

Graph Embeddings for Frame Identification

Alexander Popov

IICT, Bulgarian Academy of Sciences
Sofia, Bulgaria
alex.popov
@bultreebank.org

Jennifer Sikos

IMS, University of Stuttgart
Stuttgart, Germany
jen.sikos
@ims.uni-stuttgart.de

Abstract

Lexical resources such as WordNet (Miller, 1995) and FrameNet (Baker et al., 1998) are organized as graphs, where relationships between words are made explicit via the structure of the resource. This work explores how structural information from these lexical resources can lead to gains in a downstream task, namely frame identification. While much of the current work in frame identification uses various neural architectures to predict frames, those neural architectures only use representations of frames based on annotated corpus data. We demonstrate how incorporating knowledge directly from the FrameNet graph structure improves the performance of a neural network-based frame identification system. Specifically, we construct a bidirectional LSTM with a loss function that incorporates various graph- and corpus-based frame embeddings for learning and ultimately achieves strong performance gains with the graph-based embeddings over corpus-based embeddings alone.

1 Introduction

Frames are common scenarios, expressed by their typical participants and the predicates which evoke them. The INFECTING frame, where someone transmits an illness, has participants INFECTED_ENTITY and INFECTION_CAUSE and is evoked by predicates *infect* and *give*. A single predicate can evoke one or more frames; *give*, for instance, can evoke INFECTING (*give someone a cold*) or GIVING (*give someone a present*). The disambiguation of which frame is evoked in context is the task of *frame identification* (FrameID),

which is, in essence, a word sense disambiguation task where the senses are frames (Das et al., 2010).

The FrameNet resource (Baker et al., 1998) provides concrete definitions of frames, predicates, and participants (called *frame elements*) and is structured as a hierarchical graph. Frames towards the top of the hierarchy are more abstract (ex. INTENTIONALLY_ACT), while frames lower in the hierarchy are more granular (ex., SUBMITTING_DOCUMENTS). FrameNet's graph structure captures the relationship between frames, their predicates, and their frame elements, such that a single frame can be connected to multiple frames via different relationships (Baker and Ellsworth, 2017). Given the structure of the resource, it is surprising that the graph is not used in current FrameID systems, as the relationships in the FrameNet graph have been previously shown as relevant to frame prediction, especially in cases of unseen predicates (Das et al., 2014).

This work leverages FrameNet's graph structure to boost the performance of neural architectures for FrameID. Specifically, we construct frame embeddings from the FrameNet graph structure and use them as input to a neural network. Although frame embeddings are often used in neural architectures for FrameID, prior work only learns these embeddings from frame-annotated corpus data (Hartmann et al., 2017; Hermann et al., 2014). To our knowledge, this is the first work to incorporate embeddings composed from the FrameNet graph structure itself, thus incorporating all frame, frame element, and predicate relationships that could otherwise be missing from corpus data. We expand our frame graphs with knowledge from another lexical resource, WordNet, to achieve further gains in performance.

Our paper is structured as follows. Section 2 describes prior work in graph embedding models and FrameID. We define our model in Section 3, and

Section 4 focuses on the FrameNet and WordNet graphs, while Sections 5 and 6 explain how we use the FrameNet knowledge base to build graph embeddings. Section 7 outlines our experiments, and Section 8 gives results and further analysis of the model performance.

2 Background

2.1 Graph Embeddings

Graph algorithms, especially random walk algorithms, have been applied to prediction tasks such as word sense disambiguation (Agirre et al., 2014), measuring semantic similarity and relatedness between words (Agirre et al., 2010), and entity linking (Guo et al., 2011). These algorithms traverse over the relations and nodes in a large knowledge base such as WordNet (Miller, 1995) or taxonomies built from Wikipedia links (Cucerzan, 2007) to uncover relationships between the nodes in the knowledge base. However, the effectiveness of shallow neural networks in learning word similarity, as shown by the popular Word2Vec and GloVe models (Mikolov et al., 2013; Pennington et al., 2014), rapidly replaced traditional graph-based methods for word similarity and prediction tasks. Word representations learned in these models, called *embeddings*, provide the latent features of a word in context as low-dimensional vectors. Recent work has sought to incorporate the best of graph algorithms and embeddings by learning representations for words over a neural network while using the structural information found in knowledge bases (Goikoetxea et al., 2015; Faruqi et al., 2015).

Our method for constructing graph embeddings follows the work of Goikoetxea et al. (2015) who use random walks to build a synthetic corpus based on entries in WordNet. In this corpus, a word’s contexts are the other words in the knowledge base that it is related to. They use the corpus to generate embeddings for words with CBOW and Skipgram models.

2.2 Neural Architectures for Frame Identification

The first work to use embeddings for FrameID used the WSABIE algorithm to project frames and predicate contexts into the same shared space (Hermann et al., 2014). The authors then apply a pairwise loss to minimize the distance between the frames and their predicate instances.

Subsequent FrameID models followed, including a system that constructed frame embeddings using the Word2Vec model (Botschen et al., 2017). More recent state-of-the-art models use contextualized embeddings of frames in the BERT framework (Sikos and Padó, 2019) or joint models with semantic dependencies and frames (Peng et al., 2018). All of these prior neural architectures construct frame embeddings directly from annotated corpora, meaning the frame embeddings that are used to make predictions in the model are limited to frames that are seen in the corpora.

3 Frame Identification Model

This section describes the overall architecture of our FrameID model. We adopt a bidirectional recurrent neural network (Bi-LSTM) that accepts as input different embeddings that represent frames and predicates. This allows us to measure the effectiveness of the pre-trained embeddings (corpus-based, graph-based and combined) for an apples-to-apples comparison within the same setting. The components of our neural network include:

- an input embedding layer; the network can allow two embedding inputs per word – the embeddings from separate sources (corpus/graph-based) are concatenated;
- bidirectional recurrent layers; the forward and backward states are concatenated;
- an output vector with the same dimensionality as the frame embeddings;
- an objective function that compares the final output and the gold frame embeddings; at those steps that do not introduce frame-evoking predicates the network attempts to reconstruct the embedding for the input word itself, so that “infect” should result in the embedding of INFECTING, but “the” should just be reconstructed to the embedding of “the”.

At evaluation time, output vectors are compared via cosine similarity to the frame embeddings, where the label for the closest frame is selected as the model’s prediction. Figure 1 shows the architecture of the network.

4 FrameNet as a Graph

As described above in Section 1, FrameNet has a structure which connects frames via different semantic relations. Relations in the knowledge base include *Inheritance*, an *is-a* relation akin

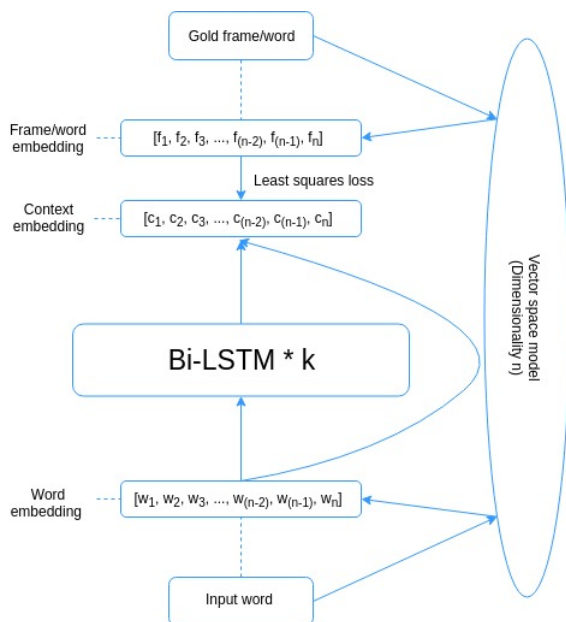


Figure 1: Bi-LSTM for FrameID. The diagram shows a model with one embedding layer, but combining with an additional embedding source is done trivially via concatenation. The embeddings used for the input and the gold frames can either come from the same or a different vector space. The size of the word/frame embeddings is n ; k is the number of hidden layers.

to hypernymy in WordNet (e.g., REVENGE inherits from REWARDS_AND_PUNISHMENTS), *Using*, which connects a concrete frame to a related, but more abstract frame (e.g. SPEED uses the MOTION frame), and *Subframe*, where the parent frame is a more complex event that subsumes the child frame(s) (e.g. CRIMINAL_PROCESS subsumes ARREST). See Ruppenhofer et al. (2006) for a more detailed description of all possible relation types.

This network of relations is used as the basis of our graph. We construct a graph where each frame is a node, and we build edges between frames according to the available relational information in FrameNet. Frame elements (FEs) are also nodes that are connected to their frames. Some FE nodes, such as AGENT, are widely connected in the graph since they are linked to a diverse set of frames. In order to have a single FE node be distinctive to a single frame, the FE nodes are given unique identifiers. For the unique identifier, the frame name is prepended to the name of the FE together with a colon separator, such as HIRING::EMPLOYEE. However, these unique FE nodes are further con-

nected to their abstract counterparts with an *is-a* relation (REVENGE::AVENGER *is-a* AVENGER), which preserves the connectedness of the FE in FrameNet and also provides some connectivity between frames that share the same FE.

4.1 Extending the FrameNet graph

We empirically find that, on its own, the graph described above does not form a dense enough graph structure. This is due to the fact that few frames can be connected through short paths since some frames contain little to no relations to others and FrameNet FEs are often very specific to particular types of scenarios.

To overcome this limitation, we mapped FrameNet to the WordNet lexical resource. WordNet groups synonymous words of the same part-of-speech category into concept clusters called *synsets* and provides lexical relations between word senses and the relations between synsets. Its dense semantic network provides a rich representation of the ecology of lexical meaning, especially when associated resources are brought into play. This allows for connecting concepts that might not be linked via lexico-semantic relations but nevertheless are mutually dependent due to contextual co-occurrence.

Mapping FrameNet predicates with the Predicate Matrix The mapping process is done in two steps. First, the verbal predicates in FrameNet are connected to their corresponding synsets in WordNet. Since one word can be mapped to several senses in WordNet, this mapping is far from straightforward. We map FrameNet predicates and WordNet synsets through an auxiliary resource called the Predicate Matrix (De Lacalle et al., 2014), which aimed to automatically extend the predicate mappings from Sem-Link (Palmer, 2009). Within the Predicate Matrix, predicates from a FrameNet frame are linked to concept nodes, which are also connected to a corresponding WordNet synset. Via these synsets, we now have access to WordNet’s semantic network, where words are interconnected through lexico-semantic relations such as hypernymy, antonymy, meronymy, as well as through other relations expressing relatedness.

Lexical fillers for FrameNet FEs via Wikipedia and BabelNet The second step in the mapping process is the automatic expansion of FrameNet by extracting typical fillers for FrameNet FEs.

For instance, SELLER is strongly linked to lexical items such as *cashier* or *salesman*, both available in WordNet. While the Predicate Matrix aligns the predicates in FrameNet to WordNet synsets, we additionally align these candidate fillers from WordNet to the FEs they instantiate in FrameNet. Aligning FrameNet’s FEs with WordNet is made possible by an automatically created extension of FrameNet (Bryl et al., 2012). The method used by the authors relies on a machine learning model linking nominal lexical fillers for the FEs in the authors’ FrameNet reference corpus to Wikipedia pages. The BabelNet resource (Navigli and Ponzetto, 2010) provides mappings between Wikipedia pages and the corresponding WordNet synsets, which are mapped to the frame-FE tuples. Because the process is automated, the correspondences are inevitably noisy. However, it provides a significant expansion of words are not directly given by the FrameNet lexicon but nevertheless intuitively related to the concepts in the frame; e.g., a *cashier* and *salesman* are implicitly related to the COMMERCE_SELL frame. In this sense, this step incorporates more world knowledge into a frame’s graph.

5 The FrameNet Graph as a Pseudo-Corpus

Once the nodes in FrameNet and WordNet are aligned, the resulting graph structure is leveraged to produce a large corpus of artificial sentences on which a vector space model can be trained. We follow the methodology of Goikoetxea et al. (2015) for the generation of a pseudo-corpus, where each line contains a sequence of node identifiers visited during one random walk of the algorithm as it moves along the graph. This is done with the UKB tool¹ and its “ukb_walkandprint” functionality. The *wemit_prob* parameter is set to 0.5, which means that in half the cases it emits a dictionary key associated with the node ID generated along the random walk, i.e., half the “words” in the artificial sentences are either predicates or frames corresponding to synsets, as established in the mapping process.

The UKB tool uses a particular dictionary format for the creation of pseudo-corpora. Each line is constructed as follows: a key, often a predicate, begins the entry, followed by its associated WordNet synsets. The result looks like the following:

¹<http://ixa2.si.ehu.es/ukb/>

```
employment 13968092-n:9 00584367-n:9
01217859-n:6 00947128-n:1
```

where the synset number is given along with a letter after the synset (“-n”), which signifies the POS category and the number (“:9”) is the number of instances of this specific sense found in a reference corpus. The *ukb_walkandprint* functionality requires this dictionary for the emission of lexical items from the visited nodes along the random walks. In the case when *wemit_prob* is set to a non-zero value, the tool will pick, with probability *wemit_prob*, one dictionary key associated with the current WordNet synset. The dictionary provides information on what items belong to which synset. If the *dict_weight* parameters are provided, the tool will also take into consideration the count-based weights for each predicate-synset pairing and emit a predicate according to the available probability distribution.

Incorporating FrameNet frames into the UKB dictionary

Because the graph now includes frames and FE relations, which are outside of the WordNet lexicon, those need to be included in the dictionary as well. Therefore, we add the frame IDs to the dictionary. This is done in two ways; first, the frames are added as keys corresponding to the synsets to which they have been mapped - below, the frame RENTING is mapped:

```
Renting 02208537-v:1 02460619-v:1 02208903-
v:2
```

Second, frames are added as values of lexical items that evoke them, as per the information encoded in the FrameNet database. For instance, the predicate *hire.v* is evoked by the RENTING frame:

```
hire.v Renting:0 02208537-v:0 02460619-v:1
02409412-v:33 Hiring:342
```

A frame ID can be emitted whenever a connected synset node is visited during a random walk, or if a frame node is visited, it might emit a lexical item connected to the frame as a key in the dictionary. Thus the pseudo-corpus includes both references to specific frames and frame-evoking expressions whose use is contextualized with respect to particular frames. The weights for the frames in the dictionary are calculated by summing up the weights

²Note that because we are interested in obtaining representations for lexical units (i.e. *lemma.pos* items), the dictionary is modified to include morphological information per each key.

for the lemma-synset pairs mapped to the frames-as-values.

Augmenting the Graph with Wikipedia

With the graph and dictionary in place, *ukb_walkandprint* is used to generate 200 million random walks. This pseudo-corpus is then concatenated to a lemmatized and POS-tagged Wikipedia dump (where each word is transformed to a *lemma.pos* lexical item), so that the embeddings can be trained on natural language text in addition to the graph-generated artificial sentences. This is a naive method to combine graph and text information, in that the model is never trained on both kinds of information in one and the same sentence, but it nevertheless allows it to encode real-text syntagmatic information within the lexical space projected from the graph structure. The Wikipedia data is approximately the same size as the pseudo-corpus.

6 Frame Embeddings

Graph embeddings of frames We now have a pseudo-corpus composed from the FrameNet/WordNet graph structure, as described in Section 5. These graph representations are constructed for all the frames in the FrameNet lexicon. We use the popular Word2Vec tool³ to generate embeddings from the pseudo-corpus. This model uses negative sampling to learn word representations, and we apply the Skipgram variant of the model where a single word is used to predict its neighboring terms in the immediate context. This produces a large inventory of graph embeddings for frames in FrameNet and lexical items, all located in the same space.

Corpus embeddings of frames To compare the graph embeddings with frame embeddings learned from a corpus, we took the freely available embeddings from Sikos and Padó (2018), in which the authors generated frame embeddings using the Word2Vec tool. Specifically, the authors took lemmatized sentences from the FrameNet annotated data and replaced each predicate (e.g., **say.v**) with the frame it evokes:

FrameNet sentence	Officials say.v he left
Frame evoked	STATEMENT
Embedding Input	official STATEMENT he left

³<https://code.google.com/archive/p/word2vec/>

This resulted in sentences of frames in context, so the example input above would not only produce an embedding for the frame STATEMENT, but also embeddings for the context words such as *official*. In addition to training embeddings on this frame corpus, the authors also provide embeddings based on the original FrameNet annotated corpus. This is important in our case, as we can use the embeddings trained on the original corpus at the input step and the embeddings trained on the frames corpus at the output.

7 Experiments

7.1 Model Setup

The inputs to our model are frame and lexical unit embeddings with 300 dimensions, and the Bi-LSTM has 1 recurrent layer. Each LSTM cell is of size 200, with 0.2 dropout applied to all its sublayers during training. We use the Adam optimizer with a least squares loss function. The training sentences are presented in batches of 128.

The graph that we use for generating the training data for the embedding models consists of 129,101 nodes and 1,146,508 edges. The nodes correspond predominantly to WordNet synsets, with the rest being frame and FE IDs. The synset and FE IDs can be part of the pseudo-corpus or can be omitted from it, depending on the parametrization of the *ukb_walkandprint* command. Depending on this choice, the embeddings will either contain representations of synsets and FEs, or will not, since they are not present as keys in the UKB dictionary.

The graph embeddings were produced by the Word2Vec tool in the Skipgram variant, which has many hyperparameters that can be tuned. We use the following hyperparameters to generate the graph embeddings: size=300; window=15; sample=1e-7; negative=5; iter=7.

7.2 Datasets

We use the annotated data from the FrameNet v1.5 full text annotations to train our models. Sentences are drawn from the balanced BNC corpus⁴, and overall there are over 11k frame-evoking predicates with their frames manually annotated. Standard training, development and test splits are defined by Das et al. (2014), where 39 documents are used in training with over 15k target predicates, 16

⁴<http://www.natcorp.ox.ac.uk/>

Model	Full Lexicon	Ambiguous	No Lexicon	Unseen
Das et al. (2014)	83.60	69.19	-	23.08
Hermann et al. (2014)	88.41	73.10	-	-
Botschen et al. (2018)	88.82	75.28	81.21	-
I:GloVe & O:FN corpus embeddings	84.76	67.41	63.94	12.36
I&O:Graph embeddings (no FE & synsets)	83.27	64.39	46.29	9.96
I&O:FN corpus embeddings	85.89	69.93	67.82	11.25
I&O:Graph embeddings	86.06	70.29	73.71	19.93
I&O:Graph embeddings + FN corpus embeddings	87.03	72.48	77.15	30.44

Table 1: Evaluation of FrameID with different frame embeddings. The table provides results obtained with the same architecture, but with different input and output embedding models; *I* stands for *input embedding model* and *O* – for *output embedding model*; + means concatenation of vectors from two embedding models.

documents for development with over 4,500 target predicates, and 23 documents for testing with 4,458 target predicates.

7.3 Evaluation Metrics

Standard FrameID systems are evaluated via several different metrics, and we evaluate our models on the most common types. “Full Lexicon” evaluation uses knowledge of the FrameNet lexicon, so that for each predicate we classify the most likely frame given the list of all possible frame candidates for that predicate. “Ambiguous” only runs evaluation on predicates that can evoke multiple frames, thus making the evaluation more challenging by removing predicates that can only evoke one frame. “No Lexicon” reports only results from the classifier, where all frames in the FrameNet database are considered possible candidates. Finally, “Unseen” is the most challenging, as it only evaluates predicates that were not seen in the training data and also does not incorporate knowledge from the lexicon.

8 Results

Results of our models are given in Table 1. The best performing model combines graph-based embeddings with corpus-based embeddings for frame prediction, where strong gains are seen over the basic model that uses corpus-based embeddings alone. “Unseen” results in particular show the largest performance gains, where the combined graph- and corpus-based embeddings allow the model to generalize over new predicates.

The model that uses the popular GloVe embeddings (Pennington et al., 2014) at the input step performs worse than the model that uses the FN

corpus embeddings at input and output. This indicates that learning the mapping between predicates and frames is much easier if their corresponding representations are drawn from identical data. The graph-based embeddings are somewhat better than those based solely on the FrameNet annotated corpus; the difference is especially pronounced in the evaluations without the lexicon. However, the graph embedding model that does not incorporate WordNet synsets and FrameNet FEs seems to perform the worst, indicating that those resources provide a strong conceptual skeleton for situating the embeddings of frames and lexical items. What is clear from the results, then, is that both corpus-based and graph-based embeddings contribute to frame prediction and that a richer structure underlying the graph-based embeddings is crucial for improved accuracy.

Our model underperforms compared to other embedding frameworks from Hermann et al. (2014) and Botschen et al. (2018), which can be explained through an examination of the input representation methods used by the different models, as well as their disambiguation strategies. The model by Hermann et al. (2014) constructs an input representation that encodes the syntactic dependency relations found within the predicate context by concatenating the embeddings for the arguments and learning a mapping to a lower-dimensional space. In this way it is similar to our recurrent neural network, but instead of learning the syntactic information implicitly, it feeds it directly, which potentially gives it an advantage. In our experimental setup this could be remedied by training a syntactic dependency parser that shares hidden layer parameters with the FrameID mod-

ule. Since the FrameNet corpus is not annotated with such data, sequential transfer learning can be employed in order to train on two different signals at different stages (Ruder, 2019).

The Botschen et al. (2018) model is most significantly different from ours in two respects: it uses multimodal embedding representations at the input (textual + visual), and it employs a softmax classifier at the output step, whereas we use MSE as a loss function. Prior work has shown that the first option is more powerful in the context of word sense disambiguation tasks (Popov, 2017). The two classification approaches can be combined, however, in a multitask learning setting to boost accuracy with respect to both (Popov, 2019). Since our main goal in this paper has been to demonstrate the benefits of embedding FrameNet concepts using a graph model, we leave the task of improving the accuracy of the framework for future work.

Nevertheless, our Bi-LSTM does perform well compared to the Das et al. (2014) system, and we get strong results in the “No Lexicon” condition, suggesting the model is able to successfully learn frame categories without any knowledge from the lexicon. The “Unseen” metric shows a complementarity between the graph- and corpus-based embeddings, which further suggests that the two sources of information encode very different lexical and world knowledge.

8.1 Linguistic Analysis

We proceed with a qualitative comparison of frame performance to establish which frames have a boost in performance with graph or corpus-based knowledge. To discriminate the best performing frames within each model, we assign a ranking function for frame performance. We run this ranking function over each model to obtain a ranked list of its best performing frames.

Ranking combines the accuracy of each individual frame $Acc(F_i)$ with an added weight for the frequency of the frame in the corpus $Freq(F_i)$ to obtain an overall rank score $Rank(F_i)$. $Acc(F_i)$ is defined as the number of correctly predicted instances of the frame over the total number of instances. $Freq(F_i)$ is the total number of frame instances over the total number of all frame instances in the test corpus. The $Freq(F_i)$ weight ensures that frames with higher counts in the corpus are ranked higher than frames that have few in-

stances, so therefore a frame with a perfect score will receive a higher rank when there are 80 instances of that frame than a perfect score with only a single instance.

We define $Rank(F_i)$ of a single frame as:

$$Rank(F_i) = Acc(F_i) + Freq(F_i) \quad (1)$$

We then apply the ranking formula to all frames F in the test data and select the 20 highest ranked frames. The highest ranking frames for each model are given in Table 2 and Table 3. Table 2 gives frames that perform better with graph-based knowledge, including AWARENESS, TIME_VECTOR, GOAL, and DEPARTING. Interestingly, AWARENESS and TIME_VECTOR both have a large number of predicates, where AWARENESS has predicates *comprehend.v*, *know.v*, and *understand.v*, and TIME_VECTOR contains mostly function words such as *after.prep*, *before.prep*, and *since.adv*. Both frames have frame elements that appear in multiple related frames, including EXPRESSOR and COGNIZER, which appear in AWARENESS and other frames relating to mental activity, and LANDMARK_EVENT in the TIME_VECTOR frame, which also appears in the closely related TEMPORAL_COLLOCATION frame. This suggests that their graph structures are large but the knowledge in these frame graphs are tied to a specific domain.

Alternatively, Table 3 shows frames that perform better in the corpus-based model, including ORIGIN, LEADERSHIP, and NATURAL_FEATURES. Their predicates are mostly restricted to one sense – ORIGIN, for instance, has predicates *jamaican.n*, *canadian.n*, and *french.a*, while NATURAL_FEATURES has predicates *mountain.n* and *continent.n*. However, their frame elements are more likely to appear across different domains. LOCALE is a frame element of the NATURAL_FEATURES frame, and it appears in POLITICAL_LOCALES which has many politically-related predicates and politically-related frames, all relatively unrelated to the concept of NATURAL_FEATURES. ENTITY is a frame element of the ORIGIN frame, and it appears in AGING and EVENTIVE_AFFECTING frames – both are also only loosely related to the concept of an ORIGIN. Frames that perform better under the corpus-based model, then, have frame elements and related frames that are more spread out and distantly connected in the graph, so it is perhaps not too sur-

Top 20 frames in graph- versus corpus-based frame embedding models. Frames in **bold** are not in the top 20 of the other model, and thus have benefited from knowledge found in either the graph (Table 2) or corpus (Table 3).

Top Graph-based Frames	G-Rank	C-Rank
BUILDINGS	1	2
QUANTITY	2	3
PEOPLE	3	4
VEHICLE	4	5
AWARENESS	5	269
KINSHIP	6	6
ASSISTANCE	7	7
INCREMENT	8	8
TIME_VECTOR	9	279
POLITICAL_LOCALES	10	9
ROADWAYS	11	11
KILLING	12	12
IMPORTANCE	13	13
WEAPON	14	14
INTENTIONALLY_ACT	15	17
ECONOMY	16	18
BUILDING	17	19
GOAL	18	400
DISCUSSION	19	20
DEPARTING	20	21

Table 2: Top graph embedding-based frames with the graph model rank (**G-rank**) and the corpus model rank (**C-Rank**)

Top Corpus-based Frames	C-Rank	G-Rank
LOCATIVE_RELATION	1	269
BUILDINGS	2	1
QUANTITY	3	2
PEOPLE	4	3
VEHICLE	5	4
KINSHIP	6	6
ASSISTANCE	7	7
INCREMENT	8	8
POLITICAL_LOCALES	9	10
ORIGIN	10	325
ROADWAYS	11	11
KILLING	12	12
IMPORTANCE	13	13
WEAPON	14	14
NATURAL_FEATURES	15	286
LEADERSHIP	16	277
INTENTIONALLY_ACT	17	15
ECONOMY	18	16
BUILDING	19	17
DISCUSSION	20	19

Table 3: Top corpus embedding-based frames with the corpus model rank (**C-Rank**) and the graph model rank (**G-rank**)

prising that under these conditions the graph embeddings do not help in prediction.

9 Conclusion

Our results here demonstrate that neural networks can achieve a significant boost when combining representations learned from corpus data with representations learned from knowledge graphs. Many frames that perform poorly in pure corpus-based models improve in the graph-based models. The graph-based model seems to learn better when there is a large set of domain-specific knowledge extracted from the frame’s graph. Alternatively, corpus models provide benefits to frames whose graph structures are more diffuse, suggesting the corpus knowledge is better at helping the model to narrow down the sense of the predicate by using the contextual cues found in the annotated data. By combining both embedding types, we achieve strong gains where the combined model can draw advantages from both graph- and corpus-

based embeddings.

Acknowledgements

Alexander Popov’s contribution has been supported by the Bulgarian Ministry of Education and Science under the National Research Programme “Young scientists and postdoctoral students” approved by DCM # 577 / 17.08.2018.

References

- Eneko Agirre, Montse Cuadros, German Rigau, and Aitor Soroa. 2010. Exploring knowledge bases for similarity. In *LREC*.
- Eneko Agirre, Oier López de Lacalle, and Aitor Soroa. 2014. Random walks for knowledge-based word sense disambiguation. *Computational Linguistics* 40(1):57–84.
- Collin Baker and Michael Ellsworth. 2017. Graph methods for multilingual framenets. In *Proceedings of TextGraphs-11: the Workshop on Graph-based Methods for Natural Language Processing*. pages 45–50.
- Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, pages 86–90.
- Teresa Botschen, Iryna Gurevych, Jan-Christoph Klie, Hatem Mousselly Sergieh, and Stefan Roth. 2018. **Multimodal frame identification with multilingual evaluation**. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, pages 1481–1491. <https://doi.org/10.18653/v1/N18-1134>.
- 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.
- Volha Bryl, Sara Tonelli, Claudio Giuliano, and Luciano Serafini. 2012. **A novel framenet-based resource for the semantic web**. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*. ACM, New York, NY, USA, SAC '12, pages 360–365. <https://doi.org/10.1145/2245276.2245346>.
- Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on wikipedia data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. pages 708–716.
- Dipanjan Das, Desai Chen, André FT Martins, Nathan Schneider, and Noah A Smith. 2014. Frame-semantic parsing. *Computational linguistics* 40(1):9–56.
- Dipanjan Das, Nathan Schneider, Desai Chen, and Noah A Smith. 2010. Probabilistic frame-semantic parsing. In *Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics*. Association for Computational Linguistics, pages 948–956.
- Maddalen Lopez De Lacalle, Egoitz Laparra, and German Rigau. 2014. Predicate matrix: extending semlink through wordnet mappings. In *LREC*. pages 903–909.
- Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard H. Hovy, and Noah A. Smith. 2015. **Retrofitting word vectors to semantic lexicons**. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*. pages 1606–1615. <http://aclweb.org/anthology/N/N15/N15-1184.pdf>.
- Josu Goikoetxea, Aitor Soroa, and Eneko Agirre. 2015. Random walks and neural network language models on knowledge bases. In *Proceedings of the 2015 conference of the North American Chapter of the Association for Computational Linguistics: Human language technologies*. pages 1434–1439.
- Yuhang Guo, Wanxiang Che, Ting Liu, and Sheng Li. 2011. A graph-based method for entity linking. In *Proceedings of 5th International Joint Conference on Natural Language Processing*. pages 1010–1018.
- Silvana Hartmann, Ilia Kuznetsov, Teresa Martin, and Iryna Gurevych. 2017. **Out-of-domain FrameNet semantic role labeling**. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Association for Computational Linguistics, Valencia, Spain, pages 471–482. <https://www.aclweb.org/anthology/E17-1045>.
- Karl Moritz Hermann, Dipanjan Das, Jason Weston, and Kuzman Ganchev. 2014. Semantic frame identification with distributed word representations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. volume 1, pages 1448–1458.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. **Efficient estimation of word representations in vector space**. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*. <http://arxiv.org/abs/1301.3781>.
- George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM* 38(11):39–41.
- Roberto Navigli and Simone Paolo Ponzetto. 2010. Babelnet: Building a very large multilingual semantic network. In *Proceedings of the 48th annual meeting of the association for computational linguistics*. Association for Computational Linguistics.
- Martha Palmer. 2009. Semlink: Linking propbank, verbnet and framenet. In *Proceedings of the generative lexicon conference*. GenLex-09, Pisa, Italy, pages 9–15.

- Hao Peng, Sam Thomson, Swabha Swayamdipta, and Noah A. Smith. 2018. [Learning joint semantic parsers from disjoint data](https://arxiv.org/abs/1804.05990). *CoRR* abs/1804.05990. <http://arxiv.org/abs/1804.05990>.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.
- Alexander Popov. 2017. Word sense disambiguation with recurrent neural networks. In *Proceedings of the Student Research Workshop associated with RANLP*, pages 25–34.
- Alexander Popov. 2019. Lexical modeling for natural language processing. In *Selected papers from the CLARIN Annual Conference 2018, Pisa, 8-10 October 2018*. Linköping University Electronic Press, 159, pages 147–160.
- Sebastian Ruder. 2019. *Neural Transfer Learning for Natural Language Processing*. Ph.D. thesis, NATIONAL UNIVERSITY OF IRELAND, GALWAY.
- Josef Ruppenhofer, Michael Ellsworth, Myriam Schwarzer-Petruck, Christopher R Johnson, and Jan Scheffczyk. 2006. Framenet ii: Extended theory and practice .
- Jennifer Sikos and Sebastian Padó. 2018. Using embeddings to compare framenet frames across languages. In *Proceedings of the First Workshop on Linguistic Resources for Natural Language Processing*, pages 91–101.
- Jennifer Sikos and Sebastian Padó. 2019. Frame identification as categorization: Exemplars vs prototypes in embeddingland. In *International Conference on Computational Semantics(IWCS)*.