Olive Oil is Made *of* Olives, Baby Oil is Made *for* Babies: Interpreting Noun Compounds using Paraphrases in a Neural Model

Anonymous ACL submission

Abstract

Automatic interpretation of the relation between the constituents of a noun compound, e.g. *olive oil* (source) and *baby oil* (purpose) is an important task for many NLP applications. Recent approaches are typically based on either noun-compound representations or paraphrases. While the former has initially shown promising results, recent work suggests that the success stems from memorizing single prototypical words for each relation. We explore a neural paraphrasing approach that demonstrates superior performance when such memorization is not possible.

1 Introduction

Automatic classification of a noun-compound (NC) to the implicit semantic relation that holds between its constituent words is beneficial for applications that require text understanding. For instance, a personal assistant asked “do I have a *morning meeting* tomorrow?” should search the calendar for meetings occurring in the morning, while for *group meeting* it should look for meetings with specific participants. The NC classification task is a challenging one, as the meaning of an NC is often not easily derivable from the meaning of its constituent words (Spärck Jones, 1983).

Previous work on the task falls into two main approaches. The first maps NCs to paraphrases that express the relation between the constituent words (e.g. Nakov and Hearst, 2006; Nulty and Costello, 2013), such as mapping *coffee cup* and *garbage dump* to the pattern $[w_1]$ CONTAINS $[w_2]$. The second approach computes a representation for NCs from the distributional representation of their individual constituents. While this approach yielded promising results, recently, Dima (2016) showed that similar performance is achieved by representing the NC as a concatenation of its constituent embeddings, and attributed it to the *lexical memorization* phenomenon (Levy et al., 2015).

In this paper we apply lessons learned from the parallel task of semantic relation classification. We adapt HypeNET (Shwartz et al., 2016) to the NC classification task, representing paraphrases using their path embedding mechanism and integrating distributional information. We use various evaluation settings, including settings that make lexical memorization impossible. The results confirm that the integrated method improves performance in these settings, but even so, the performance is mediocre for all methods, suggesting that the task is difficult and warrants further investigation.[1]

2 Background

The NC classification task is defined as follows: given a pre-defined set of relations, classify $nc = w_1 w_2$ to the relation that holds between w_1 and w_2. Past work leveraged information derived from lexical resources and corpora (e.g. Nastase and Szpakowicz, 2003; Tratz and Hovy, 2010). We focus on more recent work, which can be roughly divided into the following approaches.

2.1 Compositional Representations

In this approach, NC is represented as vector computed by applying a function to its constituents. In this section, w_1, w_2 denote the elements of a noun-compound nc, with the word vectors $\vec{v}_{w_1}, \vec{v}_{w_2} \in \mathcal{R}^n$ respectively.

Mitchell and Lapata (2010) proposed 3 simple combinations of $\vec{v}_{w_1}$ and $\vec{v}_{w_2}$ (additive, multiplicative, dilation). Others suggested to represent compositions by applying linear functions, encoded as

[1] We will make the code available upon publication.

matrices, over word vectors. Baroni and Zamparelli (2010) focused on adjective-noun compositions (AN) and represented adjectives as matrices, nouns as vectors, and ANs as their multiplication. Matrices were learned with the objective of minimizing the distance between the learned vector and the observed vector (computed from corpus occurrences) of each AN. The full-additive model (Zanzotto et al., 2010; Dinu et al., 2013) is a similar approach that works on any two-word composition, multiplying each word by a square matrix: $nc = A \cdot \vec{v}_{w_1} + B \cdot \vec{v}_{w_2}$.

Socher et al. (2012) suggested a non-linear composition model. A recursive neural network operates bottom-up on the output of a constituency parser to represent variable-length phrases. Each constituent is represented by a vector that captures its meaning and a matrix that captures how it modifies the meaning of constituents that it combines with. For a binary NC, $nc = g(W \cdot [\vec{v}_{w_1}; \vec{v}_{w_2}])$, where $W \in \mathcal{R}^{2n \times n}$ and g is a non-linear function.

These representations were used as features in NC classification, often achieving promising results (e.g. Van de Cruys et al., 2013; Dima and Hinrichs, 2015). However, Dima (2016) recently showed that similar performance is achieved by representing the NC as a concatenation of its constituent embeddings, and argued that it stems from memorizing prototypical words for each relation.

2.2 Paraphrasing

Nakov and Hearst (2006) express the semantics of an NC with multiple prepositional and verbal paraphrases. For example, *student protest* is a protest LED BY, BE SPONSORED BY, or BE ORGANIZED BY students. While early work on this approach focused on unsupervised extraction of paraphrases (e.g. Kim and Nakov, 2011; Wijaya and Gianfortoni, 2011; Hendrickx et al., 2013; Nulty and Costello, 2013),[2] paraphrases can also be used as features for NC classification.

A general problem with this approach is that the feature space is sparse, especially for infrequent NCs. Recently, Surtani and Paul (2015) suggested to represent NCs in a vector space model (VSM) using paraphrases as features. These vectors were used to classify new NCs based on the nearest neighbor in the VSM. However, the model was only tested on a small dataset and performed similarly to previous methods.

3 Model

We similarly investigate the use of paraphrasing for NC relation classification. We first learn a representation for NC paraphrases (§3.1), and then use this representation along with distributional information in the classification task (§3.2).

3.1 Paraphrase Representation

To generate a signal for the joint occurrences of w_1 and w_2, we follow the approach used by HypeNET (Shwartz et al., 2016). For an NC $w_1 w_2$, we collect all the dependency paths that connect w_1 and w_2 in the corpus. We represent each edge by the concatenation of its lemma, part-of-speech tag, dependency label and direction vectors: $\vec{v_e} = [\vec{v_l}, \vec{v_{pos}}, \vec{v_{dep}}, \vec{v_{dir}}]$. For a path p composed of edges $e_1, ..., e_k$, the edge vectors $\vec{v_{e_1}}, ..., \vec{v_{e_k}}$ are encoded using an LSTM (Hochreiter and Schmidhuber, 1997), resulting in a vector $\vec{o_p}$ representing the entire path.

We predict the label distribution for each path, and use weighted average to obtain the NC label: $r = \text{argmax}_i \frac{\sum_{p \in paths(w_1,w_2)} f_{p,(w_1,w_2)} \cdot \text{softmax}(\vec{o_p})}{\sum_{p \in paths(w_1,w_2)} f_{p,(w_1,w_2)}}$.[3]

We initialize the lemma embeddings with pre-trained embeddings and – unlike HypeNET – do not update them during training. The other component embeddings are initialized randomly and updated during training. We compute the path embeddings in advance for all the paths in the dataset, saving a path embedding matrix M_p (Figure 1).

3.2 Classification Models

Figure 1 provides an overview of the models: path-based, integrated, and integrated-NC, each which incrementally adds new features not present in the previous model. In the following sections, $\vec{x}$ denotes the input vector representing the NC. The network classifies NC to the highest scoring relation: $r = \text{argmax}_i \, \text{softmax}(\vec{o})[i]$, where $\vec{o}$ is the output layer. All networks contain a single hidden layer whose dimension is $\frac{|x|}{2}$. k is the number of relations in the dataset. See supplementary material for additional technical details.

Path-based. This model uses only the paths as features. We define the NC path vector as the weighted average of its path embeddings:

[2] See Nakov (2013), page 25, for additional methods.

[3] This differs slightly from HypeNET, which predicts a word pair's label from the frequency-weighted average of the path vectors. We conjecture that label distribution averaging allows for more efficient training of path embeddings when a single NC contains multiple paths.

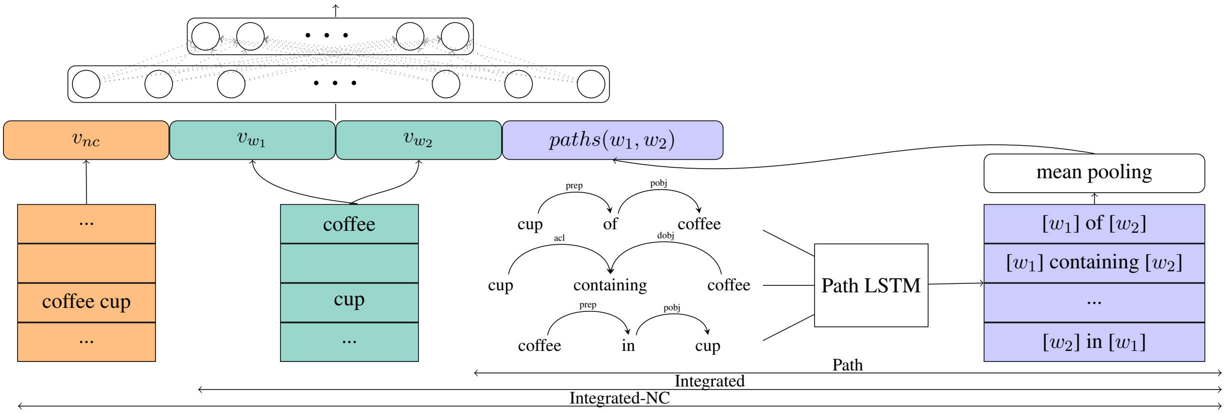

Figure 1: An illustration of the classification models for the NC *coffee cup*. The model consists of two parts: (1) the distributional representations of the NC (left, orange) and each word (middle, green). (2) the corpus occurrences of *coffee* and *cup*, in the form of dependency path embeddings (right, purple).

$\vec{v}_{paths(w_1,w_2)} = \frac{\sum_{p \in paths(w_1,w_2)} f_{p,(w_1,w_2)} \cdot (M_p)_p}{\sum_{p \in paths(w_1,w_2)} f_{p,(w_1,w_2)}}$. In this model we define $x = \vec{v}_{paths(w_1,w_2)}$.

Integrated. We concatenate w_1 and w_2's word embeddings to the path vector, to add distributional information: $x = [\vec{v}_{w_1}, \vec{v}_{w_2}, \vec{v}_{paths(w_1,w_2)}]$. Potentially, this allows the network to utilize the contextual properties of each individual constituent, e.g. assigning high probability to SUBSTANCE-MATERIAL-INGREDIENT for edible w_1s (e.g. *vanilla pudding, apple cake*).

Integrated-NC. We add the NC's *observed* vector $\vec{v}_{nc}$ as additional distributional input, providing the contexts in which $w_1 w_2$ occur as an NC: $\vec{v}_{nc} = [\vec{v}_{w_1}, \vec{v}_{w_2}, \vec{v}_{nc}, \vec{v}_{paths(w_1,w_2)}]$. Like Dima (2016), we learn NC vectors using the GloVe algorithm (Pennington et al., 2014), by replacing each NC occurrence in the corpus with a single token.

This information can potentially help clustering NCs that appear in similar contexts despite having low pairwise similarity scores between their constituents. For example, *gun violence* and *abortion rights* belong to the TOPIC relation and may appear in similar news-related contexts, while *(gun, abortion)* and *(violence, rights)* are dissimilar.

4 Evaluation

4.1 Dataset

We follow Dima (2016) and evaluate on the Tratz (2011) dataset, with 19,158 instances and two levels of labels: fine-grained (`Tratz-fine`, 37 relations) and coarse-grained (`Tratz-coarse`, 12 relations). We report results on both versions. See Tratz (2011) for the list of relations.

Dataset Splits Dima (2016) showed that a classifier based only on v_{w_1} and v_{w_2} performs on par with compound representations, and that the success comes from lexical memorization (Levy et al., 2015): memorizing the majority label of single words in particular slots of the compound (e.g. TOPIC for *travel guide*, *fishing guide*, etc.). This memorization paints a skewed picture of the state-of-the-art performance on this difficult task.

To better test this hypothesis, we evaluate on four different splits of the datasets to train, validation, and test sets: (1) **random**, in a 75:20:5 ratio, (2) **lexical-full**, in which each set consists of a distinct vocabulary (Levy et al., 2015), and (3)/(4) the **lexical-mod** (**lexical-head**) splits in which the w_1 (w_2) words are unique in each set.[4]

4.2 Baselines

Frequency Baselines. *mod freq* classifies $w_1 w_2$ to the most common relation in the train set for NCs with the same modifier ($w_1 w_2'$), while *head freq* considers NCs with the same head ($w_1' w_2$).[5]

Distributional Baselines. Ablation of the path-based component from our models: **Dist** uses only w_1 and w_2's word embeddings: $\vec{x} = [\vec{v}_{w_1}, \vec{v}_{w_2}]$, while **Dist-NC** includes also the NC embedding: $\vec{x} = [\vec{v}_{w_1}, \vec{v}_{w_2}, \vec{v}_{nc}]$. The network architecture is defined similarly to our models (Section 3.2).

Compositional Baselines. We re-train Dima's (2016) models, various combinations of NC representations (Zanzotto et al., 2010; Socher et al., 2012) and single word embeddings in a fully connected network.[6]

[4] See supplementary material for the sizes of each split.

[5] Unseen heads/modifiers are assigned a random relation.

[6] We only include the compositional models, and omit the "basic" setting which is similar to our Dist model. For the full details of the compositional models, see Dima (2016).

Dataset	Split	Best Freq	Dist	Dist-NC	Best Comp
Tratz-fine	Rand	0.319	0.692	0.673	**0.725**
	Lex$_{head}$	0.222	**0.458**	0.449	0.450
	Lex$_{mod}$	0.292	0.574	0.559	**0.607**
	Lex$_{full}$	0.066	**0.363**	0.360	0.334
Tratz-coarse	Rand	0.256	0.734	0.718	**0.775**
	Lex$_{head}$	0.225	0.501	0.497	**0.538**
	Lex$_{mod}$	0.282	0.630	0.600	**0.645**
	Lex$_{full}$	0.136	0.406	**0.409**	0.372

Table 1: Baseline results (F_1) on the various splits: **best freq**: best performing frequency baseline, **best comp**: best model from Dima (2016).

4.3 Results

Table 1 shows the performance of baseline methods (Section 4.2) on the datasets, while Table 2 compares the best-performing baseline on each dataset to our proposed methods (Section 3.2).

Dima's (2016) compositional models perform best among the baselines, and on the random split, better than all the methods. On the lexical splits, however, the baselines exhibit a dramatic drop in performance, and are outperformed by our methods. The gap is larger in the lexical-full split.

Finally, there is no gain from the added NC vector in Dist-NC and Integrated-NC.

5 Analysis

To focus on the changes from previous work, we analyze the performance of the path-based model on the Tratz-fine random split. This dataset contains 37 relations and the model performance varies across them. Some relations, such as MEASURE and PERSONAL_TITLE yield reasonable performance (F_1 score of 0.87 and 0.68). Table 3 focuses on these relations and illustrates the indicative paths that the model has learned for each relation. We compute these by performing the analysis in Shwartz et al. (2016), where each path is fed into the path-based model, and is assigned to its best-scoring relation. For each relation, we consider paths with a score ≥ 0.8.

Other relations achieve very low F$_1$ scores, indicating that the model is unable to learn them at all. Interestingly, the four relations with the lowest performance in our model [7] are also those with the highest error rate in Dima (2016), very likely since they express complex relations. For example, the LEXICALIZED relation contains non-compositional NCs (*soap opera*) or lexical items whose meanings departed from the combination of the constituent meanings. It is expected that there are no paths that indicate lexicalization. In PARTIAL_ATTRIBUTE_TRANSFER (*bullet train*), w_1 transfers an attribute to w_2 (e.g. *bullet* transfers speed to *train*). These relations are not expected to be expressed in text, unless the text aims to explain them (e.g. *train as fast as a bullet*).

Dataset	Split	Best Baseline	Path	Int	Int-NC
Tratz-fine	Rand	**0.725**	0.538	0.714	0.692
	Lex$_{head}$	0.458	0.448	**0.510**	0.471
	Lex$_{mod}$	0.607	0.472	**0.613**	0.600
	Lex$_{full}$	0.363	0.423	0.421	**0.429**
Tratz-coarse	Rand	**0.775**	0.586	0.724	0.710
	Lex$_{head}$	0.538	0.518	**0.569**	0.548
	Lex$_{mod}$	0.645	0.548	**0.646**	0.632
	Lex$_{full}$	0.409	0.472	**0.475**	0.455

Table 2: Our method results (F_1) on the various splits, compared with the best baseline performance from Table 1.

relation	path	examples
MEASURE	[w_2] varies by [w_1]	*state limit*
	2,560 [w_1] portion of [w_2]	*acre estate*
PERSONAL TITLE	[w_2] Anderson [w_1]/title	*Mrs. Brown*
	[w_2] Sheridan [w_1]/title	*Gen. Johnson*
CREATE-PROVIDE-GENERATE-SELL	[w_2] produce [w_1]	*food producer*
	[w_2] manufacture [w_1]	*engine plant*
TIME-OF1	[w_2] begin [w_1]	*morning program*
	[w_2] held Saturday [w_1]	*afternoon meeting*
SUBSTANCE-MATERIAL–INGREDIENT	[w_2] made of wood and [w_1]	*marble table*
	[w_2] material includes type of [w_1]	*steel pipe*

Table 3: Indicative paths for selected relations.

Looking closer at the model confusions shows that it often defaulted to general relations like OBJECTIVE (*recovery plan*) or RELATIONAL-NOUN-COMPLEMENT (*eye shape*). The latter is described as "indicating the complement of a relational noun (e.g., son of, price of)", and the indicative paths for this relation indeed contain many variants of "[w_2] of [w_1]", which potentially can occur with NCs in other relations. The model also confused between relations with subtle differences, such as the different topic relations. Given that these relations were conflated to a single relation in the inter-annotator agreement computation in Tratz and Hovy (2010), we can conjecture that even humans find it difficult to distinguish between them.

6 Conclusion

We used an existing neural dependency path representation to represent noun-compound paraphrases, and along with distributional information applied it to the NC classification task. Following previous work, that suggested that distributional methods succeed due to lexical memorization, we show that when lexical memorization is not possible, the performance of all methods is much worse. Adding the path-based component helps mitigate this issue and increase performance.

[7] LEXICALIZED, TOPIC_OF_COGNITION&EMOTION, WHOLE+ATTRIBUTE&FEAT, PARTIAL_ATTR_TRANSFER

References

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

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.

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*. Association for Computational Linguistics, Sofia, Bulgaria, pages 50–58. http://www.aclweb.org/anthology/W13-3206.

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.

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

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.

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.

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.

Paul Nulty and Fintan Costello. 2013. General and specific paraphrases of semantic relations between nouns. *Natural Language Engineering* 19(03):357–384.

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.

Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving hypernymy detection with an integrated path-based and distributional method. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 2389–2398. http://www.aclweb.org/anthology/P16-1226.

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.

Karen Spärck Jones. 1983. Compound noun interpretation problems. Technical report, University of Cambridge, Computer Laboratory.

Nitesh Surtani and Soma Paul. 2015. A vsm-based statistical model for the semantic relation interpretation of noun-modifier pairs. In *RANLP*. pages 636–645.

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

Derry Tanti Wijaya and Philip Gianfortoni. 2011. Nut case: What does it mean?: Understanding semantic relationship between nouns in noun compounds through paraphrasing and ranking the paraphrases. In *Proceedings of the 1st international workshop on Search and mining entity-relationship data*. ACM, pages 9–14.

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.