Noun Compound Interpretation

by

Girishkumar Ponkiya

Based on my work under guidance of:

Prof. Pushpak Bhattacharyya

Mr. Girish Keshav Palshikar (TRDDC, Pune)

Oct 21, 2021

Outline

- Introduction
- Problem definition
- Interpretation via labeling
  - FrameNet-based labels
  - FrameNet-assisted noun compound interpretation
- Interpretation via paraphrasing
  - Supervised prepositional paraphrasing using LSTM
  - Unsupervised prepositional and free paraphrasing
- Conclusion
- Future work

Introduction

- **Noun Compound**: a sequence of two or more nouns that act as a single noun.

  Example: *apple pie, student protest*

- We consider only *compositional* noun-noun compounds.

  A compound is compositional if its meaning can be expressed in terms of the meaning of its components.

- The task of identifying underlying semantic relation between the components of a noun compound is known as *interpretation*.

Motivational Example

- “Honey Singh became the latest victim of celebrity death hoax.”
- Machine Translation:
  - [Hindi] हनी सिंघ प्रसिद्ध व्यक्ति की मौत के बारे में अफवाह के ताजा शिकार बने।
    “Hanī siṅgha prasid'dha vyakti kī mauta kē bārē mēṁ aphavāha kē tājā śikāra banē.”
- Question Answering:
  - What type of rumor was spread about Honey Singh?
- Text Entailment:
  - H: Honey Singh is dead. (*False*)

Problem Definition

Paraphrasing

- Given: a noun-noun compound
- Output: paraphrase(s) of the noun compound

- Free paraphrases:
  - Paraphrasing using any set of words
  - E.g., *apple pie*: '*a pie made of apple*'

- Prepositional paraphrase:
  - Paraphrase using a preposition only
  - E.g., *apple pie*: '*a pie of apple*'

Labeling

- Given: a noun-noun compound
- Output: an abstract label chosen from a predefined set of relations
- E.g.,

  apple pie: {INGREDIENT, COOK, UTENSIL}

Challenges

- The semantic relation is implicit

  The relation of a modifier noun with the head noun in a compound is not mentioned explicitly.

- Pragmatic factors influence the interpretation

  Why *students* are 'beneficiary' in *student price*, but not in *student protest*?

- In BNC corpus, 60.3% of total noun compounds appears only once (Baldwin and Tanaka, 2004)

  Challenge for statistical approaches

(LREC 2018)

FrameNet based Semantic Relations

Interpretation via Labeling

FrameNet

- FrameNet (Baker et al., 1998) is a lexical database that shows usage of words in actual text with annotated examples
  - Based on Fillmore (1976)'s Frame Semantics
- The theory claims: "meanings of most words can be inferred on the basis of a semantic frame: a conceptual structure that denotes the type of event, relation, or entity and the involved participants"
- For instance:
  - the concept of walking typically involves a person walking (**Self_Mover**), the **Path** on which walking occurs, the **Direction** in which the walking occurs, and so on.
  - This is represented as **Self_Motion** frame
  - **Self_Mover**, **Path**, **Direction**, etc. are called frame elements (FEs)
  - This frame can be invoked by followings words (via lexical units): *advance, crawl, dash, drive, march, run, walk*, etc.

FrameNet-based Labels

- Uses frames and frame elements of FrameNet to indicate semantic relations between the component nouns
- Given a noun compound $nc = \langle w_1, w_2 \rangle$,
  - w_2: invoke a frame
  - w_1: fits in one of the FEs of the frame
  - So, annotate noun compounds with a frame and a frame element
- Added benefits:
  - One can use information from FrameNet and FrameNet annotated data
  - Hierarchy of frames and frame elements can be used to decide granularity of relations
- We automatically extracted noun compounds and their labels from FrameNet
  - Retained only those which appear in any of the existing noun compound datasets.
  - Manually verified labels
  - The resulting dataset has 1546 noun-noun compounds
- We have illustrated how we can use these semantic relations to enhance dependency parsing

FrameNet Mappings

- FrameNet data provides two types of mappings:
  - words to frames (via lexical units), and
  - frames to frame elements
- For instance,
  - *'protest'* word can invoke three frames:

    protest → {PROTEST, POLITICAL_ACTIONS, and JUDGMENT_COMMUNICATION}
  - PROTEST frame has following frame elements:

    PROTEST → {ACTION, ISSUE, PROTESTER, SIDE, DEGREE, DESCRIPTOR, DURATION, EXPLANATION, FREQUENCY, MANNER, MEANS, PLACE, PURPOSE, TIME, …}
- We use these mappings to create a candidate set

Frame and Frame Element Relations

- FrameNet includes a graph of relations between frames.
- Some of important frame relations are:

  Inheritance: close to a typical Is-A relation in an ontology — PROTEST $\xrightarrow{\text{Is-A}}$ INTENTIONALLY_ACT

  Using: the child frame presupposes the parent frame — PROTEST $\xrightarrow{\text{Uses}}$ TAKING_SIDES

  Subframe: the child frame is a subevent of a complex parent event — TRIAL $\xrightarrow{\text{Subframe}}$ VERDICT

- Along with each frame relation, FrameNet also defines relations between frame elements of the parent frame and frame elements of the child frame.

  PROTEST:PROTESTER $\xrightarrow{\text{Is-A}}$ INTENTIONALLY_ACT:AGENT

  PROTEST:PROTESTER $\xrightarrow{\text{Uses}}$ TAKING_SIDES:COGNIZER

- We use these relations to learn embeddings for frames and frame elements

(ACL-2021)

FrameNet-assisted Noun Compound Interpretation

Interpretation via Labeling

Problem Definition

- Given: a noun compound (say, "$w_1\ w_2$")
- Output:
  - Frame f: appropriate for w_2
  - Frame Element (from frame f): appropriate for w_1

- Examples:

  student protest → **PROTEST** : **PROTESTER**
  board approval → **DENY_OR_GRANT_PERMISSION** : **AUTHORITY**

- Contribution:
  - Created a tool for annotation
  - Extended our LREC-2018 dataset
  - A method for automatic prediction of frame and FE for noun compounds

Idea

- We have 1123 frames and 11,473 frame elements in FrameNet, but 1546 annotated examples.
- For better generalization, we predict over a continuous space:
  - We embed labels (frame and frame elements) into a continuous space such that the space captures relation among the labels.
  - Then, prediction in the generalized target space could help in prediction on unseen labels.
- Our system is a two-step pipeline: (a) predict a frame, and (b) predict a frame element from the frame.
- Pruning a label set:
  - We create a *candidate set* using FrameNet mappings to remove unnecessary labels.

Approach

- Given a noun compound '$w_1\ w_2$'
- Frame Prediction:
  - Get candidate frame set using FrameNet API

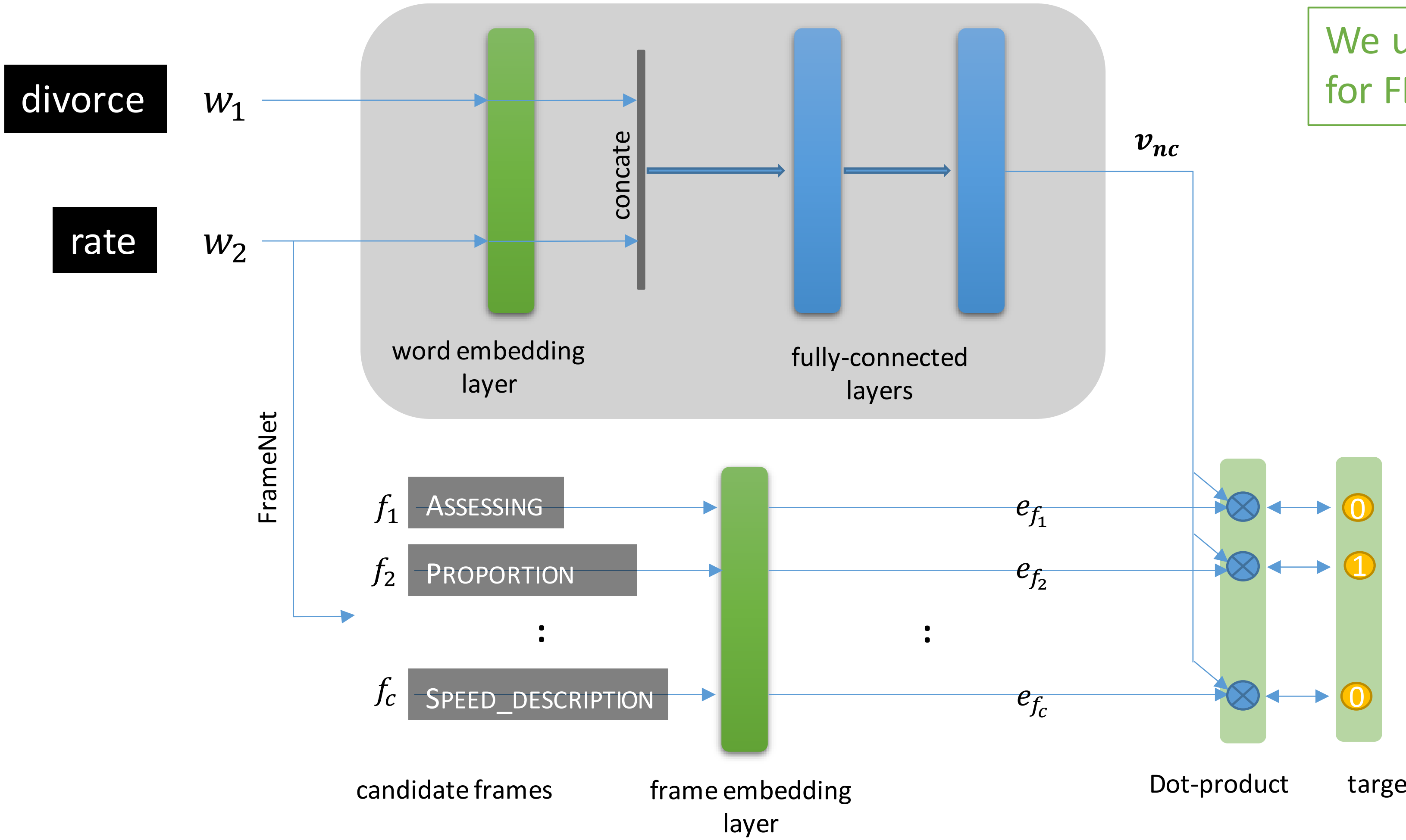
$$\text{candidate}-\text{frames} = \text{word2frame}-\text{mapping}(w_2)$$

  - Predict an appropriate frame f from the candidate set
- FE Prediction:
  - Get candidate FE set using FrameNet API

$$\text{candidate}-\text{FEs} = \text{frame2fe}-\text{mapping}(f)$$

  - Predict an appropriate FE from the candidate set

System Architecture



We use a similar model for FE prediction

$$f^* = argmax_{f_i}\ \boldsymbol{v}_{nc} \cdot \boldsymbol{e}_{f_i}$$

Frame and Frame Element Embeddings

- For learning frame embeddings, we adapt Kumar et al. (2019)'s approach:
  - Train ConvE: a multi-layer 2D-convolution network model proposed by Dettmers et al. (2018)
  - Use the definition of entities along with the relations to learn entities and relation embeddings
- We train frame and frame element embeddings separately
- We use frame and frame element embeddings to initialize the corresponding embedding layers

Experimental Setup: Training

- For each noun compound nc_i, the network outputs scores for each candidate
- During training,
  - Apply *softmax* over the score values, and compute categorical cross entropy loss
  - We fine-tune word embeddings
  - Freeze the frame/FE embedding layer for initial few epochs and them update them
- We also randomly initialize frame/FE embeddings to see usefulness of pre-trained embeddings

- Random baseline:
  - A noun compound nc with candidate set $cand(nc)$ could be predicted correctly with probability $\frac{1}{|cand(nc)|}$
  - So, accuracy for random baseline is computed using:

$$\mathrm{Acc}_{\mathrm{random}} = \frac{1}{N} \sum_{nc_i \in \mathrm{test-set}} \frac{1}{|cand(nc_i)|}$$

Experimental Setup: Evaluation Metrics

- We report Precision, Recall and F-score for our experiments
  - These values are weighted values in proportion to the number of test-examples for each label

- We also report results for…
  - *unseen-set:* a subset of all test-samples, whose output label does not have even a single example in the training set
  - In a fold with 310 test samples, *unseen-set* statistics:

    (1) 30 unique frames, covering 32 test samples, (2) 75 unique FEs, covering 82 test samples

Results (Frame Prediction)

	D1			D1+D2		
	P	**R**	**F**	**P**	**R**	**F**
Random baseline	23.99	53.04	33.04	24.88	53.29	33.92
SVM	64.45	69.45	64.81	66.64	71.86	67.16
WITHOUT frame embedding	76.81	77.13	**75.70**	78.13	78.26	**76.67**
WITH frame embeddings	77.05	76.90	75.68	78.67	78.14	76.59

Table 3: Performance of our approach for *frame prediction* compared to the baseline approach on **test set** from D1 and D1+D2 datasets. (**P**: Precision@1; **R**: Recall@1; **F**: F1-score@1)

	D1			D1+D2		
	P	**R**	**F**	**P**	**R**	**F**
Random baseline	17.68	51.83	26.37	23.08	47.88	31.15
WITHOUT frame embedding	40.38	38.42	39.07	41.29	42.42	40.82
WITH frame embeddings	46.83	43.75	**44.63**	48.25	44.98	**45.78**

Table 4: Comparison of our approach for *frame prediction* with random prediction baseline on examples with **unseen-set**, achieving zero-shot learning **for frame prediction**. (**P**: Precision@1; **R**: Recall@1; **F**: F1-score@1)

Results (FE Prediction)

	D1			D1+D2		
	P	**R**	**F**	**P**	**R**	**F**
Random baseline	13.75	9.68	11.36	14.48	10.20	11.97
SVM	43.99	49.13	43.73	46.55	50.85	45.14
WITHOUT frame element embedding	49.42	52.14	**49.28**	51.13	54.08	**50.29**
WITH frame element embeddings	48.18	48.50	46.61	50.47	50.23	49.73

Table 5: Performance of our approach for *frame element prediction* compared to the baseline approach on **test set** from D1 and D1+D2 datasets. (**P**: Precision@1; **R**: Recall@1; **F**: F1-score@1)

	D1			D1+D2		
	P	**R**	**F**	**P**	**R**	**F**
Random baseline	13.28	10.35	**11.64**	15.36	9.26	**11.56**
WITHOUT frame element embedding	2.43	2.33	2.36	3.45	2.53	2.67
WITH frame element embeddings	5.05	4.96	4.70	5.23	5.23	5.28

Table 6: Comparison of our approach for *frame element prediction* with random prediction baseline on examples with **unseen-set**. (**P**: Precision@1; **R**: Recall@1; **F**: F1-score@1)

Summary

- We proposed semantic relations which are based on frame semantics
  - We have develop a tool for annotation
- We show how we can use the FrameNet data for an automatic interpretation system
- Results for frame prediction are acceptable, but not for element prediction
- Main cause behind frame prediction: coverage issue in FrameNet
  - We plan to use additional resources, like FrameNet+, to handle coverage issue

Interpretation via Paraphrasing

Paraphrasing of Noun Compounds

- Paraphrasing involves rewriting the noun compound as a paraphrase which conveys its meaning explicitly, *e.g.*,
  - *orange juice*: "*juice made from orange*", "*juice with orange flavour*" or "*juice with orange color*"
- Free paraphrasing for noun compounds has been relatively less pursued

- Nakov (2008) uses a verb (and an optional preposition) to paraphrase a noun compound
- Given w_1 w_2, the annotators need to fill in the black: "w_1 w_2" *is a* "w_2 *that* ___ w_1"
- Extended in SemEval-2010 Task-9 (Butnariu et al., 2009): given a noun compound a list of paraphrasing verbs, produce aptness scores that correlate well with human judgement.

- SemEval-2013 Task-4 (Hendrickx et al., 2013) asked the participating teams to generate free paraphrases with score for given noun compounds.
  - The organizer observed that the top-ranked paraphrase for a given compound is often a preposition-only phrase.

Prepositional Paraphrasing of NCs

- Prepositional paraphrasing: paraphrasing using only prepositions,
  - e.g., *orange juice*: "*juice of orange*"

- Advantage: the set of prepositions is finite, limited, and pre-defined.
- Disadvantage: the information is too coarse-grained, prepositions are too ambiguous (Girju et al., 2005)

- We use 8-prepositions used by Lauer (1995)

  {*about, at, for, from, in, of, on, with*}

Motivation

- Prepositions capture significant information regarding underlying semantic relations
- Evidence:
  - Girju et al. (2005) observed that prepositions as an additional feature improved the performance of automatic labeling significantly
  - We annotated Kim and Baldwin (2005)’s dataset with prepositions and observed high correlation between prepositions and semantic relations
  - In some NLP tasks (such as English-Hindi MT), uncovering of case markers is sufficient (Paul et al., 2010; Kulkarni et al., 2012)

rice husk ⇒ husk of rice ⇒ chaaval kee bhoosee

sevā-nivṛtta ⇒ sevā se nivṛtta ⇒ retired from service

Related Work

- Supervised approaches rely on annotated data that needs to be sufficiently large and representative enough of the underlying problem.
  - The existing datasets are small in size
- Hence, unsupervised models were preferred (Lauer, 1995; Lapata and Keller, 2004)
- Lauer (1995) proposed a triples $\langle w_2, prep, w_1 \rangle$ frequency based model
  - Lauer (1995) also computed frequencies from Grolier's encyclopedia
  - Lapata and Keller (2004) used Altavista search engine and BNC corpus

(COLING 2018)

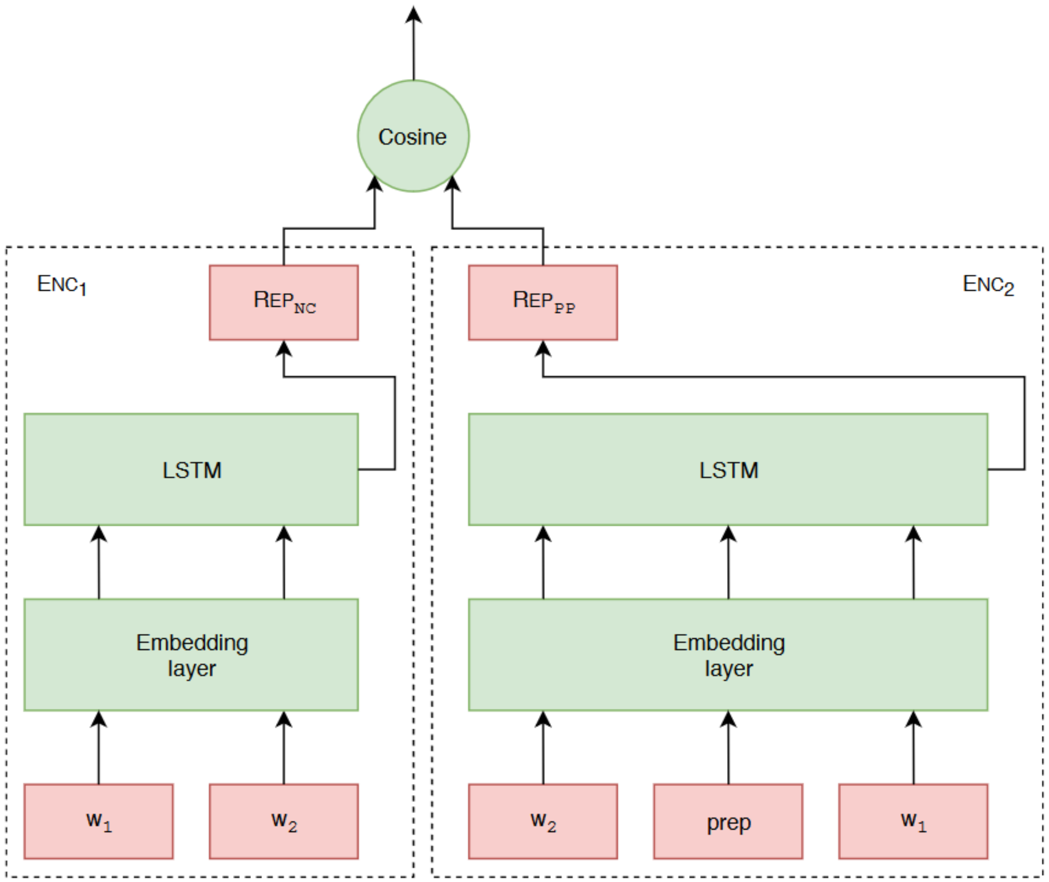
Prepositional Paraphrasing using LSTM

Interpretation via Paraphrasing

Idea

- Noun compounds and prepositional paraphrases are sequences
- We can use sequence learners to learn their representations
- We learn representations of an NC and its paraphrases, such that the representation of a noun compound is most similar to the representation of its prepositional paraphrase.

Approach



- The network consists of two encoders:
  - ENC_1 gives REP_{NC} for the noun compound NC
  - ENC_2 gives REP_{PP} for the prepositional paraphrase PP
- Higher the cosine similarity of REP_{NC} and REP_{PP}, the greater is the match between NC and PP
- For a test sample *NNC*, the correct preposition PP^* is given by

$$PP^* = \underset{PP}{\arg\max}\ \ cosine(\text{REP}_{\text{NNC}}, \text{REP}_{\text{PP}})$$

- We initialize the embedding layer with Google's pre-trained embeddings
- 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

Experimental Setup: Datasets

- Two relevant datasets: Lauer (1995) and Girju et al. (2005) with 282 and 805 examples, respectively
- Inhouse: We manually annotated noun compounds† (from Kim and Baldwin (2005)'s dataset)
  - Each noun compound was annotated by two annotators
  - The percentage inter-annotator agreement is 51.48% (Cohen's kappa $\kappa = 0.445$); 1042 examples
- To create a large dataset automatically, we adapted Lapata and Keller (2004)'s approach:

  Use a search engine to find a prepositional paraphrase with highest frequency, and consider it as a correct preposition
  - Lapata and Keller (2004) used Altavista search engine and BNC corpus. We use Netspeak.

expert [about at for from in of on with] [a an the] analysis

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%

- Noun compounds were collected from datasets of Kim and Baldwin (2005), Ó Séaghdha (2007) and Tratz and Hovy (2010)

† We released the dataset with our COLING-2018 paper

Experimental Setup: Training

- We have noun compounds with their prepositional paraphrases.
- For each noun compound, we treat the remaining (other than the given) prepositions as negative samples.
- Example: *analysis expert* → *in*
  - Positive sample: (*analysis expert*, ‘*expert <u>in</u> analysis*’)
  - Negative samples: (*analysis expert*, ‘*expert <u>from</u> analysis*’), (*analysis expert*, ‘*expert <u>about</u> analysis*’), etc.
- Two different set of experiments:
  1. Train the system on the automatically annotated dataset and evaluate performance on the various datasets
  2. (For each dataset) additionally fine-tune the trained-model using a portion of the manually annotated dataset
     - We use 75% of the dataset for tuning, and rest 25% of examples for testing.

Results

Dataset	Approach	Without Tuning			With Tuning		
		P	R	F	P	R	F
Lauer (1995)	Baseline	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 et al. (2005)	Baseline	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	Baseline	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**

Comparison of performance of our LSTM based architecture (NC-LSTM) with Dima and Hinrichs (2015)'s feed-forward neural network based architecture (*Baseline*) on different datasets
(P: Precision; R: Recall; F: F-score; weighted in proportion to the no. of test-examples for each preposition)

Analysis

- NC-LSTM performs comparably, if not better, than the baseline.
- NC-LSTM easily outperforms the baseline by a significant margin when tuned with a portion of a gold dataset.
- Improvement in tuning can be easily explained from the fact that the original training data was extracted from the web, and is noisy in nature.
- For instance, automatic step assigns ‘*tree with apple*’ as paraphrase of *apple tree,* but the correct paraphrase is ‘*tree of apple*’.
- The following are example sentences, which contributed to the count of ‘*tree with apple*’:

  “*if you combine a pine* ***tree with an apple*** *tree you do indeed get a pineapple tree.*”

  “*What do you get if you cross a Christmas* ***tree with an apple****?*”

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

Analysis

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

Figure 1: PCA visualizations of noun compounds and their prepositional paraphrases for some examples

(EMNLP 2020)

Unsupervised Prepositional and Free Paraphrasing

Interpretation via Paraphrasing

Bidirectional Encoder Representations from Transformers (BERT)

- BERT (Devlin et al., 2019) is a language model using the Transformer architecture (Vaswani et al., 2017) designed to pre-train deep bidirectional representations.

- BERT has been trained on two prediction tasks:

1. Random tokens in each sequence are replaced with the special `[MASK]` token and BERT is trained to recover the masked tokens.

   This allows representations to be conditioned on the left and right context.

2. Second, BERT is trained to predict whether the second text segment follows the first text segment.

   This improves BERT's understanding of the relationship between two text sentences.

Masked Language Model

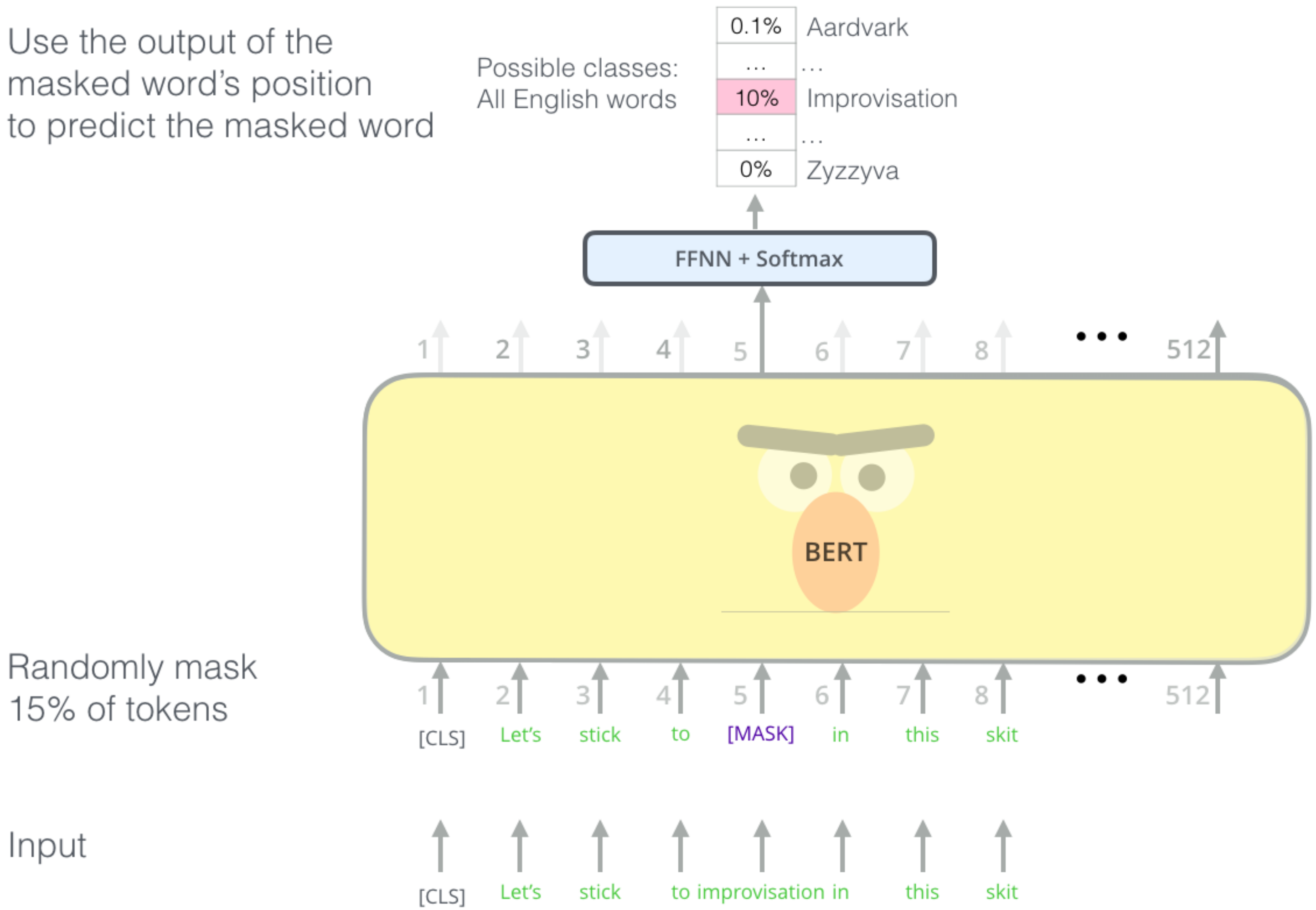


Image by Jay Alammar (copied from http://jalammar.github.io/illustrated-bert/#masked-language-model)

Idea

- BERT (Devlin et al., 2019) is a language model using the Transformer architecture (Vaswani et al., 2017) designed to pre-train deep bidirectional representations.
- As the BERT has been trained to uncover a masked token, we can use it to uncover a preposition in prepositional paraphrase of a noun compound.

- **Approach**:
  - Paraphrase the given noun compound with a blank space (indicated by a `[MASK]` token) for a preposition.
  - Use the pre-trained BERT to fill in the blank.
  - Given a noun compound w_1 w_2, we use three patterns:

	Pattern	BERT Input
#1	**w_2 <prep> w_1**	`[CLS]` w_2 `[MASK]` w_1 `[SEP]`
#2	**w_1 w_2 means w_2 <prep> w_1**	`[CLS]` w_1 w_2 means w_2 `[MASK]` w_1 `[SEP]`
#3	**a w_1 w_2 is a w_1 <prep> the w_2**	`[CLS]` a w_1 w_2 is a w_2 `[MASK]` the w_1 `[SEP]`

  - BERT provides a score for each vocabulary word. We select preposition with the highest score.

- We use Transformers library (Wolf et al., 2019), and run the experiments on Google Colaboratory.

Overall Results

Datasets →	Lauer (1995)			Girju et al. (2005)			Inhouse		
Approach↓	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Trained only on automatically annotated dataset									
Baseline	40.85	38.05	31.15	74.72	80.69	77.52	63.00	66.96	63.97
NC-LSTM	50.84	45.07	40.66	78.86	74.26	75.50	62.32	65.65	63.09
Trained on automatically annotated dataset + Fine-tuned on respective manually annotated dataset									
Baseline	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
*No training: RoBERTa-**base***									
w_2 <prep> w_1	50.82	38.02	26.74	79.28	77.22	78.15	45.89	53.47	46.94
w_1 w_2 means w_2 <prep> w_1	55.57	52.11	47.95	83.11	57.92	66.99	65.39	63.04	63.47
a w_1 w_2 is a w_1 <prep> the w_2	43.30	47.88	41.51	83.83	67.32	74.26	64.48	63.04	63.48
*No training: RoBERTa-**large***									
w_2 <prep> w_1	50.78	33.80	25.79	79.33	69.30	73.96	53.72	56.08	51.94
w_1 w_2 means w_2 <prep> w_1	51.78	47.88	43.28	87.02	72.27	78.58	72.98	72.60	**72.21**
a w_1 w_2 is a w_1 <prep> the w_2	56.06	56.33	**51.74**	88.30	72.77	79.12	68.36	67.39	67.32

Comparing performance of our RoBERTa-based unsupervised system with LSTM-based (NC-LSTM) and feed-forward neural network based (*Baseline*) supervised systems on different datasets (P: Precision; R: Recall, F: F-score)

Error Analysis

- We analysed the performance 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 *<u>of</u>* predicted by pattern-2, but pattern-3 predicted *<u>for</u>***.
- Examples (using RoBERTa-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 RoBERTa-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 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).

Free Paraphrasing

- SemEval-2013 Task-4: Free Paraphrases of Noun Compounds
  - Target: noun-noun compound, e.g. *air filter*
  - Goal: produce an explicitly ranked list of free paraphrases, *e.g.*,
    - 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*
    - …
  - Data collection: using *Amazon Mechanical Turk*.
  - Evaluation: comparison to a similar list produced by human annotators

	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

- The authors observed that the top-ranked paraphrase for a given compound is often a preposition-only phrase.

Motivation

- Abstract relations and prepositional paraphrasing have limitations on expressivity and coverage
  - Fixed set of abstract relations
  - Only 8 prepositions for prepositional paraphrasing
- For example:
  - Consider *caffeine headache* and *ice-cream headache*:
    - One would be assign the same semantic relation (CAUSEOF) or can be paraphrased with *for*
    - However, a lack of caffeine causes the former, an excess of ice-cream – the latter.
  - Similarly for *headache pills* and *fertility pills*
- Ambiguity: a *plastic saw* could be a '*saw made of plastic*' or a '*saw for cutting plastic*'

T5 framework

Every task we consider uses text as input to the model, which is trained to generate some target text. This allows us to use the same model, loss function, and hyperparameters across our diverse set of tasks including translation (green), linguistic acceptability (red), sentence similarity (yellow), and document summarization (blue). It also provides a standard testbed for the methods included in our empirical survey.

- A pre-training objective used by T5 aligns more closely with a fill-in-the-blank task where the model predicts missing words within a corrupted piece of text.
- The model is asked to replace a blank with any number of words.
  - Model can do this as it uses encoder-decoder architecture.
- We generate multiple candidates for the blank to create multiple paraphrases.
- There is one more similar framework, BART (Bidirectional and Auto-Regressive Transformers), from Facebook. [paper]

Using T5 for Free Paraphrasing

- **Templet:** $w_1\ w_2 \Rightarrow$ *"A $w_1\ w_2$ is a w_2 < extra_id_0> the w_1 . </s>"*

- Given a noun compound: say '*club house*'

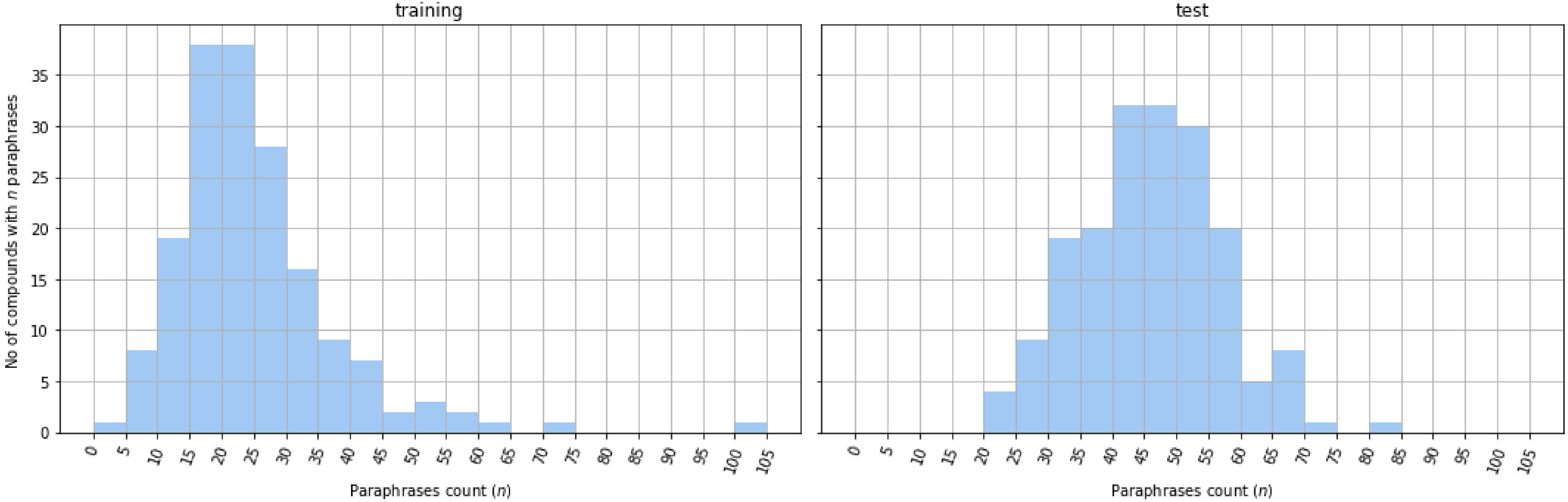
1. **T5 input**: *"A club house is a house <extra_id_0> the club . </s>"*
2. **T5 generated** the following sequences (for *k*=10):
   - *"<extra_id_0> **for** <extra_id_1>. A"*
   - *"<extra_id_0> **of** <extra_id_1>. A"*
   - *"<extra_id_0> **for** <extra_id_1> . <extra_id_1>"*
   - *"<extra_id_0> **for** <extra_id_1> house ."*
   - *"<extra_id_0> **owned by** <extra_id_1> ."*
   - *"<extra_id_0> **of** <extra_id_1> . <extra_id_1>"*
   - *"<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"*
3. Extract words between *<extra_id_0>* and *<extra_id_1>*, and use them to generate a candidate para-phrase
   - *"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"*
4. Grouping similar paraphrases, and ranking them based on the frequencies, we get (rank:paraphrase)
   - 1 *"house for a club"*
   - 1 *"house of a club"*
   - 2 *"house owned by a club"*
   - 3 *"house that belongs to a club"*
   - 3 *"house in a club"*

Evaluation

- **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.
    - It rewards a system for accurately reproducing the top-ranked human paraphrases in the "gold standard".
    - A system will achieve a higher score in a non-isomorphic match if it reproduces the top-ranked human paraphrases as opposed to lower-ranked human paraphrases.
    - The ordering of system's paraphrases is thus not important in non-isomorphic matching.
- **Isomorphic scoring** maps system paraphrases to (unmapped) paraphrases from the reference dataset, and requires systems to produce the full set of paraphrases
    - It rewards both precision and recall.
    - It rewards a system for accurately reproducing the paraphrases suggested by human judges, and for reproducing as many of these as it can, and in much the same order.
    - System's paraphrases are matched 1-to-1 with reference paraphrases on a first-come first-matched basis, so ordering can be crucial.

Number of Paraphrases per Compound

How many paraphrase should be generated?



Results

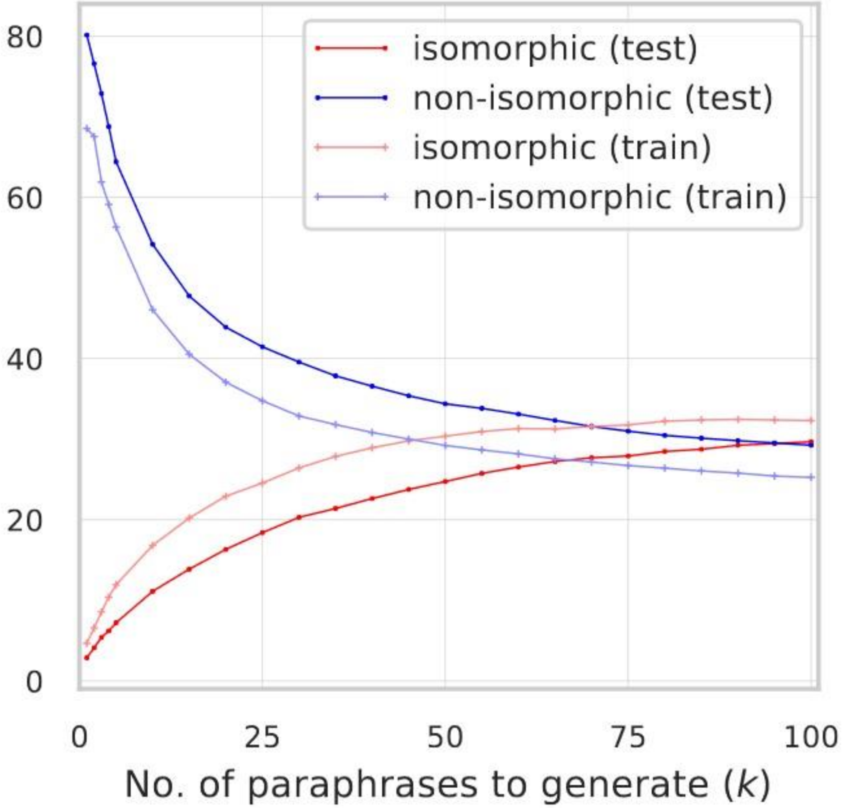


Figure 4: Performance of T5-based system for different value of k (number of paraphrases to generate) on train and test sets of SemEval-2013 Task-9 dataset.

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)

Analysis

- 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)).
- T5 generates quite a good quality set of paraphrases.
  - However, the reference list does not have matching paraphrases
- Our system allows extra words only between the component nouns.
- However, the dataset has many reference paraphrases where new words appear..
  - at the beginning: '*pay policy*' → "<u>corporate</u> policy on pay"
  - at the end of a paraphrase: '*operating system*' → "*system controls operating <u>of computer</u>*"

- Example:
  "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"
- Examples, marked with dagger-sign (†), have a partial matching (score ≤ 25%), while the rest of the listed paraphrases do not havea match.

Conclusion

- We extend ideas from language modeling for prepositional paraphrasing
  - Using an LSTM encoder along with distant supervision helps prepositional paraphrasing
- We use abstract labels as a representation for semantic relations
  - Using semantic relations which have a lexical resource associated with them have added benefits
  - We use FrameNet information to show how it can help the system to predict unseen labels
  - Our results for frame element predictions are not so good, but results for frame prediction are promising
- We use pre-trained language models to develop unsupervised paraphrasing systems
  - We pose the paraphrasing problem as fill-in-the-blank problem
  - BERT can be used to uncover a preposition as it has been trained to uncover a ‘unknown’ word
  - T5 can be used to uncover for free paraphrasing as it has been trained to uncover multiple words

Our Publications

1. **Girishkumar Ponkiya**, Pushpak Bhattacharyya, and Girish K Palshikar. FrameNet-assisted Noun Compound Interpretation. In *Proceedings of the Association of Computational Linguistics: Findings (Finding of ACL)*.
2. **Girishkumar Ponkiya**, Rudra Murthy, Pushpak Bhattacharyya, and Girish K Palshikar. inside Noun Compounds: Unsupervised Prepositional and Free Paraphrasing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings (Finding of EMNLP),* pp. 4313-4323, 2020.
3. **Girishkumar Ponkiya**, Kevin Patel, Pushpak Bhattacharyya, and Girish K Palshikar. Treat us like the sequences we are: Prepositional paraphrasing of noun compounds using LSTM. In *The 27th International Conference on Computational Linguistics (COLING 2018)*, Santa Fe, New-Mexico, USA, 2018a.
4. **Girishkumar Ponkiya**, Kevin Patel, Pushpak Bhattacharyya, and Girish K Palshikar. Towards a standardized dataset for noun compound interpretation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 3092–3097, Miyazaki, Japan, 2018b. ISBN 979-10-95546-00-9.
5. **Girishkumar Ponkiya**, Pushpak Bhattacharyya, and Girish K Palshikar. On why coarse class classification is a bottleneck for noun compound interpretation. In *13th International Conference on Natural Language Processing (ICON 2016)*, page 293, 2016.

References – 1/4

- Timothy Baldwin and Takaaki Tanaka. Translation by machine of complex nominals: Getting it right. *In Proceedings of the Workshop on Multiword Expressions: Integrating Processing*, pages 24—31. Association for Computational Linguistics, 2004.
- Renu Balyan and Niladri Chatterjee. Translating noun compounds using semantic relations. *Computer Speech & Language*, 2014.
- Ken Barker and Stan Szpakowicz. Semi-automatic recognition of noun modifier relationships. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 96—102. Association for Computational Linguistics, 1998.
- Cristina Butnariu and Tony Veale. A concept-centered approach to noun-compound interpretation. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 81—88. Association for Computational Linguistics, 2008.
- Peter Clark and Robin Boswell. Rule induction with cn2: Some recent improvements. In *Machine learning—EWSL-91*, pages 151—163, 1991.
- Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493{2537, 2011.
- Corina Dima and Erhard Hinrichs. Automatic noun compound interpretation using deep neural networks and word embeddings. *IWCS 2015*, page 173, 2015.

References – 2/4

- Pamela Downing. On the creation and use of English compound nouns. *Language*, pages 810—842, 1977.
- Barbara J Grosz, Scott Weinstein, and Aravind K Joshi. Centering: A framework for modeling the local coherence of discourse. *Computational linguistics*, 21(2):203—225, 1995.
- Su Nam Kim and Timothy Baldwin. Automatic interpretation of noun compounds using WordNet similarity. In *Natural Language Processing—IJCNLP 2005*, pages 945—956. Springer, 2005.
- Preslav Nakov. On the interpretation of noun compounds: Syntax, semantics, and entailment. *Natural Language Engineering*, 19(03):291—330, 2013.
- Parag Singla. *Markov Logic: Theory, Algorithms and Applications*. PhD thesis, University of Washington Graduate School, 2009.
- Stephen Tratz. *Semantically-enriched parsing for natural language understanding*. University of Southern California, 2011.
- Stephen Tratz and Eduard Hovy. A taxonomy, dataset, and classier for automatic noun compound interpretation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 678—687. Association for Computational Linguistics, 2010.

References – 3/4

- Tim Van de Cruys. A neural network approach to selectional preference acquisition. In *EMNLP*, pages 26–35 , 2014.
- Philip Resnik. Selectional constraints: An information-theoretic model and its computational realization. Cognition, 61(1):127—159, 1996.
- Katrin Erk. A simple, similarity-based model for selectional preferences. In ANNUAL MEETING-ASSOCIATION FOR COMPUTATIONAL LINGUISTICS, volume 45, page 216, 2007.
- Diarmuid Ó Séaghdha. Latent variable models of selectional preference. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 435—444. Association for Computational Linguistics, 2010.
- 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.
- Murhaf Fares, Stephan Oepen, and Erik Velldal. "Transfer and Multi-Task Learning for Noun–Noun Compound Interpretation." *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 2018.
- Sawan Kumar, Sharmistha Jat, Karan Saxena, and Partha Talukdar. "Zero-shot Word Sense Disambiguation using Sense Definition Embeddings." In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pp. 5670-5681. 2019.
- Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. "Convolutional 2D knowledge graph embeddings." In *Thirty-Second AAAI Conference on Artificial Intelligence*. 2018.

References – 4/4

- Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The Berkeley FrameNet project. In *COLING-ACL'98: Proceedings of the Conference*, pages 86–90, Montreal, Canada, 1998.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, June 2019. Association for Computational Linguistics.
- Preslav Nakov. Noun compound interpretation using paraphrasing verbs: Feasibility study. In Artificial Intelligence: Methodology, Systems, and Applications, pages 103–117. *Springer Berlin Heidelberg*, 2008c.
- Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 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, 2009. ISBN 9781932432312. doi: 10.3115/1621969.1621986.
- Iris Hendrickx, Preslav Nakov, Stan Szpakowicz, Zornitsa Kozareva, Diarmuid O Séaghdha, and Tony Veale. SemEval-2013 task 4: Free paraphrases of noun compounds. *Atlanta, Georgia, USA*, page 138, 2013.

Thank you

https://girishponkiya.github.io