Paraphrase to Explicate:
Revealing Implicit Noun-Compound Relations

Vered Shwartz   **Ido Dagan**
Computer Science Department, Bar-Ilan University, Ramat-Gan, Israel
vered1986@gmail.com   dagan@cs.biu.ac.il

Abstract

Revealing the implicit semantic relation between the constituents of a noun-compound is important for many NLP applications. It has been addressed in the literature either as a classification task to a set of pre-defined relations or by producing free text paraphrases explicating the relations. Most existing paraphrasing methods lack the ability to generalize, and have a hard time interpreting infrequent or new noun-compounds. We propose a neural model that generalizes better by representing paraphrases in a continuous space, generalizing for both unseen noun-compounds and rare paraphrases. Our model helps improving performance on both the noun-compound paraphrasing and classification tasks.

1 Introduction

Noun-compounds hold an implicit semantic relation between their constituents. For example, a 'birthday cake' is a cake *eaten on* a birthday, while 'apple cake' is a cake *made of* apples. Interpreting noun-compounds by explicating the relationship is beneficial for many natural language understanding tasks, especially given the prevalence of noun-compounds in English (Nakov, 2013).

The interpretation of noun-compounds has been addressed in the literature either by classifying them to a fixed inventory of ontological relationships (e.g. Nastase and Szpakowicz, 2003) or by generating various free text paraphrases that describe the relation in a more expressive manner (e.g. Hendrickx et al., 2013).

Methods dedicated to paraphrasing noun-compounds usually rely on corpus co-occurrences of the compound's constituents as a source of explicit relation paraphrases (e.g. Wubben, 2010; Versley, 2013). Such methods are unable to generalize for unseen noun-compounds. Yet, most noun-compounds are very infrequent in text (Kim and Baldwin, 2007), and humans easily interpret the meaning of a new noun-compound by generalizing existing knowledge. For example, consider interpreting *parsley cake* as a cake made of parsley vs. *resignation cake* as a cake eaten to celebrate quitting an unpleasant job.

We follow the paraphrasing approach and propose a semi-supervised model for paraphrasing noun-compounds. Differently from previous methods, we train the model to predict either a paraphrase expressing the semantic relation of a noun-compound (predicting '$[w_2]$ made of $[w_1]$' given 'apple cake'), or a missing constituent given a combination of paraphrase and noun-compound (predicting 'apple' given 'cake made of $[w_1]$'). Constituents and paraphrase templates are represented as continuous vectors, and semantically-similar paraphrase templates are embedded in proximity, enabling better generalization. Interpreting 'parsley cake' effectively reduces to identifying paraphrase templates whose "selectional preferences" (Pantel et al., 2007) on each constituent fit 'parsley' and 'cake'.

A qualitative analysis of the model shows that the top ranked paraphrases retrieved for each noun-compound are plausible even when the constituents never co-occur (Section 4). We evaluate our model on both the paraphrasing and the classification tasks (Section 5). On both tasks, the model's ability to generalize leads to improved performance in challenging evaluation settings.[1]

---

[1] The code is available at github.com/vered1986/panic

2 Background

2.1 Noun-compound Classification

Noun-compound classification is the task concerned with automatically determining the semantic relation that holds between the constituents of a noun-compound, taken from a set of pre-defined relations.

Early work on the task leveraged information derived from lexical resources and corpora (e.g. Girju, 2007; Ó Séaghdha and Copestake, 2009; Tratz and Hovy, 2010). More recent work broke the task into two steps: in the first step, a noun-compound representation is learned from the distributional representation of the constituent words (e.g. Mitchell and Lapata, 2010; Zanzotto et al., 2010; Socher et al., 2012). In the second step, the noun-compound representations are used as feature vectors for classification (e.g. Dima and Hinrichs, 2015; Dima, 2016).

The datasets for this task differ in size, number of relations and granularity level (e.g. Nastase and Szpakowicz, 2003; Kim and Baldwin, 2007; Tratz and Hovy, 2010). The decision on the relation inventory is somewhat arbitrary, and subsequently, the inter-annotator agreement is relatively low (Kim and Baldwin, 2007). Specifically, a noun-compound may fit into more than one relation: for instance, in Tratz (2011), *business zone* is labeled as CONTAINED (zone contains business), although it could also be labeled as PURPOSE (zone whose purpose is business).

2.2 Noun-compound Paraphrasing

As an alternative to the strict classification to pre-defined relation classes, Nakov and Hearst (2006) suggested that the semantics of a noun-compound could be expressed with multiple prepositional and verbal paraphrases. For example, *apple cake* is a cake *from*, *made of*, or which *contains* apples.

The suggestion was embraced and resulted in two SemEval tasks. SemEval 2010 task 9 (Butnariu et al., 2009) provided a list of plausible human-written paraphrases for each noun-compound, and systems had to rank them with the goal of high correlation with human judgments. In SemEval 2013 task 4 (Hendrickx et al., 2013), systems were expected to provide a ranked list of paraphrases extracted from free text.

Various approaches were proposed for this task. Most approaches start with a pre-processing step of extracting joint occurrences of the constituents from a corpus to generate a list of candidate paraphrases. Unsupervised methods apply information extraction techniques to find and rank the most meaningful paraphrases (Kim and Nakov, 2011; Xavier and Lima, 2014; Pasca, 2015; Pavlick and Pasca, 2017), while supervised approaches learn to rank paraphrases using various features such as co-occurrence counts (Wubben, 2010; Li et al., 2010; Surtani et al., 2013; Versley, 2013) or the distributional representations of the noun-compounds (Van de Cruys et al., 2013).

One of the challenges of this approach is the ability to generalize. If one assumes that sufficient paraphrases for all noun-compounds appear in the corpus, the problem reduces to ranking the existing paraphrases. It is more likely, however, that some noun-compounds do not have any paraphrases in the corpus or have just a few. The approach of Van de Cruys et al. (2013) somewhat generalizes for unseen noun-compounds. They represented each noun-compound using a compositional distributional vector (Mitchell and Lapata, 2010) and used it to predict paraphrases from the corpus. Similar noun-compounds are expected to have similar distributional representations and therefore yield the same paraphrases. For example, if the corpus does not contain paraphrases for *plastic spoon*, the model may predict the paraphrases of a similar compound such as *steel knife*.

In terms of sharing information between semantically-similar paraphrases, Nulty and Costello (2010) and Surtani et al. (2013) learned "is-a" relations between paraphrases from the co-occurrences of various paraphrases with each other. For example, the specific '[w_2] extracted from [w_1]' template (e.g. in the context of *olive oil*) generalizes to '[w_2] made from [w_1]'. One of the drawbacks of these systems is that they favor more frequent paraphrases, which may co-occur with a wide variety of more specific paraphrases.

2.3 Noun-compounds in other Tasks

Noun-compound paraphrasing may be considered as a subtask of the general paraphrasing task, whose goal is to generate, given a text fragment, additional texts with the same meaning. However, general paraphrasing methods do not guarantee to explicate implicit information conveyed in the original text. Moreover, the most notable source for extracting paraphrases is multiple translations of the same text (Barzilay and McKeown,
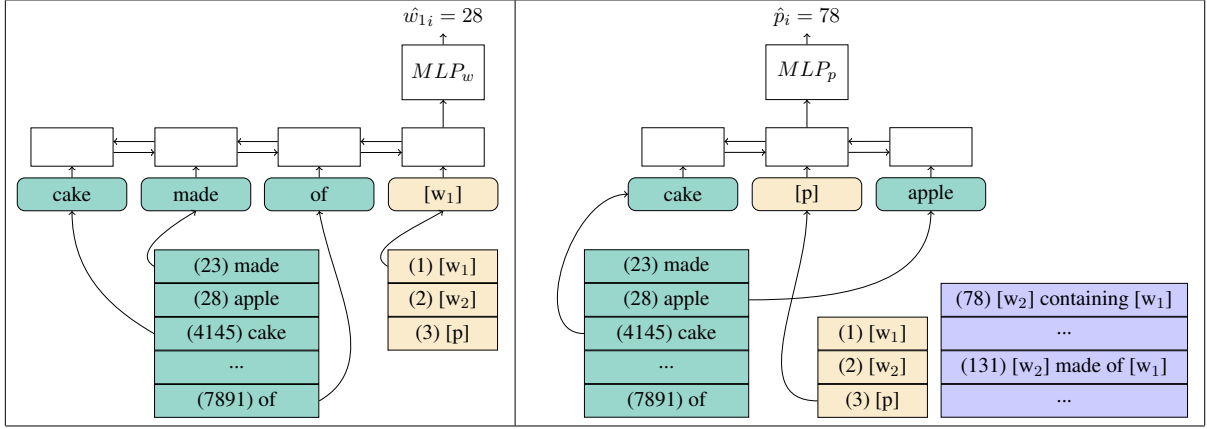
Figure 1: An illustration of the model predictions for w_1 and p given the triplet *(cake, made of, apple)*. The model predicts each component given the encoding of the other two components, successfully predicting '*apple*' given '*cake made of* [w$_1$]', while predicting '[w$_2$] *containing* [w$_1$]' for '*cake* [p] *apple*'.

2001; Ganitkevitch et al., 2013; Mallinson et al., 2017). If a certain concept can be described by an English noun-compound, it is unlikely that a translator chose to translate its foreign language equivalent to an explicit paraphrase instead.

Another related task is Open Information Extraction (Etzioni et al., 2008), whose goal is to extract relational tuples from text. Most system focus on extracting verb-mediated relations, and the few exceptions that addressed noun-compounds provided partial solutions. Pal and Mausam (2016) focused on segmenting multi-word noun-compounds and assumed an is-a relation between the parts, as extracting *(Francis Collins, is, NIH director)* from "NIH director Francis Collins". Xavier and Lima (2014) enriched the corpus with compound definitions from online dictionaries, for example, interpreting *oil industry* as *(industry, produces and delivers, oil)* based on the WordNet definition "industry that produces and delivers oil". This method is very limited as it can only interpret noun-compounds with dictionary entries, while the majority of English noun-compounds don't have them (Nakov, 2013).

3 Paraphrasing Model

As opposed to previous approaches, that focus on predicting a paraphrase template for a given noun-compound, we reformulate the task as a multi-task learning problem (Section 3.1), and train the model to also predict a missing constituent given the paraphrase template and the other constituent. Our model is semi-supervised, and it expects as input a set of noun-compounds and a set of constrained part-of-speech tag-based templates that make valid prepositional and verbal paraphrases.

Section 3.2 details the creation of training data, and Section 3.3 describes the model.

3.1 Multi-task Reformulation

Each training example consists of two constituents and a paraphrase (w_2, p, w_1), and we train the model on 3 subtasks: (1) predict p given w_1 and w_2, (2) predict w_1 given p and w_2, and (3) predict w_2 given p and w_1. Figure 1 demonstrates the predictions for subtasks (1) (right) and (2) (left) for the training example *(cake, made of, apple)*. Effectively, the model is trained to answer questions such as "what can cake be made of?", "what can be made of apple?", and "what are the possible relationships between cake and apple?".

The multi-task reformulation helps learning better representations for paraphrase templates, by embedding semantically-similar paraphrases in proximity. Similarity between paraphrases stems either from lexical similarity and overlap between the paraphrases (e.g. 'is made of' and 'made of'), or from shared constituents, e.g. '[w$_2$] involved in [w$_1$]' and '[w$_2$] in [w$_1$] industry' can share [w$_1$] = *insurance* and [w$_2$] = *company*. This allows the model to predict a correct paraphrase for a given noun-compound, even when the constituents *do not occur with that paraphrase in the corpus*.

3.2 Training Data

We collect a training set of (w_2, p, w_1, s) examples, where w_1 and w_2 are constituents of a noun-compound $w_1 w_2$, p is a templated paraphrase, and s is the score assigned to the training instance.[2]

---

[2] We refer to "paraphrases" and "paraphrase templates" interchangeably. In the extracted templates, [w$_2$] always precedes [w$_1$], probably because w$_2$ is normally the head noun.

We use the 19,491 noun-compounds found in the SemEval tasks datasets (Butnariu et al., 2009; Hendrickx et al., 2013) and in Tratz (2011). To extract patterns of part-of-speech tags that can form noun-compound paraphrases, such as '[w$_2$] VERB PREP [w$_1$]', we use the SemEval task training data, but we do not use the lexical information in the gold paraphrases.

Corpus. Similarly to previous noun-compound paraphrasing approaches, we use the Google N-gram corpus (Brants and Franz, 2006) as a source of paraphrases (Wubben, 2010; Li et al., 2010; Surtani et al., 2013; Versley, 2013). The corpus consists of sequences of n terms (for $n \in \{3, 4, 5\}$) that occur more than 40 times on the web. We search for n-grams following the extracted patterns and containing w_1 and w_2's lemmas for some noun-compound in the set. We remove punctuation, adjectives, adverbs and some determiners to unite similar paraphrases. For example, from the 5-gram '*cake made of sweet apples*' we extract the training example *(cake, made of, apple)*. We keep only paraphrases that occurred at least 5 times, resulting in 136,609 instances.

Weighting. Each n-gram in the corpus is accompanied with its frequency, which we use to assign scores to the different paraphrases. For instance, '*cake of apples*' may also appear in the corpus, although with lower frequency than '*cake from apples*'. As also noted by Surtani et al. (2013), the shortcoming of such a weighting mechanism is that it prefers shorter paraphrases, which are much more common in the corpus (e.g. count('*cake made of apples*') $\ll$ count('*cake of apples*')). We overcome this by normalizing the frequencies *for each paraphrase length*, creating a distribution of paraphrases in a given length.

Negative Samples. We add 1% of negative samples by selecting random corpus words w_1 and w_2 that do not co-occur, and adding an example (w_2, [w$_2$] is unrelated to [w$_1$], w_1, s_n), for some predefined negative samples score s_n. Similarly, for a word w_i that did not occur in a paraphrase p we add (w_i, p, UNK, s_n) or (UNK, p, w_i, s_n), where UNK is the unknown word. This may help the model deal with non-compositional noun-compounds, where w_1 and w_2 are unrelated, rather than forcibly predicting some relation between them.

3.3 Model

For a training instance (w_2, p, w_1, s), we predict each item given the encoding of the other two.

Encoding. We use the 100-dimensional pre-trained GloVe embeddings (Pennington et al., 2014), which are fixed during training. In addition, we learn embeddings for the special words [w$_1$], [w$_2$], and [p], which are used to represent a missing component, as in "cake made of [w$_1$]", "[w$_2$] made of apple", and "cake [p] apple".

For a missing component $x \in \{[p], [w_1], [w_2]\}$ surrounded by the sequences of words $v_{1:i-1}$ and $v_{i+1:n}$, we encode the sequence using a bidirectional long-short term memory (bi-LSTM) network (Graves and Schmidhuber, 2005), and take the ith output vector as representing the missing component: $bLS(v_{1:i}, x, v_{i+1:n})_i$.

In bi-LSTMs, each output vector is a concatenation of the outputs of the forward and backward LSTMs, so the output vector is expected to contain information on valid substitutions both with respect to the previous words $v_{1:i-1}$ and the subsequent words $v_{i+1:n}$.

Prediction. We predict a distribution of the vocabulary of the missing component, i.e. to predict w_1 correctly we need to predict its index in the word vocabulary V_w, while the prediction of p is from the vocabulary of paraphrases in the training set, V_p. We predict the following distributions:

$\hat{p} = \text{softmax}(W_p \cdot bLS(\vec{w_2}, [p], \vec{w_1})_2)$
$\hat{w_1} = \text{softmax}(W_w \cdot bLS(\vec{w_2}, \vec{p}_{1:n}, [w_1])_{n+1})$ (1)
$\hat{w_2} = \text{softmax}(W_w \cdot bLS([w_2], \vec{p}_{1:n}, \vec{w_1})_1)$

where $W_w \in \mathcal{R}^{|V_w| \times 2d}$, $W_p \in \mathcal{R}^{|V_p| \times 2d}$, and d is the embeddings dimension.

During training, we compute cross-entropy loss for each subtask using the gold item and the prediction, sum up the losses, and weight them by the instance score. During inference, we predict the missing components by picking the best scoring index in each distribution:[3]

$$\hat{p}_i = \text{argmax}(\hat{p})$$
$$\hat{w}_{1i} = \text{argmax}(\hat{w_1}) \quad (2)$$
$$\hat{w}_{2i} = \text{argmax}(\hat{w_2})$$

The subtasks share the pre-trained word embeddings, the special embeddings, and the biLSTM parameters. Subtasks (2) and (3) also share W_w, the MLP that predicts the index of a word.

---

[3]In practice, we pick the k best scoring indices in each distribution for some predefined k, as we discuss in Section 5.

[w$_1$]	[w$_2$]	Predicted Paraphrases	[w$_2$]	Paraphrase	Predicted [w$_1$]	Paraphrase	[w$_1$]	Predicted [w$_2$]
cataract	surgery	[w$_2$] of [w$_1$] [w$_2$] on [w$_1$] [w$_2$] to remove [w$_1$] [w$_2$] in patients with [w$_1$]	surgery	[w$_2$] to treat [w$_1$]	heart brain back knee	[w$_2$] to treat [w$_1$]	cataract	surgery drug patient transplant
software	company	[w$_2$] of [w$_1$] [w$_2$] to develop [w$_1$] [w$_2$] in [w$_1$] industry [w$_2$] involved in [w$_1$]	company	[w$_2$] engaged in [w$_1$]	management production computer business	[w$_2$] engaged in [w$_1$]	software	company firm engineer industry
stone	wall	[w$_2$] is of [w$_1$] [w$_2$] of [w$_1$] [w$_2$] is made of [w$_1$] [w$_2$] made of [w$_1$]	meeting	[w$_2$] held in [w$_1$]	spring afternoon hour day	[w$_2$] held in [w$_1$]	morning	party meeting rally session

Table 1: Examples of top ranked predicted components using the model: predicting the paraphrase given w_1 and w_2 (left), w_1 given w_2 and the paraphrase (middle), and w_2 given w_1 and the paraphrase (right).

Figure 2: A t-SNE map of a sample of paraphrases, using the paraphrase vectors encoded by the biLSTM, for example $bLS([w_2]$ made of $[w_1])$.

Implementation Details. The model is implemented in DyNet (Neubig et al., 2017). We dedicate a small number of noun-compounds from the corpus for validation. We train for up to 10 epochs, stopping early if the validation loss has not improved in 3 epochs. We use Momentum SGD (Nesterov, 1983), and set the batch size to 10 and the other hyper-parameters to their default values.

4 Qualitative Analysis

To estimate the quality of the proposed model, we first provide a qualitative analysis of the model outputs. Table 1 displays examples of the model outputs for each possible usage: predicting the paraphrase given the constituent words, and predicting each constituent word given the paraphrase and the other word.

The examples in the table are from among the top 10 ranked predictions for each component-pair. We note that most of the (w_2, paraphrase, w_1) triplets in the table do not occur in the training data, but are rather generalized from similar examples. For example, there is no training instance for "company in the software industry" but there is a "firm in the software industry" and a company in many other industries.

While the frequent prepositional paraphrases are often ranked at the top of the list, the model also retrieves more specified verbal paraphrases. The list often contains multiple semantically-similar paraphrases, such as '[w$_2$] involved in [w$_1$]' and '[w$_2$] in [w$_1$] industry'. This is a result of the model training objective (Section 3) which positions the vectors of semantically-similar paraphrases close to each other in the embedding space, based on similar constituents.

To illustrate paraphrase similarity we compute a t-SNE projection (Van Der Maaten, 2014) of the embeddings of all the paraphrases, and draw a sample of 50 paraphrases in Figure 2. The projection positions semantically-similar but lexically-divergent paraphrases in proximity, likely due to

many shared constituents. For instance, 'with', 'from', and 'out of' can all describe the relation between food words and their ingredients.

5 Evaluation: Noun-Compound Interpretation Tasks

For quantitative evaluation we employ our model for two noun-compound interpretation tasks. The main evaluation is on retrieving and ranking paraphrases (§5.1). For the sake of completeness, we also evaluate the model on classification to a fixed inventory of relations (§5.2), although it wasn't designed for this task.

5.1 Paraphrasing

Task Definition. The general goal of this task is to interpret each noun-compound to multiple prepositional and verbal paraphrases. In SemEval 2013 Task 4,[4] the participating systems were asked to retrieve a ranked list of paraphrases for each noun-compound, which was automatically evaluated against a similarly ranked list of paraphrases proposed by human annotators.

Model. For a given noun-compound $w_1 w_2$, we first predict the $k = 250$ most likely paraphrases: $\hat{p}_1, ..., \hat{p}_k = \text{argmax}_k \, \hat{p}$, where $\hat{p}$ is the distribution of paraphrases defined in Equation 1.

While the model also provides a score for each paraphrase (Equation 1), the scores have not been optimized to correlate with human judgments. We therefore developed a re-ranking model that receives a list of paraphrases and re-ranks the list to better fit the human judgments.

We follow Herbrich (2000) and learn a pairwise ranking model. The model determines which of two paraphrases of the same noun-compound should be ranked higher, and it is implemented as an SVM classifier using scikit-learn (Pedregosa et al., 2011). For training, we use the available training data with gold paraphrases and ranks provided by the SemEval task organizers. We extract the following features for a paraphrase p:

1. The part-of-speech tags contained in p
2. The prepositions contained in p
3. The number of words in p
4. Whether p ends with the special [w$_1$] symbol
5. $cosine(bLS([w_2], p, [w_1])_2, \vec{V}_p^{\hat{p}_i}) \cdot \hat{p}^{\hat{p}_i}$

where $\vec{V}_p^{\hat{p}_i}$ is the biLSTM encoding of the predicted paraphrase computed in Equation 1 and $\hat{p}^{\hat{p}_i}$

---

is its confidence score. The last feature incorporates the original model score into the decision, as to not let other considerations such as preposition frequency in the training set take over.

During inference, the model sorts the list of paraphrases retrieved for each noun-compound according to the pairwise ranking. It then scores each paraphrase by multiplying its rank with its original model score, and prunes paraphrases with final score < 0.025. The values for k and the threshold were tuned on the training set.

Evaluation Settings. The SemEval 2013 task provided a scorer that compares words and n-grams from the gold paraphrases against those in the predicted paraphrases, where agreement on a prefix of a word (e.g. in derivations) yields a partial scoring. The overall score assigned to each system is calculated in two different ways. The 'isomorphic' setting rewards both precision and recall, and performing well on it requires accurately reproducing as many of the gold paraphrases as possible, and in much the same order. The 'non-isomorphic' setting rewards only precision, and performing well on it requires accurately reproducing the top-ranked gold paraphrases, with no importance to order.

Baselines. We compare our method with the published results from the SemEval task. The SemEval 2013 baseline generates for each noun-compound a list of prepositional paraphrases in an arbitrary fixed order. It achieves a moderately good score in the non-isomorphic setting by generating a fixed set of paraphrases which are both common and generic. The MELODI system performs similarly: it represents each noun-compound using a compositional distributional vector (Mitchell and Lapata, 2010) which is then used to predict paraphrases from the corpus. The performance of MELODI indicates that the system was rather conservative, yielding a few common paraphrases rather than many specific ones. SFS and IIITH, on the other hand, show a more balanced trade-off between recall and precision.

As a sanity check, we also report the results of a baseline that retrieves ranked paraphrases from the training data collected in Section 3.2. This baseline has no generalization abilities, therefore it is expected to score poorly on the recall-aware isomorphic setting.

	Method	isomorphic	non-isomorphic
Baselines	SFS (Versley, 2013)	23.1	17.9
	IIITH (Surtani et al., 2013)	23.1	25.8
	MELODI (Van de Cruys et al., 2013)	13.0	54.8
	SemEval 2013 Baseline (Hendrickx et al., 2013)	13.8	40.6
This paper	Baseline	3.8	16.1
	Our method	**28.2**	28.4

Table 2: Results of the proposed method and the baselines on the SemEval 2013 task.

Category	%
False Positive	
(1) Valid paraphrase missing from gold	44%
(2) Valid paraphrase, slightly too specific	15%
(3) Incorrect, common prepositional paraphrase	14%
(4) Incorrect, other errors	14%
(5) Syntactic error in paraphrase	8%
(6) Valid paraphrase, but borderline grammatical	5%
False Negative	
(1) Long paraphrase (more than 5 words)	30%
(2) Prepositional paraphrase with determiners	25%
(3) Inflected constituents in gold	10%
(4) Other errors	35%

Table 3: Categories of false positive and false negative predictions along with their percentage.

Results. Table 2 displays the performance of the proposed method and the baselines in the two evaluation settings. Our method outperforms all the methods in the isomorphic setting. In the non-isomorphic setting, it outperforms the other two systems that score reasonably on the isomorphic setting (SFS and IIITH) but cannot compete with the systems that focus on achieving high precision.

The main advantage of our proposed model is in its ability to generalize, and that is also demonstrated in comparison to our baseline performance. The baseline retrieved paraphrases only for a third of the noun-compounds (61/181), expectedly yielding poor performance on the isomorphic setting. Our model, which was trained on the very same data, retrieved paraphrases for all noun-compounds. For example, *welfare system* was not present in the training data, yet the model predicted the correct paraphrases "system of welfare benefits", "system to provide welfare" and others.

Error Analysis. We analyze the causes of the false positive and false negative errors made by the model. For each error type we sample 10 noun-compounds. For each noun-compound, false positive errors are the top 10 predicted paraphrases which are not included in the gold paraphrases, while false negative errors are the top 10 gold paraphrases not found in the top k predictions made by the model. Table 3 displays the manu-

ally annotated categories for each error type.

Many false positive errors are actually valid paraphrases that were not suggested by the human annotators (error 1, "discussion by group"). Some are borderline valid with minor grammatical changes (error 6, "force of coalition forces") or too specific (error 2, "life *of women in* community" instead of "life *in* community"). Common prepositional paraphrases were often retrieved although they are incorrect (error 3). We conjecture that this error often stem from an n-gram that does not respect the syntactic structure of the sentence, e.g. a sentence such as "rinse away the oil from baby 's head" produces the n-gram "oil from baby".

With respect to false negative examples, they consisted of many long paraphrases, while our model was restricted to 5 words due to the source of the training data (error 1, "holding done in the case of a share"). Many prepositional paraphrases consisted of determiners, which we conflated with the same paraphrases without determiners (error 2, "mutation of a gene"). Finally, in some paraphrases, the constituents in the gold paraphrase appear in inflectional forms (error 3, "holding of shares" instead of "holding of share").

5.2 Classification

Noun-compound classification is defined as a multiclass classification problem: given a pre-defined set of relations, classify $w_1 w_2$ to the relation that holds between w_1 and w_2. Potentially, the corpus co-occurrences of w_1 and w_2 may contribute to the classification, e.g. '[w$_2$] held at [w$_1$]' indicates a TIME relation. Tratz and Hovy (2010) included such features in their classifier, but ablation tests showed that these features had a relatively small contribution, probably due to the sparseness of the paraphrases. Recently, Shwartz and Waterson (2018) showed that paraphrases may contribute to the classification when represented in a continuous space.

Model. We generate a paraphrase vector representation $p\vec{a}r(w_1w_2)$ for a given noun-compound w_1w_2 as follows. We predict the indices of the k most likely paraphrases: $\hat{p}_1, ..., \hat{p}_k = \text{argmax}_k\,\hat{p}$, where $\hat{p}$ is the distribution on the paraphrase vocabulary V_p, as defined in Equation 1. We then encode each paraphrase using the biLSTM, and average the paraphrase vectors, weighted by their confidence scores in $\hat{p}$:

$$p\vec{a}r(w_1w_2) = \frac{\sum_{i=1}^{k}\hat{p}^{\hat{p}_i} \cdot \vec{V}_p^{\hat{p}_i}}{\sum_{i=1}^{k}\hat{p}^{\hat{p}_i}} \quad (3)$$

We train a linear classifier, and represent w_1w_2 in a feature vector $f(w_1w_2)$ in two variants: paraphrase: $f(w_1w_2) = p\vec{a}r(w_1w_2)$, or integrated: concatenated to the constituent word embeddings $f(w_1w_2) = [p\vec{a}r(w_1w_2), \vec{w_1}, \vec{w_2}]$. The classifier type (logistic regression/SVM), k, and the penalty are tuned on the validation set. We also provide a baseline in which we ablate the paraphrase component from our model, representing a noun-compound by the concatenation of its constituent embeddings $f(w_1w_2) = [\vec{w_1}, \vec{w_2}]$ (distributional).

Datasets. We evaluate on the Tratz (2011) dataset, which consists of 19,158 instances, labeled in 37 fine-grained relations (`Tratz-fine`) or 12 coarse-grained relations (`Tratz-coarse`).

We report the performance on two different dataset splits to train, test, and validation: a random split in a 75:20:5 ratio, and, following concerns raised by Dima (2016) about lexical memorization (Levy et al., 2015), on a lexical split in which the sets consist of distinct vocabularies. The lexical split better demonstrates the scenario in which a noun-compound whose constituents have not been observed needs to be interpreted based on similar observed noun-compounds, e.g. inferring the relation in *pear tart* based on *apple cake* and other similar compounds. We follow the random and full-lexical splits from Shwartz and Waterson (2018).

Baselines. We report the results of 3 baselines representative of different approaches:
1) **Feature-based** (Tratz and Hovy, 2010): we re-implement a version of the classifier with features from WordNet and Roget's Thesaurus.
2) **Compositional** (Dima, 2016): a neural architecture that operates on the distributional representations of the noun-compound and its constituents. Noun-compound representations are learned with

Dataset & Split	Method	F_1
Tratz fine Random	Tratz and Hovy (2010)	**0.739**
	Dima (2016)	0.725
	Shwartz and Waterson (2018)	0.714
	distributional	**0.677**
	paraphrase	0.505
	integrated	0.673
Tratz fine Lexical	Tratz and Hovy (2010)	0.340
	Dima (2016)	0.334
	Shwartz and Waterson (2018)	**0.429**
	distributional	0.356
	paraphrase	0.333
	integrated	**0.370**
Tratz coarse Random	Tratz and Hovy (2010)	0.760
	Dima (2016)	**0.775**
	Shwartz and Waterson (2018)	0.736
	distributional	0.689
	paraphrase	0.557
	integrated	**0.700**
Tratz coarse Lexical	Tratz and Hovy (2010)	0.391
	Dima (2016)	0.372
	Shwartz and Waterson (2018)	**0.478**
	distributional	0.370
	paraphrase	0.345
	integrated	**0.393**

Table 4: Classification results. For each dataset split, the top part consists of baseline methods and the bottom part of methods from this paper. The best performance in each part appears in bold.

the Full-Additive (Zanzotto et al., 2010) and Matrix (Socher et al., 2012) models. We report the results from Shwartz and Waterson (2018).
3) **Paraphrase-based** (Shwartz and Waterson, 2018): a neural classification model that learns an LSTM-based representation of the joint occurrences of w_1 and w_2 in a corpus (i.e. observed paraphrases), and integrates distributional information using the constituent embeddings.

Results. Table 4 displays the methods' performance on the two versions of the Tratz (2011) dataset and the two dataset splits. The paraphrase model on its own is inferior to the distributional model, however, the integrated version improves upon the distributional model in 3 out of 4 settings, demonstrating the complementary nature of the distributional and paraphrase-based methods. The contribution of the paraphrase component is especially noticeable in the lexical splits.

As expected, the integrated method in Shwartz and Waterson (2018), in which the paraphrase representation was trained with the objective of classification, performs better than our integrated model. The superiority of both integrated models in the lexical splits confirms that paraphrases are beneficial for classification.

Example Noun-compounds	Gold	Distributional	Example Paraphrases
printing plant	PURPOSE	OBJECTIVE	[w_2] engaged in [w_1]
marketing expert *development expert*	TOPICAL	OBJECTIVE	[w_2] in [w_1] [w_2] knowledge of [w_1]
weight/job loss	OBJECTIVE	CAUSAL	[w_2] of [w_1]
rubber band *rice cake*	CONTAINMENT	PURPOSE	[w_2] made of [w_1] [w_2] is made of [w_1]
laboratory animal	LOCATION/PART-WHOLE	ATTRIBUTE	[w_2] in [w_1], [w_2] used in [w_1]

Table 5: Examples of noun-compounds that were correctly classified by the integrated model while being incorrectly classified by distributional, along with top ranked indicative paraphrases.

Analysis. To analyze the contribution of the paraphrase component to the classification, we focused on the differences between the distributional and integrated models on the Tratz-Coarse lexical split. Examination of the per-relation F_1 scores revealed that the relations for which performance improved the most in the integrated model were TOPICAL (+11.1 F_1 points), OBJECTIVE (+5.5), ATTRIBUTE (+3.8) and LOCATION/PART_WHOLE (+3.5).

Table 5 provides examples of noun-compounds that were correctly classified by the integrated model while being incorrectly classified by the distributional model. For each noun-compound, we provide examples of top ranked paraphrases which are indicative of the gold label relation.

6 Compositionality Analysis

Our paraphrasing approach at its core assumes compositionality: only a noun-compound whose meaning is derived from the meanings of its constituent words can be rephrased using them. In §3.2 we added negative samples to the training data to simulate non-compositional noun-compounds, which are included in the classification dataset (§5.2). We assumed that these compounds, more often than compositional ones would consist of unrelated constituents (*spelling bee*, *sacred cow*), and added instances of random unrelated nouns with '[w_2] is unrelated to [w_1]'. Here, we assess whether our model succeeds to recognize non-compositional noun-compounds.

We used the compositionality dataset of Reddy et al. (2011) which consists of 90 noun-compounds along with human judgments about their compositionality in a scale of 0-5, 0 being non-compositional and 5 being compositional. For each noun-compound in the dataset, we predicted the 15 best paraphrases and analyzed the errors. The most common error was predicting paraphrases for idiomatic compounds which may have

a plausible concrete interpretation or which originated from one. For example, it predicted that *silver spoon* is simply a *spoon* made of *silver* and that *monkey business* is a *business* that buys or raises *monkeys*. In other cases, it seems that the strong prior on one constituent leads to ignoring the other, unrelated constituent, as in predicting "*wedding* made of *diamond*". Finally, the "unrelated" paraphrase was predicted for a few compounds, but those are not necessarily non-compositional (*application form*, *head teacher*). We conclude that the model does not address compositionality and suggest to apply it only to compositional compounds, which may be recognized using compositionality prediction methods as in Reddy et al. (2011).

7 Conclusion

We presented a new semi-supervised model for noun-compound paraphrasing. The model differs from previous models by being trained to predict both a paraphrase given a noun-compound, and a missing constituent given the paraphrase and the other constituent. This results in better generalization abilities, leading to improved performance in two noun-compound interpretation tasks. In the future, we plan to take generalization one step further, and explore the possibility to use the biLSTM for generating completely new paraphrase templates unseen during training.

References

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*. http://aclweb.org/anthology/P01-1008.

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

Cristina Butnariu, Su Nam Kim, Preslav Nakov, Diarmuid Ó Séaghdha, Stan Szpakowicz, and Tony Veale. 2009. Semeval-2010 task 9: The interpretation of noun compounds using paraphrasing verbs and prepositions. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW-2009)*. Association for Computational Linguistics, Boulder, Colorado, pages 100–105. http://www.aclweb.org/anthology/W09-2416.

Corina Dima. 2016. *Proceedings of the 1st Workshop on Representation Learning for NLP*, Association for Computational Linguistics, chapter On the Compositionality and Semantic Interpretation of English Noun Compounds, pages 27–39. https://doi.org/10.18653/v1/W16-1604.

Corina Dima and Erhard Hinrichs. 2015. Automatic noun compound interpretation using deep neural networks and word embeddings. *IWCS 2015* page 173.

Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S Weld. 2008. Open information extraction from the web. *Communications of the ACM* 51(12):68–74.

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*. Association for Computational Linguistics, pages 758–764. http://aclweb.org/anthology/N13-1092.

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*. Association for Computational Linguistics, Prague, Czech Republic, pages 568–575. http://www.aclweb.org/anthology/P07-1072.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks* 18(5-6):602–610.

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). Association for Computational Linguistics, pages 138–143. http://aclweb.org/anthology/S13-2025.

Ralf Herbrich. 2000. Large margin rank boundaries for ordinal regression. *Advances in large margin classifiers* pages 115–132.

Nam Su Kim and Preslav Nakov. 2011. Large-scale noun compound interpretation using bootstrapping and the web as a corpus. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 648–658. http://aclweb.org/anthology/D11-1060.

Su Nam Kim and Timothy Baldwin. 2007. Interpreting noun compounds using bootstrapping and sense collocation. In *Proceedings of Conference of the Pacific Association for Computational Linguistics*. pages 129–136.

Omer Levy, Steffen Remus, Chris Biemann, and Ido Dagan. 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*. Association for Computational Linguistics, Denver, Colorado, pages 970–976. http://www.aclweb.org/anthology/N15-1098.

Guofu Li, Alejandra Lopez-Fernandez, and Tony Veale. 2010. Ucd-goggle: A hybrid system for noun compound paraphrasing. In *Proceedings of the 5th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, pages 230–233.

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*. Association for Computational Linguistics, Valencia, Spain, pages 881–893.

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(03):291–330.

Preslav Nakov and Marti Hearst. 2006. Using verbs to characterize noun-noun relations. In *International Conference on Artificial Intelligence: Methodology, Systems, and Applications*. Springer, pages 233–244.

Vivi Nastase and Stan Szpakowicz. 2003. Exploring noun-modifier semantic relations. In *Fifth international workshop on computational semantics (IWCS-5)*. pages 285–301.

Yurii Nesterov. 1983. A method of solving a convex programming problem with convergence rate o (1/k2). In *Soviet Mathematics Doklady*. volume 27, pages 372–376.

Graham Neubig, Chris Dyer, Yoav Goldberg, Austin Matthews, Waleed Ammar, Antonios Anastasopoulos, Miguel Ballesteros, David Chiang, Daniel Clothiaux, Trevor Cohn, et al. 2017. Dynet: The dynamic neural network toolkit. *arXiv preprint arXiv:1701.03980* .

Paul Nulty and Fintan Costello. 2010. Ucd-pn: Selecting general paraphrases using conditional probability. In *Proceedings of the 5th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, pages 234–237.

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)*. Association for Computational Linguistics, Athens, Greece, pages 621–629. http://www.aclweb.org/anthology/E09-1071.

Harinder Pal and Mausam. 2016. Demonyms and compound relational nouns in nominal open ie. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction*. Association for Computational Linguistics, San Diego, CA, pages 35–39. http://www.aclweb.org/anthology/W16-1307.

Patrick Pantel, Rahul Bhagat, Bonaventura Coppola, Timothy Chklovski, and Eduard Hovy. 2007. ISP: Learning inferential selectional preferences. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*. Association for Computational Linguistics, Rochester, New York, pages 564–571. http://www.aclweb.org/anthology/N/N07/N07-1071.

Marius Pasca. 2015. Interpreting compound noun phrases using web search queries. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pages 335–344. https://doi.org/10.3115/v1/N15-1037.

Ellie Pavlick and Marius Pasca. 2017. Identifying 1950s american jazz musicians: Fine-grained isa extraction via modifier composition. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, pages 2099–2109. http://aclweb.org/anthology/P17-1192.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and

E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, pages 1532–1543. http://www.aclweb.org/anthology/D14-1162.

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*. Asian Federation of Natural Language Processing, Chiang Mai, Thailand, pages 210–218. http://www.aclweb.org/anthology/I11-1024.

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 *The 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. New Orleans, Louisiana.

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*. Association for Computational Linguistics, pages 1201–1211. http://aclweb.org/anthology/D12-1110.

Nitesh Surtani, Arpita Batra, Urmi Ghosh, and Soma Paul. 2013. Iiit-h: A corpus-driven co-occurrence based probabilistic model for noun compound paraphrasing. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. volume 2, pages 153–157.

Stephen Tratz. 2011. *Semantically-enriched parsing for natural language understanding*. University of Southern California.

Stephen Tratz and Eduard Hovy. 2010. A taxonomy, dataset, and classifier for automatic noun compound interpretation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Uppsala, Sweden, pages 678–687. http://www.aclweb.org/anthology/P10-1070.

Tim Van de Cruys, Stergos Afantenos, and Philippe Muller. 2013. Melodi: A supervised distributional approach for free paraphrasing of noun compounds. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (Se-*

mEval 2013). Association for Computational Linguistics, Atlanta, Georgia, USA, pages 144–147. http://www.aclweb.org/anthology/S13-2026.

Laurens Van Der Maaten. 2014. Accelerating t-sne using tree-based algorithms. *Journal of machine learning research* 15(1):3221–3245.

Yannick Versley. 2013. Sfs-tue: Compound paraphrasing with a language model and discriminative reranking. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. volume 2, pages 148–152.

Sander Wubben. 2010. Uvt: Memory-based pairwise ranking of paraphrasing verbs. In *Proceedings of the 5th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, pages 260–263.

Clarissa Xavier and Vera Lima. 2014. Boosting open information extraction with noun-based relations. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. European Language Resources Association (ELRA), Reykjavik, Iceland.

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*. Association for Computational Linguistics, pages 1263–1271.