Looking inside Noun Compounds: Unsupervised Prepositional and Free Paraphrasing

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

Abstract

A noun compound is a sequence of contiguous nouns that acts as a single noun, although the predicate denoting the semantic relation between its components is dropped. Noun Compound Interpretation is the task of uncovering the relation, in the form of a preposition or a free paraphrase. Prepositional paraphrasing refers to the use of preposition to explain the semantic relation, whereas free paraphrasing refers to invoking an appropriate predicate denoting the semantic relation. In this paper, we propose an unsupervised methodology for these two types of paraphrasing. We use pre-trained contextualized language models to uncover the 'missing' words (preposition or predicate). These language models are usually trained to uncover the missing word/words in a given input sentence. Our approach uses templates to prepare the input sequence for the language model. The template uses a special token to indicate the missing predicate. As the model has already been pre-trained to uncover a missing word (or a sequence of words), we exploit it to predict missing words for the input sequence.

Our experiments using four datasets show that our unsupervised approach (a) performs comparably to supervised approaches for prepositional paraphrasing, and (b) outperforms supervised approaches for free paraphrasing. Paraphrasing (prepositional or free) using our unsupervised approach is potentially helpful for NLP tasks like machine translation and information extraction.

1 Introduction

Noun compounds- contiguous sequences of nouns- are common linguistic constructs. A compound is called compositional if the meaning of the compounds can be derived from the meaning of its components. The component nouns are related through a semantic relation that is constituents dependent. For instance, *'student protest'* and *'university protest'* are *protests*. However, the *student(s)* are AGENT (doer of an event), whereas *university* is LOCATION of the protest.

The task of identifying such relations between the components of a noun compound is called *noun compound interpretation (NCI)*. Such interpretation can help a wide variety of NLP tasks, like machine translation (Baldwin and Tanaka, 2004; Paul et al., 2010; Balyan and Chatterjee, 2015), question answering (Ahn et al., 2005), text entailment (Nakov, 2013), and semantic parsing (Tratz, 2011). For instance, to translate the English noun compound *'cow milk'* to Hindi, a machine translation system needs to generate the postposition *kA* (*of*) in addition to translating the individual nouns. The correct translation of the compound is *'gāya ka dūdha'* (lit. *'cow -of milk'*; *'milk of cow'*). Without understanding the underlying relation, a machine translation system might fail.

Interpretation via abstract labels (representing semantic relations) is popular in the literature. Given a noun compound, the task is to assign an abstract label from a predefined set, *e.g.*, *'student protest'*: PROTESTER. Past work has proposed a wide variety of inventories for semantic relations (Levi, 1978; Warren, 1978; Lauer, 1995; Nastase and Szpakowicz, 2003; Ó Séaghdha, 2007; Rosario et al., 2001; Barker and Szpakowicz, 1998; Vanderwende, 1994; Tratz and Hovy, 2010; Fares, 2016; Ponkiya et al., 2018a); however, there is no community agreed standard inventory.

Interpretation can be done via paraphrasing as well. Here, one can use extract words (along with component nouns) to paraphrase a noun compound, *e.g.*, *'student protest'*: *'protest by student'*, *'protest held by students'*, *etc*. The paraphrase reveals the underlying relation. A simpler version of paraphrasing, also known as *prepositional paraphras-*

ing, uses only a preposition to paraphrase a noun compound. A set of 8 prepositions by Lauer (1995) is widely used for prepositional paraphrasing, and the task is to identify a preposition which can paraphrase the given noun compound.

Another way of paraphrasing, also known as *free paraphrasing*, allows any word(s) for paraphrasing. One can use multiple paraphrases to represent the semantic relation collectively. This is a more complex and challenging task.

In this paper, we show how contextualized language models can be used for unsupervised paraphrasing of noun compounds. Specifically, we propose two unsupervised approaches for paraphrasing of noun compounds: one for prepositional paraphrasing and another for free paraphrasing. We use contextualized language models and feed template to generate possible paraphrases. Our results show that the proposed unsupervised approach gives results comparable to supervised systems for prepositional paraphrasing and outperforms supervised approaches for free paraphrasing.

2 Related Work

2.1 Prepositional Paraphrasing

Lauer (1995) used 8 prepositions for paraphrasing: *about, at, for, from, in, of, on* and *with*. They argue that the 8 prepositions are sufficient to paraphrase any compound except two categories: copula and verb-external arguments. In some NLP tasks, prepositions are sufficient to convey the meaning. For instance, Paul et al. (2010) proposed a system that first uncovers a preposition from the English noun compound before translating it to Hindi.

The problem tackled was to classify a given noun compound into one of these prepositions such that the assigned preposition can paraphrase that compound. For example, a *baby chair* is a *chair for a baby*, and *reactor waste* is *waste from a reactor*.

Lauer's approach is attractive and simple. It yields prepositions representing paraphrases directly usable in NLP applications. However, it is also problematic, since mapping prepositions with constituent nouns as inputs to abstract relations is hard, *e.g.*, *in*, *on*, and *at*, all can refer to both LOCATION and TIME.

Lauer (1995) and Lapata and Keller (2004) gave unsupervised approaches to prepositional paraphrasing of noun compounds. Both approaches used frequencies of patterns in a large corpus of the Web. Girju (2007) trained various classifiers for the task and observed that SVM performs the best.

Recently, Ponkiya et al. (2018b) have proposed an LSTM-based system which encodes nouns compounds and their candidate prepositional paraphrases such that encoding of a noun compound is the most similar to the encoding of its correct prepositional paraphrase. The system was trained in two steps: (1) distant supervision: prepared a large dataset by annotating noun compounds automatically, and trained the system on the dataset; (2) the distant supervision system was further trained on manually annotated data. The authors evaluated both systems. We use these systems as our baseline to compare the performance of our approach.

The general idea of probing the semantic/commonsense knowledge residing in language models has been recently explored by Petroni et al. (2019) and Bouraoui et al. (2020). Both the approaches use different templates for different relations, whereas we use a single pattern for a classifier. Bouraoui et al. (2020) propose a supervised approach. They use masking-objective to find templates, and train a classifier for each relation. On the other hand, our approach is entirely unsupervised.

2.2 Free Paraphrasing

Nakov (2008) argue that noun compounds are best characterized by the set of all possible paraphrasing verbs that can connect the target nouns, with associated weights, *e.g.*, *malaria mosquito* can be represented as follows: *carry* (23), *spread* (16), *cause* (12), *transmit* (9), *etc*. The numbers in the parentheses indicate the number of human annotators who proposed the respective verb. These verbs are directly usable as paraphrases, and using multiple of them simultaneously yields an appealing fine-grained semantic representation.

The authors of the present paper collected multiple possible paraphrases for noun compounds using crowd-sourcing. They used human subjects (recruited through Amazon Mechanical Turk Web Service) to get paraphrasing verbs. For a noun compound $\langle noun_1\ noun_2 \rangle$, they asked the participants to propose at least three paraphrasing verbs (optionally followed by a preposition) as shown below:

"$noun_1$ $noun_2$" is a "$noun_2$ that ... $noun_1$"

An example (as shown in 1) was also provided for

the participants' reference.

(1) The compound *neck vein* can be paraphrased as follows:

'a vein that nourishes the neck'
'a vein that runs along the neck'
'a vein that comes from the neck'
'a vein that enters the neck'
'a vein that emerges from the neck'
etc.

Following Nakov (2008)'s footsteps, Task-9 of SemEval-2010 (Hendrickx et al., 2009) proposed the following simple problem:

> Given a noun compound and a list of paraphrasing verbs, (a participating system needs to) produce aptness scores that correlate well (in terms of relative ranking) with the held out human judgments.

For the task, the training dataset contains 250 noun-noun compounds, and at least 50 AMT workers provided paraphrases for each compound. The test dataset consisted of 388 noun compounds, and at least 57 workers provided paraphrases for each compound.

For official evaluation in the shared task, Spearman rank correlation (ρ) was used to evaluate relative ordering. Additionally, Pearson correlation (r) and cosine similarity were also used to check correlation strength between scores provided by a participating system and human scores.

SemEval-2013 Task-4 (Hendrickx et al., 2013)[1] proposed the following task (free paraphrases of noun compounds):

> **Task:** Given a noun-noun compound, such as *air filter*, (the participating systems are asked to) produce an explicitly ranked list of free paraphrases, as in the following example:
>
> 1 *'filter for air'*
> 2 *'filter of air'*
> 3 *'filter that cleans the air'*
> 4 *'filter which makes air healthier'*
> 5 *'a filter that removes impurities from the air'*
> ...

The task is different from the SemEval-2010 Task-9 in mainly three ways: (a) the restriction on the paraphrases was relaxed, (b) instead of ranking, a participating system needs to generate and rank the paraphrases, and (c) the task performed by a participating system is the same as that of human annotators. Compared with the dataset for the previous task, the dataset for this new task have a far greater range of variety and richness.

Human annotators were recruited through AMT (Amazon Mechanical Turk) to prepare a dataset for the task. The annotators were asked to provide free paraphrases for each noun compound. Identical paraphrases were merged to compute their frequencies, and sorted by their frequencies. The training set contains 174 noun-noun compounds with 4,255 unique paraphrases (24.5 paraphrases on average). The test set includes 181 noun-noun compounds with 8,216 unique paraphrases (45.4 paraphrases on average).

For evaluation, the predicted paraphrases for a test example were ranked, and then the overall scores were computed by matching predicated paraphrase with the reference paraphrases. The matching was done in two ways: based on whether multiple generated paraphrases can be matched with a reference paraphrase or not. A simple baseline for the task used a fixed set of prepositional paraphrases in a fixed order. None of the four proposed systems (submitted by three teams) beat the baseline in both evaluation techniques.

All three participating systems (Van de Cruys et al., 2013; Surtani et al., 2013; Versley, 2013) were supervised. Van de Cruys et al. (2013) used a distributional model to extract word features, which were then used to train a maximum-entropy classifier. The classifier predicted a probability distribution over a set of paraphrases. A threshold was used to decide whether the paraphrases should be included in the final output or not. A higher threshold value resulted in fewer paraphrases, where a lower threshold value generated more paraphrases. It was observed that using only features of the head noun (the second word in a compound) performs better than when using feature vectors of both component nouns.

Surtani et al. (2013) used a corpus-based co-occurrence probability in predicting paraphrases. The prepositional paraphrases are quite frequent and well covered. To handle sparsity, they used prepositional paraphrase to predict a semantic relation, and then, selected verbs that mostly co-occur with that relation.

Versley (2013) retrieved mutually more similar compounds from training data, extracted templates

[1] https://www.cs.york.ac.uk/semeval-2013/task4

and fillers from paraphrases of the similar compounds. The templates were weighted by its frequency and similarity its deriving noun compound with test noun compound. The final generated paraphrases were ranked using a language model and MaxEnt model.

Recently, Shwartz and Dagan (2018) proposed a semi-supervised method by formulating paraphrasing as a multi-task learning objective. The authors first generated 250 most likely paraphrases using a neural model, and then re-ranked the paraphrases using an SVM.

3 Background

With the introduction of the Transformer networks (Vaswani et al., 2017), pre-trained language models have become a key component in advancing the state-of-the-art for many NLP tasks. BERT (Devlin et al., 2019), a transformer-based encoder, has advanced the state-of-the-art for various NLP tasks. For pre-training, BERT uses two self-supervised objectives: next sentence prediction (NSP), and masked language model (MLM). For NSP, BERT is trained to predict whether the second text segment follows the first text segment. This is hypothesized to improve BERT's understanding of the relationship between two text sentences. For MLM, given the input token sequence, a portion of tokens are replaced by a special symbol `[MASK]`, and the model is trained to recover the original tokens from the corrupted version. This allows representations to be conditioned on the left and right context. Note that BERT predicts plausible words for each `[MASK]` token independently. The success of BERT inspired many variants such as training on domain/application specific corpus (Lee et al., 2020; Beltagy et al., 2019; Huang et al., 2019; Alsentzer et al., 2019; Adhikari et al., 2019; Lee and Hsiang, 2019), training on monolingual corpora (Pires et al., 2019), incorporating knowledge graph in the input (Zhang et al., 2019), *etc*.

BERT requires a task-specific output layer. So, one needs to modify BERT's architecture to adapt it for a new task. Recent text-to-text models, such as T5 (Raffel et al., 2019) and BART (Lewis et al., 2019), use encoder-decoder architectures which share output layer for all tasks effectively eliminating the requirement to modify architecture for a new task. These models convert all NLP problems into a text-to-text format, *i.e.*, input and output for *any* NLP task (including classification task) are sequences. A text-to-text model can generate a variable length span for a single masked token because of encoder-decoder architecture. We use the T5 model to generate free paraphrases for noun compounds.

4 Our Approach

Our approach benefits from the MLM objective of contextualized language models. We use templates to rephrase a noun compound. The template uses a mask-token[2] to indicate the missing word(s). We feed the phrase to a pre-trained model and ask it to predict the missing word(s), which can replace the mask-token. We use BERT and RoBERTa to uncover a single word and T5 to uncover variable-length sequences. We use the Transformers library (Wolf et al., 2019, v2.8)[3] for the experiments.

4.1 Prepositional Paraphrasing

For prepositional paraphrasing, we need to predict the preposition inside the noun compound. We use BERT (and RoBERTa) to uncover the missing preposition. We use a template-based approach to prepare input for BERT. The template uses `[MASK]` token in place of the preposition. The following example illustrates the procedure:

1. **Input:** *apple juice*
2. **Template:** $w_1\ w_2 \Rightarrow w_2$ *[MASK]* w_1
   apple juice ⇒ "*juice [MASK] apple*"
3. BERT input: "*juice [MASK] apple*"
   BERT predicts the missing word along with the model confidence for that word from the vocabulary.

word	score
of	19.61 %
from	1.41 %
and	1.01 %
with	0.66 %
for	0.60 %
on	0.59 %
...	

4. We select a preposition with the highest score as the correct preposition.
   'apple juice' → *of*

BERT assigns a score for each vocabulary word. The score indicates the likelihood of the word to

[2]Different models represent the mask token differently, like `[MASK]`, `<MASK>`, `_MASK_1_`, *<extra_id_0>*, *etc*.

[3]https://huggingface.co/transformers

replace [MASK] token. We use scores of the 8 prepositions of our interest, and predict a preposition with the highest score as the correct preposition.

We use three patterns as templates. Table 1 shows the patterns with their realizations as BERT input. Pattern 1 is obtained from Ponkiya et al. (2018b), where the input to paraphrase encoder is similar and does not use articles. Pattern 2 provides context to Pattern 1. So, if BERT captures the semantics of a noun compound, it should help preposition uncovering. Pattern 3: Without the use of articles, we found that w_1 and/or w_2 was treated as verbs in some cases. For instance, for "*student protest is protest ___ student*", a model predicted *'##ing'* as a top choice. Adding articles in the pattern provides the clue that w_1 and w_2 should be considered as nouns.

We observed that *'a'/'an'* in input to BERT does not make much difference. This is because the MLM (masked language model) has been trained in such a way. During masking of tokens, after selecting 15% token randomly, MLM (a) replaces 80% of the chosen tokens with [MASK] token, (b) replaces 10% of chosen tokens in input sequence with a random token, and (c) and keeps 10% of chosen as it is.

4.2 Free Paraphrasing

For free paraphrasing of a noun compound, we need to generate multiple paraphrases and rank them. The paraphrases are of arbitrary lengths. Therefore, we need to generate an arbitrary number of words for each noun compound. We cannot use BERT based simple approach for free paraphrases. We use T5 model to generate such paraphrases.

We use a template to prepare input for T5. The template uses *<extra_id_0>* token (mask token in T5) to indicate a blank to be filled by T5. T5 predicts plausible k (a hyperparameter) sequences for the blank. The following example illustrates the procedure:

1. **Input:** *club house*
2. **Template:** w_1 w_2 $\Rightarrow$ "*A* w_1 w_2 *is a* w_2 *<extra_id_0> the* w_1*. </s>*"
   club house $\Rightarrow$ *"A club house is a house <extra_id_0> the club. </s>"*
3. T5 input: *"A club house is a house <extra_id_0> the club. </s>"*
   T5 generated the following sequences (for $k = 10$):
   "<extra_id_0> ***for*** *<extra_id_1>. A"*
   "<extra_id_0> ***of*** *<extra_id_1> . A"*
   "<extra_id_0> ***for*** *<extra_id_1>. <extra_id_2>"*
   "<extra_id_0> ***for*** *<extra_id_1> house ."*
   "<extra_id_0> ***owned by*** *<extra_id_1> ."*
   "<extra_id_0> ***of*** *<extra_id_1> . <extra_id_2>"*
   "<extra_id_0> ***owned by*** *<extra_id_1> house"*
   "<extra_id_0> ***that belongs to*** *<extra_id_1>"*
   "<extra_id_0> ***of*** *<extra_id_1> house."*
   "<extra_id_0> ***in*** *<extra_id_1> . A"*
4. For each generated sequence, extract words between *<extra_id_0>* and *<extra_id_1>*, and use them to generate a candidate paraphrase for the given noun compound.
   "house ***for*** *a club"*
   "house ***of*** *a club"*
   "house ***for*** *a club"*
   "house ***for*** *a club"*
   "house ***owned by*** *a club"*
   "house ***of*** *a club"*
   "house ***owned by*** *a club"*
   "house ***that belongs to*** *a club"*
   "house ***of*** *a club"*
   "house ***in*** *a club"*
5. Grouping similar paraphrases, and ranking them based on the frequencies, we get (rank:paraphrase):
   1 *"house of a club"*
   1 *"house for a club"*
   2 *"house owned by a club"*
   3 *"house that belongs to a club"*
   3 *"house in a club"*

As most paraphrases require up to 4 extra words, we set a maximum length for T5 output (step 3 in the above example) to 6. We assign the same rank to paraphrases with similar frequencies.

5 Experiments

In this section, we discuss the datasets, evaluation metrics used in our experiments.

5.1 Datasets

For Prepositional paraphrasing, Lauer (1995), Girju et al. (2005) and Ponkiya et al. (2018b) have reported preposition annotated noun compound datasets.[4] Noun compounds in these datasets have been annotated with Levi's eight prepositions.

Lauer (1995)'s dataset is very small (282 examples). Girju et al. (2005)'s dataset is not available

[4]The datasets: http://www.cfilt.iitb.ac.in/nc-dataset

	Pattern	BERT Input
1.	w_2 ____ w_1	[CLS] w_2 [MASK] w_1 [SEP]
2.	w_1 w_2 means w_2 ____ w_1	[CLS] w_1 w_2 means w_2 [MASK] w_1 [SEP]
3.	a w_1 w_2 is a w_2 ____ the w_1	[CLS] a w_1 w_2 is a w_2 [MASK] the w_1 [SEP]

Table 1: Patterns and their realizations for preposition uncovering. ($\langle w_1\ w_2 \rangle$ is a noun compound)

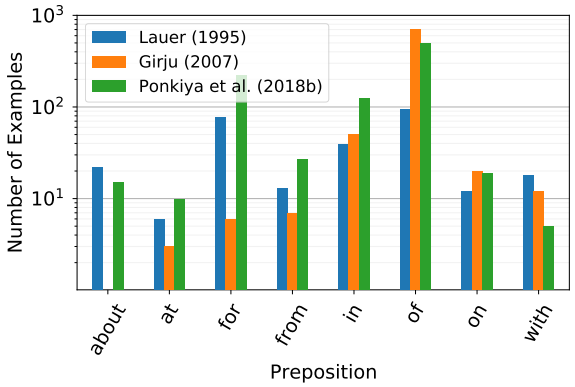

Figure 1: Noun compound distribution (per preposition) in three datasets for prepositional paraphrasing.

	Total	Min / Max / Avg
Trial/Train (174 NCs)		
Paraphrases	6,069	1 / 287 / 34.9
Unique Paraphrases	4,255	1 / 105 / 24.5
Test (181 NCs)		
Paraphrases	9,706	24 / 99 / 53.6
Unique Paraphrases	8,216	21 / 80 / 45.4

Table 2: Statistics of the trial and test sets from SemEval-2013 Task-4 dataset. (Total: number of paraphrases provided by human annotators with and without duplicates; Min / Max / Avg: the minimum / maximum / average number of paraphrases per noun compound.)

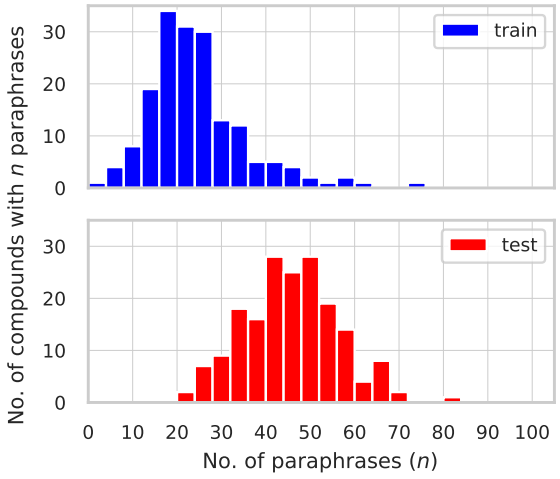

Figure 2: Distribution of paraphrase count in train and test part of SemEval-2013 Task-4 datasets.

in the public domain, but Ponkiya et al. (2018b) have created a dataset (805 examples) from the cross-lingual dataset of Girju (2007). The dataset is highly skewed – 85% examples covered by a single preposition *of*. Ponkiya et al. (2018b) also annotated noun compounds (919 examples) from Kim and Baldwin (2013)'s dataset with prepositions. Figure 1 shows distributions of prepositions for the above-mentioned three datasets. Please note that each noun compound in the above three datasets has been annotated with a single preposition.

For each dataset, Ponkiya et al. (2018b) used 25% of examples for testing. We used the same test splits to test our system. So, our results are directly comparable.

For free paraphrasing, we use SemEval-2013 Task-4 dataset.[5] The dataset contains train and test sets. The dataset provides a list of paraphrases for each noun compound. The paraphrases are ranked in order of preference. Table 2 shows the statistics of the dataset.

Figure 2 shows the histogram for the number of paraphrases per noun compound. The number of paraphrases for most noun compounds in the training set ranges from 15 to 35. The same for the test goes from 35 to 60. So, we expect higher precision for the test set (as a generate paraphrase would highly likely match with a reference paraphrase) and higher recall on the training set (as a system need not generate verity of paraphrases).

[5]Dataset available at https://www.cs.york.ac.uk/semeval-2013/task4/index.php%3Fid=data.html

5.2 Evaluation metrics

5.2.1 Prepositional Paraphrasing

The recent work by Ponkiya et al. (2018b) have reported weighted precision, recall and f-score for their experiments. So, we use the same metrics to evaluate our systems. These values are weighted values in proportion to the number of test-examples for each preposition.

Dataset → / Approach ↓	Lauer (1995) P	R	F	Girju (2007) P	R	F	Ponkiya et al. (2018b) P	R	F
Distance supervision *(Ponkiya et al., 2018b)*									
NC-FFN	40.85	38.03	31.15	74.72	80.69	77.52	63.00	66.96	63.97
NC-LSTM	50.84	45.07	40.66	76.86	74.26	75.50	62.32	65.65	63.09
Distance supervision + supervised fine-tuning *(Ponkiya et al., 2018b)*									
NC-FFN	43.97	40.85	40.09	74.20	86.14	79.72	64.91	67.39	64.40
NC-LSTM	48.72	46.48	46.21	84.74	88.61	**85.13**	73.50	72.17	71.27
BERT-base									
Pattern-1	71.80	53.52	48.95	87.38	80.69	83.80	69.30	70.00	68.60
Pattern-2	55.92	46.47	41.26	86.25	81.18	83.16	73.01	72.60	70.69
Pattern-3	41.47	43.66	36.14	88.73	74.25	79.86	66.59	66.95	66.15
BERT-large									
Pattern-1	64.85	52.11	46.62	86.39	77.22	81.47	67.86	66.52	65.19
Pattern-2	61.41	47.88	42.44	83.68	78.71	80.79	68.16	67.82	65.40
Pattern-3	51.32	45.07	35.91	86.01	75.74	80.02	67.74	65.65	65.59
RoBERTa-base									
Pattern-1	50.82	38.02	26.74	79.28	77.22	78.15	45.89	53.47	46.94
Pattern-2	55.57	52.11	47.95	83.11	57.92	66.99	65.39	63.04	63.47
Pattern-3	43.30	47.88	41.51	83.83	67.32	74.26	64.48	63.04	63.48
RoBERTa-large									
Pattern-1	50.78	33.80	25.79	79.33	69.30	73.96	53.72	56.08	51.94
Pattern-2	51.78	47.88	43.28	87.02	72.27	78.58	72.98	72.60	**72.21**
Pattern-3	56.06	56.33	**51.74**	88.30	72.77	79.12	68.36	67.39	67.32

Table 3: Prepositional Paraphrasing: Comparison of performance of our system (BERT and RoBERTa) with LSTM-based (NC-LSTM) and feed-forward neural network based (NC-FNN) systems on different datasets (P: Precision; R: Recall, F: F-score)

5.2.2 Free Paraphrasing

For a test noun compound, a system needs to generate a list of paraphrases in order of preference. The task uses two ways to match paraphrases: isomorphic matching and non-isomorphic matching.[6]

Isomorphic scoring maps each system generated paraphrase (in order of given preference) to an (unmapped) reference paraphrase one by one each. The system's paraphrases are matched 1-to-1 with reference paraphrases on a first-come first-matched basis, so ordering can be crucial. The final score is the sum of all system paraphrases, normalized with the maximum score for the reference list.

The isomorphic scoring mechanism requires a system to produce the full set of paraphrases. It rewards a system for accurately reproducing the paraphrases suggested by human judges, reproducing as many of these as possible, and in much the same order. So, it rewards both precision and recall. Isomorphic scoring was used as an official score by SemEval-2013 Task-4 for ranking of participating system.

Non-isomorphic scoring scores each system paraphrase with respect to the best match from the reference dataset, and averages these scores over-all system paraphrases. Non-isomorphic matching rewards only precision. More than one system generated paraphrases are allowed to match with a reference paraphrase. So, the ordering of a system's paraphrases is not important.

Non-isomorphic scoring rewards a system for accurately reproducing the top-ranked reference paraphrases. A system generating only one top-ranked reference paraphrase will achieve a perfect non-isomorphic score.

6 Results and Analysis

6.1 Prepositional Paraphrasing

We use BERT and RoBERTa to uncover the preposition. We compare the performance of our system with two systems used by Ponkiya et al. (2018b): (a) feed-forward neural-network (hereafter NC-FFN), and (b) LSTM-based sequence encoders (hereafter NC-LSTM).

Table 3 shows that NC-RoBERTa (our system with RoBERTa model) outperforms supervised NC-FFN and NC-LSTM on two datasets. For the third dataset, NC-BERT (our system with BERT model) outperforms non-tuned (only distant supervision) models and fine-tuned NC-FFN. However, our unsupervised approach slightly under-performs com-

[6] We use an evaluation script (*scorer*) provided by the task.

pared to the fine-tuned NC-LSTM (supervised).

For NC-BERT, the precision score is higher than the respective recall score. For NC-RoBERTa, precision and recall scores are mostly similar. We also tried combining scores from different models (base and large models) and different patterns. Overall results were similar. We have not observed improvement compared to the three patterns. So, we did not include them in this paper.

We expected Pattern-2 and Pattern-3 to perform better than Pattern-1 Pattern-2, respectively, as they provide more context (§4.1). The performance of NC-RoBERTa is as expected on all three datasets. However, we see a reverse trend for NC-BERT.

We analyzed the performance of the patterns on Ponkiya et al. (2018b)'s dataset using BERT-base and RoBERTa-large models. The dataset was prepared by annotating noun compounds from Kim and Baldwin (2005)'s dataset with prepositions. For every example, we have a semantic relation from Kim and Baldwin (2005) and a preposition from Ponkiya et al. (2018b).

We observe that the major reason behind pattern-3 underperforming compared to pattern-2 is: the correct preposition *of* predicted by pattern-2, but pattern-3 predicted *for*. Some examples are (using BASE-base model):

- PURPOSE relation: *approval process*, *takeover plan*, *merger agreement*, and *release term*.
- PRODUCT relation: *petroleum refinery*, and *gas industry*.
- SOURCE relation: *pulp price*, and *government plan*.

Out of 230 test samples, 22 are of such kind (pattern-2 correctly predicted *of*; pattern-3 predicted: *for*) for BASE-base. This degrades the precision of *for* (from 75.86 for pattern-2 to 57.14 for pattern-3) and recall of *of* (from 92.97 to 71.09). We have observed similar case with RoBERTa-large model.

This observation is in line with preposition-vs-relation mapping observed by Ponkiya et al. (2018b, see Table 2).

6.2 Free Paraphrasing

T5 comes in five versions: **small**, **base**, **large**, **3B**, and **11B** with 60 million, 220 million, 770million, 3 billion, and 11 billion parameters, respectively. We have experimented with **small**, **base** and **large** T5 models. However, the **small** model performed better. So, we report results for the *small* version of T5 model.

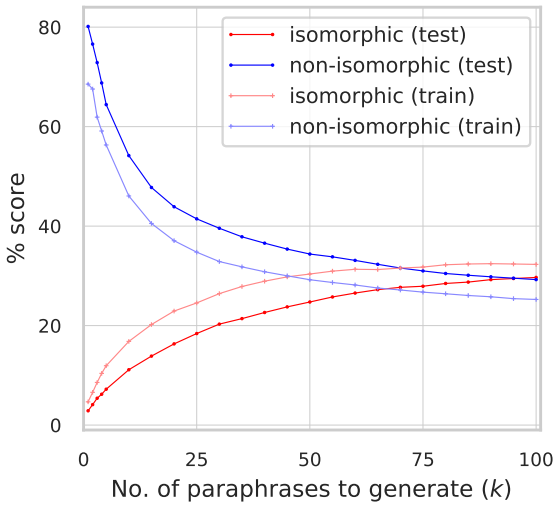

Figure 3: Performance of T5-based system for different value of k (number of paraphrases to generate) on train and test sets of SemEval-2013 Task-9 dataset.

To understand the impact of the number of generated paraphrases over scores, we evaluate our system by generating a varying number of paraphrases (k). When a system generates a smaller set of paraphrases, the generated paraphrases match with highly ranked in the reference, resulting in a higher non-isomorphic score. However, a smaller set might not cover all reference paraphrases. So, the isomorphic score takes hit. With the increase in the number of generated paraphrases, more paraphrases from the reference list were matched, hence isomorphic score increases. However, newly generated paraphrases were matched with high-ranking reference paraphrase, resulting in a decrease in non-isomorphic score.

The average number of paraphrases (per compound) is lower in the train set than in the test set. So, as explained earlier (§5.2.2, ref. Figure 2), the non-isomorphic score is higher for the test set, and the isomorphic score is higher for the training set.

We compare the performance of our T5-based system (hereafter NC-T5) with previously reported results (ref. Table 4). For a smaller value of k (number of sequences generated by T5), generated paraphrases mostly matched top-ranked reference paraphrases, resulting in a higher non-isomorphic score. With an increase in k, the system generated diverse paraphrases, helps isomorphic score. For $k = 80$ to 100, our system beats the recently reported results (by Shwartz and Dagan (2018)).

Method	isom.	n-isom.
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	13.8	40.6
Shwartz and Dagan (2018)	28.2	28.4
Our system (T5-base model)		
$k = 1$	2.87	**80.14**
$k = 2$	4.11	76.59
$k = 3$	5.39	72.87
$k = 4$	6.20	68.77
...		
$k = 80$	28.47	30.47
$k = 85$	28.74	30.12
$k = 90$	29.24	29.81
$k = 95$	29.46	29.53
$k = 100$	**29.68**	29.24

Table 4: Results of the proposed method and the baselines on the SemEval-2013 Task-4. (**isom**: isomorphic score, **n-isom**: non-isomorphic score)

NC-T5 generates quite a good quality set of paraphrases. However, the reference list does not have matching paraphrases. For example, Example 2 lists some of the system-generated paraphrases for *"pay policy"*. All examples, marked with dagger-sign (†), have a partial matching (score $\leq 25\%$), while the rest of the listed paraphrases do not have a match.

(2) *"policy on pay"* †
"policy defines pay"
"policy covering pay"
"policy governing pay"
"policy covers pay"
"policy deals with pay"
"policy describes pay"
"policy involving pay"
"policy designed to protect pay" †
"policy designed to cover pay" †
"policy designed for pay" †
"policy applicable to pay" †
"policy to protect pay" †
"policy used to cover pay" †
"policy used to pay pay" †
"policy used to protect pay" †
"policy focuses on pay" †

The dataset has many reference paraphrases where new words appear at the beginning (*e.g.*, *'pay policy'* → *"corporate policy on pay"*) or at the end of a paraphrase (*e.g.*, *'operating system'* → *"system controls operating of computer"*). However, our system allows extra words only between the component nouns.

7 Conclusion and Future Work

A noun compound can be paraphrased using the components nouns along with the predicate. The predicate indicates the semantic relation between the component nouns. We use a simple pattern for generating the predicate using a fixed pattern, *i.e.*, $w_1\ w_2 \rightarrow$ 'w_2 <*extra-words*> w_1'. One can exploit recent pre-trained language models to uncover the connecting extra-words for paraphrasing. These language models have been trained with one of the training objective being uncovering the missing words. In this paper, we propose an approach that performs noun compound paraphrasing by using these pre-trained models to uncover the missing extra words. Our approach uses these pre-trained models *as is* without any task-specific training or fine-tuning. Our approach is tested for both prepositional paraphrasing and free paraphrasing of noun compounds on various datasets. With simple patterns, our approach gives results closer to supervised systems for prepositional paraphrasing and outperforms supervised systems for free paraphrasing.

In the future, we will investigate whether fine-tuning the language models would lead to better paraphrasing. We will also study the setting where context is crucial for correct paraphrasing. We believe that given this approach is language-agnostic, it should work for other languages too. So we will also verify this belief holds for other languages.

Acknowledgements

We thank CFILT members at IIT Bombay for their valuable comments and suggestions. Special thanks to Kevin Patel and Aditya Joshi for their suggestions and feedback. This work is funded by Tata Consultancy Services Limited (TCS) under NGIE (Next Generation Information Extraction) with project code 13TCSIRC001.

References

Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin. 2019. Docbert: BERT for document classification. *arXiv preprint arXiv:1904.08398*.

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*.

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and

Matthew McDermott. 2019. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78.

Timothy Baldwin and Takaaki Tanaka. 2004. Translation by machine of complex nominals. In *Proceedings of the Workshop on Multiword Expressions Integrating Processing - MWE '04*, pages 24–31. Association for Computational Linguistics.

Renu Balyan and Niladri Chatterjee. 2015. Translating noun compounds using semantic relations. *Comput. Speech Lang.*, 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* -, pages 96–102. Association for Computational Linguistics.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3606–3611.

Zied Bouraoui, Jose Camacho-Collados, and Steven Schockaert. 2020. Inducing relational knowledge from BERT. In *AAAI*.

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)*, pages 144–147, Atlanta, Georgia, USA. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Murhaf Fares. 2016. A dataset for joint noun-noun compound bracketing and interpretation. In *Proceedings of the ACL 2016 Student Research Workshop*, pages 72–79.

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.

Roxana Girju, Dan Moldovan, Marta Tatu, and Daniel Antohe. 2005. On the semantics of noun compounds. *Comput. Speech Lang.*, 19(4):479–496.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2009. SemEval-2010 task 8. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions - DEW '09*, pages 94–99. Association for Computational Linguistics.

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.

Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. 2019. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *arXiv preprint arXiv:1904.05342*.

Su Nam Kim and Timothy Baldwin. 2005. Automatic interpretation of noun compounds using WordNet similarity. In *Lecture Notes in Computer Science*, pages 945–956. Springer Berlin Heidelberg.

Su Nam Kim and Timothy Baldwin. 2013. A lexical semantic approach to interpreting and bracketing English noun compounds. *Nat. Lang. Eng.*, 19(3):385–407.

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.

Jieh-Sheng Lee and Jieh Hsiang. 2019. PatentBERT: Patent classification with fine-tuning a pre-trained BERT model. *arXiv preprint arXiv:1906.02124*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

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

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Preslav Nakov. 2008. Noun compound interpretation using paraphrasing verbs: Feasibility study. In *Artificial Intelligence: Methodology, Systems, and Applications*, pages 103–117. Springer Berlin Heidelberg.

Preslav Nakov. 2013. On the interpretation of noun compounds: Syntax, semantics, and entailment. *Nat. Lang. Eng.*, 19(3):291–330.

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

Diarmuid Ó Séaghdha. 2007. Designing and evaluating a semantic annotation scheme for compound nouns. In *Proc. Corpus Linguistics*.

Soma Paul, Prashant Mathur, and Sushant Kishore. 2010. Syntactic construct: an aid for translating english nominal compound into hindi. In *Proceedings of the NAACL HLT Workshop on Extracting and Using Constructions in Computational Linguistics*, pages 32–38. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001.

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

Girishkumar Ponkiya, Kevin Patel, Pushpak Bhattacharyya, and Girish K Palshikar. 2018b. Treat us like the sequences we are: Prepositional paraphrasing of noun compounds using LSTM. In *The 27th International Conference on Computational Linguistics (COLING 2018)*, pages 1827–1836, Santa Fe, New-Mexico, USA.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Barbara Rosario, Marti A. Hearst, and Charles Fillmore. 2001. The descent of hierarchy, and selection in relational semantics. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, pages 247–254. Association for Computational Linguistics.

Vered Shwartz and Ido Dagan. 2018. Paraphrase to explicate: Revealing implicit noun-compound relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1200–1211.

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)*, pages 153–157, Atlanta, Georgia, USA. Association for Computational Linguistics.

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*, pages 678–687.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

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)*, pages 148–152, Atlanta, Georgia, USA. Association for Computational Linguistics.

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451.