remove the noun compound itself from its list of paraphrases. Second, the translation sometimes yields non-English phrases (a result of an error in the translation), which we automatically identify and remove using a language identification tool.[4] After filtering around half of the paraphrases, we remain with an average number of 6.71 paraphrases per compound.

- **Co-occurrence**: We treat the frequent joint occurrences of w_1 and w_2 in a corpus as paraphrases, e.g. *apple cake* may yield a paraphrase like "*cake made of apples*". Specifically, we use the paraphrases obtained by Shwartz and Dagan (2018) from the Google N-gram corpus (Brants and Franz, 2006). The paraphrases are of variable length (3-5 words), and have been pre-processed to remove punctuation, adjectives, adverbs and determiners. The averaged number of paraphrases per compound is 9.18.

2.2 Constituent Word Embeddings

To represent the constituent words, we trained various word embedding algorithms: **word2vec** (Mikolov et al., 2013) and **fastText** (Bojanowski et al., 2017), which extends word2vec by adding subword information. We used both the Skip-Gram objective (which predicts the context words given the target word) and the CBOW objective (continuous bag-of-words, predicting the target word from its context).[5] We also trained the **GloVe** algorithm (Pennington et al., 2014), which estimates the log-probability of a word pair co-occurrence. All the embeddings were trained on the English Wikipedia dump from January 2018, with various values for the window size (2, 5, 10) and the embedding dimension (100, 200, 300).

2.3 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). To train the DSMs we used the list of 18,856 *compositional* noun compounds from Tratz (2011).[6] We only used binary noun compounds, i.e. consisting of exactly two constituent nouns, and we split them to 80% train, 10% test, and 10% validation sets.

For the sake of simplicity, for the remainder of the paper we will refer to the training objective and architecture combination as the "representation", and a trained instance of the representation, with a choice of underlying word embeddings (algorithm, dimension, and window), as a DSM.

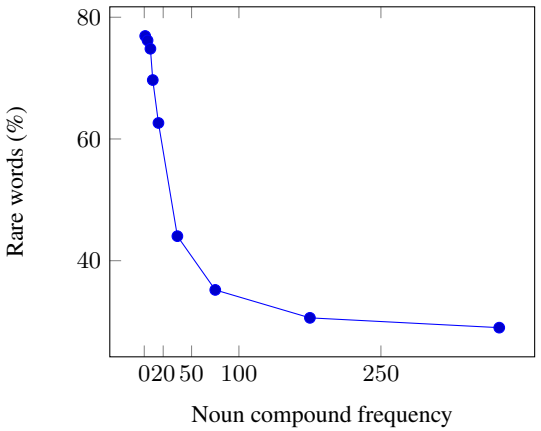

Figure 1: Averaged percent of rare words (less than 10 occurrences) among the 10 nearest neighbours of a noun compound with a given corpus frequency.

3 Experiments

We compare the various representations in 3 experiments: an analysis of the nearest neighbours of each noun compound vector (Section 3.1), an evaluation on property prediction (Section 3.2), and an evaluation on noun compound relation classification (Section 3.3).

3.1 Nearest Neighbour Analysis

Similarly to Boleda et al.'s (2013) analysis for adjective-noun compositions, we compute the 10 nearest neighbors of each noun compound in the test set and analyze the outputs. Table 1 exemplifies the nearest neighbours of two noun compounds in each representation, setting the DSM to (word2vec SG, window 5, 300d).

[4] https://pypi.org/project/guess_language-spirit/

[5] We used the Gensim implementation: https://radimrehurek.com/gensim/

[6] Omitting 351 noun compounds belonging to the LEXICALIZED, PERSONAL_NAME, and PERSONAL_TITLE classes.

syndicate representative (rare)			
Distributional			
geloios			
t.franse			
adopter(s			
ahchie			
anquish			
Compositional			
Add	**FullAdd**	**Matrix**	**LSTM**
syndicate	syndicate	f(student, representative)	f(worker, representative)
representative	f(deputy, representative)	syndicate	f(player, representative)
f(worker, representative)	f(student, representative)	f(deputy, representative)	f(crack, dealer)
f(deputy, representative)	f(player, representative)	f(worker, representative)	f(company, spokesman)
f(student, representative)	f(worker, representative)	f(player, representative)	f(industry, commissioner)
Paraphrase-based			
Co-occurrence	**Backtranslation**		
f(company, representative)	f(worker, representative)		
f(phone, representative)	f(union, representative)		
f(union, representative)	f(group, manager)		
f(marketing, representative)	f(employee, representative)		
f(labor, representative)	f(student, representative)		
army officer (frequent)			
Distributional			
army_captain			
army_major			
navy_officer			
army_general			
army_lieutenant			
Compositional			
Add	**FullAdd**	**Matrix**	**LSTM**
army	f(police, commander)	f(police, commander)	f(militia, commander)
officer	f(army, troop)	army_officer	f(police, commander)
f(army, battalion)	f(militia, commander)	f(army, troop)	f(opposition, commander)
f(army, troop)	f(army, camp)	army_general	f(military, official)
f(army, building)	army_officer	f(army, camp)	f(comrade, commander)
Paraphrase-based			
Co-occurrence	**Backtranslation**		
	f(patrol, officer)	f(army, official)	
	f(navy, officer)	f(military, spokesman)	
	f(prison, officer)	f(army, lieutenant)	
	f(fire, officer)	f(army, chief)	
	f(police, officer)	f(army, spokesman)	

Table 1: Top 5 nearest neighbour of two example noun compounds, *syndicate representative* (1 corpus occurrence) and *army officer* (13,924 occurrences) in each composition function. DSM = (word2vec SG, window 5, 300d).

3.1.1 Observed vs. Composed

The nearest distributional neighbours of *syndicate representative* in Table 1 demonstrate the well known fact that the distributional embeddings of rare terms are of low quality. The goal of the composition functions is to provide meaningful representations for ad-hoc, possibly rare compositions of nouns. They are learned as an approximation of the observed (distributional) representations of frequent noun compounds. How frequent should a noun compound be for its observed representation to be preferred over the compositional one? For example, the nearest neighbours of *army officer*, a very frequent term, indicate that its distributional embedding is meaningful.[7]

To get an approximate answer to this question, we compute the percentage of rare words (words which occurred less than 10 times in the corpus) among the 10 nearest neighbours of each noun compound, using the distributional DSMs. We average the percents across the various word embedding algorithms, dimensions, and windows. Figure 1 plots the percentage of rare neighbours by noun compound frequency. While the percent of rare words quickly drops from 75% after only a

[7] Boleda et al. (2013) found that in the case of adjective-noun compositions, observed vectors were preferred for frequent compositions, and compositional vectors for rare ones.

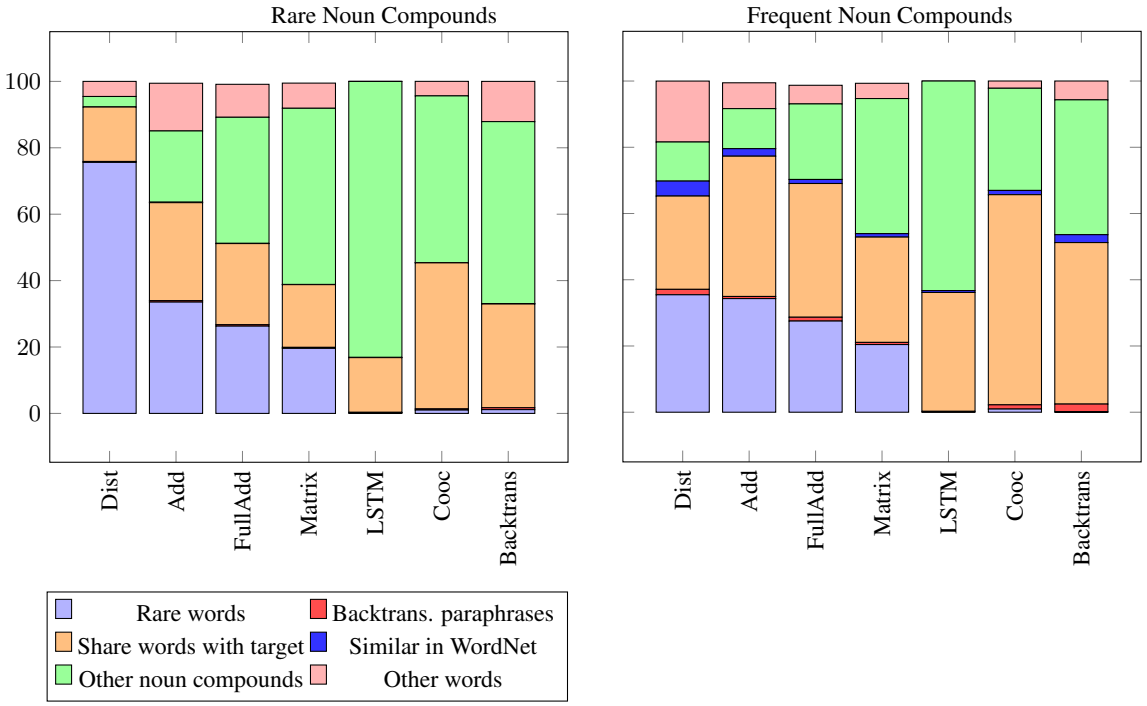

Figure 2: Categories of the top 10 neighbors of each target compound, for the 100 most rare compounds in the test set (first row) and the 100 most frequent compounds in the test set (second row). Best viewed in color.

few occurrences, even noun compounds with more than 250 occurrences have around 30% of rare neighbours.

3.1.2 Neighbour Types

We focus on the 100 most frequent compounds (between 3,235 occurrences: *city manager*, and 47,866 occurrences: *ball player*) and the 100 most rare compounds (from one occurrence, e.g. *chief joker*, to 6 occurrences, e.g. *coat shopping*).

We categorize the neighbours of a target compound into 6 categories, as exemplified for the noun compound *street level*: (1) rare words (*3bf*); (2) other noun compounds which are included in its "backtranslation" paraphrases (*ground floor*); (3) the compound's constituents or other noun compounds that share a constituent with it (e.g. *street*, *level*, and *sea level*); (4) words or noun compounds which have high WordNet similarity with the compound[8]; (5) other noun compounds (*parking garage*); and (6) other words (*stairs*). Figure 2 shows the charts of categories for each representation, averaged across DSMs.

Figure 2 shows that for the compositional representations (Add, FullAdd, Matrix), between 20% and a third of the neighbours are rare words. The percent of rare words decreases as the composition function has more parameters.[9] The nearest neighbours also typically include trivial neighbours, such as the constituents and other compounds that share a constituent with the target compound (19-30% for rare compounds and 32-43% for frequent ones). Overall, at least a half of the neighbours are trivial or meaningless. Most of the other neighbours are other noun compounds which have not been judged for correctness.

LSTM, Co-occurrence, and Backtranslation all use an LSTM to encode the noun compounds. Although their training objectives are different, they all tend to produce noun compound vectors which are very different from those of single words. This results in nearest neighbour lists which consist of mostly other compounds, either with or without shared constituents.

Very few neighbours were backtranslation paraphrases: less than 1% for most representations, and 2.32% for backtranslation of frequent compounds.

For frequent compounds, 1-2% of the neigh-

[8] Specifically, we used the Wu-Palmer similarity (Wu and Palmer, 1994), which returns a score denoting how similar two synsets are, based on the depth of their most specific ancestor in the WordNet taxonomy. We took the highest score among all the different synsets of each term, and considered a high score as > 0.25.

[9] The percents are similar for frequent and rare noun compounds. This is expected because once the composition function has been learned, the frequency of a test compound has no importance.

Representation	Used for transportation	Is a weapon	Is round	Has various colors	Made of metal
Distributional	48.0 ± 12.6	$\mathbf{57.3 \pm 14.8}$	24.8 ± 8.9	42.0 ± 12.5	41.3 ± 12.0
Add	55.8 ± 13.5	30.3 ± 20.1	46.2 ± 13.2	41.8 ± 13.1	55.1 ± 14.1
FullAdd	55.9 ± 13.4	36.8 ± 17.3	44.0 ± 13.0	48.2 ± 12.7	52.2 ± 13.0
Matrix	56.5 ± 13.9	24.0 ± 19.1	43.8 ± 13.4	49.5 ± 13.3	52.0 ± 12.9
LSTM	48.3 ± 15.8	0.0 ± 0.0	21.7 ± 17.5	37.2 ± 18.4	42.1 ± 18.6
Co-occurrence	$\mathbf{64.2 \pm 14.9}$	40.5 ± 30.1	$\mathbf{47.0 \pm 13.0}$	$\mathbf{56.9 \pm 12.8}$	$\mathbf{57.6 \pm 12.9}$
Backtranslation	58.3 ± 14.1	54.0 ± 19.5	42.1 ± 13.5	52.4 ± 13.5	57.4 ± 13.1

Table 2: Mean and standard deviation of F_1 scores across DSMs, for each representation and property combination. The majority baseline F_1 score is 0 for all properties, since it always predicts False.

Feature	Representation	Embedding	Window	Dimension	Precision	Recall	$\mathbf{F_1}$
Used for transportation	Co-occurrence	word2vec SG	10	300	74.5	78.8	76.6
Is a weapon	Backtranslation	word2vec CBOW	2	300	71.4	88.2	78.9
Is round	Co-occurrence	word2vec CBOW	10	300	56.2	87.1	68.4
Has various colors	Co-occurrence	GloVe	2	200	70.6	76.6	73.5
Made of metal	Matrix	word2vec SG	5	300	78.6	61.1	68.8

Table 3: The performance of the best setting for each property.

bours were considered similar to the target compound in WordNet. We note that this category is meaningless for rare noun compounds since most of them are not in WordNet all.[10]

3.2 Property Prediction

Do the various representations capture properties of noun compounds? To answer this question, we create a task in which we need to predict for a given noun compound whether it has a certain property or not. For example, is a *cheese wheel* round?

3.2.1 Task Definition and Data

We use the McRae Feature Norms dataset (McRae et al., 2005), which provides, for single words describing concrete nouns, the most salient properties that describe them. We follow the binary classification setting of Rubinstein et al. (2015) in which each task is focused on a single property, and negative instances are (a sample of) the concepts that do not appear with the property.

To augment this data with noun compounds, we first filtered the dataset such that it only contains constituents of noun compounds in our vocabulary. We then selected 5 of the most frequent properties ("a weapon", "round", "made of metal", "used for transportation", and "comes in different colors"). For each property, we looked for all the noun compounds that consist of a constituent annotated to holding this property, and manually annotated them to whether they also hold this property. For example, since *apple* is round, we manually judged noun compounds such as *apple pie* (also round), and *apple grower* (not round).[11] Finally, we manually added some examples from online lists (e.g. the "round objects" list in Wikipedia[12]).

We split the data to train (90% of the single words and 20% of the noun compounds), validation (10% of the single words and 20% of the noun compounds), and test (60% of the noun compounds). The training sets each contains around 500 instances. For each DSM, we train classifiers on the composed vectors of each given concept (a single word or a noun compound). We train multiple classifiers (logistic regression and SVM, with various L2 regularization values) and select the best performing classifier with respect to the validation F_1 score.

3.2.2 Results and Analysis

Table 2 shows the mean and the standard deviation of F_1 scores per representation across DSMs, for each of the properties.

The co-occurrence function stands out in its performance, and the backtranslation function is often second best. There is no clear preference among the compositional functions, except for the LSTM which is consistently worse than the others. The distributional embeddings typically per-

[10] WordNet only consists of lexicalized noun compounds, e.g. *olive oil* and *ice cream*, which tend to be frequent.

[11] We note that this semi-automatic data collection procedure might miss some salient properties of noun compounds which are not properties of their constituents.

[12] https://commons.wikimedia.org/wiki/Category:Round_objects

form among the worst. This is expected both due to the quality of the embeddings of rare noun compounds (Section 3.1) and since some of the noun compounds in the data are out-of-vocabulary. In contrast, the other representations compute ad hoc vectors for such noun compounds.

For the sake of completeness, Table 3 displays the best performing DSM for each property. There is a preference to word2vec and to a higher embedding dimension.

Looking at the errors made by the best model we found a common pattern of false positive errors. Most of them stem from multiple positive training instances that share a constituent with the target noun compound, e.g. predicting that *sprint car* is used for transportation, although its primary purpose is racing, that *kidney stone* is a weapon, that *tomato soup* is round, and that *tar ball* comes in multiple colors. We did not find a common pattern among the false negative errors.

Finally, although it is tempting to draw general conclusions as to the types of properties (e.g. attributive vs. taxonomic) that each representation captures, we refrain from doing so given the small number of properties we tested.

3.3 Relation Classification

Similarly to Dima (2016), we also evaluate the various representations on the noun compound classification task. This is a multiclass classification problem to a pre-defined set of semantic relations, e.g. *morning coffee*: TIME vs. *coffee cup*: CONTAINED.

3.3.1 Evaluation Setup

We evaluate on the Tratz (2011) dataset, which consists of 19,158 instances, labeled in 37 fine-grained relations or 12 coarse-grained relations. We follow the data splits from Shwartz and Waterson (2018), reporting performance on both the random split and the lexical split, in which there are no shared constituents between the train, validation, and test sets. Since we focus on *compositional* noun compounds, we remove the LEXICALIZED relation (which consists of many non-compositional noun compounds). We also remove the PERSONAL NAME and PERSONAL TITLE relations which consist of named entities. We train various classifiers on the vectors obtained by each DSM for a given noun compound, choosing the best performing classifier with respect to the validation F_1 score.

It is important to note that the categorization of noun compounds to a fixed inventory of semantic relations that may hold between their constituents is often subjective, making the data noisy. Previous work suggested that many noun compounds fit into more than one relation, and that some relations in the fine-grained version of the data are overlapping (Shwartz and Waterson, 2018). With that said, this data is still a useful proxy for measuring and comparing the quality of representations.

3.3.2 Results

Table 4 shows the mean and the standard deviation of F_1 scores per representation across DSMs, while Table 5 displays the best DSM for each dataset.

Compositional functions perform better. The best performing methods are FullAdd and LSTM. Examination of the per-relation F_1 scores shows that Add is, for many relations, the best performing composition function. The poor performance of the distributional DSMs may be attributed to the quality of representations for rare noun compounds, although it was also noted by Shwartz and Waterson (2018) that even when the target noun compound has a meaningful distributional vector, its most similar neighbor may have been assigned a different label by the annotators, as in *majority party*: EQUATIVE vs. *minority party*: WHOLE+PART_OR_MEMBER_OF (see the discussion in Section 4).

In contrast, it is surprising to see that the paraphrase-based DSMs did not perform as well as the compositional ones. We expected their training objective and data to drive the representations towards capturing more explicit information which could aid the classification; for instance, *glass product* has a "*product made of glass*" paraphrase in backtranslation and *night meeting* has a "*meeting held at night*" paraphrase in co-occurrence. The mediocre performance may be either due to the sparsity of such explicit paraphrases in the data or due to a sub-optimal training objective. We leave further investigation to future work.

Smaller windows are preferred. Table 5 shows a consistent preference to the small window size. DSMs with small windows are known to capture functional, rather than topical similarity between terms, which could to be beneficial for relation classification. For example, *morning workout* in

Representation	Coarse-grained Random	Coarse-grained Lexical	Fine-grained Random	Fine-grained Lexical
Distributional	44.0 ± 11.5	30.5 ± 8.5	40.8 ± 12.5	24.7 ± 6.5
Add	51.9 ± 10.5	34.7 ± 7.3	51.5 ± 10.9	30.7 ± 5.9
FullAdd	$\mathbf{54.5 \pm 10.7}$	35.7 ± 8.0	$\mathbf{53.5 \pm 11.0}$	28.8 ± 6.8
Matrix	49.1 ± 11.3	32.6 ± 8.1	47.3 ± 12.1	26.7 ± 7.2
LSTM	54.0 ± 11.8	$\mathbf{37.5 \pm 8.2}$	52.1 ± 11.9	$\mathbf{30.9 \pm 6.6}$
Co-occurrence	49.8 ± 9.7	31.4 ± 7.1	47.7 ± 10.6	24.6 ± 6.0
Backtranslation	47.2 ± 7.7	33.5 ± 6.1	44.6 ± 8.5	26.7 ± 5.1

Table 4: Mean and standard deviation of F_1 scores across word embeddings, windows and dimensions, for each composition function and dataset combination.

Dataset	Representation	Embedding	Window	Dimension	Precision	Recall	$\mathbf{F_1}$
Coarse-grained Random	LSTM	Fasttext SG	2	300	66.5	66.7	66.2
Coarse-grained Lexical	LSTM	Fasttext SG	2	200	50.2	49.0	47.5
Fine-grained Random	LSTM	Fasttext SG	2	300	64.6	65.3	63.9
Fine-grained Lexical	Matrix	word2vec SG	2	100	39.6	39.8	38.1

Table 5: The performance of the best setting for each noun compound relation classification dataset.

the train set and *night thunderstorm* in the test set are both annotated to TIME-OF1. While they are not topically related, they may appear in similar syntactic constructions related to time, e.g. "before / after / during the *morning workout* / *night thunderstorm*".

Some relations are more challenging than others. The average per-relation F_1 scores by representation varies across relations. In the fine-grained version of the dataset, the worse performance was achieved on the PARTIAL ATTRIBUTE TRANSFER relation (2.18). In these noun compounds, the modifier "transfers" an attribute to the head, as in *bullet train*, which is a fast train (fast "like a bullet"). Given the figurative nature of this relation, it is not surprising that the various representations struggle in recognizing it. In contrast, the average performance on the MEASURE relation was 71.25, as it is often enough to recognize that the modifier is a measuring unit (e.g. *hour ride*). These observations are in line with previous work (Shwartz and Waterson, 2018).

Comparison to prior work. The best previously reported F_1 scores on these datasets are: coarse-grained random: 77.5, coarse-grained lexical: 47.8, fine-grained random: 73.9, and fine-grained lexical: 42.9 (Shwartz and Dagan, 2018). They are achieved by richer models and evaluated on the full inventory of semantic relations. Furthermore, the random splits benefit from "lexical memorization", i.e. predicting the relation based on the distribution of training instances sharing a single constituent with the target noun compound (e.g., predicting TOPIC for every compound whose head is *guide*; Dima, 2016; Shwartz and Waterson, 2018). This may enhance the performance of models with direct access to the constituent embeddings (e.g. a classifier trained on their vector concatenation). For the sake of comparing between the various representations, we used only the noun compound vectors as input to the classifier.

4 Discussion

Limitations. The main limitation of composition functions is that they rely on the assumption of compositionality, which often does not hold. While in this work we focused on compositional noun compounds, the meaning of many noun compounds is not a straightforward combination of the meanings of their constituents. This happens with figurative noun compounds (e.g. *brain drain*, *family tree*), as well as some highly lexicalized ones (e.g., it is not natural to describe *ice cream* using *ice* and *cream*).

Some representations only operate on binary noun compounds, while the LSTM based representations are capable of producing vectors for variable-length noun compounds. However, we only tested binary noun compounds. It is not certain that the representations we tested would be able to address the complexity of longer noun compounds, which, among other things, also require uncovering the syntactic head-modifier structure.

Finally, we used a pre-defined list of noun com-

pounds and did not address identification, which should precede both the training and the inference of the representations. While the criteria for selecting what is considered a noun compound can be strictly syntactic, the decision on whether to use (and train) a distributional embedding for a given noun compound may be based on its frequency.

Contextualized Word Embeddings are dynamic word embeddings computed for words given their context sentence (Peters et al., 2018; Radford et al., 2018; Devlin et al., 2019). They have become increasingly popular last year, outperforming static embeddings across NLP tasks. Supposedly, such representations obviate the need to learn dedicated noun compound representations, as the vector of each constituent is computed given the other constituent.

Recently, Shwartz and Dagan (2019) found that while these representations excel at detecting non-compositional noun compounds, they perform much worse at revealing implicit information such as the relationship between the constituents. Moreover, looking into these models' predictions of substitute constituents shows that even when they recognize a constituent is not used in its literal sense (e.g. in non-compositional compounds), the representation of its (often rare) non-literal sense is not always meaningful. Overall, contextualized word embeddings do not completely solve the problem of obtaining meaningful representations for noun compounds, but they do offer a step forward.

5 Conclusions

We trained numerous noun compound representations and compared their quality through a series of tasks and analyses. Our results confirm that distributional representations lose quality as the frequency of the noun compound in the corpus decreases, making dynamic representations imperative. Among such representations, those with more computational power were preferred. There was no single representation that performed best across tasks. The paraphrase-based representations performed better on property identification, while those trained to approximate the distributional representations performed better on relation classification. Two interesting future research directions would be to design a representation with multiple training objectives, and to build it on top of contextualized word representations.

Acknowledgments

The author is supported by the Clore Scholars Programme (2017).

References

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 597–604, Ann Arbor, Michigan. Association for Computational Linguistics.

Regina Barzilay and R. Kathleen McKeown. 2001. Extracting paraphrases from a parallel corpus. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*.

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

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

Thorsten Brants and Alex Franz. 2006. Web 1t 5-gram version 1.

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, Volume 1 (Long Papers)*, Minneapolis, Minnesota. Association for Computational Linguistics.

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. Association for Computational Linguistics.

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

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for*

Computational Linguistics: Human Language Technologies, pages 758–764. Association for Computational Linguistics.

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

Roxana Girju. 2007. Improving the interpretation of noun phrases with cross-linguistic information. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 568–575, Prague, Czech Republic. Association for Computational Linguistics.

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

Su Nam Kim and Timothy Baldwin. 2006. Interpreting semantic relations in noun compounds via verb semantics. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 491–498. Association for Computational Linguistics.

Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. 2017. Paraphrasing revisited with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 881–893, Valencia, Spain. Association for Computational Linguistics.

Ken McRae, George S Cree, Mark S Seidenberg, and Chris McNorgan. 2005. Semantic feature production norms for a large set of living and nonliving things. *Behavior research methods*, 37(4):547–559.

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.

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*.

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

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

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, Spain. Association for Computational Linguistics.

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

Dana Rubinstein, Effi Levi, Roy Schwartz, and Ari Rappoport. 2015. How well do distributional models capture different types of semantic knowledge? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 726–730, Beijing, China. Association for Computational Linguistics.

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, Australia. Association for Computational Linguistics.

Vered Shwartz and Ido Dagan. 2019. Still a pain in the neck: Evaluating text representations on lexical composition. In *Transactions of the Association for Computational Linguistics (TACL)*, page (to appear).

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

Richard Socher, Brody Huval, D. Christopher Manning, and Y. Andrew Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and*

Computational Natural Language Learning, pages 1201–1211. Association for Computational Linguistics.

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

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

John Wieting and Kevin Gimpel. 2017. Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *arXiv preprint arXiv:1711.05732*.

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

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

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

A Noun Compound Classification Labels

The following table displays the semantic relations in the Tratz (2011) dataset. Each coarse-grained relation (highlighted in gray), is followed by the fine-grained relations that it unites. Each fine-grained relation contains an example noun compound (see Section 3.3).

CAUSE	
experiencer-of-experience	*company strategy*
PURPOSE	
purpose	*labor market*
create-provide-generate-sell	*aid center*
mitigate&oppose	*fishing quota*
perform&engage_in	*acquisition fund*
organize&supervise&authority	*fire commissioner*
TIME	
time-of1	*fourth-quarter income*
time-of2	*rating period*
LOC_PART_WHOLE	
location	*water spider*
whole+part_or_member_of	*society member*
ATTRIBUTE	
equative	*winter season*
adj-like_noun	*core tradition*
partial_attribute_transfer	*lemon soda*
OTHER	
measure	*percentage change*
lexicalized	*action hero*
other	*trade conflict*
OBJECTIVE	
objective	*biotechnology research*
CAUSAL	
subject	*government figure*
justification	*genocide trial*
creator-provider-cause_of	*refining margin*
means	*car bombing*
COMPLEMENT	
relational-noun-complement	*police power*
whole+attribute&feature&quality_value_is_characteristic_of	*earth tone*
CONTAINMENT	
part&member_of_collection&config&series	*stock portfolio*
contain	*studio lot*
variety&genus_of	*tuberculosis strain*
amount-of	*work load*
substance-material-ingredient	*cedar chalet*
OWNER_EMP_USE	
user_recipient	*subway platform*
employer	*government technocrat*
owner-user	*government surplus*
TOPICAL	
personal_name	*Sarah Boyle*
topic_of_cognition&emotion	*security fear*
topic_of_expert	*cancer expert*
obtain&access&seek	*finance plan*
personal_title	*Minister Kennedy*
topic	*property deal*