Treat us like the sequences we are: Prepositional Paraphrasing of Noun Compounds using LSTM

Girishkumar Ponkiya[†]**, Kevin Patel**[†]**, Pushpak Bhattacharyya**[†] and **Girish K Palshikar**[‡]
[†]Indian Institute of Technology Bombay, Mumbai
[‡]TCS Research, Tata Consultancy Services, Pune
{girishp,kevin.patel,pb}@cse.iitb.ac.in, gk.palshikar@tcs.com

Abstract

Interpreting noun compounds is a challenging task. It involves uncovering the underlying predicate which is dropped in the formation of the compound. In most cases, this predicate is of the form VERB+PREP. It has been observed that uncovering the preposition is a significant step towards uncovering the predicate.

In this paper, we attempt to paraphrase noun compounds using prepositions. We consider noun compounds and their corresponding prepositional paraphrases as parallelly aligned sequences of words. This enables us to adapt different architectures from cross-lingual embedding literature. We choose the architecture where we create representations of both noun compound (source sequence) and its corresponding prepositional paraphrase (target sequence), such that their similarity is high. We use LSTMs to learn these representations. We use these representations to decide the correct preposition. Our experiments show that this approach performs considerably well on different datasets of noun compounds that are manually annotated with prepositions.

1 Introduction

A noun compound is a sequence of two or more nouns which have a well-defined meaning when written together. For example, *orange juice*, *colon cancer*, *research paper submission*, *paper submission deadline*, *etc.* The fact that "*juice is made from orange*" is hidden in *orange juice*. Noun compound interpretation deals with uncovering such hidden relations.

Noun compounds are usually interpreted in two ways: labelling and paraphrasing. Labelling involves assigning a semantic relation to a noun compound *e.g.*, *student protest*: AGENT, *orange juice*: MADEOF, *etc.*. These relations come from a set of a predefined taxonomy of semantic relations (Lauer, 1995; Warren, 1978; Barker and Szpakowicz, 1998; Girju et al., 2003; Tratz and Hovy, 2010; Ponkiya et al., 2018). Such detailed, fine-grained information can be useful for downstream tasks such as machine translation (Baldwin and Tanaka, 2004; Balyan and Chatterjee, 2015), question answering (Ahn et al., 2005), text entailment (Nakov, 2013), *etc.* Unfortunately, there is a lack of standard taxonomy. There is no consensus on which set of labels should be uniformly used.

On the other hand, paraphrasing involves rewriting the noun compound as a paraphrase which conveys its meaning explicitly, *e.g.*, *orange juice*: "*juice made from orange*" or "*juice with orange flavour*". Generic paraphrasing has been relatively less pursued (Butnariu et al., 2009; Hendrickx et al., 2013).

Prepositional paraphrasing – paraphrasing using only prepositions, *e.g.*, *orange juice*: "*juice of orange*" – is a simpler version of generic paraphrasing. The advantage of prepositional paraphrasing as compared to labelling is that the set of prepositions is finite, limited and pre-defined. However, the shortcoming is that the information is too coarse-grained for downstream tasks.

Prepositional paraphrasing of noun compounds is a useful subtask to solve. Our preliminary analysis reveals a strong correlation between labels of certain taxonomies and prepositions (refer Section 3 for more details). This was also observed by Girju et al. (2005), who concluded that knowledge of preposition can aid labelling. Also, it filters out the verbs that go with the preposition for verb+preposition

(VERB+PREP) paraphrasing. For example, if the preposition used in paraphrase is *at*, the verbs that go along with mainly be of type LOCATION. Thus, it can aid verb+preposition paraphrasing. Moreover, uncovering of a preposition is enough for some NLP application like Hindi-to-English translation (Kulkarni et al., 2012). Thus, prepositional paraphrasing is a useful first step towards noun compound interpretation.

Deep learning has made tremendous progress in various fields. One of the significant contributions of deep learning in NLP is word embeddings. They are dense real-valued representations of words learnt in an unsupervised manner. Their use has advanced the state in many applications (word sense disambiguation - Rothe and Schütze (2015), named entity recognition - Lample et al. (2016), sentiment classification - Tang et al. (2014), sarcasm detection - Joshi et al. (2016), *etc.*). Word embeddings enable words to share statistical strength. For instance, pattern learnt for the word *hotel* could be used to a good extent for the word *motel*, since they are semantically similar (which is elicited by the similarity between their corresponding word embeddings). This motivates us to investigate the use of word embeddings for noun compound interpretation.

Another significant contribution of deep learning to the field of NLP is the introduction of Encoder Decoder architectures (Bahdanau et al., 2015) for different tasks involving sequences. In such models, an input sequence of words is converted to a sequence of corresponding embeddings, which is then encoded into a single embedding via a part of the network known as Encoder. The single embedding is known as sequence embedding, sequence representation, *etc.* The second part of the network, the Decoder, then uses this sequence embedding in tandem with other information to produce the desired output. Such architectures are commonly used in machine translation, where the sequence embeddings are learnt such that the embeddings of a sentence in a source language and its parallelly aligned sentence in target language have high similarity.

As far as prepositional paraphrasing is concerned, one can observe that both the noun compounds as well as their corresponding paraphrases are parallelly aligned sequences of words. Thus, we can also make use of various sequence learning models from deep learning such as RNN (Werbos, 1990; Rumelhart et al., 1986), LSTM (Hochreiter and Schmidhuber, 1997), *etc.* as encoders to learn their representations.

Thus, in this paper, we raise the following question:

Can sequence representations, learned from word embeddings of components of a noun compound, help in its prepositional paraphrasing?

We attempt to answer this question in the following manner: we encode a noun compound and its prepositional paraphrase through two different LSTMs. Then, we train a network such that the encodings of a noun compound and its corresponding prepositional paraphrase have high similarity. Using this core intuition, our approach is able to generate the correct prepositional paraphrase. We evaluated our approach on four different datasets. Our approach performs reasonably well on a relatively small dataset, and outperforms others on the larger datasets.

The rest of the paper is organized as follows: Section 2 furnishes the necessary background. Section 3 motivates the need for prepositional paraphrasing. Section 4 details the sequence learning based architecture for learning representations of noun compounds and prepositional paraphrases, which we use to generate the correct prepositional paraphrase. Section 5 provides the experimental setup: the datasets used and the training/testing procedure used. Section 6 discusses the results of our experiments and some analysis, followed by a conclusion and future work.

2   Background and Related Work

A relation between the components of a noun compound can be represented in either of the following two ways: (1) Labelling: assigning a relation from a predefined set of semantic relations (*e.g.*, *apple juice*: MADE_OF), or (2) Paraphrasing: using a paraphrase to convey the underlying semantic relation (*e.g.*, *apple juice*: *"juice extracted from an apple"* or *"juice with apple flavor"*).

Over the years, many sets of semantic relations have been proposed (Levi, 1978; Warren, 1978; Lauer, 1995; Ó Séaghdha, 2007; Rosario et al., 2002; Barker and Szpakowicz, 1998; Vanderwende, 1994;

Tratz and Hovy, 2010; Ponkiya et al., 2018). Most of these sets have been created with the assumption that most predicates are of the form verb+preposition. For example, The CAUSED-BY relation in Levi (1978), or the USER/RECIPIENT+THINGUSED/RECEIVED relation in Tratz and Hovy (2010) which paraphrases to *thing used by user*. One may observe that these labels are motivated from verb+preposition constructions. We believe that preposition uncovering will help in the verb+preposition paraphrasing of noun compounds.

A system for automatic uncovering of the predicate can be developed in two ways: rule-based or statistical. A rule-based system will involve linguistic analysis of both w_1 and w_2. In this case, we may end up with more exceptions than the rules. Thus statistical approaches are the way to go. Among statistical approaches, supervised approaches rely on annotated data that needs to be sufficiently large and representative enough of the underlying problem. However, such datasets are rare, and the ones that do exist are small and heavily skewed, which makes the learning more difficult. For example, ≈80% of noun compounds in Girju (2007)'s dataset are annotated with the preposition *of*. The problems arising in such a supervised setting have been well studied by Girju et al. (2005).

On the other hand, Lauer (1995) and Lapata and Keller (2004) have proposed unsupervised models for identifying preposition for an interpretation of noun compounds. Lauer (1995) models $P(prep|n_1, n_2)$ (probability of preposition given two components of a noun compound) using frequency of triples $\langle n_1, prep, n_2 \rangle$. In an alternative model to handle sparsity, she used the following:

$$prep^* = \arg \max_{prep} \sum_{t1 \in cats(n_1), t_2 \in cats(n_2)} P(t_1|prep)P(t_2|prep)$$

where t_1 and t_2 represent particular concepts from the sets of concepts $cats(n_1)$ and $cats(n_2)$, respectively. These sets come from Roget's thesaurus. She also used a lexical frequency based approach in the above formula using pattern searching from Grolier's encyclopedia. Lapata and Keller (2004) implemented the same lexical based model, but they used Altavista search engine and BNC corpus for computing the probability factors.

Recently, Dima and Hinrichs (2015) proposed a feed-forward neural network based system for noun compound interpretation via labelling. Their network takes concatenated word-vectors of components of the noun compound as input, and predicts one of the labels from the Tratz and Hovy (2010)'s label set. We also adapt their system to our problem setting for a comparison.

3 Why Prepositional Paraphrasing of Noun Compounds?

Uncovering a hidden relation or ellipsis from a construct is an important problem in NLP. For instance, when a customer searches for *15 inch laptop*, he/she is actually searching for *a laptop with 15-inch display*. But, when a customer searches for *30 inch tyres*, he/she is actually searching for *tyres with 60 inch diameter*.

Noun Compounds are one of the many such constructs where the relation is hidden. For example, *"WHO Geneva headquarter"* is *"(the) headquarter of WHO located in Geneva."* One way of explicitly revealing the hidden relation is to paraphrase the noun compound using a verb+preposition construction. For example, *city hospital*: *"hospital located in a/the city"*, *mango juice*: *"juice extracted from mango"*, *etc*.

Prepositions are ambiguous, and they have their own selectional preferences (Zapirain et al., 2013). Once, a preposition and its corresponding sense are known, the space of verb+preposition constructions that can be used for paraphrasing is severely reduced (illustrated in Table 1), thereby making the task of verb+preposition paraphrasing easy. Thus we believe prepositional paraphrasing to be an important step towards verb+preposition paraphrasing of noun compounds.

Noun compound interpretation via labelling is another avenue where prepositional paraphrasing is useful. We annotated Kim and Baldwin (2005)'s dataset with the correct paraphrasing prepositions. A comparison of preposition vs. the annotated semantic relation (Table 2) shows that knowing preposition for a noun compound helps the system in identifying the finer semantic relations. For instance, 29 samples in the annotated data have *from* preposition. Following is the distribution of those samples:

Correct Preposition	Intended Meaning	Possible Verb+Prep Constructions	Noun Compounds	Example Verb+Prep Paraphrases
from	Source	resulted from, produced from, grown in	forest product farm product bonds yield	product produced from forest product grown in farm yield resulted from the bonds
from	Purpose	provided during	pain relief	relief provided during pain

Table 1: Demonstrating how knowledge of prepositions with intended meaning can restrict the space of verb+preposition constructions

	about	at	for	from	in	of	on	with
AGENT	0	0	0	0	0	3	0	0
BENEFICIARY	0	0	4	1	2	3	0	0
CAUSE	0	0	6	3	15	29	3	0
CONTAINER	0	0	1	0	3	12	0	0
CONTENT	0	0	3	2	3	29	1	1
DESTINATION	0	0	0	0	0	2	0	0
EQUATIVE	0	1	4	0	0	1	0	0
INSTRUMENT	0	0	5	0	0	2	0	1
LOCATED	0	0	3	0	0	8	0	0
LOCATION	0	1	2	0	9	17	0	0
MATERIAL	0	0	0	1	0	11	0	0
OBJECT	1	0	7	0	16	55	1	0
POSSESSOR	0	0	8	1	0	21	0	0
PRODUCT	1	0	3	0	0	28	0	0
PROPERTY	3	1	11	0	5	15	1	0
PURPOSE	1	1	89	3	6	41	2	0
RESULT	0	0	3	0	0	2	0	0
SOURCE	1	0	10	13	23	38	3	0
TIME	0	2	4	0	15	3	0	0
TOPIC	9	5	74	5	45	227	8	3

Table 2: Comparison of semantic relations (SR) and prepositions in Kim and Baldwin (2005)'s dataset manually annotated with prepositions. Row labels are SRs and column labels are prepositions. Each entry indicates the number of examples labelled with corresponding semantic relation and preposition.

SOURCE: 13, TOPIC: 5, CAUSE: 3, PURPOSE: 3, CONTENT: 2, BENEFICIARY: 1, MATERIAL: 1, POSSESSOR: 1, and 0 for remaining 12 relations. The annotation frequencies indicate that, for a noun compound with *from* as preposition, SOURCE is a highly likely *e.g.*, *marketing profit*, *forest product*, *tax revenue*, *etc*, whereas the 12 relations with 0 counts are extremely unlikely. Girju et al. (2005) used true prepositions as an additional feature in their classifier for fine semantic relation prediction and reported significant improvement in their performance. Thus, knowledge of preposition will definitely help in predicting and labelling finer relations.

In some NLP tasks, prepositional paraphrasing of a noun compound yields sufficient information for solving the task. Knowledge of Hindi case markers is sufficient for proper Hindi to English translation of Hindi noun compounds (a subset of *samāsa*)(Kulkarni et al., 2012).

4 Approach

As discussed earlier, we learn representations of noun compounds and their prepositional paraphrases, such that the cosine similarity between the representation of a noun compound and the representation of

the corresponding prepositional paraphrase is high. To do this, we use LSTMs (Hochreiter and Schmidhuber, 1997) based encoders. The architecture of our network is shown in Figure 1.
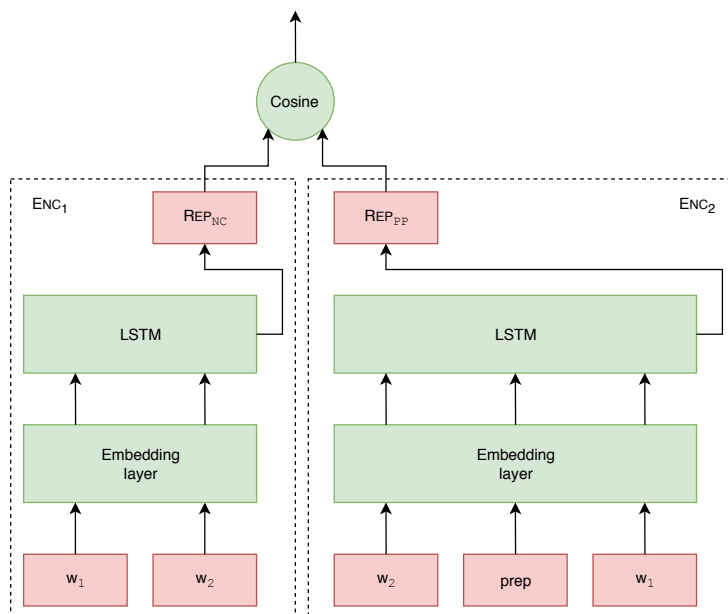

Figure 1: Architecture for learning representations of noun compounds (REP_{NC}) and prepositional paraphrases (REP_{PP}).

The network consists of two encoders: ENC_1 and ENC_2. ENC_1 embeds the constituents of an input noun compound and encodes those embeddings using an LSTM to get a representation REP_{NC} for the noun compound. ENC_2 embeds the words in an input prepositional paraphrase and encodes those embeddings using LSTM to get a representation REP_{PP} for the prepositional paraphrase. The architecture then computes the cosine similarity of these representations, which generates a value in the range $[-1, 1]$. The higher the similarity, the greater is the match between the noun compound and the prepositional paraphrase.

We initialize the embedding layer with Google's pre-trained embeddings[1]. For the 8-prepositions, we use 1-hot representations. We add padding of 0's to make them of same dimensions as that of word embeddings. The embedding layer is not updated during the training.

To calculate the correct prepositional paraphrase for a test noun compound NNC, we use the following procedure:

- Generate representation REP_{NNC} using the encoder ENC_1

- Generate a list of candidate prepositional phrases using the predefined set of prepositions

- For each candidate prepositional paraphrase CPP, generate the corresponding representation REP_{CPP} using the encoder ENC_2.

- Return that prepositional paraphrase PP* such that cosine similarity between REP_{NNC} and $\text{REP}_{\text{PP*}}$ is maximum. Mathematically,

$$\text{PP}^* = \arg\max_{\text{CPP}} \; cosine(\text{REP}_{\text{NNC}}, \text{REP}_{\text{CPP}})$$

---

[1]Pre-trained embeddings available at `https://code.google.com/archive/p/word2vec/`
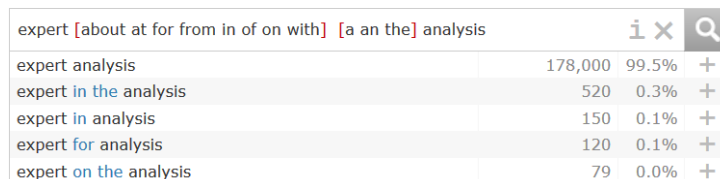
5   Experimental Setup

5.1   Datasets

For prepositional paraphrasing of noun compounds, Lauer (1995) and Girju et al. (2005) have proposed datasets. These datasets contain noun compounds annotated with correct preposition.

Lauer (1995)'s dataset is not publicly available. Thus, we extracted test samples from her thesis (Lauer, 1995). We dropped 118 samples out of 400 samples, as they had been marked as non-prepositional, and use the remaining 282 samples for our test experiments. Do note that this dataset is too small for effective learning. We have included it only for the sake of completeness.

Similarly, Girju et al. (2005)'s dataset is not available online[2]. Thus, we use Girju (2007)'s dataset (which has cross-lingual information along with English noun compounds) and extract relevant information to create a dataset. We extracted 832 samples. As our experiments are for Lauer (1995)'s set of 8 prepositions, we dropped examples annotated with prepositions that do not belong to this set. We use the remaining 805 examples for our experiments.

In addition to the above two datasets, we prepared a dataset by manually annotating noun compounds with Lauer (1995)'s prepositions. For annotation, we use 1,779 noun compounds (without labels) from Kim and Baldwin (2005)'s dataset. Each example was annotated by two annotators. The inter-annotator agreement is 51.48% (Cohen's kappa κ =0.36). For experiments, we have used only 1042 examples[3] where both annotators agreed upon. From here on, we refer to this dataset as Inhouse dataset.

In order to make learning more effective, we needed a lot more training samples. We adapted Lapata and Keller (2004)'s approach which use the web to automatically annotate noun compounds. We, however, use Netspeak[4]. We collected a list of noun compounds from existing noun compound interpretation datasets (Kim and Baldwin, 2005; Ó Séaghdha, 2007; Tratz and Hovy, 2010) and query each of them with a set of prepositions (as shown in Figure 2). The preposition with the highest frequency is chosen to form the *correct* prepositional paraphrase. For example, as shown in Figure 2, '*expert <u>in</u> (the) analysis*' has the highest frequency among all candidate paraphrases of *analysis expert*. So, we choose *in* as a correct preposition for the noun compound *analysis expert*. This dataset is used for training all systems.

expert [about at for from in of on with] [a an the] analysis		i ✕	🔍
expert analysis	178,000	99.5%	+
expert in the analysis	520	0.3%	+
expert in analysis	150	0.1%	+
expert for analysis	120	0.1%	+
expert on the analysis	79	0.0%	+

Figure 2: Example query result using Netspeak

5.2   Training

To train our system, we have noun compounds with their preposition paraphrases. Our objective is to learn representations such that similarity of a noun compound representation is higher with correct preposition compared with its similarity with any other prepositional paraphrase. Thus, for each noun compound, we treat the remaining prepositions as negative examples. For example, correct preposition for *analysis expert* is *in*. So, we treat (*analysis expert, 'expert in analysis'*) as a positive example and other pairs – like (*analysis expert, 'expert from analysis'*), (*analysis expert, 'expert about analysis'*), (*analysis expert, 'expert at analysis'*), *etc.* – as negative examples. We create such examples from the training dataset mentioned above.

We perform two different set of experiments. In one kind of experiments, we train our system (explained in Section 4) on the automatically annotated dataset and evaluate performance on the various datasets. In another type, for each dataset, we additionally fine-tune the trained-model using a portion of

---

[2]We requested the authors for the Girju et al. (2005)'s dataset, but they did not respond positively.
[3]The dataset is available at `http://www.cfilt.iitb.ac.in/nc-dataset`
[4]A search engine on Web 1T 5-gram available at `http://www.netspeak.org`.

the dataset. We use 75% of the dataset for tuning, and rest 25% of examples for testing. In both cases, we use the same test-set for a fair comparison. We discuss the performance of the models in the next section.

5.3 Evaluation metrics

We report Precision, Recall and F-score for our experiments. These values are weighted values in proportion to a number of test-examples for each preposition. For instance, following is a formula for computing (weighted) precision:

$$\text{Precision} = \sum_{prep} P_{prep} * \frac{N_{prep}}{N}; \quad P_{prep} = \frac{TP_{prep}}{TP_{prep} + FP_{prep}}$$

where P_{prep} is the precision score, TP_{prep} is the number of true-positives and FP_p is the number of false-positives for a preposition $prep$. N_{prep} is the number of instances with preposition $prep$ in the test set, and N is the total number of instances in a test-set.

To compare the results with previous work, we report micro-averaged accuracy[5]. The following formula computes micro-accuracy:

$$\text{Accuracy} = \frac{\text{Number of correcly classified instances}}{\text{Total number of instances in test-test}}$$

6 Results and Analysis

We compare the performance of our approach (referred to as NC-LSTM hereafter) with the performance reported by Lauer (1995), Lapata and Keller (2004), and Girju (2007) and performance computed for Dima and Hinrichs (2015)'s approach (referred to as NC-FFN hereafter) on different datasets.

We first compare NC-LSTM with NC-FFN. Table 3 shows that NC-LSTM performs comparably, if not better, than NC-FNN. Also, NC-LSTM easily outperforms NC-FFN by a significant margin when the network parameters are further tuned with a portion of the dataset. Thus, this also shows the importance of fine-tuning. This is easily explained by the fact that the original training data was extracted from the web, and is noisy in nature.

Dataset	Approach	Without Tuning			With Tuning		
		P	R	F	P	R	F
Lauer (1995)	NC-FFN	40.85	38.03	31.15	43.97	40.85	40.09
	NC-LSTM	50.84	45.07	40.66	48.72	46.48	**46.21**
Girju (2007)	NC-FFN	74.72	80.69	77.52	74.20	86.14	79.72
	NC-LSTM	76.86	74.26	75.50	84.74	88.61	**85.13**
Inhouse Dataset	NC-FFN	63.00	66.96	63.97	64.91	67.39	64.40
	NC-LSTM	62.32	65.65	63.09	73.50	72.17	**71.27**

Table 3: Comparison of performance of our LSTM based architecture (NC-LSTM) with Dima and Hinrichs (2015)'s feed-forward neural network based architecture (NC-FNN) on different datasets (P: Precision; R: Recall, F: F-score)

To demonstrate the effect of noisy data, we provide an analysis of the noun compound *apple tree*. The correct paraphrase of this noun compound is *'tree of apple'*. However, the system considers the paraphrase *'tree with apple'* to be correct. The following example sentences, which contributed to the count of *'tree with apple'*, demonstrate some errors made by the system:

---

[5]Mathematically, micro-averaged accuracy and weighted recall are same. We reported accuracy so that it can be compared with prior work. We reported F-score as it is the standard metric for classification evaluation. We reported recall as it used for computing f-score.

1. *if you combine a pine **tree with an apple** tree you do indeed get a pineapple tree.*
2. *What do you get if you cross a Christmas **tree with an apple?***

Clearly, the implicit assumption that *apple* is attached to the first occurrence of *tree* made by automatic web-based annotation algorithm which created the training dataset in sentence 1 is incorrect. Similarly, the implicit assumption that *apple* is attached to *tree* in sentence 2 is incorrect. Such errors add noise to the training dataset. However, using the tuning set fixes such errors, thereby improving performance considerably.

To check the extent to which such errors can impact effective training of the system, we estimate the accuracy of the automatic annotation process. We do this by testing the accuracy of automatic annotation on the common examples between automatically annotated and each of the three manually annotated datasets. The statistics of the common examples are shown in Table 4.

Dataset	#of Common Examples	#of Common Examples with Matching Labels
Lauer (1995)	31	12
Girju et al. (2005)	9	8
Inhouse Dataset	434	317

Table 4: Statistics of common examples between automatically created dataset and each of the three gold-standard dataset.

Table 5 compares reported results of various approaches with results from our approach. It shows that NC-LSTM outperforms Girju (2007)'s approach on their own dataset. We are unable to perform well on Lauer (1995) dataset. However, as mentioned earlier, given the relatively small size of the data, we refrain from commenting about our system's performance on the basis of this result.

Approach	L95	G05	Inhouse
Lauer (1995)	40.00		
Lapata and Keller (2004)	**55.71**		
Girju et al. (2005)	46.09	56.22	
NC-FFN	40.85	86.14	67.39
NC-LSTM	46.48	**88.61**	**72.17**

Table 5: Accuracy of various system reported in the literature. (L95: Lauer (1995); G05: Girju et al. (2005); Inhouse: Our manually created Inhouse dataset)

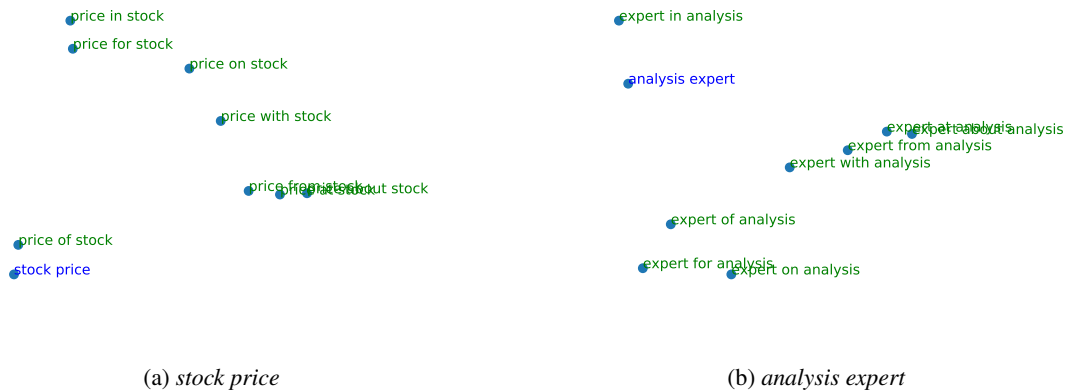(a) *stock price*　　　　　　(b) *analysis expert*

Figure 3: PCA visualizations of noun compounds and their prepositional paraphrases for some examples

Figure 3 shows visualizations of noun compounds and their candidate prepositional paraphrases. One can observe from Figure 3a that among all candidate paraphrases, the embedding of *'price of stock'* is the closest to the embedding of *stock price*. Similarly, Figure 3b shows that among all candidate paraphrases, the embedding of *'expert in analysis'* is the closest to the embedding of *analysis expert*.

7   Conclusion and Future Work

In this paper, we proposed a novel way to perform prepositional paraphrasing of noun compounds. We learn representations for noun compounds and their corresponding prepositional paraphrases using LSTMs, such that cosine similarity of a noun compound and a prepositional paraphrase correlates with their semantic similarity. Our proposed method, when trained on noun compound data automatically annotated using the web, performs reasonably well on existing datasets. In the future, we would investigate adapting this approach for verb+preposition paraphrasing, *i.e.* given a noun compound *orange juice*, generate the verb+preposition paraphrase *'juice made from orange'*.

References

Kisuh Ahn, Johan Bos, David Kor, Malvina Nissim, Bonnie L Webber, and James R Curran. 2005. Question answering with QED at TREC 2005. In *TREC*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations (ICLR 2015)*.

Timothy Baldwin and Takaaki Tanaka. 2004. Translation by machine of complex nominals: Getting it right. In *Proceedings of the Workshop on Multiword Expressions: Integrating Processing*, pages 24–31.

Renu Balyan and Niladri Chatterjee. 2015. Translating noun compounds using semantic relations. *Computer Speech & Language*, 32(1):91–108.

Ken Barker and Stan Szpakowicz. 1998. Semi-automatic recognition of noun modifier relationships. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 96–102.

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*, pages 100–105.

Corina Dima and Erhard Hinrichs. 2015. Automatic noun compound interpretation using deep neural networks and word embeddings. *IWCS 2015*, pages 173–183.

Roxana Girju, Adriana Badulescu, and Dan Moldovan. 2003. Learning semantic constraints for the automatic discovery of part-whole relations. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 1–8.

Roxana Girju, Dan Moldovan, Marta Tatu, and Daniel Antohe. 2005. On the semantics of noun compounds. *Computer speech & language*, 19(4):479–496.

Roxana Girju. 2007. Improving the interpretation of noun phrases with cross-linguistic information. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, volume 45, page 568.

Iris Hendrickx, Preslav Nakov, Stan Szpakowicz, Zornitsa Kozareva, Diarmuid O Séaghdha, and Tony Veale. 2013. Semeval-2013 task 4: Free paraphrases of noun compounds. *Atlanta, Georgia, USA*, page 138.

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

Aditya Joshi, Vaibhav Tripathi, Kevin Patel, Pushpak Bhattacharyya, and Mark Carman. 2016. Are word embedding-based features useful for sarcasm detection? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1006–1011. Association for Computational Linguistics.

Su Nam Kim and Timothy Baldwin. 2005. Automatic interpretation of noun compounds using wordnet similarity. In *Natural Language Processing–IJCNLP 2005*, pages 945–956. Springer.

Amba Kulkarni, Soma Paul, Malhar Kulkarni, Anil Kumar, and Nitesh Surtani. 2012. Semantic processing of compounds in indian languages. In *COLING*, pages 1489–1502.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California, June. Association for Computational Linguistics.

Mirella Lapata and Frank Keller. 2004. The web as a baseline: Evaluating the performance of unsupervised web-based models for a range of nlp tasks. In *the Proceedings of the Human Language Technology Conference/North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*.

Mark Lauer. 1995. *Designing statistical language learners: Experiments on compound nouns*. Ph.D. thesis, Ph. D. thesis, Macquarie University.

Judith N Levi. 1978. *The syntax and semantics of complex nominals*. Academic Press New York.

Preslav Nakov. 2013. On the interpretation of noun compounds: Syntax, semantics, and entailment. *Natural Language Engineering*, 19(03):291–330.

Diarmuid Ó Séaghdha. 2007. Annotating and learning compound noun semantics. In *Proceedings of the 45th Annual Meeting of the ACL: Student Research Workshop*, pages 73–78.

Girishkumar Ponkiya, Kevin Patel, Pushpak Bhattacharyya, and Girish K Palshikar. 2018. Towards a standardized dataset for noun compound interpretation. In *Language Resources and Evaluation Conference*, Miyazaki, Japan.

Barbara Rosario, Marti A Hearst, and Charles Fillmore. 2002. The descent of hierarchy, and selection in relational semantics. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 247–254.

Sascha Rothe and Hinrich Schütze. 2015. Autoextend: Extending word embeddings to embeddings for synsets and lexemes. In *Proceedings of the ACL*.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning representations by back-propagating errors. *nature*, 323(6088):533.

Duyu Tang, Furu Wei, Bing Qin, Ting Liu, and Ming Zhou. 2014. Coooolll: A deep learning system for twitter sentiment classification. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 208–212.

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*, pages 678–687.

Lucy Vanderwende. 1994. Algorithm for automatic interpretation of noun sequences. In *Proceedings of the 15th conference on Computational linguistics-Volume 2*, pages 782–788.

Beatrice Warren. 1978. Semantic patterns of noun-noun compounds. *Acta Universitatis Gothoburgensis. Gothenburg Studies in English Goteborg*, 41:1–266.

Paul J Werbos. 1990. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560.

Beñat Zapirain, Eneko Agirre, Lluís Màrquez, and Mihai Surdeanu. 2013. Selectional preferences for semantic role classification. *Computational Linguistics*, 39(3).