FrameNet-assisted Noun Compound Interpretation

Girishkumar Ponkiya[†], **Diptesh Kanojia**[‡], **Pushpak Bhattacharyya**[†]
[†]Indian Institute of Technology Bombay, Mumbai
[‡]Centre for Translation Studies, University of Surrey, United Kingdom
{girishp,diptesh,pb}@cse.iitb.ac.in

Girish K Palshikar
TCS Research, Tata Consultancy Services, Pune
gk.palshikar@tcs.com

Abstract

Given a noun compound (NC), we address the problem of predicting the appropriate semantic label linking the constituents of the NC. This problem is called *Noun Compound Interpretation (NCI)*. We use FrameNet as a semantic label repository. For example, given the noun compound (*board approval*), we predict the frame (DENY_OR_GRANT_PERMISSION, as per FrameNet) as appropriate and the semantic role of the modifier word (AUTHORITY) as the semantic label linking *board* and *approval*; the resulting label is DENY_OR_GRANT_PERMISSION:AUTHORITY.

Our semantic label repository is very large ($\approx$ 11k labels) compared to the NC data available for training (approx 1900). Thus, learning in this case, especially for unseen semantic labels, is hard. We propose to solve this problem by predicting semantic labels in a *continuous label embedding space*, which is novel. This embedding space is created by learning label embeddings using the FrameNet data. The embeddings are then used to train two separate models – one for predicting Frames and the other for FEs. As the label embedding space captures the semantics of the labels, using these embeddings enables generalizing well on unseen labels, thus achieving zero-shot learning. Our preliminary investigations show that the proposed approach performs well for unseen labels, achieving 5% and 2% points improvements over baselines for the frame and FE prediction, respectively. The study shows the promise of the use of continuous space embeddings for noun compound interpretation and points to the need for further investigation.

1 Introduction

A noun compound is a sequence of two or more nouns that act as a single entity with well-defined meaning (*e.g.*, *paper submission*, *colon cancer*, *etc.*). Semantic relations between the component nouns are implicit. For instance, the information that ‘it is a *juice* made from *orange*’ is hidden in *orange juice*. Uncovering this semantic relation is called the problem of *Noun Compound Interpretation (NCI)*. NCI needs ML, as the task faces the challenge of ambiguity, and disambiguation by rules is well nigh impossible because of multifarious complex underlying language phenomena. The proposition of storing NCs and doing table lookup for interpretation is also impractical due to a large number of NCs and the challenge of high productivity (new nouns and NCs get created frequently, *e.g.*, *corona vaccine* is a relatively new NC).

Often, the exact relation, sentiment, *etc.* are also governed by contextual pragmatics. For instance, the sentiment towards *tax money* depends on who the beneficiary is, which again depends on the predicate. The predicate *give* could indicate negative sentiment (for the tax-payer), whereas the predicate *receive* would indicate positive sentiment (for the government). Due to such instances, NLP tasks such as machine translation (Baldwin and Tanaka, 2004; Balyan and Chatterjee, 2015), textual entailment (Nakov, 2013), question answering (Ahn et al., 2005), *etc.* suffer when they encounter noun compounds. For example, from the below text-question pairs, a system would need to interpret the underlying semantics within the compound, to answer the question correctly.

(a) *“student protest”*: *“who is protesting?”*,

(b) *“fee-hike protest”*: *“why protest?”*, and

(c) *“university protest”*: *“where is the protest?”*

In this work, we interpret only *compositional* noun-noun compounds. A noun-noun compound is categorised as compositional, if the meaning of the compound can be composed from the semantics of the individual noun units present.

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 2901–2911
August 1–6, 2021. ©2021 Association for Computational Linguistics

From a relation representation perspective, noun compounds are interpreted in two ways: via *labelling* and *paraphrasing*. Labelling involves assigning an abstract semantic relation from a predefined set, for example, *orange juice*: MadeOf, *hillside home*: Location, *etc.* There are many inventories of predefined semantic relations.

We use the FrameNet based labels proposed by Ponkiya et al. (2018a). As per their convention, the head noun of a compound invokes the frame, and the modifier noun fits in one of the frame elements of the invoked frame, *vide 'board approval'* in the abstract.

There are more than 11,000 FEs in FrameNet, and we have about 1900 training examples. Thus, the average number of examples for each label is quite small, and many labels do not have a training example. In summary, the contributions of this paper are three-fold:

1. We embed FrameNet entities in a continuous space, perform prediction in the continuous space to generalize over unseen labels, and show performance improvement on the unseen labels.

2. We create a noun-compound annotation tool that assists annotators in providing manual labels, and we release it publicly.

3. Using the above tool, we extend the dataset released by Ponkiya et al. (2018a) with 326 more manually-annotated gold samples, and release it for further research.

The rest of the paper is organized as follow: Section 2 discuss related work, Section 3 gives an overview of foundations for the work. Section 4 details our approach. Section 5 provides experimental details: the dataset used and training/testing setup. Section 6 discusses the results and analysis, followed by a conclusion and future work. The code, dataset and the tool can be downloaded from `http://www.cfilt.iitb.ac.in/nc-dataset`.

2 Related Work

A relation between the components of a noun compound (say, *chocolate cake*) can be represented in one of the following two ways: (1) assigning a relation from a predefined set of semantic relations (MadeOf), or (2) using a paraphrase to convey the underlying semantic relation (*"cake made using chocolates"* or *"cake with chocolate flavor"*).

Noun-compound (NC) interpretation via **labelling** is the most commonly used methodology for NC interpretation. Scholars have proposed many inventories of semantic relations (Levi, 1978; Warren, 1978; Vanderwende, 1994; Lauer, 1995; Barker and Szpakowicz, 1998; Ó Séaghdha, 2007; Rosario et al., 2001; Tratz and Hovy, 2010; Fares, 2016; Ponkiya et al., 2018a). A recent FrameNet-based inventory by Ponkiya et al. (2018a) proposed FEs (Frame Elements) from FrameNet as labels (or, semantic relations). They released a dataset by annotating each noun compound with a frame and a frame element; and proposed this annotation for predicate 'nominalization'. However, it also works for most of the cases of 'predicate deletion'.

For automatic labelling, Dima and Hinrichs (2015) and Fares et al. (2018)'s architecture is similar to ours. Dima and Hinrichs (2015) proposed a feed-forward neural network-based approach. This network takes concatenated embeddings of component nouns as an input and predicts one of the labels from the Tratz and Hovy (2010)'s label set. Fares et al. (2018) used a similar feed-forward network to predict two types of relations. This network, however, shares initial layers and separates output layers for each label type.

NC interpretation via **paraphrasing** is another methodology that contains approaches such as prepositional and free paraphrasing. Prepositional paraphrasing, *i.e.*, paraphrasing using a preposition, for example, *student protest*: "*protest by student(s)*", is a relatively well-attended problem (Lauer, 1995; Lapata and Keller, 2004; Ponkiya et al., 2018b). All the above approaches proposed for prepositional paraphrasing use the fixed-set of eight prepositions proposed by Lauer (1995). The other set of approaches, *i.e.*, free paraphrasing, however, has not received much attention. Apart from two SemEval tasks (Butnariu et al., 2009; Hendrickx et al., 2013), it does not have much literature available. A recent study (Ponkiya et al., 2020) expresses paraphrasing as a "fill-in-the-blank" problem, and utilizes pre-trained language models, for the task of noun-compound interpretation.

3 Foundations

Levi (1978) performed a linguistic study to understand how noun compounds are generated. They call such compounds *nominal compounds*. This theory puts nominal compounds into two categories, based on the compounding process, as

1. **Predicate Deletion:** Here, a predicate between the components is dropped to create a compound.

   For example, *apple pie* is a "pie made from apple." The predicate made from is dropped in this case. Similarly, for *elbow injury*, *gas pipeline*, *etc*.

2. **Predicate Nominalization:** Here, the head noun is a nominalized form of a verb, and the modifier is an argument of the verb.

   For example, "The union demonstrated against the price hike..." becomes "The union demonstration against the price hike..."

   Verbal noun as head: *student demonstration*, *government approval*, *opposition objection*, *etc*.

   Verb form as head: *student protest*, *government support*, *competition schedule*, *etc*.

Levi (1978) also proposed a set of abstract predicates[1] for the former category, but no relation for the latter category. Later, Ó Séaghdha (2007) revised this inventory and proposed a two-level hierarchy of semantic relations.

Ponkiya et al. (2018a) proposed a method to use FrameNet based labels for noun compounds. Here, the head noun invokes a frame, and the modifier noun fits in one of the slots of the frame. They also prepared a dataset by annotating each noun compound with a frame and a frame element. Ponkiya et al. (2018a) proposed this annotation for predicate nominalization, which also works for most cases of predicate deletion.

3.1 FrameNet

FrameNet[2] (Baker et al., 1998) is a taxonomy based on Fillmore's theory of Frame Semantics. This theory claims that most words' meanings can be inferred based on a semantic frame: a conceptual structure that denotes an abstract event, relation, or entity and the involved participants. For example, the concept of questioning involves a person asking a question (SPEAKER), person/people begin questioned ADDRESSEE, the content of the question MESSAGE, and so on. In FrameNet, such a concept is represented by QUESTIONING frame. The participating entities, such as SPEAKER, ADDRESSEE, MESSAGE, *etc*., are called *frame elements* (FEs). Such frames are invoked in running text via words known as *lexical units*. Some of the lexical units for the QUESTIONING frame are *ask, grill, inquire, inquiry, interrogate, query*, *etc*. FrameNet data provides two types of linkages between entities:

(a) relations: linking among frames or among FEs, and

(b) mappings: linking from words to frames and from frames to FEs.

3.1.1 Relations

FrameNet includes a graph of relations between frames along with relations among frames. Some of the important frame relations are:

- **Inheritance:** close to a typical Is-A relation, *e.g.*, PROTEST $\xrightarrow{\text{Is-A}}$ INTENTIONALLY_ACT
- **Using:** the child frame presupposes the parent frame, *e.g.*, PROTEST $\xrightarrow{\text{Uses}}$ TAKING_SIDES
- **Subframe:** the child frame is a subevent of a complex parent event, *e.g.*, TRIAL $\xrightarrow{\text{Subframe}}$ VERDICT

Along with each frame relation, FrameNet also consists of relations between FEs of parent-child frames. Following are illustrative examples for FE relations for the above frame relations:

- PROTEST:PROTESTER $\xrightarrow{\text{Is-A}}$ INTENTIONALLY_ACT:AGENT.
- PROTEST:PROTESTER $\xrightarrow{\text{Uses}}$ TAKING_SIDES:COGNIZER
- TRIAL:JUDGE $\xrightarrow{\text{Subframe}}$ VERDICT:JUDGE

3.1.2 Mappings

FrameNet data provides two types of mappings: (a) words to frames (via lexical units), and (b) frames to FEs. For instance, *protest* word can invoke three frames: PROTEST, POLITICAL_ACTIONS, and JUDGMENT_COMMUNICATION. Similarly, using the frame to FEs mapping, we can list all FEs of the PROTEST frame, which are ACTION, ISSUE, PROTESTER, SIDE, DEGREE, DESCRIPTOR, DURATION, EXPLANATION, FREQUENCY, MANNER, MEANS, PLACE, PURPOSE, TIME, *etc*.

[1] RDP (Recoverable Deleted Predicates)

[2] https://framenet.icsi.berkeley.edu

In our work, we utilize *Relations* for the generation of frame and frame element embeddings (§3.2). We, further, utilize the *Mappings* to prune the search space (§4.1).

3.2 Knowledge Graph Embeddings

A Knowledge Graph $\mathcal{G}$ is a set of relations $\mathcal{R}$ defined over a set of entities $\mathcal{E}$. Formally, it is comprised of a set of N triples (h, r, t), where h and t are called *head* and *tail* entities, and r denotes a relation among them.

$$\mathcal{G} = \{(h, r, t) : h, t \in \mathcal{E}; r \in \mathcal{R}\}$$

Knowledge Graphs are widely used to store knowledge in a structured format, and they play an important role in representation learning. Methods for learning representations for both entities $\mathcal{E}$ and relations $\mathcal{R}$ have been explored (Wang et al., 2017) with an aim to represent graphical knowledge.

Various algorithms for representation learning have been proposed, which help tasks such as link prediction *etc*. TransE (Bordes et al., 2013) is a method that models relationships by interpreting them as translations operating on the low-dimensional embeddings of the entities. We use the ConvE (Dettmers et al., 2018) algorithm to get embeddings of frames and frame elements. For the training of ConvE, we treat all relations from FrameNet as triples of a knowledge graph.

3.2.1 ConvE

Convolution-based **E**mbeddings is a multi-layer 2D-convolution network model proposed by Dettmers et al. (2018). It usages fewer parameters, yet efficient compared to similar models. It defines the scoring function (for each relation r) as follows:

$$\psi_r(e_h, e_t) = f(vec(f([\overline{e}_h; \overline{e}_r]w))W)e_t \qquad (1)$$

where, e_h, e_r and e_t are embeddings of head h, relation r and tail t, respectively, $\overline{x}$ denotes reshaping of vector x to a matrix, f is a rectified linear unit (relu) function, vec converts a matrix into a flat vector, w is convolution kernel, and W is the parameter of a fully connected layer.

For training, it applies logistic sigmoid function $\sigma(\cdot)$ to the scores, and minimize the binary cross-entropy computed using the following formula:

$$L_{CE} = -\frac{1}{N}\sum_i (t_i \cdot \log(p_i) + (1 - t_i) \cdot \log(1 - p_i)) \qquad (2)$$

where, $p = \sigma(\psi_r(e_h, e_t))$ and t is 1 when $(h, r, t) \in \mathcal{G}$, 0 otherwise.

ConvE uses two embedding layers: one for entities and the other for relations, which initializes the embeddings layers randomly. The embeddings layers get updated during the training. At the end of the training, the embedding layers contain the embeddings for entities and relations.

4 Our Approach

FrameNet has 1223 frames and 11,473 frame elements. However, the existing dataset for FrameNet-based noun compound interpretation does not have examples for many frames and frame elements. However, unlike other relation inventories, we have FrameNet taxonomy, which can help in building a better model. We first explain our frame prediction approach and then extend the same for FE prediction.

4.1 System Architecture

We encode a given noun compound $nc = w_1\ w_2$ (say, *divorce rate*) using a feed-forward network to get vector v_{nc}. Using FrameNet API, we create a set of candidate frames that can be invoked by w_2 (*rate* → {Assessing, Proportion, Speed_description, etc.}). For each candidate frame f_i, we take its frame embedding e_{f_i} from the frame embedding layer. We take the dot product of v_{nc} with embedding e_{f_i} of each frame f_i to compute the score for the frame. For testing, we use the following formula to predict a frame:

$$f^* = \text{argmax}_i\ v_{nc} \cdot e_{f_i} \qquad (3)$$

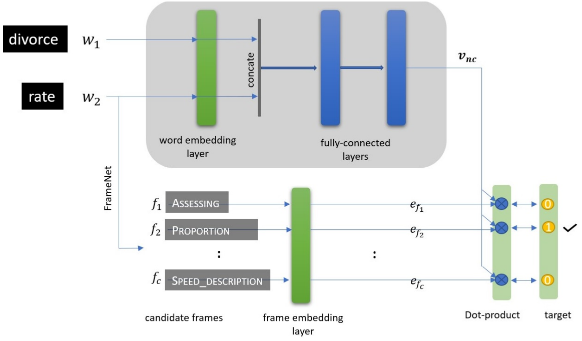


Figure 1: Basic system architecture illustrating frame prediction for *divorce rate*.

Dima and Hinrichs (2015) and Fares et al. (2018) use a simple feed-forward network. In our model, if we remove the frame/FE embedding matrix and use them as weights of one more dense layer (after the "fully-connected layers"), *our model becomes identical to theirs*. In doing so, **(a)** our model will NOT need any extra computation to compute the score of labels that are NOT part of the candidate set, and **(b)** the back-prorogation does not have to pass through an additional layer which might not be effective.

We implement these models in PyTorch (Paszke et al., 2017). We initialize the word embedding layer with Google's pre-trained embeddings[3] and initialize the frame embedding layer with random values, in one case, for baseline, and pre-trained frame embedding, in another case.

We use the same architecture to train another model for FE prediction, replace the frame embedding layer with an FE embedding layer, and candidates FEs are the FEs from all candidate frames. We take all FEs as a candidate set if no such mapping is found.

4.2 Frame and Frame Element Embeddings

Inspired by Kumar et al. (2019)'s approach for the task of Word Sense Disambiguation (WSD), we propose a similar approach to perform NC interpretation. Our approach uses the definition of entities (along with the relations) to learn entity embeddings and relation embeddings. It uses an encoder (Bi-LSTM) to encode the definition of an entity and uses encoded representation as an embedding of the entity for ConvE. During the training, it also optimizes both: the encoder and ConvE. After the training, the encoding of definitions is taken as entity embeddings.

We train ConvE twice to get frame and frame element embeddings separately. ConvE training is independent of the main training.

5 Experimental Setup

In this section, we explain our dataset, baseline, training, and evaluation metrics.

5.1 Dataset Creation and Analysis

We use the dataset released by Ponkiya et al. (2018a) as D1. The dataset contains 1546 noun-noun compounds with two labels: frame and FE. The dataset was created by extracting noun compound along with labels from the FrameNet data. As the extraction is automatic and the manual step only confirms the correctness of the labelling, the labels are not exhaustive. For instance, a noun compound *student demonstration* has been annotated with PROTEST:PROTESTER. However, the following labels are also applicable: REASONING:ARGUER and CAUSE_TO_PERCEIVE:ACTOR. So, we annotate more examples with all possible labels.

Manual Annotation

We manually annotate 326 noun compounds, and call it D2. We extend D1 by merging these examples from D2 to perform our experiments. The annotation is performed by one of the authors and hence does not warrant discussion on the inter-annotator agreement. However, please allow us to point out that our annotations are still manually performed by a human, which begets the consideration of these annotations to be gold-standard. The author chose the examples from Tratz and Hovy (2010)'s dataset randomly. During the annotation process, we found some difficulties because of the coverage issue of the FrameNet. The word-to-frame mapping in FrameNet has a coverage issue, and it has been widely reported in the literature (Pavlick et al., 2015; Botschen et al., 2017).

We categorize the coverage issues into the following:

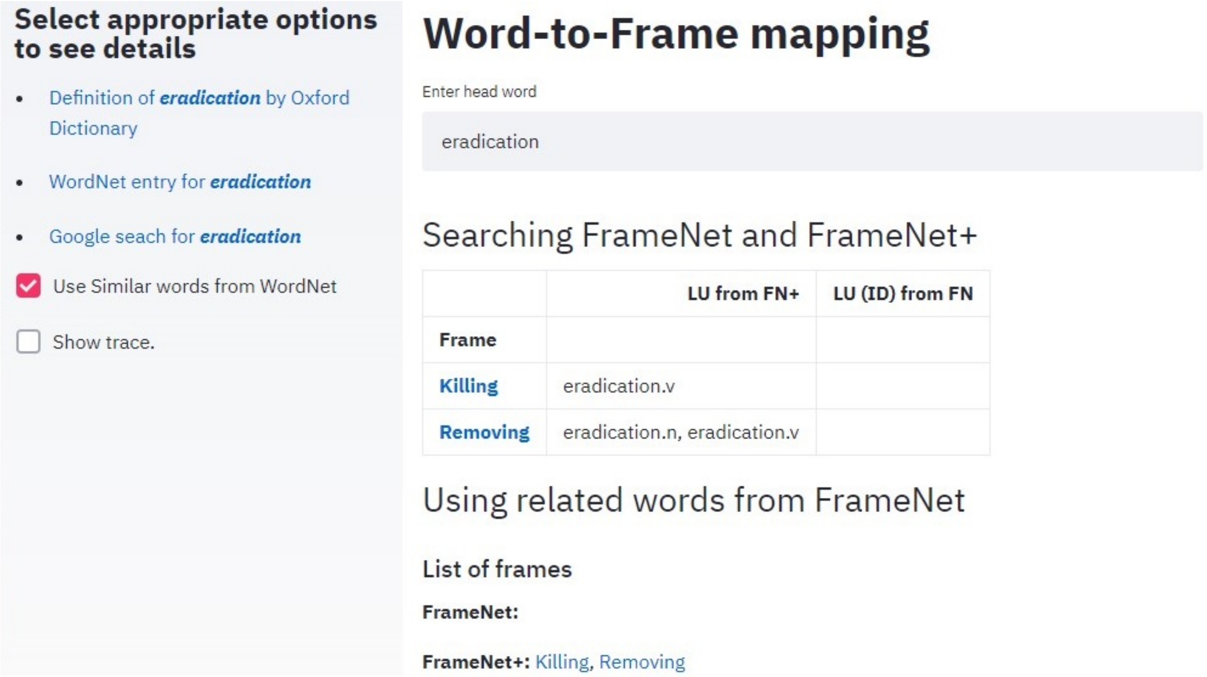
No Candidate Frames: The word-to-frame mapping returned no candidate frame. In some cases, we could find a frame with manual effort (ref. Table 1). However, despite manual efforts, some cases, we could not find an appropriate frame all the time (*e.g.*, *star autograph*, *employee misconduct*, *etc*.).

No Suitable Frame in the Candidate Set: In this set, word-to-frame mapping retrieved candidate frames, but none of the candidates was found to be appropriate. For example, candidate frames for *heat* returned by the mapping are: CAUSE_TEMPERATURE_CHANGE

Word	Manually extracted frames
funding	FUNDING
authorities	LEADERSHIP, AUTHORITY
eradication	REMOVING
harvesting	FOOD_GATHERING
analyst	PEOPLE_BY_VOCATION

Table 1: Examples where FrameNet data does not contain an appropriate mapping but we manually find suitable frames.

[3] https://code.google.com/archive/p/word2vec/



Select appropriate options to see details

- Definition of ***eradication*** by Oxford Dictionary
- WordNet entry for ***eradication***
- Google seach for ***eradication***

☑ Use Similar words from WordNet

□ Show trace.

Word-to-Frame mapping

Enter head word

eradication

Searching FrameNet and FrameNet+

	LU from FN+	LU (ID) from FN
Frame		
Killing	eradication.v	
Removing	eradication.n, eradication.v	

Using related words from FrameNet

List of frames

FrameNet:

FrameNet+: Killing, Removing

Figure 2: Tool interface to assist annotator in finding a frame. FrameNet does not have mappings for *eradication*, however FrameNet+ maps it to KILLING and REMOVING frames.

and CHANGE_OF_TEMPERATURE. However, none of the two frames is appropriate for *body heat*.

In some cases, we could find an appropriate frame that was not a part of the candidate set. For example, for noun compound *ulcer drug*, candidate frames are INTOXICANTS and CAUSE_HARM, but the appropriate frame is CURE.

No Suitable Frame Element in the Frame: We could find an appropriate frame, but no frame element from the frame is appropriate. For instance, the BUSINESS frame is suitable for *retail operation*, but no frame element from the frame is suitable for the modifier noun *retail*.

NC Annotation Tool

To handle the first two cases (finding of a frame), we use synonyms from WordNet (Miller, 1994) and FrameNet+ data (Pavlick et al., 2015). To simplify the annotation process, we develop a tool (Figure 2) that makes the annotation process easier.

We split each dataset – D1 and D1+D2 – randomly for 5-fold validation. Each fold contains three disjoint sets: training set (60% compounds), validation set (20% compounds), and test set (20% compounds). We use the same folds across all experiments, so results across different models are comparable.

5.2 Frame and Frame Element Embeddings

To get frame embeddings, we consider frames as entities and frame relations from FrameNet as relations between the entities. Then we train ConvE (§4.2) to learn frame embeddings. We use these entity embeddings to initialize the frame embedding layer. Table 2 shows the ten most similar frames for FRIENDLY_OR_HOSTILE frame based on cosine similarity between frame embeddings. Similarly, we get embeddings of frame elements using frame elements and their relations in FrameNet.

5.3 Baseline

The first baseline is a **random prediction**: the probability of predicting a label from a candidate set is uniform. We take expected counts to compute metrics. For instance, we compute random accuracy using the following formula:

$$\text{Acc}_{\text{random}} = \frac{1}{N} \sum_{nc_i \in \text{Test-set}} \frac{1}{|\text{candidates}(nc_i)|} \quad (4)$$

We also use **Support Vector Machines** (SVM) (Cortes and Vapnik, 1995) as a baseline approach for this task. We use the *sklearn* library (Pedregosa et al., 2011) with default parameters for

#	Score	Name	Definition
0	1.0000	FRIENDLY_OR_HOSTILE	Two parties or individuals are on the same sid...
1	0.9763	BE_IN_AGREEMENT_ON_ACTION	Two (or more) people (the Parties, also encoda...
2	0.9749	MAKE_AGREEMENT_ON_ACTION	Two (or more) people (the Parties, also encoda...
3	0.9748	DISTRIBUTED_POSITION	This frame involves a static (primarily spatia...
4	0.9743	TERMS_OF_AGREEMENT	A condition that is set out in an Agreement im...
5	0.9742	WHOLES_AND_PARTS	A Whole is made up of multiple Parts, which th...
6	0.9737	MAKE_COMPROMISE	Two (or more) people (the Parties, also encoda...
7	0.9729	SOCIAL_EVENT_INDIVIDUALS	This frames describes a social event where the...
8	0.9725	CO-ASSOCIATION	Two or more individuals have a relationship by...
9	0.9725	PUBLIC_SERVICES	This frame concerns permanent organizations (t...
10	0.9716	BE_IN_AGREEMENT_ON_ASSESSMENT	The Cognizers have a similarity (or dissimilar...

Table 2: 10 most similar frames for FRIENDLY_OR_HOSTILE frame using frame embeddings.

this approach. The input for the SVM-based approach is the concatenated vector of individual lexical units.

We provide results for **another baseline** approach where we use the same architecture (§4.1) with random initialization for frame/FE embeddings.

5.4 Training

Given a noun compound, we get candidate labels using FrameNet mapping. We compute scores for candidate labels and compare them with the target to compute loss value. We minimize categorical cross-entropy with stochastic gradient descent (with momentum). Frame/FE embedding layer remains fixed (non-trainable) for the initial few epochs. For stopping criteria, we monitor performance on the validation set.

5.5 Evaluation

We report weighted **P**recision, **R**ecall, and **F1**-score for our experiments. The weight values for each label is in proportion to the number of test examples for the label. Following is a formula for computing (weighted) precision:

$$\text{Precision} = \sum_{l} P_l * \frac{N_l}{N} \tag{5}$$

$$P_l = \frac{\text{TP}_l}{\text{TP}_l + \text{FP}_l} \tag{6}$$

where, P_l is the precision score, TP_l is the number of true-positives and FP_l is the number of false-positives for a label l. N_l is the number of instances with label l in the test set, and N is the total number of instances in a test set.

The above metrics are based on the top prediction. We also report accuracy at k, which treats a prediction as a true prediction if the correct label is in top k predicted labels. We report (micro-averaged) accuracy, computed using the following formula:

$$\text{Acc} = \frac{\text{No. of correctly classified instances}}{\text{Total instances in test-test}} \tag{7}$$

We compute all of these metrics for frame and frame element independently. We use the Scikit-learn library (Pedregosa et al., 2011) to compute all of these metrics.

6 Results and Analysis

The reported results are averaged across 5-fold cross-validation. We define a subset of all test samples, whose output label does not have any samples in the training set, as *unseen*-set. In a fold (of D1) with 310 test samples, the following are the statistics about the "unseen set": (1) 30 unique frames, covering 32 test samples, (2) 75 unique FEs, covering 82 test samples. *These cases are challenging to handle.* Our prediction in continuous space helps in such cases, as the target space embeds the labels.

Table 3 reports the performance of the baselines compared to our system for frame prediction on the entire test-set. Our system beats the random baseline and SVM model across all metrics by a significant margin. Similarly, as observed from Table 4, it can be seen that frame prediction on the "unseen" set improvises over the random baseline in both the cases, *viz.,* WITH and WITHOUT frame embeddings. The performance improvement is quite significant over both the datasets (D1 and

	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)

D1+D2). We attribute the performance improvement in both cases to the fact that our model captures frame semantics for a frame in a continuous space, better than the baseline metrics. However, as seen in Table 3, the model which tries to predict WITHOUT frame embeddings shows comparable results with the model which predicts WITH frame embeddings. This marginal improvement does not seem to be significant, and the results for both these cases are almost similar. Eventually, with the improvement in dataset size, we expect the system with frame embeddings to perform better than the system without frame embeddings.

We discuss the task of frame prediction above and present our results for frame element prediction here. As it can be observed from Table 5, random baseline and SVM based models are outperformed for the task of frame element prediction as well, when compared with the results of our methodology. The improvements over both datasets (D1 and D1+D2) are at least 5% (D1+D2/SVM). With the extended dataset (D1 + D2), the performance of our approach shows an improvement across all three (precision, recall and f-score) measures. In Table 3, for frame prediction, we observe an increase in the stronger baseline score (SVM) when the extended dataset is used. However, we observe that our results for frame element prediction show that the model which uses frame embeddings is significantly outperformed by the model which does not use frame embeddings. Upon manual analysis of our train set, we find that our datasets have multiple cases where the number of examples per frame element is very few (sometimes even 1, as discussed below). This results in a data skew where the sample would either be used for training or testing, thus rendering the model either untrained for that test case, or no testing of the model trained for that single frame element. In Table 6, we see that the random baseline outperforms our method because of the data skew discussed here. There are multiple frame elements that are present in the unseen test data for which the model has not been trained at all. We do not report SVM performance in Table 6 and Table 4, since the precision, recall, and f-score for SVM were all 0. This performance can be attributed to the fact that SVM does not perform well with unseen examples, and in this case, does not perform at all.

For frame element prediction, in Table 5, we observe that the extended dataset helps improve the over quality of predictions with an improved score for each approach, including the baseline. These results signify that the dataset extension does indeed help the task of NC interpretation.

In Figure 3, we see that the average number of candidate FEs for a test sample is only 26. Without even a single training example, our system is able to correctly predict 51.22% (more than half) of unseen samples among the top-7 predictions, which is higher than the top-1 accuracy of any baseline approach on the complete test-set. With an increase

	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)

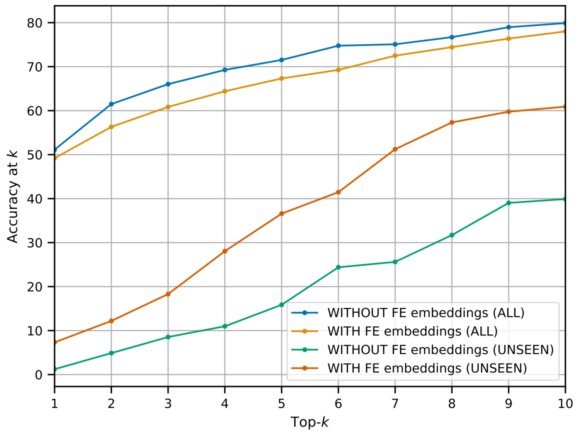


Figure 3: Accuracy@k for frame element prediction (WITH and WITHOUT frame element embeddings; on full test set and *unseen*-set)

in k, the margin between the performance of our system and baseline remains nearly the same on the whole test set. However, on unseen labels, the system with FE embeddings significantly outperforms the baseline, with an increase in k.

Overall, we observe a significant improvement in results with the help of our method. We also show that our extended dataset does help improve the performance of models in both cases.

7 Conclusion and Future Work

In this paper, we proposed a novel method for using FrameNet for NC interpretation in a continuous space. We use FrameNet mappings (word to frame and frame to frame element) to prune our search space. Our approach – prediction in continuous space – outperforms the random baseline and a stronger baseline approach. We show that the label embeddings generated using our approach help in the generalisation over unseen labels.

We annotated more noun compounds and analysed the issue in finding frame and frame elements. We create and release a tool that assists annotators in frame identification, for further research. We also show that extending the dataset created with our tool improves the system performance. Our experiments evaluate our proposed method on a small annotated dataset compared to the overall number of labels. We extend this existing dataset by annotating more NCs for various labels. Our study on the coverage issue for the annotation process helps develop a tool that assists the annotators in finding an appropriate frame. We provide promising results for the task of frame prediction. We analyse our results and discuss them in detail with respect to both frame and frame element prediction tasks.

In the future, we aim to find other ways of using FrameNet data for this task. We would also like to investigate why our approach provides promising results for the task of frame prediction but not for frame element prediction. We would like to explore more approaches to predict the frame elements effectively. We believe that using FrameNet embeddings can prove to be helpful for other tasks.

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*.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *COLING-ACL '98: Proceedings of the Conference*, pages 86–90, Montreal, Canada.

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.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795.

Teresa Botschen, Hatem Mousselly-Sergieh, and Iryna Gurevych. 2017. Prediction of frame-to-frame relations in the FrameNet hierarchy with frame embeddings. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 146–156, Vancouver, Canada. Association for Computational Linguistics.

Cristina Butnariu, Su Nam Kim, Preslav Nakov, Diarmuid Ó Séaghdha, Stan Szpakowicz, and Tony Veale. 2009. SemEval-2010 task 9. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions - DEW '09*, pages 100–105. Association for Computational Linguistics.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

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

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. 2018. Transfer and multi-task learning for noun–noun compound interpretation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1488–1498.

Charles J. Fillmore. 1976. Frame semantics and the nature of language. *Ann NY Acad Sci*, 280(1 Origins and E):20–32.

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.

Sawan Kumar, Sharmistha Jat, Karan Saxena, and Partha Talukdar. 2019. Zero-shot word sense disambiguation using sense definition embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5670–5681. 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.

George A Miller. 1994. Wordnet: An electronic lexical database.

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

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

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*.

Ellie Pavlick, Travis Wolfe, Pushpendre Rastogi, Chris Callison-Burch, Mark Dredze, and Benjamin Van Durme. 2015. FrameNet+: Fast paraphrastic tripling of FrameNet. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 408–413, Beijing, China. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Girishkumar Ponkiya, Rudra Murthy, Pushpak Bhattacharyya, and Girish Palshikar. 2020. Looking inside noun compounds: Unsupervised prepositional and free paraphrasing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4313–4323, Online. Association for Computational Linguistics.

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.

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.

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.

Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Trans. Knowl. Data Eng.*, 29(12):2724–2743.

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

Diarmuid Ó Séaghdha. 2007. Annotating and learning compound noun semantics. In *Proceedings of the 45th Annual Meeting of the ACL: Student Research Workshop on - ACL '07*, pages 73–78. Association for Computational Linguistics.