HHMM at SemEval-2019 Task 2: Unsupervised Frame Induction using Contextualized Word Embeddings

Saba Anwar[⋆], **Dmitry Ustalov**[†], **Nikolay Arefyev**[‡,§], **Simone Paolo Ponzetto**[†],
Chris Biemann[⋆], and **Alexander Panchenko**[⋆,◇]

[⋆]Language Technology Group, Department of Informatics, University of Hamburg, Germany
[◇]Skolkovo Institute of Science and Technology, Russia
{anwar,biemann,panchenko}@informatik.uni-hamburg.de
[†]Data and Web Science Group, University of Mannheim, Germany
{dmitry,simone}@informatik.uni-mannheim.de
[‡]Samsung R&D Institute Russia
[§]Lomonosov Moscow State University, Russia
narefyev@cs.msu.ru

Abstract

We present our system for semantic frame induction that showed the best performance in Subtask B.1 and finished as the runner-up in Subtask A of the SemEval 2019 Task 2 on unsupervised semantic frame induction (Qasem-iZadeh et al., 2019). Our approach separates this task into two independent steps: verb clustering using word and their context embeddings and role labeling by combining these embeddings with syntactical features. A simple combination of these steps shows very competitive results and can be extended to process other datasets and languages.

1 Introduction

Recent years have seen a lot of interest in computational models of frame semantics, with the availability of annotated sources like Prop-Bank (Palmer et al., 2005) and FrameNet (Baker et al., 1998). Unfortunately, such annotated resources are very scarce due to their language and domain specificity. Consequently, there has been work that investigated methods for unsupervised frame acquisition and parsing (Lang and Lapata, 2010; Modi et al., 2012; Kallmeyer et al., 2018; Ustalov et al., 2018). Researchers have used different approaches to induce frames, including clustering verb-specific arguments as per their roles (Lang and Lapata, 2010), subject-verb-object triples (Ustalov et al., 2018), syntactic dependency representation using dependency formats like CoNLL (Modi et al., 2012; Titov and Klementiev, 2012), and latent-variable PCFG models (Kallmeyer et al., 2018).

The SemEval 2019 task of semantic frame and role induction consists of three subtasks: (A)

learning the frame type of the highlighted verb from the context in which it has been used; (B.1) clustering the highlighted arguments of the verb into specific roles as per the frame type of that verb, e.g., *Buyer*, *Goods*, etc.; (B.2) clustering the arguments into generic roles as per VerbNet classes (Schuler, 2005), without considering the frame type of the verb, i.e., *Agent*, *Theme*, etc.

Our approach to frame induction is similar to the word sense induction approach by Arefyev et al. (2018), which uses tf–idf-weighted context word embeddings for a shared task on word sense induction by Panchenko et al. (2018). In this unsupervised task, our approach for clustering mainly consists of exploring the effectiveness of already available pre-trained models.[1] Main contributions of this paper are:

1. a method that uses contextualized distributional word representations (embeddings) for grouping verbs to frame type clusters (Subtask A);

2. a method that combines word and context embeddings for clustering arguments of verbs to frame slots (Subtasks B.1 and B.2).

The key difference of our approach with respect to prior work by Arefyev et al. (2018) and Kallmeyer et al. (2018) is that we have only used pre-trained embeddings to disambiguate the verb senses and then combined these embeddings with additional features for semantic labeling of the verb roles.[2]

---

[1]HHMM is an abbreviation for Hansestadt Hamburg, Mannheim, and Moscow. It is chosen to avoid confusion with hidden Markov models.

[2]Our code is available at https://github.com/uhh-lt/semeval2019-hhmm.

The remainder of the paper is organized as follows. The methodology and the results for each subtask are discussed in Sections 2, 3 and 4 respectively, followed by the conclusion in Section 5.

2 Subtask A: Grouping Verbs to Frame Type Clusters

In this subtask, each sentence has a highlighted verb, which is usually the predicate. The goal is to label each highlighted verb according to the frame evoked by the sentence. The gold standard for this subtask is based on the FrameNet (Baker et al., 1998) definitions for frames.

2.1 Method

Since sentences evoking the same frame should receive the same labels, we used a verb clustering approach and experimented with a number of pre-trained word and sentence embeddings models, namely Word2Vec (Mikolov et al., 2013), ELMo (Peters et al., 2018), Universal Sentence Embeddings (Conneau et al., 2017), and fast-Text (Bojanowski et al., 2017). This setup is similar to treating the frame induction task as a word sense disambiguation task (Brown et al., 2011).

We experimented with embedding different lexical units, such as verb (V), its sentence (context, C), subject-verb-object (SVO) triples, and verb arguments. Combination of context and word representations (C+W) from Word2Vec and ELMo turned out to be the best combination in our case.

We used the standard Google News Word2Vec embedding model by Mikolov et al. (2013). Since this model is trained on individual words only and the SemEval dataset contained phrasal verbs, such as *fall back* and *buy out*, we have considered only the first word in the phrase. If this word is not present in the model vocabulary, we fall back to a zero-filled vector. When aggregating a context into a vector, we used the tf–idf-weighted average of the word embeddings for this context as proposed by Arefyev et al. (2018). We tuned these weights on the development dataset.

We used the ELMo contextualized embedding model by Peters et al. (2018) that generates vectors of a whole context. Similarly to fast-Text (Bojanowski et al., 2017), ELMo can produce character-level word representations to handle out-of-vocabulary words. In all our experiments we used the same pre-trained ELMo model available

Method		Pu F$_1$	B^3 F$_1$
⚱	w2v[C+W]norm	**76.68**	**68.10**
⧖	ELMo[C+W]norm	77.03	69.50
☗	Cluster Per Verb	73.78	65.35
🏆	Winner	78.15	70.70

Table 1: Our results on Subtask A: Grouping Verbs to Frame Type Clusters. Purity F$_1$-score is denoted as *Pu F$_1$*, B-Cubed F$_1$-score is denoted as *B^3 F$_1$*. ⚱ denotes *our* final submission (# 536426), ⧖ denotes *our* post-competition result, ☗ denotes a baseline, and 🏆 denotes the submission of the winning team.

on TensorFlow Hub.[3] Among all the layers of this model, we used the mean-pooling layer for word and context embeddings.

2.2 Results and Discussion

We experimented with different clustering algorithms provided by scikit-learn (Pedregosa et al., 2011), namely agglomerative clustering, DBSCAN, and affinity propagation. After the model selection on the development dataset, we have chosen agglomerative clustering for further evaluation. Although both ELMo and Word2Vec showed the best results on the development dataset with single linkage, we opted average linkage after analyzing t-SNE plots (van der Maaten and Hinton, 2008).

Table 1 shows our results obtained on Subtask A. Our final submission (⚱) used agglomerative clustering of normalized vectors obtained by concatenating the context and verb vectors from the Word2Vec model. In particular, we found that the best performance is attained for Manhattan affinity and 150 clusters. During our post-competition experiments (⧖), we found that ELMo performed better than Word2Vec when a higher number of clusters, 235, was specified.

3 Subtask B.1: Clustering Arguments of Verbs to Frame-Specific Slots

In this subtask, each sentence has a set of highlighted nouns or noun phrases corresponding to the slots of the evoked frame. Additionally, each sentence is provided with the same highlighted verb as in Subtask A (Section 2). The goal is to label each highlighted verb according to the evoked frame and to assign each highlighted to-

---

[3] https://tfhub.dev/google/elmo/2

Method	Pu F$_1$	B^3 F$_1$
Agglomerative Clustering		
🖊 Subtask A: w2v[C+W] Subtask B.2: ID	**62.10**	**49.49**
Logistic Regression		
⚡ Subtask A: w2v[C+W]norm Subtask B.2: ELMo[C+W+V]+ID+B+123	66.81	55.61
⌛ Subtask A: ELMo[C+W]norm Subtask B.2: w2v[C+W+V]+ID+B+123	68.22	58.61
🏵 Cluster Per Dependency Role	57.99	45.79
🏆 Winner	62.10	49.49

Table 2: Our results on Subtask B.1: Clustering Arguments of Verbs to Frame-Specific Slots. Purity F$_1$-score is denoted as *Pu F$_1$*, B-Cubed F$_1$-score is denoted as *B^3 F$_1$*. 🖊 denotes *our* final submission (# 535483), ⚡ denotes a supervised *Logistic Regression* submission that does not comply to the task rules, ⌛ denotes *our* post-competition result, 🏵 denotes a baseline, and 🏆 denotes the submission of the winning team.

ken a frame-specific semantic role identifier. The gold standard for this subtask is annotated with FrameNet frames and roles (Baker et al., 1998).

3.1 Method

Since Subtask B.1 asks to assign role labels to highlighted tokens as per the frame type of the verb, we attempted this by merging the output of verb frame types from Subtask A (Section 2) and the output of generic role labels from Subtask B.2 (Section 4). We used UKN (unknown) slot identifier for the tokens present in Subtask B.1, but missing in Subtask B.2.

3.2 Results and Discussion

Table 2 shows the results from merging our solutions for Subtasks A and B.2, as described in Sections 2 and 4, correspondingly. For our final submission (🖊), we merged the frame types obtained by clustering the Word2Vec embeddings of the sentence (context, C) and verb (word, W), and the role labels obtained by clustering the vector of inbound dependencies (ID). However, we observed that the logistic regression model demonstrated better performance in Subtask B.2 than any clustering technique we tried, including our final submission (🖊) and the baselines. But this performance was further improved by combining the results from post-competition experiments of Subtask A and Subtask B.2 (⌛).

4 Subtask B.2: Clustering Arguments of Verbs to Generic Roles

In Subtask B.2, similarly to Subtask B.1 (Section 3), each sentence has a set of highlighted nouns or noun phrases that correspond to the slots of the evoked frame. The goal is to label each highlighted token with a high-level generic class, such as *Agent* or *Patient*. However, unlike Subtask B.1, the verb frame labeling part is omitted. The gold standard for this subtask is annotated as according to the VerbNet classes (Schuler, 2005).

4.1 Method

When addressing this subtask, we experimented with combining the embeddings of the word (W) filling the role, its sentence (context, C), and the highlighted verb (V). To handle the out-of-vocabulary roles in the case of Word2Vec embeddings, each role was tokenized and embeddings for each token were averaged. If a token is still not present in the vocabulary, then a zero-filled vector was used as its embedding. During prototyping we developed several features that improved the performance score, namely *inbound dependencies* (ID), which represent the dependency label from the head to the role (dependent) and two trivial baselines: *Boolean* (B) and *123*.

We built a negative one-hot encoding feature vector to represent the *inbound* dependencies of the word corresponding to the role. Thus, 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. During our experiments for the development test, we also used the outbound dependencies, which represent the dependency label from the role (head) to the dependent words. So we used −1 for inbound and 1 for outbound. But since they did not perform well in comparison to inbound dependencies, they were not considered for submitted runs.

For the *Boolean* baseline, given the position of the verb in the sentence p_v and the position of the target token p_t, we assign the role 0 to t if $p_v < p_t$, otherwise 1. For the *123* baseline, we assign its index to each highlighted slot filler. For example, if five slots need to be labelled, the first one will be labelled as 1 and the last one will be labelled as 5.

4.2 Results and Discussion

Table 3 shows our results on Subtask B.2. We found that the trivial Boolean approach outper-

Method	Pu F_1	B^3 F_1
Agglomerative Clustering		
♀ w2v[C]+ID	**62.00**	**42.10**
⧖ ELMo[C]+ID	50.37	34.89
Logistic Regression		
⚡ ELMo[C+W+V]+ID+B+123	73.14	57.37
⧖ w2v[C+W+V]+ID+B+123	74.36	58.83
Cluster Per Dependency Role	56.05	39.03
🎨 Boolean Baseline	67.16	46.78
Inbound Dependencies (ID)	66.05	45.77
🏆 Winner	64.16	45.65

Table 3: Our results on Subtask B.2: Clustering Arguments of Verbs to Generic Roles. Purity F_1-score is denoted as *Pu F_1*, B-Cubed F_1-score is denoted as *B^3 F_1*. ♀ denotes *our* final submission (# 535480), ⚡ denotes a supervised *Logistic Regression* submission that does not comply to the task rules, ⧖ denotes *our* post-competition result, 🎨 denotes a baseline, and 🏆 denotes the submission of the winning team.

formed LPCFG (Kallmeyer et al., 2018) and all the standard baselines, including cluster per dependency role (`OneClustPerGrType`), on the development dataset.[4]

Similarly to our solution for Subtask A (Section 2), we tried different clustering algorithms to cluster arguments of verbs to generic roles and found that the best clustering performance is shown by agglomerative clustering with Euclidean affinity, Ward's method linkage, and two clusters. Our final submission (♀) used the combination of inbound dependencies and Word2Vec embedding for sentence (context, C), which performed marginally better than the cluster per dependency role (`OneClustPerGrType`) baseline, but still not better than such trivial baselines as *Boolean* or 123. Replacing Word2Vec with ELMo in our post-competition experiments have lowered the performance further.

In order to estimate our upper bound of the performance, we compared our best-performing clustering algorithm, i.e., agglomerative clustering, to a logistic regression model. We found that 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, we did not observe a noticeable difference between the per-

formance of the underlying embedding models.

As the model was trained on the development dataset that contained 20 roles in contrast to the test set which contained 32 roles, this approach has its limitations due to this difference of the number and meaning of roles. We believe that the performance could be improved using semi-supervised clustering methods, yet during prototyping with the pairwise-constrained k-Means algorithm (Basu et al., 2004) we did not observe any performance improvements.

5 Conclusion

We presented an approach for unsupervised semantic frame and role induction that uses word and context embeddings. It separates the task into two independent steps: verb clustering and role labelling, using combination of these embeddings enhanced with syntactical features. Our approach showed the best performance in Subtask B.1 and also finished as the runner-up in Subtask A of this shared task, and it can be easily extended to process other datasets and languages.

---

[4]On the development dataset for Subtask B.2, the *Boolean* baseline demonstrated B-Cubed F_1 = 57.98, while LPCFG and cluster per dependency role yielded F_1 = 40.05 and F_1 = 50.79, correspondingly.

References

Nikolay Arefyev, Pavel Ermolaev, and Alexander Panchenko. 2018. How much does a word weight? Weighting word embeddings for word sense induction. In *Computational Linguistics and Intellectual Technologies: Papers from the Annual International Conference "Dialogue"*, pages 68–84, Moscow, Russia. RSUH.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet Project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, ACL '98, pages 86–90, Montréal, QC, Canada. Association for Computational Linguistics.

Sugato Basu, Arindam Banerjee, and Raymond J. Mooney. 2004. Active Semi-Supervision for Pairwise Constrained Clustering. In *Proceedings of the 2004 SIAM International Conference on Data Mining*, SDM 2004, pages 333–344, Lake Buena Vista, FL, USA. SIAM.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Susan Windisch Brown, Dmitriy Dligach, and Martha Palmer. 2011. VerbNet Class Assignment as a WSD Task. In *Proceedings of the Ninth International Conference onComputational Semantics*, IWCS 2011, pages 85–94, Oxford, UK.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Laura Kallmeyer, Behrang QasemiZadeh, and Jackie Chi Kit Cheung. 2018. Coarse Lexical Frame Acquisition at the Syntax–Semantics Interface Using a Latent-Variable PCFG Model. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, *SEM 2018, pages 130–141, New Orleans, LA, USA. Association for Computational Linguistics.

Joel Lang and Mirella Lapata. 2010. Unsupervised Induction of Semantic Roles. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 939–947, Los Angeles, CA, USA. Association for Computational Linguistics.

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

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., Harrahs and Harveys, NV, USA.

Ashutosh Modi, Ivan Titov, and Alexandre Klementiev. 2012. Unsupervised Induction of Frame-Semantic Representations. In *Proceedings of the NAACL-HLT Workshop on the Induction of Linguistic Structure*, pages 1–7, Montréal, QC, Canada. Association for Computational Linguistics.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31(1):71–106.

Alexander Panchenko, Anastasia Lopukhina, Dmitry Ustalov, Konstantin Lopukhin, Nikolay Arefyev, Alexey Leontyev, and Natalia Loukachevitch. 2018. RUSSE'2018: A Shared Task on Word Sense Induction for the Russian Language. In *Computational Linguistics and Intellectual Technologies: Papers from the Annual International Conference "Dialogue"*, pages 547–564, Moscow, Russia. RSUH.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, et al. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Matthew 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, Volume 1 (Long Papers)*, NAACL-HLT 2018, pages 2227–2237, New Orleans, LA, USA. Association for Computational Linguistics.

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*, Minneapolis, MN, USA. Association for Computational Linguistics.

Karin Kipper Schuler. 2005. *VerbNet: A Broadcoverage, Comprehensive Verb Lexicon*. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA, USA.

Ivan Titov and Alexandre Klementiev. 2012. A Bayesian Approach to Unsupervised Semantic Role Induction. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL 2012, pages 12–22, Avignon, France. Association for Computational Linguistics.

Dmitry Ustalov, Alexander Panchenko, Andrei Kutuzov, Chris Biemann, and Simone Paolo Ponzetto. 2018. Unsupervised Semantic Frame Induction using Triclustering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, ACL 2018, pages 55–62, Melbourne, VIC, Australia. Association for Computational Linguistics.