Node Placement in Argument Maps: Modeling Unidirectional Relations in High & Low-Resource Scenarios

Iman Jundi, Neele Falk, Eva Maria Vecchi, and **Gabriella Lapesa**
Institute for Natural Language Processing
University of Stuttgart, Germany
first[-middle].last@ims.uni-stuttgart.de

Abstract

Argument maps structure discourse into nodes in a tree with each node being an argument that supports or opposes its parent argument. This format is more comprehensible and less redundant compared to an unstructured one. Exploring those maps and maintaining their structure by placing new arguments under suitable parents is more challenging for users with huge maps that are typical in online discussions.

To support those users, we introduce the task of *node placement*: suggesting candidate nodes as parents for a new contribution. We establish an upper-bound of human performance, and conduct experiments with models of various sizes and training strategies. We experiment with a selection of maps from Kialo, drawn from a heterogeneous set of domains.

Based on an annotation study, we highlight the ambiguity of the task that makes it challenging for both humans and models. We examine the unidirectional relation between tree nodes and show that encoding a node into different embeddings for each of the parent and child cases improves performance. We further show the few-shot effectiveness of our approach.

1 Introduction

Online discussions can have huge numbers of contributors and contributions, making the discussion hard to follow for new users. Getting an overview of a discussion and finding points of interest for a new user might be hard in such an unstructured format which is also prone to redundancy. **Argument maps**, in their simplest form, structure arguments into a tree with each node being a pro or contra argument for its parent node (also an argument, see Figure 1). Relying on the structure of the map, users can dive deeper into specific aspects of an argument and collectively add more arguments to support or oppose it: this improves the overall quality of the discourse and at the same time, triggers

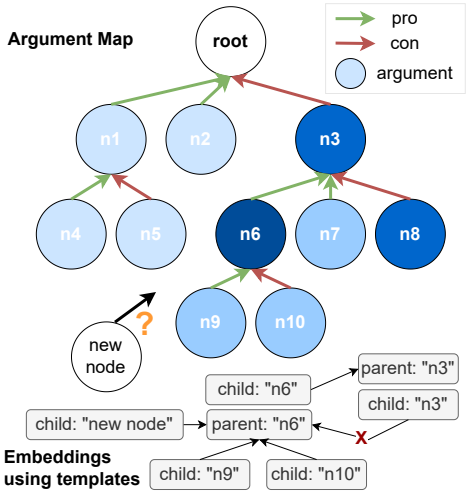


Figure 1: **Node placement** finds for a new node suitable parent nodes in an argument tree. The darker shades represent better candidates making the best ranking: n6, n3, n8, n7,.. When representing the nodes it is beneficial to decouple parent and child representations so that $n6_{child} \rightarrow n3_{parent}$ but $n3_{child} \not\rightarrow n6_{parent}$

the generation of new ideas and the continued discussion of existing ones. For an argument map to serve its purpose, it is essential to keep a somewhat clean structure, but this can be challenging for larger maps since finding where to add a new contribution can become a tedious task, and where the user initially decides to add their contribution based on their limited exploration of the map might be a sub-optimal choice.

To support in creating argument maps, we propose **node placement** as a new task defined as finding suitable candidates from an argument tree to be the parent of an argument. Deciding if an argument is pro or contra its parent is not a focus here as it does not constitute a bottleneck when adding a new contribution (binary decision vs. choosing the suitable parent from possibly hundreds of nodes). A number of nodes could be suitable as parents at varying or more similar degrees making the task inherently ambiguous (nodes n3, n8 are equally

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 5854–5876

suitable in Figure 1). The effect of this could be loosened by presenting the user with top-n recommendations (visualized using color shades similar to Figure 1). The task then could be employed to support users in two ways: 1) exploration: the user enters a short keywords-argument and based on its suggested node placement, finds the most relevant places in the map to explore; 2) optimization: after a user is done writing the argument, node placement suggestions are used to better place the final new contribution (example in Figure A.5). The task could be also seen as a first step to automatically and incrementally build argument maps from unstructured discussions or to enrich existing argument maps based on those discussions.

We use publicly available argument maps from Kialo[1] (where users manually & collectively maintain discussions in maps) and conduct an **annotation study** on a sample of nodes with 10 challenging candidates per node, in order to gain insights about the task and estimate human performance. We highlight the challenging nature of the task even with this low number of candidates. We formulate the task as a ranking problem and conduct **modeling experiments** using sentence-transformers with large and small models and a variety of intermediate-task training. We show that more intermediate-task training yields better results, and that the performance of the *large trained models is on par with humans* on our annotated samples. We highlight the **unidirectional** nature of the relationship between child and parent nodes showing that it is beneficial to decouple the parent and child representations of the same node. To address this, we propose *using different textual templates for the child vs. parent representation of a node* (see embeddings in Figure 1), and show a *boost in performance* as a result (∼4, 3 points for top1, top5 accuracy of the large model). We further examine the data-efficiency of our training strategies in **low-resource** scenarios where the number of maps and/or the size of maps are small. We show that the zero-shot performance is still relatively good and is consistently improved with few-shot training even with a small number of samples and that *using templates is especially beneficial for a smaller number of samples*.

Our **main contributions** are summarized as: 1) Defining a new task, node placement in argument maps; 2) Estimating human performance on the task through an annotation study; 3) Conducting modeling experiments, proposing a simple approach to tackle unidirectional relations between text pairs, and employing this to improve the performance of the proposed task; 4) Demonstrating the effectiveness of our training in low-resource scenarios. A detailed analysis of the results is also conducted to gain insights into our task and method. Our code is made publicly available[2].

2 Related Work

Node Placement in Argument Maps The task is related to two widely explored tasks in Argument Mining: argument retrieval and modeling argument relationships. Argument retrieval can be viewed as a more general form of node placement, in which a system should provide relevant arguments given a controversial question or topic (Stab et al., 2018; Reimers et al., 2019; Bondarenko et al., 2021) or a suitable counter-argument given an input argument (Wachsmuth et al., 2018). With respect to general argument retrieval, our task tackles a finer-grained problem: finding suitable positions in the argument tree. Regarding the modeling of the relationship between arguments (Stab and Gurevych, 2017), e.g. support/attack, only few works consider it in the context of a full, structured debate. To automatically construct argument graphs, Lenz et al. (2020) use structured debates to classify relationship between argumentative units. Agarwal et al. (2022) model the relationship between arguments as a polarity prediction task using the tree-structure, and exploiting the ancestors of a node to classify support/attack relationships.

Retrieval & Ranking: Cross-encoders like BERT (Devlin et al., 2019) can be used to score pairs of sentences, but this does not scale well for large number of candidates in retrieval & ranking tasks. Siamese networks (Bromley et al., 1993) (also called bi-encoders) have long been used to create embeddings for efficiently tackling those tasks using contrastive learning. *sentence-transformers* (Reimers and Gurevych, 2019) employs this by using BERT or other Transformer models and utilizing labeled data while ConSERT (Yan et al., 2021) and SimCSE (Gao et al., 2021) also utilize unlabeled data. The original BERT can be used to encode each sample (of a pair) into a vector for more efficiency, but the resulting embeddings have high

[1] https://www.kialo.com

[2] https://github.com/imanjundi/argument-relations

similarity in general which BERT-flow (Li et al., 2020) and WhiteningBERT (Huang et al., 2021) tackle using normalization. We utilize bi-encoders and pay attention to the high similarity issue with a thorough analysis of the effect our approach has on the embedding space.

Templates & low-resource: Templates were recently heavily used in prompts to tap into the knowledge encoded in large PLMs and to make use of their few-shot capabilities by using a task-information template (Petroni et al., 2019; Brown et al., 2020). They were also used to fine-tune PLMs in a few-shot setup (Schick and Schütze, 2021; Tam et al., 2021; Liu et al., 2022) while others attempted to do away with them (Logan IV et al., 2022; Karimi Mahabadi et al., 2022; Tunstall et al., 2022), but they were mainly employed to directly solve NLP tasks and not to learn embeddings. Prompts were used for the latter more recently (Jiang et al., 2022) with contrastive learning. We simply utilize templates with no language modeling training or inference and show they are beneficial with contrastive learning to learn embeddings in high and low-resource.

3 Data

We use argument maps from Kialo, an online platform on which people engage in discussions on specific topics or statements. In a discussion about a controversial thesis topic, the thesis acts as a root node under which further and increasingly more specific arguments for or against this point of view can be added. An example of how the original data looks like is shown in Figure 2 (taken from Kialo [3]). It shows how the tree of arguments evolves for the root node or thesis "Video game storytelling should portray gender equality." Users can navigate through the tree to find aspects of the discussion that they are interested into or to find a good node to attach their new input to. Each new argument can in turn be attacked or supported with a variety of different arguments. Thus every debate in Kialo represents a unidirected tree, where each edge represents a support or attack relation (henceforth, *pro* and *con*).

We rely on data from Agarwal et al. (2022) and use a total of 1,378 maps covering a wide variety of topics: politics, technology, ethics, etc. (overview in Appendix Figure A.2). The majority contain up to 200 nodes, but a quarter of the data are large-scale discussions (up to 6k nodes) for which an automatic support is especially beneficial (complete analysis in Section A.1).

[3] https://www.kialo.com

4 Annotation Study

To have a better understanding of the task and data, generalize a baseline of human performance, and, estimate the difficulty and cost humans encounter with such a task, we conduct an annotation study.

Design: We employed 3 annotators with a background in NLP and Social Sciences (details in Section A.2. The annotators were presented a specific contribution to a discussion[4] – the *child* – and 10 candidate parents selected from the discussion's argument map to which the child could attach. The annotators were tasked to classify each of the candidates with one of the following labels: BEST PARENT (count 1), SUITABLE PARENT (max. 4), or LESS SUITABLE PARENT. The annotation guidelines and an example are provided in Appendix Figures A.6 and A.7.

In order to control for an appropriate variety of candidates that a user might encounter, the candidates consisted of the actual parent, 6 candidates closely related in the tree to the child (with a maximum path distance of 3), and 3 randomly selected candidates from the full tree. In total, the annotated dataset consists of 200 child instances. The instances selected were evenly split between pro relations with its parent and con. The nodes were sampled from small and large-scale maps (90 to 2500 nodes). The topics of the maps are *environment*, *economy*, *gender*, *politics* and *immigration*.

To better understand the annotators' approach to the task, we asked them to provide their confidence scores for each annotation, as well as short-answer motivation for a subset of 100 annotations.

Annotation Results: We measure the annotator agreement using weighted Kappa (κ_w) (Cohen, 1960) as we would like to account for the seriousness of the disagreements, i.e. disagreeing about BEST and SUITABLE should be penalized less than BEST and LESS SUITABLE. The annotators have a fair to moderate agreement of 0.387. While we can conclude that the participants generally agreed, κ_w in the lower range of agreement is an indication of the difficult and subjective nature of this task, despite the clear guidelines and training.

[4] The topic of the discussion was also provided to the annotators for context in each instance.

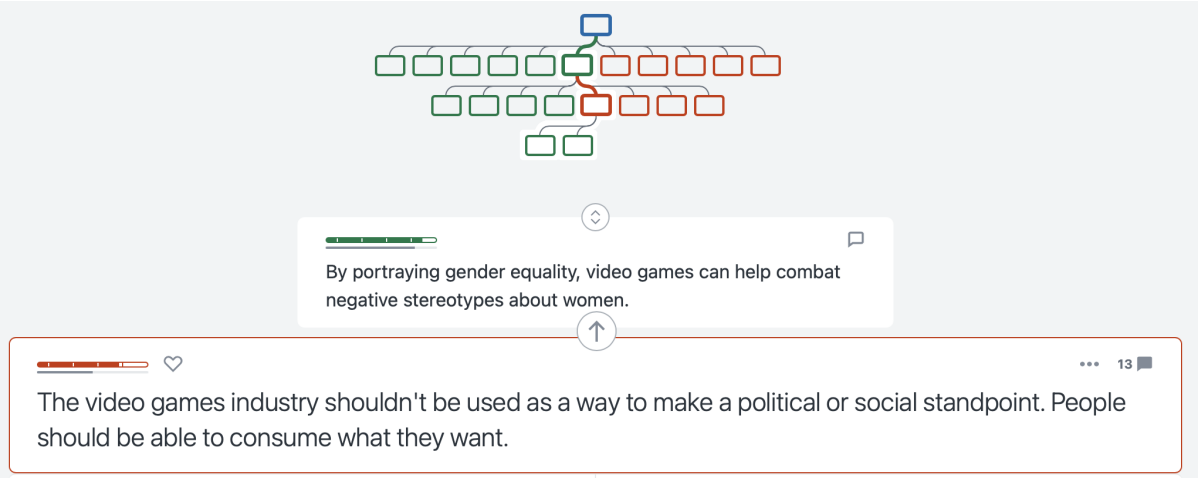


Figure 2: A snippet of an argument tree discussing the claim "Video game storytelling should portray gender equality." which corresponds to the root node.

To calculate the performance, we convert the human annotated labels for each child into a score and thus obtain a ranked list of all candidates[5]. Table 1 shows the aggregated performance of the annotators in being able to select the best parents for each child (metrics in Section A.3). The average top1 is just under 50%, meaning the participants were not necessarily able to select the best parent among the 10 candidates with ease. On the other hand, the average top5 is quite high. This contrast suggests that the best parent is often ambiguous; while distinguishing between a set of those that might be the best parent (conflating BEST and SUITABLE) and those that are certainly not (LESS SUITABLE) is less ambiguous and easier to define for the participants.

	top1	top5	MRR
all	0.480	0.935	0.664
pro	0.436	0.941	0.640
con	0.525	0.929	0.688

Table 1: Average performance of human annotators for the full annotated sample (all), for all nodes with a pro relation, and for all with a con relation. Performance is computed based on the actual parent of the child node.

We find that for the participants it was easier to determine the best parent in cases of a con relationship between parent and child. Contributions to a discussion of this class are generally presented as negations to specific points in the parent comment, and most likely contain high lexical overlap. For example, the child *The boys referred to appear to be having fun, rather than trying to hurt each other*, in response to the parent *It shows the harm boys can do when people allow bad behaviour because "boys will be boys."* Contributions that are instead of a pro relationship to the parent are more often an elaboration or extension of the argument made in the parent comment, likely resulting in less pronounced links between parent and child. For example, the child *There is poor cooperation between the Commission and national financial regulators*, in response to the parent *The Emissions Trading System is susceptible to fraud.*

Our analysis of the motivation behind the annotators' choice shows it can be divided into 3 categories, in decreasing frequency: (1) *Process of elimination*, described often as "most obvious", "best fit", or "makes most sense"; (2) *Linguistic overlap*, reported as similar use of terminology or structure; and (3) *Logical connection*, in which participants found a direct child/parent relationship, such as an elaboration or offering examples.

Note that the task given to the annotators is rather simple in comparison to a real-world application where dozens, if not hundreds, of options across the full argument map would need to be considered. That said, the three annotators averaged a total of 31 hours to complete the 200 annotations. Clearly, this cost can be greatly reduced for users or moderators of the argument maps with a filtered shortlist of candidates provided by a model trained on our proposed task: node placement.

[5]Each label corresponds to a number (BEST:1, SUITABLE:2, LESS SUITABLE:3) and the score for a candidate is the average of all three numbers. The lower the score, the higher the candidate will be ranked.

5 Main Modeling Experiments

The task can be formulated as a **ranking** problem where a score is predicted for each candidate node and used to rank all candidate nodes. We use a bi-encoder to scale to the huge number of nodes that each map might contain (up to 6k nodes, c.f. Section 3). Using a cross-encoder that scores each pair of nodes is not feasible to apply on all nodes, but could still be used to refine the ranking of the top-*n* candidates based on the scores from the bi-encoder. This re-ranking step is out-of-scope given the noisy data available and the ambiguous nature of the task, as seen in the annotation study, which makes judging the final ranking not feasible.

Unidirectional Relation Between Nodes
Common sentence or document embedding methods assume a bi-directional relation. For example, in the case of semantic similarity: if sentence1 $x1$ is similar to sentence2 $x2$ then sentence2 is also similar to sentence1 so:

$$F(x1, x2) = y \leftrightarrow F(x2, x1) = y$$

This is clearly not the case when representing parent/child relation so:

$$F(x1, x2) = y \not\leftrightarrow F(x2, x1) = y$$

This makes encoding the tree nodes into an embedding space challenging since the training should make the child nodes (c1,...,cn) closer to their parent node (p), but when p is considered as a child node with the aim of predicting its parent, it should still be closer to its parent (g) than to (c1,..,cn). The latter should be in this case regarded as negative training examples. The recursive structure of the tree might increase the effect of this issue since all nodes are eventually connected to the root.

5.1 Templates

Motivated by the successful use of prompts in related work, we use templates to better represent the unidirectional relation, exploit the stance label and utilize the knowledge encoded in the model. By encoding those signals textually through templates, they are passed through the model which allows for effective interaction with other features while keeping the approach simple. Our templates are:

parent/child: differentiate the parent vs. child by using `parent:"text"` when considering the node as a candidate parent for another node vs. `child:"text"` when considering the node as the child. This allows us to have two different embeddings for each node. The resulting training data has the same size as the original.

pro/con: represents pro & con child nodes using `pro:"text"` & `contra:"text"` which we add to parent/child template samples for training, and use parent/child template (main template) for evaluation. Using only pro/con templates would otherwise complicate evaluation since it results in two different rankings of the candidates: one when considering the node as pro for a candidate and one for con. The resulting training data is 2x the original size.

all: includes pro/con templates and 3 templates that use similar keywords while combining child & parent text during training e.g. `pro:"text" parent: "text"` (see Appendix Table A.2 for all templates). The resulting training data is 5x the original size.

5.2 Experimental Setup

Each argument in a map is encoded into an embedding using sentence-transformers[6] (Reimers and Gurevych, 2019). Cosine similarity is used between the embeddings of a node and all possible candidates to calculate the scores. Experiments use **models with varying size and intermediate-task training** (Pruksachatkun et al., 2020) based on a large model, MPNet (Song et al., 2020) and a smaller one, MiniLM (Wang et al., 2020)[7]: *mpnet* without intermediate training, *nli-mpnet* with intermediate training on MNLI and SNLI, *paraphrase-mpnet* with additional paraphrase data, *all-mpnet* with additional QA and other data, and finally *all-mini* with similar training but based on MiniLM (models overview in Appendix Table A.1).

The argument maps in the dataset are split into 80% train, 20% test (1102 and 276 maps each). No hyperparameter search was done and no validation set was used to avoid influencing the few-shot performance by knowing hyperparameter values based on extra validation data that isn't available in few-shot (see Section 6.1). The maps from which the items of the annotation studies were sampled are part of the test set. 5 different train/test random splits are used and the average performance on the various test sets is reported in the main results. Each node with its actual parent constitute a training sample resulting in ∼211k training pairs. The models are trained using a batch contrastive loss where the actual parent of a node is considered a positive sample and all other parents in a random

[6] https://github.com/UKPLab/sentence-transformers

[7] https://www.sbert.net/docs/pretrained_models.html

batch are considered negatives[8]. The models are trained for 1 epoch and then evaluated on the test set by calculating the metrics for each leaf child node and averaging over all those nodes in the map. We report the average for all maps. Evaluation is also done on the annotated samples to compare to human performance. We do not attempt to optimize hyperparameters to avoid influencing our few-shot experiments (Section 6.1). To have a more detailed estimation of the task difficulty and modeling performance, we report the average of a variety of metrics: top1, top5 accuracy and MRR (metrics description in Section A.3).

5.3 Results & Analysis

model	top1	top5	MRR
mpnet	.2859	.5884	.4259
nli-mpnet	.2864	.5935	.4277
paraphrase-mpnet	.2955	.5993	.4372
all-mpnet	**.3064**	**.6239**	**.4525**

Table 2: Results with varying intermediate-task training

Table 2 shows the results for MNPet without the use of any templates. The performance improves with more intermediate-task training for all metrics. Best performance is achieved using **all-mpnet** (more generic and larger training data), which we use in all following experiments.

Table 3 shows that training improves on the zero-shot performance and performs comparable to or better than the human performance on the 200 annotated samples. The task, however, remains challenging in general mostly because of its inherent ambiguity (as was shown in the annotation study) and because of the noisy data that is available. Using the parent/child template further boosts the performance by ∼4, 3 points for top1, top5 respectively. Adding pro/con templates improves the performance only slightly. This might be because the signal about the type of relation is not as important in solving the task or that this signal is not utilized properly. Using more templates in *all* also does not improve the performance. It is hard to estimate how much more improvement is still possible since the human performance is estimated in a controlled setup and the highest performance here is already on bar with it or exceeds it. Similar observations can be made for the smaller model **all-mini** in Appendix Table A.4, except that the boost from using parent/child is smaller and the best performance still lags behind that of humans especially for top1. Based on those findings, we focus on *parent/child* template in the following analysis of the results.

Agreement between model predictions and human annotations is moderate (κ_w=0.459) for zero-shot and is somewhat increased with training (0.491) meaning the model has a general agreement with humans about the ranking independent of what the actual parent is in the original Kialo data.

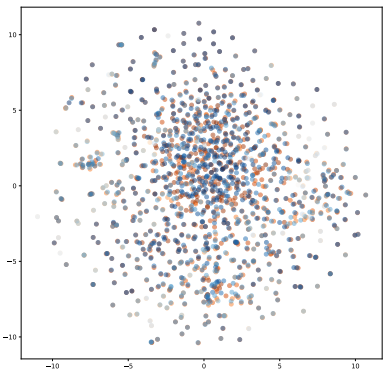


Figure 3: **Embedding space** of nodes in a sample map (lighter color for deeper nodes). Using templates allows for a variation of the same node for child (blue) vs. parent (orange). The variation is higher for nodes closer to the root (darker blue and orange) as opposed to the overlapping visualization of deeper nodes (light gray).

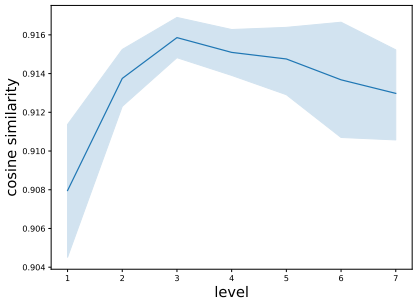


Figure 4: Cosine **similarity between parent and child** embeddings of the **same node** when parent/child template is used. The top nodes have a slightly lower similarity and more distinct embeddings for parent vs. child.

Embedding Space (Parent vs. Child)
To analyse the effect of parent/child templates on the embedding space, we visualize in Figure 3 the embedding space for the nodes when using parent/child template. The visualization of the parent (orange) vs. child (blue) is more distinct for the

[8]MultipleNegativesRankingLoss in sentence-transformers

model	test (20%) all possible candidates			200 samples 10 candidates		
	top1	top5	MRR	top1	top5	MRR
zero-shot	.2491	.5467	.3897	.430	.900	.628
no template	$.3017_{\pm.0063}$	$.6178_{\pm.0120}$	$.4472_{\pm.0083}$	$.4770_{\pm.0076}$	$.9220_{\pm.0057}$	$.6640_{\pm.0040}$
parent/child (*)	$.3366_{\pm.0073}$	$\mathbf{.6467}_{\pm.0119}$	$.4786_{\pm.0087}$	$.5180_{\pm.0091}$	$\mathbf{.9410}_{\pm.0042}$	$.6906_{\pm.0069}$
pro/con (*)	$\mathbf{.3376}_{\pm.0073}$	$.6457_{\pm.0100}$	$\mathbf{.4793}_{\pm.0086}$	$\mathbf{.5210}_{\pm.0156}$	$.9380_{\pm.0045}$	$\mathbf{.6934}_{\pm.0105}$
all (*)	$.3357_{\pm.0067}$	$.6439_{\pm.0099}$	$.4771_{\pm.0083}$	$.5190_{\pm.0152}$	$.9340_{\pm.0089}$	$.6915_{\pm.0093}$
human				.480	.935	.664

Table 3: Performance after **high-resource** is competitive with humans. parent/child template boosts the performance, whereas more templates are not as beneficial. (*) uses templates.

nodes that are closer to the root (darker color) in contrast to the deeper nodes which their parent & child visualizations are overlapping (light gray). This shows that representations for child vs. parent is more distinct for the more generic nodes at the top of the tree. This difference is not that significant as shown in Figure 4, where we visualize cosine similarity between the parent and child embedding of the same node averaged over all nodes in maps from the test set. As expected, the similarity is still high in general at $\sim$0.9 and the top nodes in the tree have slightly lower similarity (0.01) which might still be important in improving the performance.

model	cosine similarity
zero-shot	0.3210
no template	0.2738
parent/child	
(child, child)	0.2962
(child, parent)	0.0321
(parent, parent)	0.0462

Table 4: Average **cosine similarity between embeddings** for all nodes in each map.

Table 4 shows the average cosine similarity between all nodes for *zero-shot* and *no template* where we can see that the training decreases the similarity on average. When the similarity is calculated for the various combination in the case of training using parent/child templates, the training seems to have a different effect: the similarity stays higher when comparing arguments using only child template *(child, child)*, but the parent embeddings are more distinct and less similar to each other *(parent, parent)* and to child embeddings *(child, parent)* of all other nodes (not specifically actual children).

Finally, we compute cosine similarity between the embeddings of each child c and its actual parent p using (`child:"c"`, `parent:"p"`) getting an average of .5496, whereas that of (`child:"p"`, `parent:"c"`) is much lower at .4951. This shows that using parent/child template indeed leads to a better representation of the unidirectional relation $c \rightarrow p$ and $p \not\rightarrow c$.

model	test			samples		
	all	pro	con	all	pro	con
zero-shot	.5467	**.5587**	.5398	.900	**.921**	.879
no template	.6239	.6089	**.6462**	.930	**.941**	.919
parent/child	.6539	.6373	**.6749**	.940	**.950**	.929
human				.935	**.941**	.929

Table 5: **top5** according to relation type (**pro/con**) is better for pro compared to con for zero-shot. The performance of con is noticeably improved with training.

Pro vs. Con Performance: Table 5 shows the detailed results of one train/test split according the type of relation between the argument its parent. As expected, zero-shot performance is better for *pro* compared to *con* since the pro relation is similar to entailment and other relations used to construct positive samples in tasks the model was trained on. This changes after training (*no template*) where *con* performance improves more than *pro* (+.11 vs. +.05), which can be partially due to more data available for con vs. pro ($\approx 162k$ vs. $120k$ or 57% vs. 43%). Using templates gives a similar small boost for both. We see a similar pattern for top1 (Appendix Table A.5) except here con performance is similar to pro for zero-shot but after training, the performance of con is again better than pro. The pattern is similar after training to that of human performance on *pro* and *con* for both top1 and top5.

6 Few-shot Experiments

Our approach could be used for similar tasks, for which obtaining the scale of data that we used here is not feasible. Motivated by this, we investigate the data efficiency of our approach, analyze the results, and where to better invest resources.

6.1 Few-shot Experimental Setup

Random samples from the training set are used with varying numbers of maps (#maps) and numbers of nodes used from each map (#nodes) in (8, 16, 32, 64) where the final number of samples used for

#nodes	#maps	template	test top1	test top5	test MRR	samples top1	samples top5	samples MRR
zero-shot			.2491	.5467	.3897	.430	.900	.628
8	8	no template	$.2540_{\pm.0017}$	$.5507_{\pm.0022}$	$.3939_{\pm.0013}$	$.4310_{\pm.0065}$	$.9100_{\pm.0050}$	$.6309_{\pm.0032}$
		parent/child	$.2595_{\pm.0091}$	$.5537_{\pm.0123}$	$.3980_{\pm.0105}$	$.4230_{\pm.0239}$	$.9140_{\pm.0096}$	$.6201_{\pm.0142}$
		pro/con	$.2709_{\pm.0051}$	$.5655_{\pm.0057}$	$.4103_{\pm.0054}$	$.4420_{\pm.0315}$	$.9190_{\pm.0022}$	$.6331_{\pm.0172}$
		all	$\mathbf{.2802}_{\pm.0061}$	$\mathbf{.5845}_{\pm.0106}$	$\mathbf{.4218}_{\pm.0084}$	$\mathbf{.4640}_{\pm.0139}$	$\mathbf{.9280}_{\pm.0076}$	$\mathbf{.6510}_{\pm.0095}$
16	16	no template	$.2680_{\pm.0052}$	$.5727_{\pm.0065}$	$.4100_{\pm.0057}$	$.4470_{\pm.0045}$	$.9130_{\pm.0057}$	$.6405_{\pm.0043}$
		parent/child	$.2815_{\pm.0063}$	$.5853_{\pm.0140}$	$.4233_{\pm.0083}$	$.4670_{\pm.0301}$	$.9280_{\pm.0027}$	$.6545_{\pm.0168}$
		pro/con	$.2859_{\pm.0054}$	$.5948_{\pm.0106}$	$.4293_{\pm.0068}$	$.4780_{\pm.0186}$	$\mathbf{.9390}_{\pm.0102}$	$.6605_{\pm.0112}$
		all	$\mathbf{.2890}_{\pm.0015}$	$\mathbf{.6027}_{\pm.0049}$	$\mathbf{.4339}_{\pm.0016}$	$\mathbf{.4730}_{\pm.0091}$	$.9320_{\pm.0027}$	$\mathbf{.6590}_{\pm.0068}$
32	32	no template	$.2838_{\pm.0040}$	$.5932_{\pm.0058}$	$.4266_{\pm.0041}$	$.4600_{\pm.0094}$	$.9250_{\pm.0094}$	$.6524_{\pm.0069}$
		parent/child	$.2882_{\pm.0037}$	$.5985_{\pm.0050}$	$.4320_{\pm.0037}$	$.4710_{\pm.0082}$	$.9220_{\pm.0045}$	$.6563_{\pm.0033}$
		pro/con	$.2858_{\pm.0033}$	$.5972_{\pm.0030}$	$.4293_{\pm.0028}$	$\mathbf{.4830}_{\pm.0104}$	$\mathbf{.9230}_{\pm.0084}$	$\mathbf{.6626}_{\pm.0045}$
		all	$\mathbf{.2923}_{\pm.0059}$	$\mathbf{.6018}_{\pm.0068}$	$\mathbf{.4347}_{\pm.0060}$	$.4680_{\pm.0202}$	$.9280_{\pm.0084}$	$.6551_{\pm.0125}$
64	64	no template	$.2892_{\pm.0030}$	$.5974_{\pm.0042}$	$.4317_{\pm.0036}$	$\mathbf{.4890}_{\pm.0089}$	$\mathbf{.9220}_{\pm.0057}$	$\mathbf{.6651}_{\pm.0032}$
		parent/child	$.2925_{\pm.0044}$	$.5999_{\pm.0052}$	$.4348_{\pm.0041}$	$.4800_{\pm.0079}$	$.9200_{\pm.0106}$	$.6624_{\pm.0050}$
		pro/con	$.2906_{\pm.0047}$	$.5969_{\pm.0054}$	$.4319_{\pm.0037}$	$.4820_{\pm.0045}$	$.9170_{\pm.0135}$	$.6629_{\pm.0037}$
		all	$\mathbf{.3018}_{\pm.0025}$	$\mathbf{.6025}_{\pm.0035}$	$\mathbf{.4416}_{\pm.0020}$	$.4650_{\pm.0106}$	$.9160_{\pm.0171}$	$.6495_{\pm.0122}$
full dataset		none	**.3064**	**.6239**	**.4525**	**.485**	.930	**.669**
human performance						.480	**.935**	.664

Table 6: Few-shot Results. Few-shot improves over zero-shot in all cases. Using templates helps narrow the gap between low-resource and training on the *full dataset* with a boost that is larger for smaller #samples

training (#samples=#node×#maps) varies between 64 to 4096. 5 random samples for each combination of (#nodes×#maps) are used and the average performance is reported. Various templates are again investigated as their effect is expected to be different for low-resource.

True Few-shot

We refrain from using extra unlabeled data or extra samples as dev set to report *true few-shot* performance (Perez et al., 2021). We use default hyperparameters (Appendix section A.4) and batch size=8 (smallest number of nodes available for training per map).

6.2 Few-shot Results & Analysis

We see in Table 6 that using the same training paradigm proves to be effective for low-resource. Few-shot training improves on zero-shot in all cases with and without templates and no degradation in performance is observed due to overfitting even with a small number of samples (where few-shot is more prone to overfitting). Training with *parent/child* template improves the performance, especially for a lower number of #samples. For 64x64 the templates still improve the performance on the test but don't improve on the annotated samples, however, the performance there is already close to human performance. The *pro/contra* template also helps boost the performance and the best performance is achieved when using a combination of various templates (*all*). When comparing for each #samples the performance when trained with no templates vs. all, we see that using templates helps narrow the gap between low-resource and high-resource (*full dataset*) with a boost that is larger for smaller #samples. Similar findings can be seen for the smaller model (Appendix Table A.11) except using more templates is not as effective there, especially with larger #samples.

The pro/con and all templates are more helpful here than when training on the full dataset (Table 3). This might be due to an augmentation effect since each sample is used in a parent/child template as well as other templates resulting in a training size that is 2x and 5x the original size (for pro/con, all respectively). Such augmentation would be more beneficial in more low-resource cases. To verify how this compares to the model seeing the samples more often, we train the model for double the amount (2 instead of 1 epoch) without any templates and those results (Appendix Table A.6) are comparable to *no template* with 1 epoch and worse than pro/con. The same is seen for *all* vs. 5 epochs which also holds when training with a *parent/child* template for 5 epochs in which case the performance of *all* is still better although to a lesser degree (Appendix Table A.7, A.8). Those initial results demonstrate the usefulness of templates with the potential to further improve the performance with template engineering or template search which were out of scope here.

This shows that the use of templates with contrastive learning is an effective approach in low-resource: the parent/child signal can be effectively exploited even at a low #samples and incorporating more templates in the training is a promising direction to bridge the low to high-resource gap.

#nodes	#maps	top1	top5	MRR
8	8	$.2611_{\pm.0099}$	$.5535_{\pm.0153}$	$.3992_{\pm.0116}$
16	16	$.2803_{\pm.0066}$	$.5840_{\pm.0167}$	$.4220_{\pm.0090}$
32	32	$.2875_{\pm.0035}$	$.5994_{\pm.0057}$	$.4318_{\pm.0043}$
64	64	$.2946_{\pm.0017}$	$.6048_{\pm.0040}$	$.4381_{\pm.0016}$

Table 7: **foo/bar** template is comparable to parent/child.

Template Semantics: Motivated by research done on the effect of prompt semantics (Le Scao and Rush, 2021; Webson and Pavlick, 2022), we employ templates with no semantic meaning (*foo/bar*) using `foo:"text"` for child and `bar:"text"` for parent. Table 7 shows comparable results for foo/bar vs. parent/child (the same is seen when training with the full dataset Appendix Table A.9), and a similar effect is seen when using various templates in Appendix Table A.10. This is in line with findings about prompt-based fine-tuning (Webson and Pavlick, 2022) that is shown to yield good performance with irrelevant and misleading prompts.

Number of Maps vs. Number of Nodes
We investigate here where resources are more useful either when annotating and creating a dataset or when limiting training size and computing resources. For the same #samples (e.g. 128), different #maps and #nodes per map can be used (e.g. 16×8 or 8×16). We show in Table 8 a comparable combination of #node×#maps to investigate the effect each has on the performance and whether it is more beneficial to have few big maps or many small maps for training. We see better performance with more #nodes per map compared to more #maps with fewer nodes. This is probably because the more #nodes are available, the better negative samples are possible for better training.

#nodes	#maps	top1	top5	MRR
8	16	$.2556_{\pm.0056}$	$.5570_{\pm.0089}$	$.3967_{\pm.0065}$
16	8	$.2601_{\pm.0030}$	$.5622_{\pm.0026}$	$.4017_{\pm.0027}$
8	32	$.2688_{\pm.0039}$	$.5725_{\pm.0032}$	$.4108_{\pm.0033}$
32	8	$.2689_{\pm.0056}$	$.5714_{\pm.0066}$	$.4103_{\pm.0055}$
8	64	$.2711_{\pm.0061}$	$.5731_{\pm.0085}$	$.4122_{\pm.0061}$
64	8	$.2803_{\pm.0017}$	$.5876_{\pm.0042}$	$.4227_{\pm.0014}$
16	32	$.2759_{\pm.0041}$	$.5809_{\pm.0047}$	$.4177_{\pm.0042}$
32	16	$.2761_{\pm.0034}$	$.5821_{\pm.0016}$	$.4186_{\pm.0021}$
16	64	$.2798_{\pm.0035}$	$.5865_{\pm.0049}$	$.4218_{\pm.0043}$
64	16	$.2885_{\pm.0051}$	$.5970_{\pm.0074}$	$.4308_{\pm.0057}$
32	64	$.2851_{\pm.0041}$	$.5934_{\pm.0051}$	$.4274_{\pm.0042}$
64	32	$.2881_{\pm.0024}$	$.5983_{\pm.0051}$	$.4312_{\pm.0037}$

Table 8: Few-shot of **comparable #maps vs. #nodes**. Slightly favorable outcome for more #nodes (16x8) over more #maps (8x16) especially for low #samples (128).

7 Conclusion & Contributions

We propose and evaluate a solution to support in creating argument maps, contributing: 1) At the *methodological* level, we define the new task of node placement in argument maps, and conduct an annotation study to establish the human performance on the task gaining insights about factors that affect the choice of suitable parents for a node. 2) At the *experimental* level, we present modeling results with different training setups and base models, showing that templates can be used to improve the representations and are beneficial in high and low-resource scenarios. 3) At the level of *application potential*, the task could be adapted using top-n candidates by highlighting the nodes based on their predicted score similar to Figure 1. This allows for a more intuitive user interaction and loosens the effect of the ambiguity inherent in the task.

8 Limitations

- Our work focuses on data from **one platform**, Kialo, which contains cleaner and higher quality arguments from a diverse range of topics and domains. How our approach performs on data from other platforms or more specialized domains (e.g. deliberations about policy) has to be investigated in the future.
- The vast majority of data available is **English** which makes conducting and evaluating multilingual experiments not feasible even with language transfer (see Appendix Section A.1).
- The dataset used in the training and evaluation has only **one correct position** although there might be multiple suitable parents. Given the large scale of the data and the huge number of nodes per tree, annotating all suitable parents would've require a very-large-scale unfeasible annotation. This could be investigated in future-work with the support of our models.
- The design of our annotation study does not take into consideration the **structure of the tree**. This might have made the task more challenging for the annotators. Reconstructing or representing the tree structure without revealing the actual parent (since the majority of the candidates are close relatives) is challenging when limiting the candidate parents to 10. Further refinement of the annotation study is left for future work along with the inclusion of the structure in the modeling.

- Although small models are shown to perform relatively well and are recommended to use when computation resources are limited, the models that perform, in our experiments, on par with humans are **large models** that are costly to train. Employing parameter efficient fine-tuning methods might be of interest here.
- We use only manually designed **templates** as a simple approach that required no extra training or engineering. How the results compare to using automated template/prompt engineering methods is also left for future work. Including prompt-based fine-tuning might be also of interest to investigate in combination with contrastive training although language modeling training would require more computational resources.
- Our task definition excludes the prediction of **pro/con** relation as less important, but the pro/con template information might be useful for this. More evaluation and analysis is needed to verify that.
- **Extra analysis** that was out-of-scope to include in this paper might be of interest: e.g. the effect of topic, the degree of a node, and semantic similarity to siblings on model or human performance.

9 Ethics Statement

We use available data from previous research. Automated tools to support in the exploration and creation of argument maps might be biased to favor arguments that are explored more often or that have more prominent styles as they are seen more often in the data as parents. This might lead to decreased suggestions as parents of those arguments that have underrepresented styles or using jargon/slang. This in turn leads to those arguments being less discussed and explored as they have less number of contribution. It's important to take this into consideration and investigate any such effects before and after employing such models in real-world applications.

Acknowledgements

We acknowledge funding by the Bundesministerium für Bildung und Forschung (BMBF) through the project E-DELIB. We thank Michael Roth, Sebastian Padó, and Sean Papay for providing feedback about the paper.

References

Vibhor Agarwal, Sagar Joglekar, Anthony P. Young, and Nishanth Sastry. 2022. Graphnli: A graph-based natural language inference model for polarity prediction in online debates. In *The ACM Web Conference (TheWebConf)*.

Alexander Bondarenko, Lukas Gienapp, Maik Fröbe, Meriem Beloucif, Yamen Ajjour, Alexander Panchenko, Chris Biemann, Benno Stein, Henning Wachsmuth, Martin Potthast, and Matthias Hagen. 2021. Overview of touché 2021: Argument retrieval. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 450–467, Cham. Springer International Publishing.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1993. Signature verification using a siamese time delay neural network. In *Advances in Neural Information Processing Systems 6, [7th NIPS Conference, Denver, Colorado, USA, 1993]*, pages 737–744. Morgan Kaufmann.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, Minneapolis, Minnesota.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Junjie Huang, Duyu Tang, Wanjun Zhong, Shuai Lu, Linjun Shou, Ming Gong, Daxin Jiang, and Nan Duan. 2021. WhiteningBERT: An easy unsupervised sentence embedding approach. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 238–244, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ting Jiang, Jian Jiao, Shaohan Huang, Zihan Zhang, Deqing Wang, Fuzhen Zhuang, Furu Wei, Haizhen Huang, Denvy Deng, and Qi Zhang. 2022. Promptbert: Improving bert sentence embeddings with prompts.

Rabeeh Karimi Mahabadi, Luke Zettlemoyer, James Henderson, Lambert Mathias, Marzieh Saeidi, Veselin Stoyanov, and Majid Yazdani. 2022. Prompt-free and efficient few-shot learning with language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3638–3652, Dublin, Ireland. Association for Computational Linguistics.

Teven Le Scao and Alexander Rush. 2021. How many data points is a prompt worth? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2627–2636, Online. Association for Computational Linguistics.

Mirko Lenz, Premtim Sahitaj, Sean Kallenberg, Christopher Coors, Lorik Dumani, Ralf Schenkel, and Ralph Bergmann. 2020. Towards an Argument Mining Pipeline Transforming Texts to Argument Graphs. In *Proceedings of the 8th International Conference on Computational Models of Argument*, volume 326 of *Frontiers in Artificial Intelligence and Applications*, pages 263–270, Perugia, Italy. IOS Press.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Online. Association for Computational Linguistics.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *CoRR*, abs/2205.05638.

Robert Logan IV, Ivana Balazevic, Eric Wallace, Fabio Petroni, Sameer Singh, and Sebastian Riedel. 2022. Cutting down on prompts and parameters: Simple few-shot learning with language models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2824–2835, Dublin, Ireland. Association for Computational Linguistics.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True few-shot learning with language models. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 11054–11070.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel R. Bowman. 2020. Intermediate-task transfer learning with pretrained language models: When and why does it work? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5231–5247, Online. Association 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, Benjamin Schiller, Tilman Beck, Johannes Daxenberger, Christian Stab, and Iryna Gurevych. 2019. Classification and clustering of arguments with contextualized word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 567–578, Florence, Italy. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Christian Stab and Iryna Gurevych. 2017. Parsing argumentation structures in persuasive essays. *Computational Linguistics*, 43(3):619–659.

Christian Stab, Tristan Miller, Benjamin Schiller, Pranav Rai, and Iryna Gurevych. 2018. Cross-topic argument mining from heterogeneous sources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3664–3674, Brussels, Belgium. Association for Computational Linguistics.

Derek Tam, Rakesh R. Menon, Mohit Bansal, Shashank Srivastava, and Colin Raffel. 2021. Improving and simplifying pattern exploiting training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4980–4991,

Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Lewis Tunstall, Nils Reimers, Unso Eun Seo Jo, Luke Bates, Daniel Korat, Moshe Wasserblat, and Oren Pereg. 2022. Efficient few-shot learning without prompts. *CoRR*, abs/2209.11055.

Henning Wachsmuth, Shahbaz Syed, and Benno Stein. 2018. Retrieval of the best counterargument without prior topic knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 241–251, Melbourne, Australia. Association for Computational Linguistics.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Albert Webson and Ellie Pavlick. 2022. Do prompt-based models really understand the meaning of their prompts? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2300–2344, Seattle, United States. Association for Computational Linguistics.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. ConSERT: A contrastive framework for self-supervised sentence representation transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5065–5075, Online. Association for Computational Linguistics.

A Appendix

A.1 Data Details

The following section gives more details about the argument maps we use in this work.

The snapshot from Agarwal et al. (2022) contains a total of 1,560 argument maps. Using an automatic language-detection tool on a sample of the content of the map we assign a language to each map. The vast majority is English with very few other languages: 21 German, 6 Spanish, 5 French and 4 Italian. This makes conducting multilingual experiments even for mere evaluation challenging. As a result, we filter out all maps that are not-English and all with less than 19 nodes. Figure A.1 shows the distribution of maps with different amounts of nodes, the smallest having 19 nodes and the largest 6,252 nodes. Most

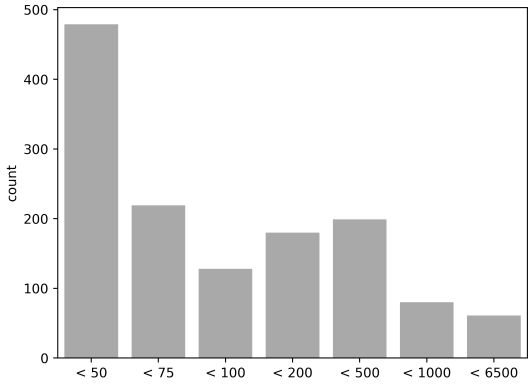


Figure A.1: Distribution of maps with different bin ranges of number of nodes per map.

argument maps is associated with a number of topic tags, which can be selected by the user creating a new argument tree on a specific thesis. We merge similar tags into more coarse-grained topics such that every map can be associated with one specific topic. Figure A.2 depicts the number of maps per general topic, showing that the data covers a variety of different domains but also that more specific topics occur less frequent (e.g. animals).

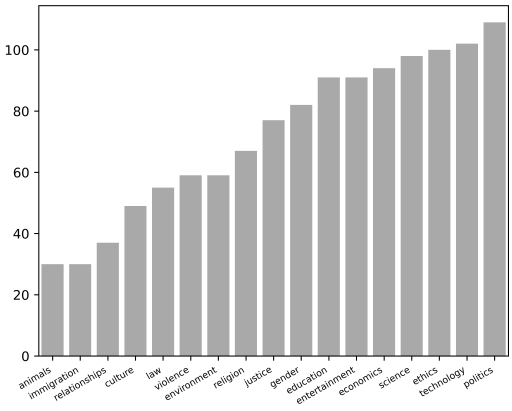


Figure A.2: Number of maps per coarse-grained topic.

Figure A.3 gives an idea about how many actual *parent nodes* are available. Most parent nodes have between 1 and 3 children with some exceptions having a very large amount of children (e.g. one node has 411 direct children).

Figure A.4 compares the distribution of nodes that act as a pro vs. nodes that act as a con. The distributions do not completely overlap as the majority of the data is slightly biased towards con.

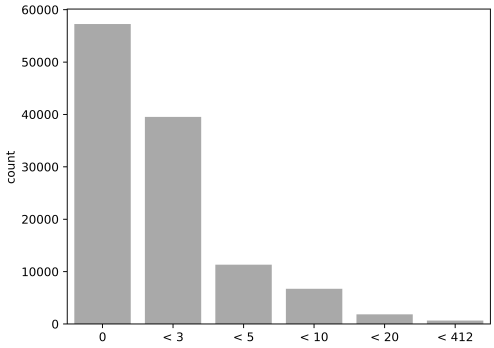


Figure A.3: Distribution of nodes with different bin ranges of number of children.

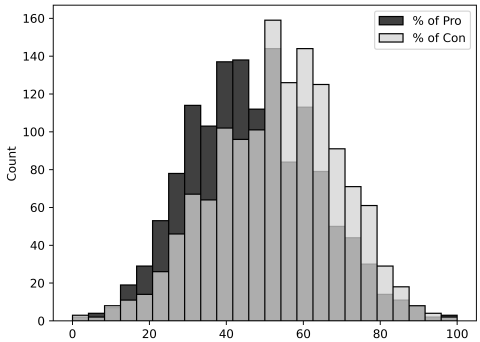


Figure A.4: Comparison between relative amount of pros vs. cons. Distribution of maps with a certain percentage of pro/con.

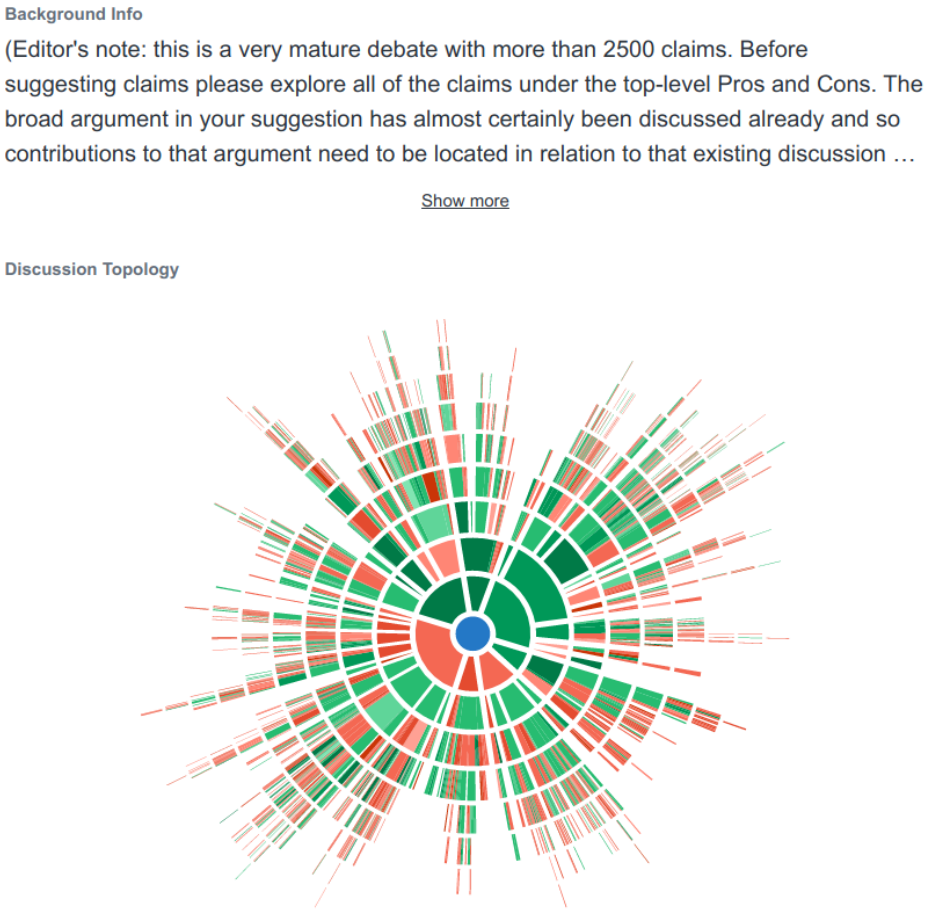


Figure A.5: Discussion Topology of Universal Basic Income (UBI)[9]. Users are advised to explore the map first which could be made easier by allowing the user to enter a short keyword-argument and then highlighting possible parents for this argument. Varying color intensity could be used to visualize multiple arguments that are relevant at varying degrees. After the user is finished with exploration, the final argument could possibly be moved to more suitable parents that are automatically suggested to keep the discussion well structured.

A.2 Annotation Study

The 4 authors annotated 20 samples while developing the guideline. We further recruited 3 student assistant as annotators, who have been paid 12,87 Euro per hour. The student assistants were Master Students of Computational Linguistics and Digital Humanities and have all participated in an Argument Mining course. Two annotators were female, one male. All have a very high level of English proficiency (one native speaker). Countries of origin: Canada, Pakistan, Germany. The annotators were aware that the data from the annotation study was used for the research purposes of our project.

Kialo Argumentation Annotation Task

Task Overview

The goal of this task is to find comment pairs that are most likely to have the relationship "parent-child" in an argument setting. More specifically, given a target comment (*child*), you must select the most likely *parent* of this comment from a list of candidates, aka, what the given comment is in response to.

It is important to note that the comments are from a debate/deliberation forum, and so the child can either be in support of the parent statement, *pro*, or contrary to the parent, *con*. It is not necessary to annotate that relation, however it is important to keep in mind when annotating.

Data

The forum in consideration is here: https://www.kialo.com/

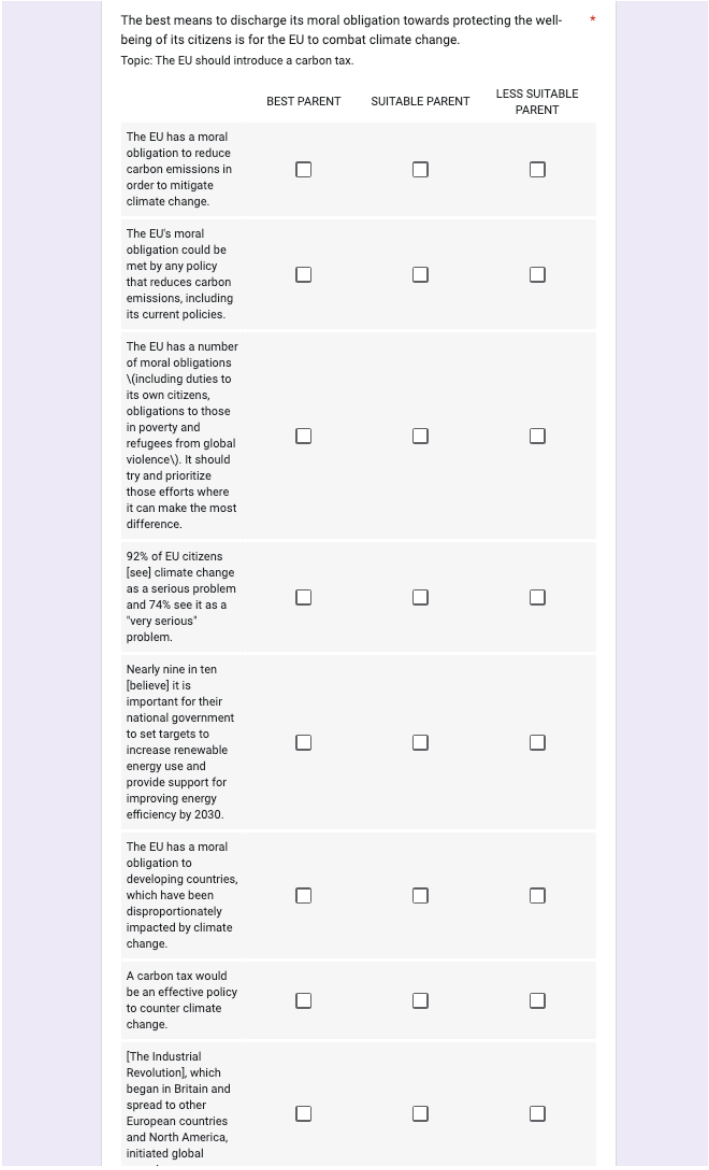
Please go through a few examples and familiarize yourself with the structure of the argument maps.

Annotation Guidelines

1. You will be given a target comment (aka the *child*). E.g., *"Censorship leads to narrow mindedness by preventing sincere and open discussion."*
2. You will then be presented X candidates for the *parent* comment, i.e., the comment to which the *child* is in response to.
3. You will have to annotate each candidate with one of three categories:
    a. Best parent (count 1): This is the candidate that you believe is the **most likely parent** to the target comment, e.g., *"There should be no limits on freedom of speech."* (Note, the parent-child relationship here is that of support, so *pro*).
    b. Other suitable parent (max count 4): These are those that you consider alternatives to the best parent, runners-up in other words.
    c. Less suitable parent (no max count): These are those in which you do not see a connection with the target comment in an argument setting. The motives could vary, such as different topics, no logical connection between the two, etc.

Strategy: we suggest you first split all candidates into categories (b) and (c) in the first sweep, then rank those in (b) to select (a).

Figure A.6: Annotation guidelines provided to the annotators. The participants were additionally trained with a small pilot study, assisted by one of the authors, to familiarize themselves better with the task.



The best means to discharge its moral obligation towards protecting the well-being of its citizens is for the EU to combat climate change. *

Topic: The EU should introduce a carbon tax.

	BEST PARENT	SUITABLE PARENT	LESS SUITABLE PARENT
The EU has a moral obligation to reduce carbon emissions in order to mitigate climate change.	☐	☐	☐
The EU's moral obligation could be met by any policy that reduces carbon emissions, including its current policies.	☐	☐	☐
The EU has a number of moral obligations \(including duties to its own citizens, obligations to those in poverty and refugees from global violence\). It should try and prioritize those efforts where it can make the most difference.	☐	☐	☐
92% of EU citizens [see] climate change as a serious problem and 74% see it as a "very serious" problem.	☐	☐	☐
Nearly nine in ten [believe] it is important for their national government to set targets to increase renewable energy use and provide support for improving energy efficiency by 2030.	☐	☐	☐
The EU has a moral obligation to developing countries, which have been disproportionately impacted by climate change.	☐	☐	☐
A carbon tax would be an effective policy to counter climate change.	☐	☐	☐
[The Industrial Revolution], which began in Britain and spread to other European countries and North America, initiated global	☐	☐	☐

Figure A.7: Sample of an annotation instance as presented to annotators. The first sentence is the *child*, the *topic* of the discussion is presented immediately below the child, and the *candidates* are then listed. The annotators can select 1 BEST PARENT, a maximum of 4 SUITABLE PARENTs, and the rest are LESS SUITABLE PARENTs.

A.3 Metrics

For one sample:
top1: accuracy at rank 1. 1 if actual answer is at rank 1, 0 otherwise
top5: accuracy at rank 5. 1 if actual answer is in the top-5 ranked, 0 otherwise
MRR: the mathematical inverse of the rank of the actual answer in the ranked predictions
The metrics are averaged for all samples so MRR for Q samples:

$$MRR = \frac{1}{Q} \sum_{i=1} \frac{1}{rank_i}$$

A.4 Models & Training Details

base model	size	name in experiments	intermediate-task training	name in huggingface
MPNet	110M	mpnet	none	microsoft/mpnet-base
		nli-mpnet	MNLI + SNLI	sentence-transformers/nli-mpnet-base-v2
		paraphrase-mpnet	(above) + paraphrase	sentence-transformers/paraphrase-mpnet-base-v2
		all-mpnet	(above) + QA & more	sentence-transformers/all-mpnet-base-v2
MiniLM	33M	mini	(above)	sentence-transformers/all-MiniLM-L6-v2

Table A.1: Models used in the experiments [10]

Software: We use sentence-transformers[11] for our experiments. Our code is made publicly available[12].
Hardware: NVIDIA RTX A6000 with 48G memory is used for training and inference.
Average runtime: training for 1 epoch using the full training dataset takes around (in minutes):
for MPNet
0:22 with no template or parent/child template
0:44 with pro/con template (double data size)
1:44 with all templates
for MiniLM
0:08 with no template or parent/child template
0:14 with pro/con template (double data size)
0:34 with all templates
Hyperparameters: Default hyperparameters are used to avoid influencing few-shot results which also kept computational cost minimal. The hyperparameters used are following:
batch size = 64, learning rate = 2e-5 with 10% of training steps as warm-up steps.

A.5 Templates

name	#templates	template
parent/child (main template)	1	`parent: "parent text"` `child: "child text"`
pro/con	2	above + `pro: "child text"` or `contra: "child text"`

all	5	above +
		`"child text" parent: "parent text"`
		`child: "child text" parent: "parent text"`
		`pro: "child text" parent: "parent text"` or
		`contra: "child text" parent: "parent text"`

Table A.2: Templates used in the experiments. #templates is the number of possible templates to apply per sample.

Other templates with a more expressive form yielded similar results:
`This sentence: "child text" is child`

[11] https://github.com/UKPLab/sentence-transformers
[12] https://github.com/imanjundi/argument-relations

name	#templates	template
foo/bar (main template)	1	`foo: "parent text"` `bar: "child text"`
pro/con	2	above + `baz: "child text"` or `qux: "child text"`

all	5	above +
		`"child text" bar: "parent text"`
		`foo: "child text" bar: "parent text"`
		`baz: "child text" bar: "parent text"` or
		`qux: "child text" bar: "parent text"`

Table A.3: Templates used to analyse the effect of template semantic.

`"child text" is a child of "parent text"`

A.6 Supplementary Results

model	test (20%) all possible candidates			200 samples 10 candidates		
	top1	top5	MRR	top1	top5	MRR
zero-shot	.2519	.5448	.3917	.400	.900	.611
no template	$.2806_{\pm.0082}$	$.5737_{\pm.0123}$	$.4183_{\pm.0095}$	$.4180_{\pm.0144}$	$.9070_{\pm.0057}$	$.6207_{\pm.0077}$
parent/child (*)	$.2900_{\pm.0081}$	$.5876_{\pm.0125}$	$.4285_{\pm.0098}$	$.4190_{\pm.0082}$	$.9100_{\pm.0061}$	$.6214_{\pm.0041}$
pro/con (*)	$.2917_{\pm.0078}$	$\mathbf{.5898}_{\pm.0114}$	$.4301_{\pm.0093}$	$\mathbf{.4340}_{\pm.0082}$	$\mathbf{.9210}_{\pm.0089}$	$\mathbf{.6322}_{\pm.0058}$
all (*)	$\mathbf{.2932}_{\pm.0076}$	$\mathbf{.5897}_{\pm.0128}$	$\mathbf{.4316}_{\pm.0096}$	$.4280_{\pm.0144}$	$.9180_{\pm.0097}$	$.6281_{\pm.0068}$
human				.480	.935	.664

Table A.4: Results after **high-resource** training of **all-mini model**. Training improves the performance, but it still lags behind human performance. Using parent/child template boosts the performance although not much as with MPNet (Table 3), and adding more templates slightly improves the performance. (*) denotes training with templates.

model	test			samples		
	all	pro	con	all	pro	con
zero-shot	.2491	.2446	.2482	.430	.426	.434
no template	.3064	.2793	**.3277**	.485	.465	**.505**
parent/child	.3441	.3068	**.3689**	.515	.455	**.576**
human				.480	.436	**.525**

Table A.5: **top1** according to relation type (**pro/con**). Comparable performance for pro and con for zero-shot. Those could be the more straightforward cases where there is a higher similarity between the parent & child as compared to other nodes in the tree (fewer other good potential parents), so p@1 would be similar for both pro & con. The performance of con is noticeably improved with training and is better overall than pro.

#nodes	#maps	top1	top5	mrr
8	8	$.2570_{\pm.0017}$	$.5571_{\pm.0035}$	$.3980_{\pm.0020}$
16	16	$.2776_{\pm.0075}$	$.5844_{\pm.0103}$	$.4199_{\pm.0080}$
32	32	$.2860_{\pm.0027}$	$.5967_{\pm.0017}$	$.4296_{\pm.0023}$
64	64	$.2869_{\pm.0030}$	$.5933_{\pm.0044}$	$.4284_{\pm.0027}$

Table A.6: Longer few-shot training for *2 epochs* without a template. Performance is still worse than pro/con with 1 epoch training.

#nodes	#maps	top1	top5	mrr
8	8	$.2633_{\pm.0018}$	$.5650_{\pm.0034}$	$.4047_{\pm.0021}$
16	16	$.2831_{\pm.0079}$	$.5925_{\pm.0091}$	$.4253_{\pm.0079}$
32	32	$.2836_{\pm.0028}$	$.5946_{\pm.0030}$	$.4272_{\pm.0029}$
64	64	$.2828_{\pm.0031}$	$.5906_{\pm.0027}$	$.4242_{\pm.0027}$

Table A.7: Longer few-shot training for **5 epochs** without a template. Performance is still worse than *all* template with 1 epoch training.

#nodes	#maps	top1	top5	mrr
8	8	$.2788_{\pm.0057}$	$.5825_{\pm.0059}$	$.4206_{\pm.0054}$
16	16	$.2860_{\pm.0053}$	$.5984_{\pm.0091}$	$.4299_{\pm.0065}$
32	32	$.2855_{\pm.0026}$	$.5964_{\pm.0019}$	$.4293_{\pm.0021}$
64	64	$.2976_{\pm.0039}$	$.6014_{\pm.0022}$	$.4383_{\pm.0035}$

Table A.8: Longer few-shot training for **5 epochs** with *parent/child* template. Performance is still worse than *all* template with 1 epoch training.

model	test			samples		
	top1	top5	MRR	top1	top5	MRR
parent/child	.3441	.6539	.4853	.54	.94	.70
foo/bar	.3432	.6529	.4856	.52	.94	.69

Table A.9: **Full training** results using a **meaningless template** (foo/bar) are similar to that of a meaningful one (parent/child).

#nodes	#maps	template	top1	top5	MRR
8	8	all	$.2802_{\pm.0061}$	$.5845_{\pm.0106}$	$.4218_{\pm.0084}$
		all meaningless	$.2797_{\pm.0088}$	$.5827_{\pm.0127}$	$.4214_{\pm.0103}$
16	16	all	$.2890_{\pm.0015}$	$.6027_{\pm.0049}$	$.4339_{\pm.0016}$
		all meaningless	$.2904_{\pm.0030}$	$.6051_{\pm.0034}$	$.4356_{\pm.0026}$
32	32	all	$.2923_{\pm.0059}$	$.6018_{\pm.0068}$	$.4347_{\pm.0060}$
		all meaningless	$.2901_{\pm.0045}$	$.6011_{\pm.0035}$	$.4337_{\pm.0038}$
64	64	all	$.3018_{\pm.0025}$	$.6025_{\pm.0035}$	$.4416_{\pm.0020}$
		all meaningless	$.3018_{\pm.0035}$	$.6045_{\pm.0047}$	$.4421_{\pm.0039}$

Table A.10: **Few-shot** results using **multiple meaningless templates**. The results are similar to that of meaningful templates.

			test			samples		
#nodes	#maps	template	top1	top5	MRR	top1	top5	MRR
zero-shot			.2519	.5448	.3917	.400	.900	.611
8	8	none	$.2531_{\pm.0009}$	$.5473_{\pm.0017}$	$.3929_{\pm.0009}$	$.4020_{\pm.0084}$	$.8990_{\pm.0042}$	$.6117_{\pm.0047}$
		parent/child	$.2513_{\pm.0051}$	$.5305_{\pm.0058}$	$.3846_{\pm.0059}$	$.3900_{\pm.0094}$	$.8730_{\pm.0045}$	$.5943_{\pm.0053}$
		pro/con	$.2564_{\pm.0049}$	$.5390_{\pm.0063}$	$.3904_{\pm.0050}$	$.3870_{\pm.0076}$	$.8820_{\pm.0110}$	$.5932_{\pm.0054}$
		all	$\mathbf{.2647}_{\pm.0060}$	$\mathbf{.5498}_{\pm.0081}$	$\mathbf{.3996}_{\pm.0062}$	$\mathbf{.3950}_{\pm.0106}$	$\mathbf{.9010}_{\pm.0042}$	$\mathbf{.5994}_{\pm.0055}$
16	16	none	$.2586_{\pm.0024}$	$.5536_{\pm.0039}$	$.3983_{\pm.0027}$	$\mathbf{.4100}_{\pm.0061}$	$\mathbf{.9020}_{\pm.0045}$	$\mathbf{.6146}_{\pm.0034}$
		parent/child	$.2668_{\pm.0031}$	$.5537_{\pm.0039}$	$.4026_{\pm.0034}$	$.3850_{\pm.0158}$	$.8950_{\pm.0071}$	$.5952_{\pm.0064}$
		pro/con	$.2674_{\pm.0048}$	$.5580_{\pm.0072}$	$.4040_{\pm.0052}$	$.3950_{\pm.0187}$	$.8990_{\pm.0055}$	$.6018_{\pm.0064}$
		all	$\mathbf{.2704}_{\pm.0033}$	$\mathbf{.5627}_{\pm.0045}$	$\mathbf{.4077}_{\pm.0038}$	$.3880_{\pm.0241}$	$.8970_{\pm.0076}$	$.6012_{\pm.0111}$
32	32	none	$.2651_{\pm.0014}$	$.5590_{\pm.0047}$	$.4040_{\pm.0023}$	$\mathbf{.4070}_{\pm.0104}$	$\mathbf{.9010}_{\pm.0042}$	$\mathbf{.6114}_{\pm.0055}$
		parent/child	$.2690_{\pm.0020}$	$.5604_{\pm.0020}$	$.4054_{\pm.0017}$	$.3950_{\pm.0094}$	$.8990_{\pm.0042}$	$.6044_{\pm.0045}$
		pro/con	$\mathbf{.2703}_{\pm.0034}$	$\mathbf{.5611}_{\pm.0015}$	$\mathbf{.4074}_{\pm.0023}$	$.3920_{\pm.0125}$	$.9010_{\pm.0055}$	$.6046_{\pm.0089}$
		all	$.2672_{\pm.0021}$	$.5571_{\pm.0025}$	$.4039_{\pm.0013}$	$.3790_{\pm.0108}$	$.8980_{\pm.0097}$	$.5953_{\pm.0073}$
64	64	none	$.2663_{\pm.0025}$	$.5601_{\pm.0011}$	$.4054_{\pm.0020}$	$\mathbf{.4030}_{\pm.0057}$	$\mathbf{.9130}_{\pm.0045}$	$\mathbf{.6132}_{\pm.0046}$
		parent/child	$\mathbf{.2675}_{\pm.0015}$	$\mathbf{.5624}_{\pm.0019}$	$\mathbf{.4068}_{\pm.0010}$	$.3990_{\pm.0185}$	$.9100_{\pm.0100}$	$.6109_{\pm.0120}$
		pro/con	$.2673_{\pm.0034}$	$.5599_{\pm.0035}$	$.4055_{\pm.0032}$	$.3940_{\pm.0108}$	$.9100_{\pm.0117}$	$.6070_{\pm.0074}$
		all	$.2612_{\pm.0024}$	$.5520_{\pm.0018}$	$.3990_{\pm.0024}$	$.3750_{\pm.0154}$	$.8960_{\pm.0164}$	$.5925_{\pm.0108}$
full dataset		none	**.2860**	**.5824**	**.4256**	**.430**	**.905**	**.625**

Table A.11: **Few-shot Results** of **all-mini model**. Few-shot improves over zero-shot in all cases. Using templates is not as effective for smaller models especially for larger #samples

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
8

☑ A2. Did you discuss any potential risks of your work?
9

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

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

B ☑ Did you use or create scientific artifacts?

3, 5.2

☑ B1. Did you cite the creators of artifacts you used?
3, 5.2

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☐ 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.?
A1

☑ 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.
5.2, 6.1

C ☑ Did you run computational experiments?

5, 6

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
A.4

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?
A.4

☑ 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?
5, 6

☑ 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.)?
A.4

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?
4

☑ 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.?
A.2 (figure A.7, A.8)

☑ 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)?
4, A.2

☑ 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?
A.2

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☑ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
A.2