Acquiring Frame Element Knowledge with Deep Metric Learning for Semantic Frame Induction

Kosuke Yamada[1] **Ryohei Sasano**[1,2] **Koichi Takeda**[1]
[1]Graduate School of Informatics, Nagoya University, Japan
[2]RIKEN Center for Advanced Intelligence Project, Japan
yamada.kosuke.v1@s.mail.nagoya-u.ac.jp,
{sasano,takedasu}@i.nagoya-u.ac.jp

Abstract

The semantic frame induction tasks are defined as a clustering of words into the frames that they evoke, and a clustering of their arguments according to the frame element roles that they should fill. In this paper, we address the latter task of argument clustering, which aims to acquire frame element knowledge, and propose a method that applies deep metric learning. In this method, a pre-trained language model is fine-tuned to be suitable for distinguishing frame element roles through the use of frame-annotated data, and argument clustering is performed with embeddings obtained from the fine-tuned model. Experimental results on FrameNet demonstrate that our method achieves substantially better performance than existing methods.

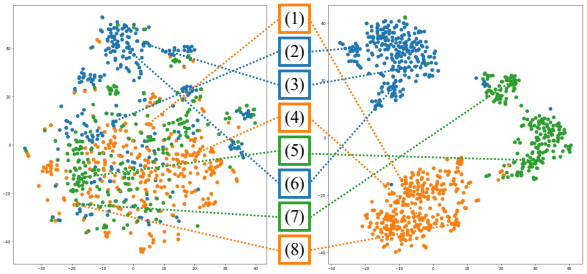
1 Introduction

A semantic frame is a coherent conceptual structure that describes a particular type of situation or event along with its participants and props. FrameNet (Ruppenhofer et al., 2016) is a representative resource, in which semantic frames define a set of frame-specific roles called frame elements (FEs). FrameNet comprises a list of semantic frames, sets of frame-evoking words, and collections of frame-annotated examples. Table 1 lists examples of frame-annotated sentences for the GIVING frame in FrameNet. For each sentence, a frame-evoking word is annotated with the GIVING frame, and its arguments are annotated with FEs such as Donor, Theme, and Recipient.

Because manually arranging such frame resources on a large scale is labor intensive, there have been many studies on automatic induction of frame resources. Most of these studies have assumed only verbs as frame-evoking words and divided the frame induction task into two sub-tasks: verb clustering, which groups verbs according to the frames that they evoke, and argument clustering, which groups arguments of verbs according to their FE roles (Anwar et al., 2019; Ribeiro et al., 2019). This study addresses the argument clustering task and acquires frame element knowledge for semantic frame induction.

1. [(1)Theme It] was handed in [(2)Donor by a couple of children] this morning.
2. [(3)Donor I] will now donate [(4)Theme the money] [(5)Recipient to charity].
3. [(6)Donor Your gift] gives [(7)Recipient children and families] [(8)Theme hope for tomorrows].

Table 1: Examples of verbs that evoke the GIVING frame in FrameNet



(a) Vanilla BERT (b) Fine-tuned BERT w/ Triplet

Figure 1: 2D t-SNE mappings of average BERT embeddings of argument tokens, which are labeled with Donor, Theme, or Recipient, in examples of verbs that evoke the GIVING frames in FrameNet. The numbers in parentheses correspond to the examples in Table 1.

As with many natural language processing tasks, methods using contextualized embeddings such as ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019) have been proposed for argument clustering tasks. However, these methods have been reported to perform worse than methods based on syntactic relations (Anwar et al., 2019; Ribeiro et al., 2019). We assume that this is because vanilla BERT, i.e., BERT without fine-tuning, is more influenced by factors such as a whole sentence's meaning and does not emphasize information that captures differences in semantic roles. Figure 1(a) shows a 2D

Findings of the Association for Computational Linguistics: ACL 2023, pages 9356–9364

t-SNE (Maaten and Hinton, 2008) projection of the average BERT embeddings of argument tokens in examples of the GIVING frame in FrameNet. We can see that these embeddings are not adequately clustered according to their semantic roles.

Hence, in this study, we propose the use of deep metric learning to fine-tune a contextual word embedding model so that instances of the same FEs are placed close together while other instances are placed farther apart in the embedding space. Figure 1(b) shows a 2D projection of the average BERT embeddings of argument tokens after fine-tuning with our proposed method based on the triplet loss. We can confirm that instances of the same FEs are located close to each other. This suggests that deep metric learning enables fine-tuning of BERT to obtain embedding spaces that better reflect human intuition about FEs.

2 Acquiring Frame Element Knowledge with Deep Metric Learning

To acquire frame element knowledge for semantic frame induction, we work on argument clustering, which is the task of grouping arguments of frame-evoking words according to their roles in the evoked frame. We introduce two argument clustering methods that cluster argument instances using their contextualized word embeddings. To achieve higher performance methods, we assume the existence of frame-annotated data and propose to fine-tune a contextualized word embedding model using deep metric learning.

2.1 Deep Metric Learning

Deep metric learning is a method of learning deep learning models on the embedding space in such a way that instances with the same label are placed closer together and instances with different labels are placed farther apart (Kaya and Bilge, 2019; Musgrave et al., 2020). By applying this to the contextualized word embedding model, it is expected that argument instances with similar roles learn to be closer together, and argument instances with different roles learn to be farther apart. We use the representative triplet (Weinberger and Saul, 2009) and ArcFace losses (Deng et al., 2019) from two major approaches: the distance-based and classification-based approaches, respectively.

Triplet loss This loss function is commonly used in deep metric learning, in which the distance to a triplet of instances can be learned directly using three encoders. Specifically, it performs learning such that the distance between an anchor instance $\boldsymbol{x}_a$ and a negative instance $\boldsymbol{x}_n$, which are taken from different classes, is to be larger than a certain margin m plus the distance between the anchor instance $\boldsymbol{x}_a$ and a positive instance $\boldsymbol{x}_p$. The squared Euclidean distance is typically used as the distance function D. The triplet loss is defined as follows:

$$L_{\text{tri}} = \max\left(D\left(\boldsymbol{x}_a, \boldsymbol{x}_p\right) - D\left(\boldsymbol{x}_a, \boldsymbol{x}_n\right) + m, 0\right). \quad (1)$$

ArcFace loss This loss has been used as a de facto standard in face recognition. It modifies the softmax-based cross-entropy loss for typical n-class classifiers. Specifically, it applies l_2 regularization to the i-th class weight $\boldsymbol{w}_i$ and the embedding of the i-th class instance $\boldsymbol{x}_i$. The angle between $\boldsymbol{w}_i$ and $\boldsymbol{x}_i$ is denoted as θ_i. An angular margin m and a feature scale s are introduced as hyperparameters to simultaneously enhance the intra-class compactness and inter-class discrepancy. The ArcFace loss is defined as follows:

$$L_{\text{arc}} = -\log \frac{e^{s\cdot\cos(\theta_i+m)}}{e^{s\cdot\cos(\theta_i+m)} + \sum_{j=1,j\neq i}^{n} e^{s\cdot\cos\theta_j}}. \quad (2)$$

2.2 Argument Clustering Methods

We introduce two argument clustering methods: a cross-frame clustering of argument instances across frames and an intra-frame clustering of frame-wise argument instances.

2.2.1 Cross-Frame Method

The cross-frame method is a method used by Anwar et al. (2019) and Ribeiro et al. (2019), in which FEs are regarded as general semantic roles independent of frames, and the argument instances are grouped by roles across frames. For example, both Donor in the GIVING frame and Agent in the PLACING frame are similar roles in the meaning of "a person who acts on an object." Taking advantage of this property, the cross-frame method clusters the argument instances to form role clusters without considering the frame that each word evokes and then combines the frame and the role clusters into the FE clusters. In this method, we apply group-average clustering based on the Euclidean distance, which is a hierarchical clustering algorithm.[1]

The cross-frame method performs fine-tuning of contextualized word embedding models across frames by using the triplet and ArcFace losses. For

[1] See Appendix A for the number of clusters.

the triplet loss, a positive instance is one with the same FE as the anchor instance, while a negative instance is one with FEs of different frames or different FEs of the same frame as the anchor instance. The ArcFace loss is used to classify instances on an FE basis so that the model trains the metric across frames rather than within a particular frame.

2.2.2 Intra-Frame Method

Since the cross-frame method treats FEs as roles independent of frames even though FEs are frame-specific roles, there are two possible drawbacks as described below. We thus propose the intra-frame method that treats FEs as frame-specific roles.

As the first drawback, the cross-frame method causes the division of argument instances of the same FE into too many clusters. For example, the GIVING frame has only three FEs, but the cross-frame method is likely to split instances into more clusters due to the nature of clustering across frames. To overcome this drawback, the intra-frame method focuses on clustering the argument instances for each frame. The method also uses group-average clustering.

As the second drawback, the fine-tuning of the cross-frame method may not provide the optimal embedding space for argument roles, because it learns to keep instances with similar roles in different frames away from each other. For example, Donor in the GIVING frame and Agent in the PLACING frame are similar, but the cross-frame method keeps these instances away because they are regarded as different roles. Hence, the intra-frame method learns to keep away only between instances of different FEs of the same frame. For the triplet loss, this is achieved by limiting negative instances to be different FEs in the same frame. For the ArcFace loss, this is achieved by training classification for the number of FE types in a frame.

3 Experiment

To confirm the usefulness of fine-tuning with deep metric learning, we experimented with an argument clustering task. This study focuses on argument clustering to induce FEs for frame-evoking verbs. Given the true frame that a verb evokes and the true positions of its argument tokens in the example sentences, we cluster only its arguments to generate role clusters. Then, we merge the true frame and the role clusters to obtain the final FE clusters.

	#Frames	#FEs	#Examples	#Instances
Set 1	212	641	21,433	42,544
Set 2	212	623	24,582	47,629
Set 3	213	637	35,468	71,617
All	637	1,901	81,493	161,790

Table 2: Statistics of the FrameNet-based dataset used in three-fold cross-validation.

3.1 Settings

Dataset The dataset in our experiment was created by extracting example sentences, in which the frame-evoking word was a verb, from FrameNet 1.7.[2] The FEs in FrameNet are divided into two types: core FEs, which are essential for frames, and non-core FEs. Our experiment targeted only the core FEs, as in QasemiZadeh et al. (2019). The examples were divided into three sets so that those of the verbs that evoke the same frames were in the same set. Table 2 lists the dataset statistics. We performed three-fold cross-validation with the three sets as the training, development, and test sets. Note that the frames to be trained and those to be clustered do not overlap because the sets are divided on the basis of frames.

Comparison Methods We used BERT[3] from Hugging Face (Wolf et al., 2020) to obtain contextualized word embeddings. We compared a total of six different methods, which use the cross-frame method or the intra-frame method for each of the three models, the vanilla model (**Vanilla**) and two fine-tuned models (**Triplet**, **ArcFace**).[4]

We also compared our methods with the two unsupervised methods used in Subtask-B.1 of SemEval-2019 Task 2 (QasemiZadeh et al., 2019).[5] Anwar et al. (2019) performed group-average clustering by using a negative one-hot encoding feature vector to represent the inbound dependencies of argument words. Ribeiro et al. (2019) applied graph clustering by Chinese whispers (Biemann, 2006) with the average ELMo (Peters et al., 2018) embeddings of argument tokens. We also prepared two baselines: **Boolean** and **Dependency-relationship**. The Boolean method clusters argument instances based on whether they appear before or after the

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

[3] https://huggingface.co/bert-base-uncased

[4] See Appendix B for the detailed settings of these methods.

[5] The SemEval-2019 Task 2 dataset is no longer available, as described on its official website; thus, we excluded that dataset from the experiments.

Method		#C	PU / IPU / PIF	BCP / BCR / BCF
Boolean		411	70.7 / 85.9 / 77.6	61.4 / 79.6 / 69.4
Dependency-relationship		2,032	84.6 / 70.6 / 77.0	78.2 / 56.9 / 65.9
Anwar et al. (2019)		415	59.2 / 75.8 / 66.5	49.0 / 67.0 / 56.6
Ribeiro et al. (2019)		628	65.3 / 74.6 / 69.6	55.0 / 64.4 / 59.3
Clustering	**Model**			
Cross-frame method (group-average clustering)	Vanilla	628	55.2 / 87.5 / 67.6	46.5 / 81.1 / 59.0
	Triplet	543	80.0 / 92.9 / 86.0	73.0 / 88.8 / 80.1
	ArcFace	594	81.7 / 91.5 / 86.2	74.9 / 86.8 / 80.3
Intra-frame method (group-average clustering)	Vanilla	636	54.9 / 88.9 / 67.9	46.2 / 83.1 / 59.4
	Triplet	646	**90.1 / 95.0 / 92.5**	**85.5 / 91.9 / 88.6**
	ArcFace	631	90.0 / 94.3 / 92.1	85.4 / 90.9 / 88.1

Table 3: Experimental results for argument clustering over three-fold cross-validation. Each value in the table is the average over three trials. #C indicates the final number of clusters.

verb. For example, in the second example sentence "[I] will now donate [the money] [to charity]." in Table 1, the word "I" belongs to the *before* cluster, while "the money" and "to charity" belong to the *after* cluster. The Dependency-relationship method clusters argument instances based on dependency labels. In the case of the same example sentence as above, "I" belongs to a cluster indicating a noun subject, "the money" belongs to a cluster indicating an object, and "to charity" belongs to a cluster indicating an oblique nominal. We use stanza (Qi et al., 2020) as a dependency parsing tool.[6]

Metrics For evaluation metrics, we used PURITY (PU), INVERSE PURITY (IPU), and their harmonic mean, F-SCORE (PIF) (Zhao and Karypis, 2001), as well as B-CUBED PRECISION (BCP), RECALL (BCR), and their harmonic mean, F-SCORE (BCF) (Bagga and Baldwin, 1998).

3.2 Results

Table 3 summarizes the experimental results. The cross-frame and intra-frame methods with the Triplet and ArcFace models showed a remarkable performance improvement compared to those with the Vanilla model. In particular, the intra-frame method with the Triplet model obtained a high score of 92.5 for PIF and 88.6 for BCF. Also, while there was no difference between the intra-frame and cross-frame methods with the Vanilla model, we can confirm the efficacy of the intra-frame methods with the fine-tuned models. There was little difference in scores with the deep metric learning models. We consider that they achieved similar scores as a result of satisfactory learning because both models learn margin-based distances.

As for the comparison to previous methods, the methods with the Vanilla model underperformed the baseline methods with syntactic features, but our methods with the fine-tuned models outperformed them considerably. This result also confirms the usefulness of the fine-tuned models through deep metric learning. Among the previous methods, although the two baselines performed better than the methods in Anwar et al. (2019) and Ribeiro et al. (2019), this was an expected result because the experiment by Anwar et al. showed that the Boolean method obtained higher scores than their method. Note that our experiment only considered core FEs. The trends that baselines with syntactic features performed well may not be going to hold in experiments that consider non-core FEs.

We also visualized the embeddings to understand them intuitively. Figure 2 shows a 2D t-SNE projection of the average contextualized embeddings of the argument tokens. With the Vanilla model, clumps of instances can be seen for each FE, but instances for the same FE are entirely scattered, and the instances for different FEs in the same frame are mixed together. On the other hand, with the fine-tuned models, the instances are clustered for each FE. We can see that the instances with the cross-frame Triplet model are tightly grouped by FEs than those with the intra-frame Triplet model. However, the FEs are still independent of each frame, and it is important to distinguish instances of different FEs in the same frame. The intra-frame Triplet model distinguishes more instances with different roles in the same frame than the cross-frame

[6] https://stanfordnlp.github.io/stanza/

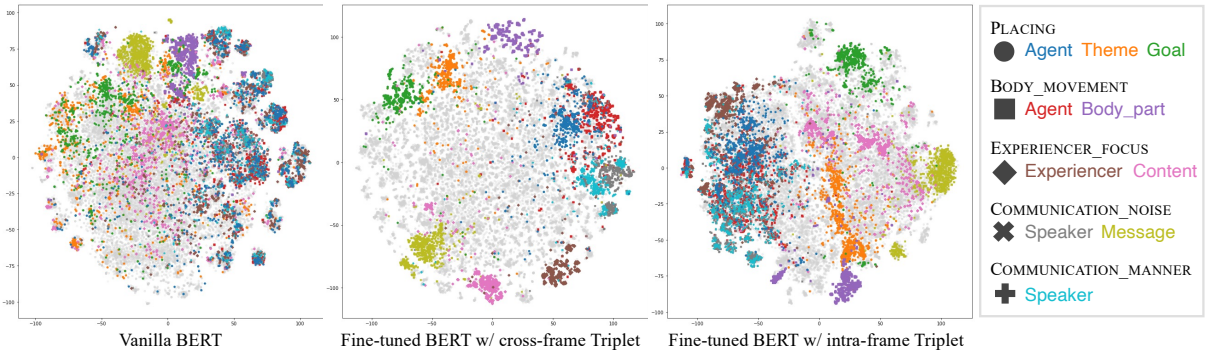


Figure 2: 2D t-SNE projections of the average embeddings of argument tokens with the Vanilla, cross-frame Triplet, and intra-frame Triplet models. The top 10 FEs with the highest numbers of instances are highlighted.

Triplet model does, such as instances of Theme and Goal in the PLACING frame. Furthermore, with the intra-frame Triplet model, we can see instances of similar roles clustered together across frames such as instances of Speaker in the COMMUNICATION_NOISE frame and Agent in the PLACING frame. These results confirm the usefulness of the fine-tuning of the intra-frame method.

4 Conclusion

We have addressed argument clustering for semantic frame induction. We proposed a method that uses deep metric learning to fine-tune contextualized embedding models and applied the resulting fine-tuned embeddings to perform argument clustering. We also introduced intra-frame methods that exploit the property that FEs are frame-specific. Experimental results showed that fine-tuned models with deep metric learning are promising and that intra-frame methods perform quite well. Especially, the intra-frame method with the Triplet model achieved high scores of 92.5 for PIF and 88.6 for BCF.

Although only core frame elements are covered in this study, it would be ideal to acquire non-core frame element knowledge as well. Since many non-core frame elements are shared among different frames and are likely to be easier to learn than core frame elements, our methods are expected to achieve competitive performance for non-core frame elements as well. We would like to confirm it in future work. The ultimate goal of this research is to automatically build frame knowledge resources from large text corpora. We will need to merge our method with methods that cluster verbs according to the frames that they evoke (Yamada et al., 2021, 2023) and predict the positions of argument tokens. In addition, we will consider how to apply our method to large text corpora.

Limitations

As we only used English FrameNet as the dataset for our experiment, it is unclear how well our method would work with other languages or corpora. However, because the method is neither language- nor corpus-specific, fine-tuning may lead to better results with other datasets. Also, the method relies on a semantic frame knowledge resource, and annotation will thus be required if it is applied to languages without such resources. This study only considers core frame elements and does not show results for non-core frame elements.

Acknowledgements

This work was supported by JST FOREST Program, Grant Number JPMJFR216N and JSPS KAKENHI Grant Numbers 21K12012 and 23KJ1052.

References

Saba Anwar, Dmitry Ustalov, Nikolay Arefyev, Simone Paolo Ponzetto, Chris Biemann, and Alexander Panchenko. 2019. HHMM at SemEval-2019 task 2: Unsupervised frame induction using contextualized word embeddings. In *Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval 2019)*, pages 125–129.

Amit Bagga and Breck Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (ACL-COLING 1998)*, pages 79–85.

Chris Biemann. 2006. Chinese whispers: An efficient graph clustering algorithm and its application to natural language processing problems. In *Proceedings of TextGraphs: the First Workshop on Graph*

Based Methods for Natural Language Processing (TextGraphs 2006), pages 73–80.

Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. 2019. ArcFace: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2019)*, pages 4690–4699.

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

Mahmut Kaya and Hasan Şakir Bilge. 2019. Deep metric learning: A survey. *Symmetry*, 11(9):1066.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *Proceedings of the 5th International Conference on Learning Representations (ICLR 2017)*.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605.

Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. 2020. A metric learning reality check. In *Proceedings of the 16th European Conference on Computer Vision (ECCV 2020)*, pages 681–699.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2018)*, pages 2227–2237.

Behrang QasemiZadeh, Miriam R. L. Petruck, Regina Stodden, Laura Kallmeyer, and Marie Candito. 2019. SemEval-2019 task 2: Unsupervised lexical frame induction. In *Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval 2019)*, pages 16–30.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations (ACL 2020)*, pages 101–108.

Eugénio Ribeiro, Vânia Mendonça, Ricardo Ribeiro, David Martins de Matos, Alberto Sardinha, Ana Lúcia Santos, and Luísa Coheur. 2019. L2F/INESC-ID at SemEval-2019 task 2: Unsupervised lexical semantic frame induction using contextualized word representations. In *Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval 2019)*, pages 130–136.

Josef Ruppenhofer, Michael Ellsworth, Myriam Schwarzer-Petruck, Christopher R Johnson, and Jan Scheffczyk. 2016. *FrameNet II: Extended theory and practice*. International Computer Science Institute.

Kilian Q Weinberger and Lawrence K Saul. 2009. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10(2).

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP 2020)*, pages 38–45.

Kosuke Yamada, Ryohei Sasano, and Koichi Takeda. 2021. Semantic frame induction using masked word embeddings and two-step clustering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021)*, pages 811–816.

Kosuke Yamada, Ryohei Sasano, and Koichi Takeda. 2023. Semantic frame induction with deep metric learning. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2023)*, pages 1833–1845.

Xiao Zhang, Rui Zhao, Yu Qiao, Xiaogang Wang, and Hongsheng Li. 2019. AdaCos: Adaptively scaling cosine logits for effectively learning deep face representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2019)*, pages 10823–10832.

Ying Zhao and George Karypis. 2001. Criterion functions for document clustering: Experiments and analysis. Technical report, Retrieved from the University of Minnesota Digital Conservancy.

A How to Determine Number of Clusters

Here, we explain how to determine the number of clusters in cross-frame and intra-frame methods. In the cross-frame method, it is determined from the ratio of the number of FEs to the number of frames in the development set.

In contrast, the intra-frame method uses criteria across frames because the number of frames is not easy to decide on a frame-by-frame basis. The termination criterion for clustering is the point at which there are no more cluster pairs for which the distance between clusters is less than a threshold θ that all frames share. The threshold θ is gradually

decreased from a sufficiently large value, and the average number of clusters over all frames is set to a value that is closest to the average number of different FEs in each frame in the development set.

B Detailed Settings for Our Methods

Here, we describe the detailed settings, including hyperparameters, of the methods in our experiment. All embeddings were processed with l_2 normalization to match the ArcFace requirement. In fine-tuning, the batch size was 16, the learning rate was 1e-5, and the number of epochs was 10. The candidate margins were 0.1, 0.2, 0.5, and 1.0 for the triplet loss and 0.01, 0.02, 0.05, and 0.1 for the ArcFace loss. The feature scale for ArcFace was 64. We explored only the margin because Zhang et al. (2019) showed that the behaviors of the margin and scale are similar. The optimization algorithm was AdamW (Loshchilov and Hutter, 2017).

In the experiment, the epochs and margins for fine-tuning and the number of clusters for clustering were determined by the development set. The most plausible model for fine-tuning was determined from ranking similarities to ensure clustering-independent evaluation. Specifically, we took an argument instance as a query instance; then, we computed the cosine similarity of the embeddings between the query instance and the remaining argument instances, and we evaluated the instances' similarity rankings in descending order. For a metric, we chose the recall. It computes the average match rate between true instances, which are instances of the same FE as the query instance, and predicted instances, which are obtained by extracting the same number of top-ranked instances as the number of true instances. The embedding of the model with the highest score was used for clustering.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
"Limitations" is found in the section after Conclusion without the section number.

☒ A2. Did you discuss any potential risks of your work?
No potential risk to our work

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1 Introduction

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

3 Experiment

☑ B1. Did you cite the creators of artifacts you used?
3 Experiment

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
URLs with licenses and terms are given for the artifacts used in the paper.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
In the three-part cross-validation, the average numbers of frame types, FE types, example sentences, and argument instances for the three sets were 212, 634, 27,161, and 53,930, respectively.

C ☑ Did you run computational experiments?

3 Experiment

☒ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
It is not described because deep metric learning using GPU is lightweight and can be implemented without using much CPU memory.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
3.1 Settings

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
3.1 Settings

☒ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Although stanza is used for normalization in part of the preprocessing, it is not described because it is a process that does not directly affect the evaluation score and is not the essence of this paper.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.