Background Research: Frame Embeddings

Hermann, K. M., Das, D., Weston, J., & Ganchev, K. (2015). Semantic Frame Identification with Distributed Word Representations (pp. 1448–1458).

General Idea of Frame Disambiguation: Given labeled data, project the set of word representations for the syntactic context around a predicate to a low dimensional representation.

1. Represent the syntactic context of a predicate instance as a matrix $\mathbb{R}^{nk}$ of word embeddings with size n for all the k possible syntactic dependents. For example, in the sentence "He runs the company." The context vector has the embedding of "He" at the subject position, the embedding of "company" at the direct object position, and zeros everywhere else.
2. At inference time, the predicate-context is mapped to a low dimensional space. The nearest frame label is chosen as the correct label.
3. The mapping M is a linear transformation, and it is learned using the $WSABIE$ algorithm.

General Idea of Frame Embedding:

1. The embeddings of frame labels are learned using the $WSABIE$ algorithm. For F frames in total, we represent all the embeddings as $\mathbb{R}^{F \times m}$ where each frame is represented by one m-dimensional vector.
2. The objective function is a weighted approximate-rank pairwise (WARP) loss (Weston et al. (2011)).

   $\sum_x \sum_y L(rank_y(x)) max(0, \gamma + s(x,y) - s(x,\hat{y}))$

   x, y are the training inputs and their corresponding correct frames, and $\hat{y}$ are negative frames, γ is the margin.

   $rank_y(x)$ is the rank of the positive frame y relative to all the negative frames:

   $rank_y(x) = \sum_{\hat{y}} I(s(x,y) \leq \gamma + s(x,\hat{y}))$

   $L(x)$ converts the rank to a weight.
3. The objective function is minimized through stochastic gradient. For speed, the computation of $rank_y(x)$ is replaced with an approximation of $(F-1)/N$ where F is the total number of frames and N is the number of items $\hat{y}$ that are sampled until a violation is found, i.e., $max(0, \gamma + s(x,y) - s(x,\hat{y})) > 0$

<u>Hyperparameters:</u>

- embedding dimension 100
- maximum number of negative samples: 100
- epochs: 1000
- initial representation of predicate and context: concatenation of pretrained dependency- based word embeddings

---

Botschen, T., Mousselly Sergieh, H., & Gurevych, I. (2017). Prediction of Frame-to-Frame Relations in the FrameNet Hierarchy with Frame Embeddings (pp. 146–156)

General Idea of Frame Embedding: Word2Vec

1. Replace all predicates with their frames. Then find the embeddings for the frames using the skip-gram model.

Hyperparameters

- embedding dimension: 300
- minimal word frequency: 10
- negative sampling of noise words: 5
- window size 2
- initial learning rate: 0.025
- iterations: 10

General Idea of F2F Relations Embeddings:

1. **Prototypical relation embeddings**: Find all the vector offsets for any two frames in a relation-specific subset of frames G. The embedding for a relation is the mean of all the vector offsets of G.
2. **TransE embeddings**: Generate transE embeddings for frames and relations in the knoweldge graph (f_1, r, f_2) such that $transE(f_1) + transE(r) \approx transE(f_2)$
3. **Neural network for relation selection**: Identify the best F2F relation r between two frames. Within the NN, the initial vector representations of the two frames are combined into an internal dense layer c, followed by the calculation of the cosine similarity between this combination and the relation r.

   A negative relation r' is sampled randomly and its vector representation is fed into the NN to yield a second cosine similarity.

   The NN minimizes the ranking loss: $loss = max(0, m - cos(c, r) + cos(c, r'))$ where m is a margin and cos is the cosine similarity.

Achieved the best result in predicting relations between frames using the NN approach in combination with the knowledge-based TransE embeddings as input representations.

Sikos, J., & Padó, S. (2018, August). Using embeddings to compare framenet frames across languages. In *Proceedings of the First Workshop on Linguistic Resources for Natural Language Processing* (pp. 91-101).

Preprocessing FrameNet Corpus:

1. multi-word frame-evoking elements (FEEs) are concatenated into a single token because presence of multi-word FEEs that evoke different frames from the single-word FEEs that they

contain.
2. concatenate named entities that span multiple tokens into a single span.
3. replaces each occurrence of a FEEs by the name of the frame it evokes to produce the “Frame Corpus”.

Generating Frame Embeddings: learn 300-dimensional monolingual FEE-level and frame-level embeddings using the word2vec CBOW method.

- too few iterations and too few negative samples result in noisy embeddings.
- top ten nearest neighbor frame embeddings for two frames locate in the same, monolingual space because they have similar semantic meaning.

Comparison of FrameNet and SALSA frame embeddings:

- cosine similarity
- dimensionality reduction to two dimensions

Interesting Results:

- Amongst the very similar frames, frames tend to be belong to abstract domains such as Cognition or describing abstract properties of events.
- For similar frames, there are little correlation between FrameNet and SALSA.

My comment:

- Due to the lack of complete annotations which causes no reliable embeddings can be learned, which results in low similarity between English and German frames even though they have similar semantic meaning. This means that it might be difficult to expand FrameNet with frames from other languages that have fewer annotations.

Alhoshan, W., Batista-Navarro, R. T., & Zhao, L. (2019). Semantic Frame Embeddings for Detecting Relations between Software Requirements. In *Proceedings of the 13th International Conference on Computational Semantics-Student Papers* (pp. 44-51).

Method of generating frame embeddings:

1. First, generate word embeddings for each LU using continuous bag-of-words (CBOW) learning method of Word2Vec. LU of different POS tag should have different word embeddings.
2. The frame embedding of a particular frame is the average of the word vectors of the LUs that evoke it.

Measuring frame-to-frame semantic relations:

1. cosine similarity metric
2. Normalized Euclidean and Manhattan metrics (due to sparse data points that result in large distance scores) - 0.50 as a threshold value.

The result shows that cosine metric provides more reliable relatedness scores that are close to the registered human-judgement. The Euclidean and Manhattan metrics measure the similarity of these two frames depending on how often they occurred in the corpus, whereas cosine similarity measures only the angle of their vector representations.

My comment:

The paper describes how semantic frame embeddings can identify relations between frames, but not the exact type of relations. This means that certain type of relations that are infrequent in the training annotation sets may not be identified because the robustness of the frame embeddings depends on the frequency of frames. Moreover. I believe instead of taking the average, the authors should take the weighted average of the word vectors for obtaining frame embeddings.

Wang, W. Y., & Yang, D. (2015, September). That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 2557-2563).

The authors generate two types of embeddings for categorizing tweets.

1. Continuous word embeddings with additional data augmentation. Specifically, in the embedding vocabulary W, we search for the k-nearest-neighbor (knn) word w for a query term using cosine similarity between query $\vec{Q}$ and target word vectors $\vec{W}$: $argmax_{w \in W} cosine(\vec{Q}, \vec{W})$. For each word in a tweet, we query the external Google News lexical embeddings, and replace them with their knn words to create a new training instance.
2. Frame semantic embeddings with additional data augmentation: a continuous bag-of-frame model to represent each semantic frame using Word2Vec. Same data augmentation approach to create additional instances with these semantic frame embeddings.

My comment:

The most notable takeaway from this paper is the data augmentation approach that expands the annotation dataset used for generating frame embeddings.

Ustalov, D., Panchenko, A., Kutuzov, A., Biemann, C., & Ponzetto, S. P. (2018). Unsupervised semantic frame induction using triclustering. *arXiv preprint arXiv:1805.04715*.

Frame Induction: A triclustering task where the induced frames are based on graph clustering of SVO triples.

1. Extract Subject-Verb-Object triples with a dependency parser
2. Generate embeddings of size V^3 for each SVO triples given vocabulary V and a d-dimensional word embeding model $v \in V$. Dense representations of the triples are obtained by concatenating the word vectors corresponding to the elements of each triple.
3. Generate the undirected graph $\mathbb{G}$ by constructing cosine-similarity weighted edges between each pair of triple through KNN.
4. Apply WATSET fuzzy graph clustering algorithm to retrieve clusters of similar triples (frame clusters).
5. Aggregate the subjects, the verbs, and the objects of the contained triples into separate sets. As the result, each cluster is transformed into a triframe.

My comment:

1. Focus on clustering of SVO triples, i.e., a frame is defined by a head and two slots. Therefore, it is not appropriate for instances that have a variable number of arguments.
2. The study didn't touch on how different clusters are assigned a distinct frame.

QasemiZadeh, B., Petruck, M. R., Stodden, R., Kallmeyer, L., & Candito, M. (2019, June). SemEval-2019 Task 2: Unsupervised Lexical Frame Induction. In *Proceedings of the 13th International Workshop on Semantic Evaluation* (pp. 16-30).

Evaluation: distributional similarities between generated unsupervised labeled data and human annotated reference data measured by BCUBED F-SCORE.

Task A: Clustering Verbs: Identify verbs that evoke the same frame. Unlike word sense induction, the task assumes that the sense inventory is defined independent of word forms, which means two things:

1. different word forms can be mapped to the same label (frame)
2. the same label (frame) can be mapped to several word forms.

Task B.1: Unsupervised Frame Semantic Argument Labeling: Labeling of frames and core FEs. Given a set of arugments as input, the verbs are clustered and assigned to a set of unsupervised frame labels, and the arguments are labeled with frame elements. For any pair of frame labels π_x and π_y, the set of assigned roles R_x to arguments under π_x are independent from R_y under π_y ($R_x \cap R_y = \emptyset$).

Task B.2: Unsupervised Case Role Labeling: Arguments of a verb are clustered according to universal and generic semantic roles (e.g. Agent, Patient). The role labels are comparable to the roles from VerbNet 3.2.

Anwar, Saba & Ustalov, Dmitry & Arefyev, Nikolay & Paolo Ponzetto, Simone & Biemann, Chris & Panchenko, Alexander. (2019). HHMM at SemEval-2019 Task 2: Unsupervised Frame Induction using Contextualized Word Embeddings.

Task A:

Word and Sentence embeddings of verb, sentence/context, subject-verb-object (SVO) triples, and verb arguments.

- Google News Word2Vec embedding
- ELMo - generates vectors of a whole context.
- Universal Sentence Embeddings
- fastText

Combination of context and word representations from Word2Vec and ELMo turned out to be the best combination in our case.

Clustering Algorithms:

- agglomerative clustering - Manhattan affinity and 150 clusters (Best)
- DBSCAN
- affinity propagation

Task B.1:

- Merge the output of verb frame types from Subtask A and the output of generic role labels from Subtask B.2.
- used UKN (unknown) slot identi- fier for the tokens present in Subtask B.1, but missing in Subtask B.2.

Task B.2:

- combining the embeddings of the word (W) filling the role, its sentence (context, C), and the highlighted verb (V).
- Inbound dependencies: a negative one-hot encoding feature vector to represent the inbound dependencies of the word corresponding to the role. For each dependency of the given role (in case of a multi-word expression), we fill -1 if the dependency relationship holds, otherwise 0 is filled.
- Best result when clustered by agglomerative clustering with Euclidean affinity, Ward's method linkage, and two clusters.

My comment:

The study shows that in the case of general role labeling, "the combination of sentence (context, C), target word (W), and verb (V) vectors, enhanced with our other features, shows substantially better results than a simple clustering model". However it didn't fully elaborate on the reasons. I believe it is due to the fact that word vector W and verb vectors V do not confidently represent the semantic meanings of the verb and the roles without the understanding of the entire sentence, which is representented by the context C.

Arefyev, N., Sheludko, B., Davletov, A., Kharchev, D., Nevidomsky, A., & Panchenko, A. (2019). Neural GRANNy at SemEval-2019 Task 2: A combined approach for better modeling of semantic relationships in semantic frame induction. *SemEval@NAACL-HLT*.

Vector representations of target word:

1. dense vector representations of the target word in a context obtained from hidden layers of BERT model
    - didn't discriminate different senses of the same verb. Better groups synonyms.
    - BERT is better than ELMo as BERT takes into account the whole context in all of its layers,
2. sparse TF-IDF BOW vectors from substitutes generated for the target word by BERT masked LM.
    - Generate K most probable verb substitutes using a masked LM based on the bert-base-uncased model, lemmatize then build TF-IDF bag-of-words vectors.
    - problems with clustering together similar senses of different verbs
    - better at disambiguating homonyms

Clustering of Embeddings:

1. K-means
2. DBScan
3. Affinity Propagation
4. Agglomerative clustering (Best result but explanation is not given)

Ribeiro, E., Mendonça, V., Ribeiro, R.A., Matos, D.M., Sardinha, A., Santos, A.L., & Coheur, L. (2019). L2F/INESC-ID at SemEval-2019 Task 2: Unsupervised Lexical Semantic Frame Induction using Contextualized Word Representations. *SemEval@NAACL-HLT*.

Induction Approach for Clustering Verbs and Arguments

1. Generate the contextualized representation of each instance (verbs / arguments) to be clustered using ELMo and BERT.
2. Build a graph (for clustering) using the cosine distance with a varying threshold that determines the number of neighbors.
3. Obtain the cluster using the Chinese Whispers algorithm on the graph.

My Comment: Didn't create the embeddings specific to frames. It didn't quite mention in details how the labels are assigned to the clusters.