ACCEPT at SemEval-2023 Task 3: An Ensemble-based Approach to Multilingual Framing Detection

Philipp Heinisch
Bielefeld University
pheinisch@techfak.uni-bielefeld.de

Moritz Plenz
Heidelberg University
plenz@cl.uni-heidelberg.de

Anette Frank
Heidelberg University
frank@cl.uni-heidelberg.de

Philipp Cimiano
Bielefeld University
cimiano@techfak.uni-bielefeld.de

Abstract

This paper describes the system and experimental results of an ensemble-based approach to multilingual framing detection for the submission of the ACCEPT team to the SemEval-2023 Task 3 on Framing Detection (Subtask 2). The approach is based on an ensemble that combines three different methods: a classifier based on large language models, a classifier based on static word embeddings, and an approach that uses external commonsense knowledge graphs, in particular, ConceptNet. The results of the three classification heads are aggregated into an overall prediction for each frame class.

Our best submission yielded a micro F_1-score of 50.69% (rank 10) and a macro F_1-score of 50.20% (rank 3) for English articles. Our experimental results show that static word embeddings and knowledge graphs are useful components for frame detection, while the ensemble of all three methods combines the strengths of our three proposed methods. Through system ablations, we show that the commonsense-guided knowledge graphs are the outperforming method for many languages.

1 Introduction

The SemEval 2023 Task 3 by Piskorski et al. (2023) (Subtask 2) consisted in detecting different types of frames in newspaper articles in six different languages (English, French, German, Italian, Polish, and Russian). Beyond these seen languages, three more languages were added in the testing phase (Spanish, Greek, and Georgian).

Different framings represent different perspectives on one and the same reported event. Being able to categorize them helps to cluster such news articles and guides the analyses of different aspects (Card et al., 2015). In the particular task at hand, the applied generic MediaFrames set (Boydstun et al., 2014) categorizes articles into 14 frame classes: “Economic”, “Capacity and resources”, “Morality”, “Fairness and equality”, “Legality, constitutionality and jurisprudence”, “Policy prescription and evaluation”, “Crime and punishment”, “Security and defense”, “Health and safety”, “Quality of life”, “Cultural identity”, “Public opinion”, “Political” and “External regulation and reputation”.

In this paper, we describe the system and experimental results for submitting team ACCEPT to the shared task. Our approach consists of an ensemble of three different methods, each relying on a fully connected classification layer on top of i) a large language model (LM), ii) static word embeddings and iii) a Graph Neural Network (GNN) architecture exploiting a commonsense knowledge graph (CSKG), ConceptNet in particular. To handle inputs in multiple languages, as required by the task, we explore language-specific and language-agnostic methods, and determine the best configuration for each language. As an interesting result, we found that the overall best configuration is language-specific by using commonsense knowledge graphs for inference, while LMs do not perform well without further guidance.

We publish our highly adjustable ensemble framework, including the three proposed methods at https://github.com/phhei/SemEval233FramingPublic.

2 Related work

The challenge of detecting and tracing frames in newspaper articles goes back to seminal work of Boydstun et al. (2014) and Card et al. (2015), who provided a dataset with articles annotated with frames. One of the first automatic approaches to inferring frames for articles was proposed by Naderi and Hirst (2017), who relied on recurrent networks based on static word embeddings. This architecture performed with an accuracy of up to 71% in distinguishing between the five most frequent MediaFrames on the text-span level. Building on these first encouraging results, the task of frame detection also received attention in the field of argument mining (Ajjour et al., 2019; Ruckdeschel and Wiedemann, 2022) and even aspect-tailored language generation with pretrained large language models (Schiller et al., 2021; Chen et al., 2021) that were also successfully applied for frame classification (Jurkschat et al., 2022).

Beyond exploiting text only, some methods in argument mining have integrated external knowledge and found it to improve performance on tasks such as the prediction of the quality of argumentative conclusions (Heinisch et al., 2022), or the prediction of sentiment-related human needs categories following the hierarchy of human needs (Maslow, 1943) and basic motives (Reiss, 2004), in the analysis of narratives (Paul and Frank, 2019). However, to the best of our knowledge, the inclusion of background knowledge for framing prediction, as proposed in our work, has not been investigated before.

Inspired by the success of combining different approaches to the task of frame detection (Heinisch and Cimiano, 2021) and related shared tasks (Herath et al., 2020; Raj et al., 2020; Luo et al., 2022), we experiment with various method selections and hyperparameter settings to explore advantages and disadvantages across different methods for detecting frames in newspaper texts.

3 Dataset

We used the provided dataset for the shared task by (Piskorski et al., 2023), containing 2,069 propaganda or political news article published between 2020 and 2022, split into 1,251 articles for training, 361 articles for development, and 457 for training. Each article is annotated with 1-10 out of 14 different frame classes. While the annotators select some frame classes frequently (e.g. “Political” and “Security and Defense”), other frame classes are about four times less selected (e.g.: “Cultural identity” and “Public opinion”).

The articles are multi-lingual, including six languages in all splits (English, French, German, Italian, Polish, and Russian) and three languages (Spanish, Greek, and Georgian) only in the test split. The average number of tagged frames classes per article differs across languages (e.g., 1.2 frame classes per Russian article and 5.9 frame classes per Polish article, both averaged on the test split).

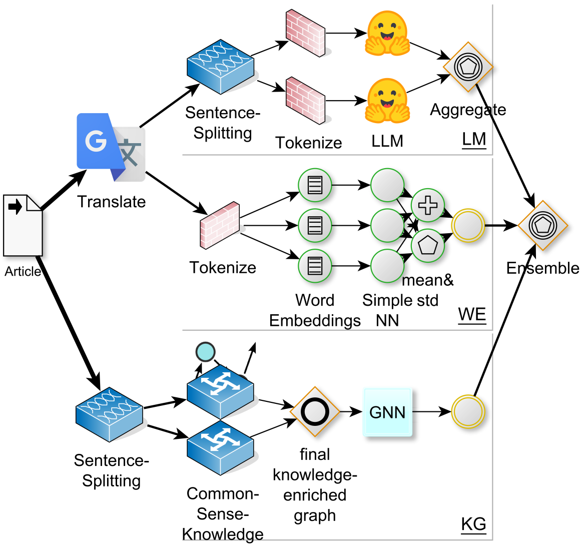

Figure 1: Our general ensemble approach in our standard configuration

4 System Overview

We present an ensemble approach that combines three different multilabel classification models to predict the frames of a given news article. The methods are based on large language models (LMs), static word embeddings (WE) and GNNs using commonsense knowledge subgraphs (KG), as illustrated in Figure 1. They are described in detail below.

Method using Large Language Models (LM) We fine-tune existing pre-trained large language models for the task of framing detection. As many language models have an input size limit of 512 tokens, which is not sufficient to process the majority of the articles, besides experimenting with special language models that are able to process longer sequences (Beltagy et al., 2020; Zaheer et al., 2020; Guo et al., 2022), we chunk the input into sentences and accumulate the predictions for each sentence. We use a standard linear classification layer on top of the language model for multilabel classification.

Method using Static Word Embeddings (WE) Motivated by Heinisch and Cimiano (2021), we experimented with static word embeddings. We map each token to a static word embedding using the English-only GloVe-Embeddings (Pennington et al., 2014) as well as the multilingual word embeddings of Lample et al. (2018). We further encode the word embeddings by a neural feed-forward layer. To adapt to the varying length of the input texts, we compute the component-wise average and standard deviation for the encodings of all words (reflecting the content, length, and variance of the input text). We concatenate this average vector and the deviation vector, processed by a simple feed-forward classification layer.

Method using a Knowledge Graph model (KG) We designed this model to incorporate commonsense knowledge by retrieving subgraphs of ConceptNet (CN) (Speer et al., 2017) that are related to chunks of text, and processing them using graph neural networks (GNNs). In CN, nodes are *concepts* in free-form text, and edge labels specify *relations* between pairs of concepts.

In order to extract relevant subgraphs from CN, we rely on a method proposed in Plenz et al. (2023). The method extracts contextualized subgraphs from CN that enrich the texts with relevant commonsense knowledge that i) matches the meaning of the texts, including ii) potentially implicit inferences. In order to select contextually relevant CN subgraphs, the method internally computes shortest paths weighted by semantic similarity, which strengthens the semantic relatedness and hence the relevance of the extracted knowledge subgraphs, compared to traditional unweighted (shortest) path searches. Semantic similarity is estimated using SBERT (Reimers and Gurevych, 2019).

In order to apply this method to longer texts, we first split the articles into sentences and construct contextualized CN-subgraphs between all pairs of subsequent sentences. We then merge the obtained graphs into one single graph that spans the entire article, by merging matching nodes from different graphs. Some CN relations have clear head and tail concepts (e.g. `IsA`), i.e., CN contains directed edges. However, for the GNN we consider all edges to be undirected, to ease knowledge propagation in the graph. GNNs can not process text, and hence, we need to compute meaningful node vectors that can be processed by a GNN. We apply semantic embeddings obtained from SBERT (Reimers and Gurevych, 2019). Note that the SBERT model is frozen, i.e. the weights of the SBERT model itself are not updated during training.

To solve the frame prediction task, we train diverse GNN types on the CN subgraph(s), where each layer consists of the GNN layer followed by a nonlinearity. In the final layer, we apply sigmoid instead of the non-linearity used in the other layers.

Addressing multilingual frame detection For multilingual frame detection, we apply different strategies. In a **language-agnostic approach**, we leverage high-quality MT systems in order to translate non-English articles to English.[1] This has the advantage that the training data available in English for frame detection grows considerably (from 433 articles to 1,238 articles). In a **language-sensitive approach** we leverage multilingual static word embeddings and multilingual pretrained language models that map input texts to a cross-lingual embedding space that can be used for language-agnostic further processing.

In the knowledge graph model approach, we pursue a **hybrid strategy**: We first use a multilingual SBERT to extract contextualized subgraphs from the English part of CN. Then, we use a monolingual SBERT to compute the node features that the GNN processes.

5 Experimental Setup and Results

We participated in *Subtask 2: Framing Detection* and experimented with various configurations. Our ensemble framework is implemented in Python 3.9 based on the `pytorch`-library (Paszke et al., 2019). In training, we used a standard learning rate of $5e-4$ or $5e-5$ in case of LM-based method. We trained up to eight epochs using an early-stopping criterion to load the best-performing checkpoint with respect to the product of micro-F_1 and macro-F_1 on the development data.

To aggregate the frame predictions of our ensemble, we average the predictions made by the applied methods. Additionally, in the case of our final submissions, we experimented with weighting the frame prediction of each method by introducing a trainable weight vector.

In inference, we determine optimal probability thresholds to decide when to select a frame as observable in an article. We compute one threshold for each frame class separately using the development data as well. However, to avoid an empty or unrealistically large set of finally predicted frames, we enforce selecting between 1 and 8 frames per article based on a ranked list of frames, sorted by their prediction likelihood.

[1] We used the Cloud Translation API `https://cloud.google.com/translate/docs`

5.1 Selected setup of the different methods

Based on the best-performing single method setups with respect to the micro-F_1 on the development split, we fix the hyperparameters described below (Appendix B.3 reports a detailed hyperparameter study of the single methods).

For our method based on **large language models**, we used the `transformers` library (Wolf et al., 2020) and experimented with several transformers as different versions of RoBERTa and transformers which are able to process longer sequences than 512 tokens. We obtained the best results with `roberta-base`, applied for every single sentence of the article. To aggregate the returned set of frame predictions, we applied the harmonic mean.

For the method using **static word embeddings**, we used the `glove.840B.300d`[2] word embeddings. To introduce non-linearity in the learning module of this method, we experimented with different activation functions that we applied after the linear word vector encoding layer. We observed outperforming results with the tanh-function on the development split.

For the **Knowledge Graph model**, we use the `spaCy`-library by Honnibal et al. (2020) to split each article into sentences, for all languages except Georgian. Since Georgian is not supported by `spaCy`, we split the Georgian articles by newlines, which correspond to paragraphs. Most paragraphs are relatively short, consisting of only a few sentences, so this crude approximation is still capable of splitting articles into reasonable segments. With the obtained sentences, we use the subgraph extraction method described in Section 4 to enrich the articles with contextually relevant ConceptNet (Speer et al., 2017) subgraphs. To compute the similarity between article segments and CN concepts, we employ a multilingual SBERT (Reimers and Gurevych, 2020) model[3]. Independently from the articles' language, the extracted CN subgraphs consist of English concepts only. We, therefore, use a monolingual English SBERT (Reimers and Gurevych, 2019) model[4] to obtain node features.

We use the language-agnostic approach for language models and static word embeddings in our final submissions for the shared task and the ablation study. Thus, we used all (translated) articles for training, development, and testing, respectively. For the knowledge graph model, we always use the hybrid strategy that processes all articles in their original language (see Section 4).

5.2 Results

In this section, we present our results, reporting first the scores of our submitted systems in Section 5.2.1. Since the submission system of the shared task restricts the submission to one prediction file per language, we present other method combinations and ablation studies in Section 5.2.2, including multiple runs per configuration to estimate variance.

To measure the performance of our systems, we follow the official metrics used by the shared task, namely the aggregated F_1-score across all 14 frame classes.

5.2.1 Shared Task Results

In order to select which configuration of our proposed ensemble to submit for each language, we rank the configurations for each language according to the micro-F_1 score on the development split. In this split, the ensemble combining all three methods, as well as an ensemble combining static word embeddings and the knowledge graph model, were determined as the best configuration for at least one out of the six development languages. To estimate the transferability of our system to the three test-only languages, we run experiments by excluding all Polish and German articles from training, monitoring the prediction performance of our system on these held-out languages. Choosing the top-ranked ensemble configuration for each language, we finally trained the superior ensembles with a new train-dev-split which adds 50% of the development data to the training data. Table 1 reports our final scores for the final configurations submitted.

Regarding the micro-F_1 score, we accomplish scores between 22% (Georgian), and 50.98% (Polish) by combining all three methods with the weighted average, placing us between ranks 10. (English, French, and Spanish) and 17. (Polish). According to the macro-F_1 score, we range between 24.85% (Russian) and 50.2% (English) by combining only static word embeddings and knowledge graphs and all three weighted methods to-

[2] `https://nlp.stanford.edu/data/glove.840B.300d.zip` – to process articles in their original languages, we have to switch to MUSE (`https://github.com/facebookresearch/MUSE`)

[3] We use `distiluse-base-multilingual-cased-v2` from `https://www.sbert.net`

[4] The SBERT model we use is `all-mpnet-base-v2` from `https://www.sbert.net`

Language	Ensemble	F1-micro	F1-macro
English	LM*+WE+KG	50.69	50.20
Italian	LM*+WE+KG	49.47	43.92
Russian	WE+KG	25.37	24.85
Polish	LM+WE+KG	50.98	49.03
French	LM*+WE+KG	46.87	42.85
German	LM+WE+KG	49.61	45.97
Spanish*	LM*+WE+KG	38.81	38.66
Greek*	LM*+WE+KG	35.52	36.95
Georgian*	LM+WE+KG	22.00	28.98

Table 1: Results of official submission for the shared task. In the case of an ensemble of three methods, we applied an aggregator of weighted average. LM* marks LM modules truncating input articles exceeding 64 sentences, exclusively applied on the final submissions.

Configuration			∅ languages	
LM	WE	KG	micro	macro
✓	✓	✓	43.88	39.79
✗	✓	✓	40.91	39.48
✓	✗	✓	40.70	36.53
✓	✓	✗	36.11	26.52
✓	✗	✗	24.74	15.92
✗	✓	✗	41.68	37.48
✗	✗	✓	**45.31**	**41.12**

Table 2: Ablation study – language models and word embeddings using translations of non-English articles.

gether, respectively. Compared to submissions from other participating teams, our approach stands out in being relatively accurate for rare frame classes, resulting ranks ranging between rank 3 (English) and 15 (German) according to the micro-F_1 score.

5.2.2 System exploration beyond official task results

To gain a better understanding of the contribution, advantages, and disadvantages of each method, we perform an ablation study of our ensemble using the official train-dev-split, reporting F_1-scores on the test data, averaging results over 10 random seeds and all languages.

While using the provided train-dev-split, the ensemble of all three methods has a language-averaged F_1-performance of 43.88% and 39.79% for the frame-class-micro-average and frame-class-macro-average, respectively. We observe lower F_1 scores by removing one or more methods from our ensemble: removing the language model lowers the micro-F_1 score by 2.97%-points, discarding the static word embeddings decreases the micro-F_1 score by 3.18%-points, and removing the knowledge graph model results in losing 7.77%-points regarding the micro-F_1 score. Hence, the knowledge graph model has the highest impact for both micro-F_1 and macro-F_1. We observe the same pattern when analyzing our three methods as single classifiers, starting with a mediocre performance of 24.74% micro-F_1 using an LM only, 41.68% micro-F_1 using static word embeddings, and 45.31% micro-F_1 using the knowledge graph model. Hence, using a single knowledge-graph model outperforms the ensemble that, in addition, includes the two remaining methods, by 1.43%-points and 1.33%-points for the frame-class-micro-average and frame-class-macro-average, respectively. However, this observation does not hold for each particular language. The complete ensemble performs slightly better for German and Spanish and especially provides better results for rare frame classes in Italian and Polish. Despite those edge cases, the knowledge graph model performs well, especially in English, yielding the overall best results with 55.26% and 50.46% for micro-F_1 and macro-F_1, respectively, in this language.

However, there is one language (Georgian) in which the knowledge graph model worsens the overall performance: the best micro-F_1 score of 33.61% is observable with static word embeddings standalone, and the configuration with the best macro-F_1 score of 30.17% uses an ensemble without knowledge graph models. Since Georgian is an under-resourced language (resulting in the absence of sentence tokenizers) that is not related to any of the other languages, processing the non-translated articles using the knowledge graph model does potentially incur errors. Additionally, the usefulness of a commonsense knowledge base that is dominated by the western culture is limited in the case of Georgian due to the existence of cultural differences in commonsense (Anacleto et al., 2006) – also observable in an uncommon frame class distribution in this language due to often higher macro-F_1 scores in comparison to the micro-F_1 scores.

Appendix B lists further system explorations outside the competition.

6 Conclusion

Predicting the occurring frame classes in long multilingual articles is challenging due to the variety of articles and differences between languages and cultures. This paper describes our participation in the SemEval-2023 Task 3 for which we proposed an ensemble combining three different methods (large language models, static word embeddings, and knowledge graphs). In order to handle inputs in all relevant languages, we leveraged machine translation software as a preprocessing step or alternatively relied on pre-trained multilingual language models that map texts into a language-agnostic semantic space.

Ablation analysis of our ensemble suggests that using a graph neural network on extracted commonsense knowledge subgraphs provides the strongest results for most of the nine languages. This shows the strength of knowledge-based methods for frame detection in multilingual settings.

Acknowledgements

This work has been funded by DFG within the project ACCEPT, which is part of the priority program "Robust Argumentation Machines" (RATIO).

References

Yamen Ajjour, Milad Alshomary, Henning Wachsmuth, and Benno Stein. 2019. Modeling frames in argumentation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2922–2932, Hong Kong, China. Association for Computational Linguistics.

Junia Anacleto, Henry Lieberman, Marie Tsutsumi, Vânia Neris, Aparecido Carvalho, Jose Espinosa, Muriel Godoi, and Silvia Zem-Mascarenhas. 2006. Can common sense uncover cultural differences in computer applications? In *Artificial Intelligence in Theory and Practice*, pages 1–10, Boston, MA. Springer US.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150*.

Amber E. Boydstun, Dallas Card, Justin Gross, Paul Resnick, and Noah A. Smith. 2014. Tracking the development of media frames within and across policy issues.

Dan Busbridge, Dane Sherburn, Pietro Cavallo, and Nils Y Hammerla. 2019. Relational graph attention networks.

Dallas Card, Amber E. Boydstun, Justin H. Gross, Philip Resnik, and Noah A. Smith. 2015. The media frames corpus: Annotations of frames across issues. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 438–444, Beijing, China. Association for Computational Linguistics.

Wei-Fan Chen, Khalid Al Khatib, Benno Stein, and Henning Wachsmuth. 2021. Controlled neural sentence-level reframing of news articles. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2683–2693, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mandy Guo, Joshua Ainslie, David Uthus, Santiago Ontanon, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. 2022. LongT5: Efficient text-to-text transformer for long sequences. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 724–736, Seattle, United States. Association for Computational Linguistics.

Philipp Heinisch and Philipp Cimiano. 2021. A multi-task approach to argument frame classification at variable granularity levels. *it - Information Technology*, 63(1):59–72.

Philipp Heinisch, Anette Frank, Juri Opitz, Moritz Plenz, and Philipp Cimiano. 2022. Overview of the validity and novelty prediction shared task. In *Proceedings of the 9th Workshop on Argument Mining*, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Mahen Herath, Thushari Atapattu, Hoang Anh Dung, Christoph Treude, and Katrina Falkner. 2020. AdelaideCyC at SemEval-2020 task 12: Ensemble of classifiers for offensive language detection in social media. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1516–1523, Barcelona (online). International Committee for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python. Technical report, ExplosionAI GmbH, Alexanderstr. 7, 10178 Berlin, Germany.

Lena Jurkschat, Gregor Wiedemann, Maximilian Heinrich, Mattes Ruckdeschel, and Sunna Torge. 2022. Few-shot learning for argument aspects of the nuclear energy debate. In *Proceedings of the Language Resources and Evaluation Conference*, pages 663–672, Marseille, France. European Language Resources Association.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Xiang Luo, Yanqing Niu, and Boer Zhu. 2022. TCU at SemEval-2022 task 8: A stacking ensemble transformer model for multilingual news article similarity. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1202–1207, Seattle, United States. Association for Computational Linguistics.

Abraham Harold Maslow. 1943. A theory of human motivation. *Psychological review*, 50(4):370.

Nona Naderi and Graeme Hirst. 2017. Classifying frames at the sentence level in news articles. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 536–542, Varna, Bulgaria. INCOMA Ltd.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Debjit Paul and Anette Frank. 2019. Ranking and selecting multi-hop knowledge paths to better predict human needs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3671–3681, Minneapolis, Minnesota. Association for Computational Linguistics.

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, Doha, Qatar. Association for Computational Linguistics.

Jakub Piskorski, Nicolas Stefanovitch, Giovanni Da San Martino, and Preslav Nakov. 2023. Semeval-2023 task 3: Detecting the category, the framing, and the persuasion techniques in online news in a multi-lingual setup. In *Proceedings of the 17th International Workshop on Semantic Evaluation*, SemEval 2023, Toronto, Canada.

Moritz Plenz, Juri Opitz, Philipp Heinisch, Philipp Cimiano, and Anette Frank. 2023. Similarity-weighted construction of contextualized commonsense knowledge graphs for knowledge-intense argumentation tasks. In *Proceedings of the 61th Annual Meeting of the Association for Computational Linguistics*, Toronto, Canada. Association for Computational Linguistics.

Mayank Raj, Ajay Jaiswal, Rohit R.R, Ankita Gupta, Sudeep Kumar Sahoo, Vertika Srivastava, and Yeon Hyang Kim. 2020. Solomon at SemEval-2020 task 11: Ensemble architecture for fine-tuned propaganda detection in news articles. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1802–1807, Barcelona (online). International Committee for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.

Steven Reiss. 2004. Multifaceted nature of intrinsic motivation: The theory of 16 basic desires. *Review of general psychology*, 8(3):179.

Mattes Ruckdeschel and Gregor Wiedemann. 2022. Boundary detection and categorization of argument aspects via supervised learning. In *Proceedings of the 9th Workshop on Argument Mining*, pages 126–136, Online and in Gyeongju, Republic of Korea. International Conference on Computational Linguistics.

Benjamin Schiller, Johannes Daxenberger, and Iryna Gurevych. 2021. Aspect-controlled neural argument generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–396, Online. Association for Computational Linguistics.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15*, pages 593–607. Springer.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 4444–4451. AAAI Press.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. *International Conference on Learning Representations*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi 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 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*, pages 38–45, Online. Association for Computational Linguistics.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33.

A Implementation of our ensembling framework

Our implementation of the ensembling approach is highly adjustable by separating the task of frame prediction into an arbitrary number of (different configured) methods that should be applied and an optionally trainable aggregator which combines all frame predictions of each applied method to a final frame prediction.

A.1 Implemented aggregators

Our framework implements several kinds of aggregators: besides calculating the average, we introduce the harmonic mean and several pooling aggregators. We also introduce an option to learn weights for each module in order to learn an effective prioritization of N module predictions.

In particular, given N frame probability predictions $\{n = (p_n(c_1), \cdots, p(c_{14})|n \in \mathcal{N}\}$, we implemented and experiment with the following aggregators:

- Average-Aggregators
  - Standard-Mean: $p(c_i) = \frac{\sum_{n \in \mathcal{N}} p_n(c_i)}{N}$
  - Harmonic-Mean: $p(c_i) = \frac{N}{\sum_{n \in \mathcal{N}} \frac{1}{p_n(c_i)}}$
- Pooler-Aggregators (potentially exclude modules from weight adaptation when their predictions are not pooled during training)
  - Max-Pooling: $p(c_i) = \max(\{p_n(c_i)|n \in \mathcal{N}\})$
  - Min-Pooling: $p(c_i) = \min(\{p_n(c_i)|n \in \mathcal{N}\})$
- SoftPooler-Aggregators
  - SoftMax-Pooling: $p(c_i) = \max(\{p_n(c_i)|n \in \mathcal{N}\}) - \alpha \frac{\sum_{m \in \mathcal{N}} |p_m(c_i) - \max(\{p_n(c_i)|n \in \mathcal{N}\})|^\beta}{N}$
  - SoftMin-Pooling: $p(c_i) = \min(\{p_n(c_i)|n \in \mathcal{N}\}) + \alpha \frac{\sum_{m \in \mathcal{N}} |p_m(c_i) - \min(\{p_n(c_i)|n \in \mathcal{N}\})|^\beta}{N}$

A.2 Training and loss

In training, we applied a standard multi-class loss, namely the cross entropy loss for each frame class. However, due to our multi-module architecture, we have several options *where* to calculate the loss, resulting in different flows of backpropagation. Let

$$Loss = \lambda Loss_{\text{final}} + (1 - \lambda) \left(\omega_p \sum_{p \in P} Loss_p \right) \tag{1}$$

with $Loss_{\text{final}}$ as loss using the final aggregated frame prediction and $Loss_p$ as loss applied for every single module separately with a method-specific scaling factor ω_p in order to address different sensitivities regarding learning rates. One option is to set $\lambda = 1$, which only applies the loss at the end by using the final aggregated frame prediction, which trains the single modules only indirectly and potentially suffers from the combination of random weight initialization and different sensitivities regarding learning rates. Another option is to set $\lambda = 0$. In this case, we have an approach that only tunes its single parts but not the aggregation of those. To combine the advantages of both options, we follow the strategy of first tuning the single modules separately and incrementally applying the loss toward the aggregation of the modules to avoid unwanted training effects by distortion due to a single modules with a random weight initialization by smoothly increasing λ from 0 to 1.

A.3 Inference

For inference, a frame class is predicted when its final predicted probability exceeds a frame-class individual probability threshold. This threshold is F_1-score-optimized using the development data.

The restriction of predicting at least one at most eight frame classes for an article results from a manual inspection of the development data. While all articles are labeled with at least one frame class, the largest observed frame set contains ten out of 14 frame classes. However, nine and ten frame classes are long-tail exceptions (five and one articles in the entire dataset, respectively). Hence, for the sake of precision, we limited the maximum amount of predicted frame classes to eight.

B Further system explorations outside competition

B.1 Further studies regarding the submitted systems

Since we were limited to submitting only a single system (configuration/ run) for each language in Section 5.2.1, we explore the performance across our three preferred method ensembles, based on the insights on the development split, in this section. We average the results over ten runs per method ensemble in order to cancel out the effects of the random weight initialization of our non-pretrained neural parts, trained with the modified train-development split explained in Section 5.2.1, having 90% of all articles as training data and the remaining 10% of all articles as development data (excluding test data).

Table 3 lists or results. While the scores are comparable to the submission results listed in Table 1 for many languages, the detailed insights from this study counter our insights from only looking at the development data. Introducing the weighted average (and hence, more trainable parameters) worsens the generalizability in most cases. The decrease is often minor (especially in English) but tends to be severe in the unseen languages (in Spanish, the micro-F_1 score is reduced by 2.21%-points, and in Georgian, the micro-F_1 score is reduced by 3.14%-points). Especially the detriment of Georgian frame detection using trained weights for the average aggregation is explainable by the learned emphasis on the knowledge graph model, which has a poor performance in Georgian. Another insight is the increased micro-F_1 score when we ensemble only the static word embeddings and the knowledge graph model in English (52.36%, increase of 0.72%-points), in Polish (59.16%, increase of 0.28%-points) and in Greek (38.49%, increase of 0.06%-points) but not in Russian (24.7%, decrease of 2.88%-points).

B.2 Further insights into the ablation study

In order to provide further insights into our ablation study presented in Section 5.2.2, we present in Table 4 an extension of Table 2 by showing the performances for each language including the standard derivation across ten runs using the official train-dev-split. An interesting insight from this detailed view is the high derivation when relying on large language models. We observe that large language models are often stuck in local optima with simple repetitive prediction patterns. Depending on the preferred set of predicted frames, the F_1-scores vary much.

B.3 Hyperparameter studies of our three methods

B.3.1 Large language models

To analyze large language models further, we explore in Table 5 additional settings, trained on the official train-dev-split and evaluated on the test split across five runs, averaged on all nine languages. To include a test regarding the language-sensitive approach, we use the multilingual pretrained `xlm-roberta-base` instead of `roberta-base`.

The scores on test data contrast the performance on development data due to a stronger tendency to fall back into repetitive predictions for articles and topics, which are neither seen in training nor for model and threshold selection. While using the average as the aggregation of split predictions or applying the language-sensitive approach outperforms the language-agnostic harmonic-mean-averaged setting by +4.45%-points and +5%-points, respectively, regarding the micro-F_1 score, other tendencies were confirmed by the test split. For example, not chunking the article into sentences and hence, missing the later text parts of large articles reduces the overall performance (-5.6%-points and -3.05%-points regarding the micro-F_1 and macro-F_1, respectively).

B.3.2 Static word embeddings

To get deeper insights into our static word embeddings processed by our shallow neural net, we explore in Table 6 additional settings, trained on the official train-dev-split and evaluated on the test split across five runs, averaged on all nine languages. To include a test regarding the language-sensitive approach, we use the multilingual adjusted `MUSE`-word-vectors (`https://`

		LM+WE+KG w/ avg	LM+WE+KG w/ weighted avg	WE+KG w/ avg
English	micro	51.64 ± 2.0	51.67 ± 3.1	**52.36** ± 1.7
	macro	**47.97** ± 3.3	47.66 ± 3.1	47.81 ± 2.8
Italian	micro	**53.43** ± 1.9	51.22 ± 3.7	51.94 ± 1.7
(translated for LM/WE)	macro	**46.62** ± 1.9	45.06 ± 3.6	44.65 ± 3.6
Russian	micro	**27.58** ± 2.9	27.07 ± 2.9	24.7 ± 1.4
(translated for LM/WE)	macro	**25.28** ± 2.0	24.19 ± 2.9	22.98 ± 1.4
Polish	micro	58.88 ± 2.0	57.54 ± 2.6	**59.16** ± 3.2
(translated for LM/WE)	macro	**54.90** ± 1.7	53.51 ± 2.5	52.19 ± 4.2
French	micro	**47.79** ± 1.3	47.06 ± 1.6	47.78 ± 1.4
(translated for LM/WE)	macro	**47.20** ± 2.5	45.12 ± 2.4	45.84 ± 2.6
German	micro	**54.51** ± 1.3	52.25 ± 3.5	53.51 ± 1.2
(translated for LM/WE)	macro	**50.53** ± 1.8	47.95 ± 3.3	47.48 ± 2.9
Spanish*	micro	**40.61** ± 0.6	38.40 ± 4.0	39.13 ± 2.0
(translated for LM/WE)	macro	**38.10** ± 1.1	36.43 ± 3.9	37.34 ± 1.6
Greek*	micro	38.43 ± 2.9	37.47 ± 4.6	**38.49** ± 3.1
(translated for LM/WE)	macro	**36.96** ± 1.6	35.83 ± 3.0	35.96 ± 2.3
Georgian*	micro	**23.08** ± 2.4	19.94 ± 2.0	19.29 ± 1.9
(translated for LM/WE)	macro	**24.97** ± 3.2	23.54 ± 3.8	21.79 ± 3.7

Table 3: Results for 90%-10% train-dev-split, averaged over 10 seeds. Languages marked with * were without training and development data

github.com/facebookresearch/MUSE) instead of the GloVe-word-vectors (https://nlp.stanford.edu/data/glove.840B.300d.zip). However, MUSE does not support Polish and Georgian. For those two languages, we always use translated articles, even for the language-sensitive approach.

The results show interesting trends. While the most complex activation functions (tanh, Sigmoid) are superior on the development split, the performance on the test split votes for the simple activation functions as the ReLU-function (+4.2%-points and +5.28%-points regarding the micro-F_1 and macro-F_1, respectively). Even disabling an activation function (equal to the use of a linear identity activation function) outperforms the tanh-activation function on the test split (+3.81%-points and +1.17%-points regarding the micro-F_1 and macro-F_1, respectively). The option of splitting the articles into sentences, predicting the frame for each sentence, and then averaging all predictions tends to reduce the misses of infrequent frame classes with respect to the test split (+1.37%-points and +3.05%-points regarding the micro-F_1 and macro-F_1, respectively). Although these contrasting insights regarding the performance on the test data, the low performance using the language-sensitive approach is confirmed (-6.45%-points and -10.02%-points regarding the micro-F_1 and macro-F_1, respectively). The multi-lingual MUSE-word-vectors are more sparse than the English GloVe-word-vectors and suffer from a smaller text corpus.

B.3.3 Knowledge graph model

We process the extracted CN subgraphs with GNNs, which can leverage the graph's structure as well as linguistic content via the SBERT embeddings. However, we do *not* additionally incorporate graph features explicitly.

As GNN layer we experiment with GCN (Kipf and Welling, 2017), R-GCN (Schlichtkrull et al., 2018), GAT (Veličković et al., 2018) and R-GAT (Busbridge et al., 2019). GCN and GAT are edge label agnostic, i.e., they only consider node features. By contrast, R-GCN and R-GAT train one set of weights for each edge type. Some relations in CN are rare, but we group similar relations for R-GCN and R-GAT, which ensures sufficient training data for each group of relations. Furthermore, for frame prediction it is perhaps not necessary to have a fine-grained differentiation between different relations, e.g. the difference between HasSubevent and HasFirstSubevent is most likely not relevant for frame prediction.

Grouping of CN relations for R-GCN and R-GAT We group CN relations into sets for the R-GCN and R-GAT to avoid sparsity for some of them. We test two different groupings: We either (i) split the relations in two groups, depending on whether they describe dissimilarity or similarity between concepts or (ii) group similar relations together, yielding four groups in total.

Configuration		3 Methods	2 Methods			1 Method		
LM		✓	✗	✓	✓	✓	✗	✗
WE		✓	✓	✗	✓	✗	✓	✗
KG		✓	✓	✓	✗	✗	✗	✓
English	micro	51.64 ± 2.3	52.8 ± 1.2	46.38 ± 7.5	39.02 ± 8.9	31.83 ± 7.4	47.08 ± 1.7	**55.26** ± 1.6
	macro	46.69 ± 3.3	50.27 ± 1.5	40.66 ± 8.3	27.30 ± 7.7	18.51 ± 7.2	41.96 ± 1.9	**50.46** ± 2.5
Italian	micro	52.02 ± 2.8	50.2 ± 1.0	49.25 ± 5.1	43.22 ± 5.1	25.37 ± 6.6	49.28 ± 3.7	**54.19** ± 1.4
(translated for LM/WE)	macro	**45.65** ± 4.0	44.82 ± 1.2	41.38 ± 4.9	28.66 ± 6.9	14.14 ± 5.9	43.1 ± 3.8	45.44 ± 1.6
Russian	micro	27.55 ± 2.0	22.75 ± 1.8	22.99 ± 3.4	22.14 ± 3.7	11.67 ± 4.5	21.58 ± 3.3	**28.73** ± 2.3
(translated for LM/WE)	macro	24.42 ± 3.2	22.71 ± 1.8	21.05 ± 3.1	17.63 ± 4.1	10.48 ± 4.7	22.07 ± 2.8	**26.75** ± 2.1
Polish	micro	53.14 ± 4.0	53.91 ± 1.9	53.83 ± 1.9	44.64 ± 11.2	37.62 ± 11.5	54.19 ± 1.1	**55.01** ± 2.0
(translated for LM/WE)	macro	**49.47** ± 4.8	48.99 ± 1.8	47.58 ± 3.1	31.82 ± 10.4	23.04 ± 9.1	47.38 ± 3.2	48.99 ± 1.4
French	micro	46.31 ± 2.9	47.05 ± 0.8	45.68 ± 2.6	32.81 ± 4.6	26.62 ± 5.8	41.01 ± 1.3	**52.30** ± 1.8
(translated for LM/WE)	macro	45.25 ± 4.3	46.78 ± 1.0	42.49 ± 3.6	24.19 ± 7.6	14.2 ± 5.6	39.43 ± 2.5	**50.43** ± 2.2
German	micro	53.02 ± 4.0	53.03 ± 2.2	51.72 ± 3.3	45.23 ± 10.9	34.43 ± 9.2	**54.75** ± 2.9	52.45 ± 2.2
(translated for LM/WE)	macro	48.49 ± 4.8	**50.23** ± 1.7	46.99 ± 3.5	31.63 ± 9.5	21.06 ± 8.1	48.37 ± 4.7	47.82 ± 1.6
Spanish*	micro	**43.55** ± 2.5	36.61 ± 2.1	37.37 ± 4.6	31.69 ± 3.7	22.37 ± 3.7	34.28 ± 2.9	42.81 ± 2.6
(translated for LM/WE)	macro	**40.70** ± 3.1	37.28 ± 2.2	35.42 ± 4.3	25.44 ± 6.7	13.52 ± 2.7	35.8 ± 2.4	39.10 ± 1.7
Greek*	micro	37.79 ± 3.1	36.52 ± 2.2	40.33 ± 3.5	34.55 ± 8.9	20.84 ± 4.5	39.38 ± 2.5	**43.33** ± 2.0
(translated for LM/WE)	macro	35.52 ± 3.5	35.93 ± 2.4	36.39 ± 4.4	21.88 ± 6.7	11.22 ± 4.3	35.1 ± 2.3	**38.67** ± 1.5
Georgian*	micro	20.92 ± 3.0	20.86 ± 1.6	18.72 ± 4.6	31.87 ± 6.3	12.09 ± 3.6	**33.61** ± 2.3	20.73 ± 2.1
(translated for LM/WE)	macro	21.89 ± 4.3	21.29 ± 3.0	18.83 ± 4.3	**30.17** ± 5.6	17.14 ± 7.4	26.15 ± 2.0	22.40 ± 3.1

Table 4: Ablation results with official train-dev-test split, averaged over 10 seeds. Languages marked with * were without training and development data.

Configuration			∅ languages	
Agg	Split	Translate	micro	macro
HM	✓	✓	24.74	15.92
Avg	✓	✓	+4.45	**+2.93**
Min	✓	✓	-4.08	-4.02
n/a	✗	✓	-5.60	-3.05
n/a	✗	✗	-1.50	-1.88
HM	✓	✗	**+5.00**	+1.39
Avg	✓	✗	+0.46	-1.88

Table 5: Ablation study for large language models standalone (average across five runs), tested with different aggregators (harmonic mean, average, and min-pooling) as well as with the language-agnostic approach (translated articles) and language-sensitive approach

Configuration			∅ languages	
σ	Split	Translate	micro	macro
tanh	✗	✓	41.68	37.48
tanh	✓	✓	+1.37	+3.05
tanh	✗	✗	-6.45	-10.02
Sigmoid	✗	✓	-2.23	-6.33
ReLU	✗	✓	**+4.20**	**+5.28**
Linear	✗	✓	+3.81	+1.17

Table 6: Ablation study for static word embeddings standalone (average across five runs), tested with different activation functions (σ) as well as with the language-agnostic approach (translated articles) and language-sensitive approach

For option (i) the relations are grouped as follows:

- `Antonym`, `DistinctFrom`
- `MadeOf`, `CreatedBy`, `DefinedAs`, `ReceivesAction`, `CausesDesire`, `MotivatedByGoal`, `MannerOf`, `Desires`, `Causes`, `HasA`, `PartOf`, `HasSubevent`, `HasLastSubevent`, `HasFirstSubevent`, `HasPrerequisite`, `HasProperty`, `UsedFor`, `CapableOf`, `AtLocation`, `LocatedNear`, `Synonym`, `FormOf`, `HasContext`, `IsA`, `SymbolOf`

and for option (ii) the following:

- `DistinctFrom`, `Antonym`
- `MadeOf`, `DefinedAs`, `HasA`, `PartOf`, `MannerOf`, `HasProperty`, `Synonym`, `FormOf`, `IsA`, `SymbolOf`
- `CreatedBy`, `UsedFor`, `ReceivesAction`, `CausesDesire`, `MotivatedByGoal`, `Desires`, `CapableOf`, `HasContext`, `AtLocation`, `LocatedNear`
- `HasSubevent`, `HasLastSubevent`, `HasFirstSubevent`, `HasPrerequisite`, `Causes`

We grouped relations based on our intuition, so the grouping might be suboptimal. However, we doubt that different groupings would drastically improve performance.

Hyperparameters For the shared task we evaluated the best hyperparameters for the GNN on the dev set. The best setting was a GAT (Veličković et al., 2018) with 2 layers, 8 attention heads, a hidden dimension of 128 and leaky ReLU activation. To obtain a graph-representation we apply sum-pooling over the nodes, i.e. we sum all final node representations.

Results To gain additional insight, Table 7 shows a selection of hyperparameters evaluated on the test set.

The results show that the GCN and GAT outperform their respective relational counterparts R-GCN and R-GAT. This indicates that relations are less important for the task of frame prediction. Furthermore, the coarser grouping option (i) showed better performance than option (ii), again indicating that indeed fine-grained differences between relations are not necessary for frame prediction.

Configuration	∅ languages	
	micro	macro
ST (GAT)	45.31	41.12
GCN	+0.05	+0.13
GAT w/ 1 head	-0.67	-1.22
R-GAT (i)	-0.6	-1.38
R-GAT (ii)	-1.87	-1.93
R-GCN (i)	-1.64	-1.61
R-GCN (ii)	-2.32	-1.73
w/ mean pool	+0.32	+1.00
w/ ReLU	-0.77	-0.81
3 layers	-0.69	-0.80

Table 7: Ablation study for knowledge graph models standalone. Results are relative to the Shared Task (ST) setting described in §B.3.3 (averaged over ten seeds). The subsequent rows show relative results when changing hyperparameter(s) ST setting (averaged over five seeds). The number behind R-GCN and R-GAT refers to the grouping on CN relations as described in §B.3.3. Due to GPU memory restrictions the R-GATs are evaluated with one attention head only.

We also observe that mean-pooling outperforms sum-pooling (i.e. averaging / summing over all nodes to obtain a graph representation). This is in contrast to our findings on the development set, where sum-pooling performed better.