May 2020

Argumentative Relation Classification with Background Knowledge

Debjit PAUL [a], Juri OPITZ [a], Maria BECKER [a] Jonathan KOBBE [b] Graeme HIRST [c] Anette FRANK [a]

[a] *Department of Computational Linguistics, Heidelberg University*
[b] *Data and Web Science, University of Mannheim*
[c] *Department of Computer Science, University of Toronto*

Abstract. A common conception is that the understanding of relations that hold between argument units requires knowledge beyond the text. But to date, argument analysis systems that leverage knowledge resources are still very rare. In this paper, we propose an unsupervised graph-based ranking method that extracts relevant multi-hop knowledge from a background knowledge resource. This knowledge is integrated into a neural argumentative relation classifier via an attention-based gating mechanism. In contrast to prior work we emphasize the selection of *relevant* multi-hop knowledge, and apply methods to *automatically enrich* the knowledge resource with missing knowledge. We assess model performance on two datasets, showing considerable improvement over strong baselines.

Keywords. argumentative relation classification, commonsense knowledge relations, multihop knowledge paths, knowledge graph completion, graph-based ranking

1. Introduction

Automatically identifying relations between argumentative text units (e.g., *support* and *attack* relations) has attracted much attention [1,2,3,4]. *Argumentative relation classification* (henceforth *ARC*) is the task of determining the type of relation that holds between two argumentative units (AUs, for short). This task has some overlap with *stance detection*, but differs in important aspects: while stance detection aims at determining the relation of AUs *towards a topic* or conclusion, argumentative relation classification analyzes relations *between argumentative units*. In this work we consider both *argument-topic relations* and *argument-argument relations* – since only a system that captures both types of relations can be applied in a real debate. We propose a ranking-based knowledge-knowledge-enhanced argumentative relation classification approach that we successfully apply to both (closely related) argumentative relation classification tasks.

Defining abstract semantic patterns is one way to explain argumentative relations [5]. In Fig. 1 Arg_1 implies that x is ***good for*** landlords, while Arg_2 implies that x is ***bad for*** tenants, with x = *'rise in price'*. This pattern can indicate *attack*. But Arg_2 states that *x should be limited* and thus the correct relation is *support* (Arg_1,Arg_2). Hence, we not only need good analysis of the text, but also further, so-called commonsense knowledge about the events, entities and relations mentioned in it, in order to gain true understanding

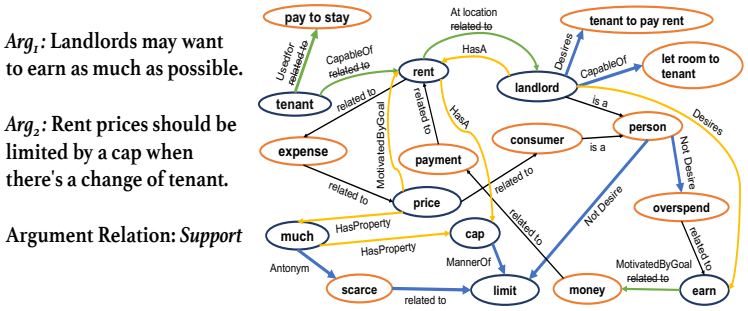


Figure 1. A subgraph extracted from ConceptNet. Blue edges portray relevant knowledge paths from ConceptNet. Concepts from the text in blue; intermediate nodes in orange. Yellow color edges: Our *on-the-fly* knowledge-base completion method infers ConceptNet relations. Green color edges: The knowledge-base completion feature of our method replaces *'related to'* relations with more specific ConceptNet relations.

of an argument. For example, we need to know that *landlords* and *tenants* are in a relation where one pays the other, with conflicting interests in the amount to be paid.

In this work we propose to leverage commonsense knowledge from ConceptNet [6] in order to connect pairs of concepts in argumentative units with implicit background knowledge relations. Fig. 1 shows a semantic (sub)graph with nodes representing concepts and edges (e.g., *'not desire'*) indicating relations between them. The graph captures semantic relations between entities (*tenant – landlord*) and properties (*much – limited*).

Our hypothesis is that capturing commonsense knowledge relations within and between AUs is essential for deeper understanding of arguments, especially for aspects of practical reasoning, cf. [7]. We investigate this hypothesis by devising a system that constructs subgraphs over pairs of AUs based on relevant concepts and multi-hop knowledge from the ConceptNet graph [6]. We propose a graph-based ranking method to extract relevant paths from these subgraphs that connect the argumentative units. Further, we dynamically enrich these graphs to counter sparsity problems when analyzing texts. Finally, we leverage knowledge from WordNet definitions to expand the meaning of words. E.g., a *tenant "... pays rent to use ... a building ... that is owned by someone else."*

Our contributions are: (i) We show that our graph-based method that extracts relevant commonsense knowledge and selectively integrates it into the model improves over a strong neural *and* a linear argumentative relation classification system on two datasets with different relation types; (ii) we show that enriching knowledge resources 'on the fly' can further improve results; and (iii) we provide an enhanced dataset for *support/attack* classification derived from Debatepedia. Our code and datasets will be made public.

2. Related Work

Argumentative Relation Classification (ARC) has been addressed in various works: [3] identify argument component types (*premise*, *claim*, *major-claim*) and argumentative relations (*support*, *non-support*) using structural, lexical and syntactic features using production rules, similar to [8]. Their system is extended by [9], who exploit the context of argumentative statements. [10] use a joint approach that, given a pre-segmented text, reconstructs the argument structure. This includes identifying the argumentative role (*pro* or *opposing*) of each segment and the argumentative function of each relation (*support* or

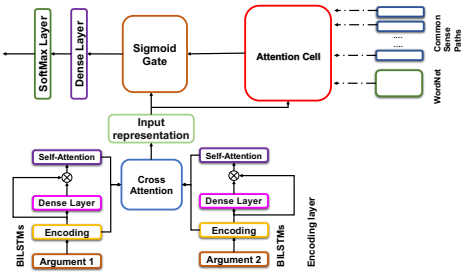
attack). [11] propose the first end-to-end approach to solve argument component and relation identification, comparing a joint model to a pipeline system. [4] propose an end-to-end approach for argument structure reconstruction. Similar to [10], they predict whether there is an argument relation between AU pairs and whether it is *support* or *attack*. While they predict relations jointly with the argument component type, they predict the relation label independently. We also classify relations between AUs into *support* and *attack*. However, [12] show that systems applied to this task tend to focus on discourse clues instead of the content and can be easily fooled when relying on discourse indicators. We therefore adopt an experimental setting that focuses only on the content of AUs.

Recent approaches to argumentative relation classification (ARC) have been built on Siamese networks [13,14,15]. In our work we devise a strong neural system with self- and cross-attention as a novel baseline for ARC. But in contrast to previous work, we leverage commonsense knowledge for ARC and extend this system with a mechanism to inject full-fledged knowledge paths that we select from a background knowledge graph.

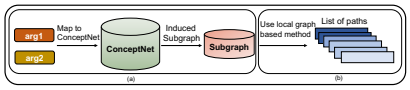
Background Knowledge for Argumentation When humans are debating (*Should rent prices be capped?*), they make use of background knowledge. Often, this knowledge belongs to the "content [that] is not expressed explicitly but resides in the mind of communicator and audience" [16,17]. Yet, few approaches have tried to leverage such knowledge in computational argumentation models, especially when it comes to commonsense knowledge (CSK). Previously, [14] investigate the impact of CSK in argumentative relation classification using linguistic and knowledge graph features derived from DBpedia and ConceptNet. They connect AUs via the knowledge graph, they use quantitative features that they derive from the established knowledge paths (edges only, i.e., deprived from concepts) to predict the argumentative relation between them. They extract a huge number of connecting paths, which they aggregate to patterns of relation types occurring in them. While [14] use only (features over) isolated relations (edges) that connect pairs of AUs, without filtering them by relevance, our work will filter and weight the knowledge paths and will include concepts (nodes) on the paths.

Besides CSK, other knowledge sources have been leveraged for argumentation. For example, Wikipedia articles [18], SNLI data [19] or sentiment lexica [20] have proven to be effective. [21] shows that the *Generative Lexicon* [22] captures relevant commonsense knowledge for argument mining in its qualia roles, such as physical or telic properties. However, such lexicons are hard to create and existing resources are little. [23] derive embeddings for FrameNet frames and entities from Wikidata to solve the Argument Reasoning Comprehension task [24]. They find small improvements from adding this knowledge and conclude that external knowledge alone is insufficient for improving argumentative reasoning. We are solving a related but different task and use different resources for injecting *commonsense knowledge*. Most importantly, while [23] integrate pre-trained embeddings computed over FrameNet and WikiData graphs at the token level, we pursue targeted knowledge selection from ConceptNet by inducing knowledge subgraphs between AUs that we extract from case-specific multi-hop knowledge paths using graph-based ranking.

Our goal is to take a step beyond the prior work by (i) studying how *relevant* knowledge can be selected that is tailor-cut to solving the relation classification task, by (ii) refining the extracted knowledge and leveraging an in-depth encoding of the paths, and finally by (iii) efficiently integrating this knowledge in a strong ARC approach.



(a) **ARK:** Argumentative relation classification (ARC) with self-attention and knowledge (ARK)



(b) Commonsense Knowledge Extraction. Left: Subgraph Construction. Right: Ranking & Path Selection

Figure 2. (a) ARC model with knowledge (ARK) and (b) Commonsense Knowledge Extraction

3. Argumentative Relation Classification with Commonsense Knowledge

We propose a neural Argumentative Relation Classification (ARC) system that (i) encodes pairs of argumentative units (AUs) using a cross-sentence attention mechanism over attentive BiLSTM encoders to understand their contextual features and structures; (ii) we leverage commonsense knowledge by linking concepts from the AUs to concepts from ConceptNet, and construct instance-specific subgraphs from which we extract relevant knowledge paths using graph-based ranking methods; finally (iii), we incorporate lexical knowledge from WordNet – *Synonyms* and definitions – to expand the meaning of terms in the AUs. Recently, [25] and [26] proposed methods to select multi-hop knowledge paths for reading comprehension and human needs classification: the former use heuristics, the latter graph-based measures for selection. In our work, we construct a knowledge subgraph over AUs and use local graph measures to select relevant knowledge for predicting the correct argumentative relation class. The selected knowledge paths along with *Synonyms* and definitional knowledge are encoded and incorporated into the relation prediction component. We use an attention cell that jointly encodes the encoded argument pair representations and the selected knowledge paths to predict implicit knowledge relations during inference. Figure 2 gives an overview of the model.

3.1. Argumentative Relation Classifier

The core of our model consists of three components: (1) encoding layer, (2) attention layer with *self-attention and cross-attention*, (3) output layer. The BiLSTM encoder takes two AUs $arg \in [arg1, arg2]$ as inputs: sequences of tokens $w_1^{arg}, ..., w_n^{arg}$ (or $w_{1:n}^{arg}$).

Encoding Layer We map the sequence of tokens of both AUs to sequences of word representations using word embeddings, and encode them with a single-layer BiLSTM.[1]

Attention Layer We apply ***self-attention*** to capture the contribution of each token in the argument [27]. We obtain argument representations x^{arg1} and x^{arg2} by taking the weighted sum of the attention scores and the hidden states that were generated by the BiLSTM.

We capture the relevance of the hidden representations of the arguments with ***cross-attention***. We calculate soft attention weights, this time across arguments and taking into account the self-attention weighted token representations from (1) and (2):

[1] The final state of the forward and backward pass is composed by taking the max over each dimension.

$$\hat{h_i}^{arg1} = \frac{\sigma(x_i^{arg2} h_i^{arg1})}{\sum_{j=1}^{N} \sigma(x_j^{arg2} h_j^{arg1})} \qquad \hat{h_i}^{arg2} = \frac{\sigma(x_i^{arg1} h_i^{arg2})}{\sum_{j=1}^{M} \sigma(x_j^{arg1} h_j^{arg2})} \tag{1, 2}$$

$$x_i^{arg1} = \sum_{j=1}^{N} \hat{h_j}^{arg1} h_i^{arg1}; \quad x_i^{arg2} = \sum_{j=1}^{M} \hat{h_j}^{arg2} h_i^{arg2} \tag{3}$$

with N, M the number of tokens in $arg1$ and $arg2$.

Output Layer We apply a final dense layer followed by softmax to predict the classes *support* or *attack*. As input y_i to this final layer we concatenate the output representations x_i^{arg1} and x_i^{arg2} from the cross-attention layer, and their difference vector $x_i^{arg1} - x_i^{arg2}$ and feed them through a projection layer: $y_i = ReLU(W_y[x_i^{arg1}; x_i^{arg2}; x_i^{arg1} - x_i^{arg2}] + b_y)$.

3.2. Commonsense Knowledge Extraction for Argumentative Relation Classification

Models for ARC will often require knowledge that is not overtly stated in the AUs or their context [28]. We aim to solve this issue by leveraging commonsense and lexical knowledge from resources such as ConceptNet and WordNet.

We begin by extracting connections between concepts mentioned in pairs of AUs from ConceptNet. For each pair we (i) collect all potentially relevant relations and concepts in a subgraph and (ii) select the top-ranked paths using local graph measures. Figure 2b, gives an overview of the extraction method.

Subgraph Construction For each pair $arg1$, $arg2$ we construct a subgraph $G' = (V', E')$ from ConceptNet $G = (V, E)$ by initializing V' with all concepts $c_{arg1} \in arg1$ and $c_{arg2} \in arg2$. To do so, we remove stop words, lemmatize tokens and perform n-gram matching of the remaining tokens to concepts in G. Similar to the subgraph construction in [25] and [26], we extend G' by including all concepts contained in the shortest paths between all concepts $c_i \in V'$ as well as all neighbouring nodes of concepts c_{arg} from $arg1$ and $arg2$. The final subgraph G' collects all edges E' from E that have both endpoints in V'.

Ranking and Selecting Paths We apply a two-step method: (i) **Collect top-n concepts**: Although most concepts in the AUs may be useful, considering all of them may introduce noise. For example, in Figure 1, the concept *possible* in arg_0 is not especially relevant in the given context. Therefore, we filter and collect the top-n concepts from each AU arg_i by ranking all the concepts $c_{arg_i} \in arg_i$ using personalized page rank [29] given the subgraph G' and all concepts $c_{arg_j} \in arg_j$ ($i \neq j$), i.e., the concepts mentioned in the other argumentative unit. (ii) **Select top-k paths**: We then collect all shortest paths between the remaining concepts (of length ≤ 4 hops). We rank each node in the path with *closeness centrality* [30] scores. We select the top-k paths that connect any pair of filtered concepts $c_{arg1} \in arg1$ and $c_{arg2} \in arg2$, which we denote as ***S*elected Knowledge Paths (SKP)**.

3.3. Knowledge Graph Completion (KGC)

Knowledge graphs are incomplete, so we expect them to be more effective after a knowledge base completion step. For ConceptNet, this task has been addressed using link pre-

diction [31,32]. [33] apply a pre-trained transformer model that learns to generate concepts as *phrase objects*, given an existing seed phrase (subject) and a ConceptNet relation label. On ConceptNet, they generate phrase objects with up to 91.7% precision. Human evaluations shows that the produced knowledge is novel and of high quality.

In contrast, [34] propose an open-world multi-label relation classification system[2] to predict ConceptNet *relation types* for given pairs of concepts. This system addresses specific properties of ConceptNet, such as the complexity of argument types and relation ambiguity. It encodes pairs of arguments (here, concepts) using word embedding inputs and an RNN component. The model constructs a joint representation that is projected to an output layer to predict one or several of 14 ConceptNet relation types (or none).

We adjust this classifier by: (1) refining the relation space and (2) pre- and postfiltering of concepts. Analyses in [34] show that the relation types *HasPrerequisite*, *HasSubevent* and *HasFirstSubevent* often co-occur, which indicates their ambiguity. To enhance the separation of these classes, we restructure the relation inventory. We retrain the classifier on the adapted dataset and perform pre- and postfiltering of concepts to reduce uninformative instances. Filtering steps include (i) TF-IDF filtering of concepts; (ii) excluding concepts covering more than two words; and (iii) type-based PoS sequence filtering on argument phrases. This enhances the performance by +9 pp. to 77 F1 score.

We apply the adapted KGC system to our data to predict and label *direct* links between any concepts detected in the AU pairs. We denote the predicted **extended knowledge relations EK** (for enhanced knowledge) and add it to the system (ARK+EK). In addition, we replace any *RelatedTo* triple in the Selected Knowledge Paths (SKP) with the predicted ConceptNet relations from EK and denote the result as **SKP***. E.g., an original triple is *umpire RelatedTo call* while the predicted triple is *umpire HasA call*. We update SKP to SKP*and combine it with additional predicted relations: ARK+SKP*+EK.

Lexical Knowledge WordNet[3] [35] is a widely used lexical resource. It defines the meaning of words and their relations for English. We employ WordNet's lexical knowledge by mapping each lemmatized token from the AUs to the WordNet graph, selecting the most frequent sense. We extract its SYNONYMS and sense definition. We denote WordNet knowledge as WN and knowledge acquired from WN as **Lexical Knowledge LK**.

3.4. Injecting Knowledge for ARC

We leverage commonsense knowledge for the ARC task from three sources: structured knowledge from ConceptNet via *Selected Knowledge Paths (SKP)* and *Enriched Knowledge (EK)*, and unstructured *Lexical Knowledge (LK)* from WordNet. SKP, EK and LK (SYNONYMS & Definitions) can all be represented as sets of (multi- or single-hop) paths $p_{1:l}$, i.e., sequences (of length l) of nodes (concepts) and edges (relation types). For LK, each path $p_{1:l}$ consists of the sequence of words from the sense definition of word w.[4]

Encoding Layer We use a single-layer BiLSTM to obtain encodings ($h^{k,i}$) for each knowledge path (h^k the encoded knowledge path, i the path index).

[2]https://gitlab.cl.uni-heidelberg.de/mbecker/corec—commonsense-relation-classifier

[3]https://wordnet.princeton.edu/

[4]We use the most frequent sense of w, as defined in WordNet. We embed each path $p^{k,i}_{1:l}$ with pretrained GloVe [36] embeddings ($k \in$ {SKP, EK, LK}).

Attention Cell We define a cell that allows the model to attentively encode the knowledge paths (see Figure 2a). We use an attention layer, where each encoded knowledge path interacts with the argument representations x^{arg} (4) (to receive attention weights ($\hat{h}^{k,i}$) from (5). In (5) we use sigmoid to calculate attention weights,

$$x_i^{arg} = [x_i^{arg1}; x_i^{arg2}; x_i^{arg1} - x_i^{arg2}] \qquad \widetilde{h}^{k,i} = \sigma(x_i^{arg} h^{k,i}), \quad \hat{h}^{k,i} = \frac{\widetilde{h}^{k,i}}{\sum_{j=1}^{N} \widetilde{h}^{k,j}} \qquad (4, 5)$$

To obtain the argument-aware commonsense knowledge representation x_i^k, we pass the output of the attention layer through a feedforward layer. W_k, b_k are trainable parameters.

$$x_i^k = ReLU(W_k(\sum_{j=1}^{N} \hat{h}^{k,j} h^{k,i}) + b_k) \qquad o_i = sigmoid(W_z[x_i^{arg}; x_i^k] + b_z) \qquad (6, 7)$$

To distill the selected and weighted knowledge into the model, we concatenate the argument x_i^{arg} and the knowledge x_i^k representation and process it by a dense layer (Eq. 8), with $\odot$ element-wise multiplication, $b_{\widetilde{y}_z}$ and $W_{\widetilde{y}_z}$ trainable parameters, y_i from *Output Layer*. Then, a sigmoid gate helps the model select when to incorporate knowledge x_i^k (Eq. 8).

$$z_i = softmax(W_{\widetilde{y}_z}(o_i \odot y_i + (1 - o_i) \odot x_i^k) + b_{\widetilde{y}_z}) \qquad (8)$$

We finally pass the representation to a softmax classifier to form a probability distribution over the two classes *attack* and *support*.

4. Experiments

4.1. Data There is are only few datasets for the ARC task. We use these two datasets:[5]
Student Essays. This well-established dataset comprises argumentative essays in English written by students. We use the extended v.02 with 402 essays [4]. An issue with this data is that many of the relations can be easily identified by observing shallow discourse clues (*however*, *moreover*). Therefore, we we use the more difficult *content-based* setup [12], where the relations between argumentative units have to be determined without looking at the textual discourse context of unit clauses.
Debatepedia The Debatepedia website[6] collects user-generated debates that each contain several arguments in favor of or opposed to the debate's topic. Topics are usually formulated as polar questions. [1] created a small dataset from Debatepedia consisting of 200 pairs of topics (questions) and associated pro vs. con arguments, as well as further dependent pairs of pro and con arguments among each topic. But the pairing of coherent pro and con arguments is difficult to establish automatically. We thus restrict ourselves to pairs of directly connected questions and pro/con arguments. To construct high-quality data, we manually reformulate the questions to statements. If an argument is in favor of the debated topic, the claim *supports* the topic. Else it *attacks* it.

[5] Below we summarize the data statistics:

Student Essay	train:	2803 / 273 (support / attack)	dev:	1017 / 132 (support / attack)
Debatepedia	train:	3240 / 3251 (support / attack)	dev:	1121 / 1042 (support / attack)

[6] http://www.debatepedia.org

4.2. Linear Classifier Baseline Among other text classification tasks, linear SVMs have been successfully applied to ARC [37,38,4,39]. Next to our neural system we thus implement an SVM model w/ and w/o knowledge enhancement. Below we describe text classification features used by our baseline SVM and explain ways of modeling and abstracting the knowledge paths to make them accessible for the SVM.

Text features. We feed the SVM a concatenation of the uni- and bigram (TF-IDF) representation of (i) source, (ii) target and (iii) the text overlap of source and target. We also concatenate averaged GloVe vectors to the bag-of-words feature representation; the vectors are separately averaged over (i), (ii) and (iii). We further concatenate to the vector the element-wise subtraction and multiplication of the averaged source from the averaged target GloVe vector, to model the argumentative relation as a directional vector.

Modeling paths as features. We investigate whether the extracted and selected knowledge paths (SKP) can improve the SVM classifier. But encoding paths is not straightforward for an SVM compared to encoding sequential paths with a recurrent NN. We thus apply the following steps: we represent every selected path as the mean vector of the token-wise GloVe vectors in a path. We then retrieve different path selections, e.g., the mean vector of all paths or the path-vector with the maximum and minimum norm. To determine the optimal selection jointly with the optimal SVM margin, we run a greedy hyper-parameter search on the development data. Details will be provided with the code.

4.3. Training Details **Objective** During training we minimize the cross-entropy loss between the predicted and the actual distribution. We use Adam optimizer [40] with an initial learning rate of 0.001, and batch size of 8/32 for Student Essays/Debatepedia. We use pretrained GloVe [36], ELMo [41] embeddings, a hidden size of 100 for all Dense Layers and L2 regularization with $\lambda = 0.01$. We use $k = 3$ for selecting top-ranked paths. For filtering the number of concepts with *personalized page rank* we use $n \leq 5$ concepts per AU. **Metrics** We report macro-averaged Precision (P), Recall (R), F1 scores.

5. Results

We examine 8 different systems: **random** baseline guesses labels according to the training data label distribution. **SVM** is a knowledge-agnostic linear classifier baseline. When we add selected knowledge paths via aggregation features, we denote this as **SVM+CN** (w/ knowledge from ConceptNet, including SKP* and EK) and as **SVM+CWN** (for the latter (+CN) extended with WordNet). **BiLSTM** is a neural knowledge-agnostic baseline and **Bi-ATT** denotes the BiLSTM with self- and co-attention (see Fig. 2a *w/o* Attention Cell and Sigmoid Gate). By further enriching Bi-ATT with knowledge paths through the Attention Cell, we obtain our main model: **ARK** (again in different varieties: **+CN**, etc.).

Table 1a reports our experiment results in averaged scores over five runs. Our models enhanced with knowledge (including SVM) perform significantly better ($p<0.05$) compared to their baselines, and similarly for ARK+CWN vs. KOB2019.

Knowledge helps The results show that adding selected knowledge to any of our baseline models improves their overall performance on both datasets and for both types of embeddings. Our full model **ARK** profits most from the added knowledge when compared to its knowledge-agnostic counterpart **Bi-ATT** (using ELMo: +4.27 pp. (percentage points) macro F1 in Student essays; +4.6 in Debatepedia; when using GloVe: +4.33

Table 1. (a) Classification results and (b) ablation study over K-path selection methods & K-graphs.

(a) Classification results. Bi-ATT = BiLSTM+Attention model, ARK = ARC model + Knowledge, where CN = ConceptNet (incl. SKP* + EK); WN = WordNet; CWN = ConceptNet (with SKP* + EK) + WordNet. Superscripts mark significant improvement (✓) or not (✗) of the result relative to the model the index names.

		Student essays			Debatepedia		
Model	**WE**	**P**	**R**	**F1**	**P**	**R**	**F1**
(1) random	-	49.68	49.66	49.65	50.04	50.03	50.01
(2) BiLSTM	G_{300d}	53.53	52.89	53.13	55.67	55.68	55.63
(3) KOB2019	G_{300d}	52.79	51.85	52.05 $^{(2)}$✗	58.06	57.75	57.04 $^{(2)}$✓
(4) KOB2019	ELMo	55.72	53.16	54.37 $^{(2)}$✓	59.16	59.17	59.11 $^{(2)}$✓
(5) SVM	G_{300d}	54.11	52.59	52.95	54.73	54.71	54.52
(6) SVM + CN	G_{300d}	54.11	54.23	54.17	56.12	56.00	55.58
(7) SVM + CWN	G_{300d}	55.80	56.38	**56.06** $^{(5,3)}$✓	56.60	56.57	**56.37** $^{(5)}$✓$^{(3)}$✗
(8) Bi-ATT	G_{300d}	54.46	53.31	53.70	56.20	56.19	56.18
(9) ARK + WN	G_{300d}	57.68	55.71	56.44	57.49	57.48	57.48
(10) ARK + CN	G_{300d}	57.64	57.71	57.67	57.38	57.25	57.31
(11) ARK + CWN	G_{300d}	60.70	55.55	**58.03** $^{(8,3)}$✓	58.78	58.43	**58.60** $^{(8,3)}$✓
(12) Bi-ATT	ELMo	56.44	54.77	55.16	59.10	59.08	59.09
(13) ARK + WN	ELMo	57.13	56.26	56.69	63.00	62.70	62.85
(14) ARK + CN	ELMo	59.13	**58.68**	58.89 $^{(12)}$✓	63.64	63.45	63.50 $^{(12)}$✓
(15) ARK + CWN	ELMo	**63.43**	55.90	**59.43** $^{(12,4)}$✓	63.72	63.65	**63.69** $^{(12,4)}$✓

(b) Ablation study over KnowledgePath (KPATH) selection methods & Knowledge Graphs (K). Models: random: 3 randomly chosen paths (= no selection); LK: Lexical Knowledge; SKP = Selected Knowledge Paths; SKP* = SKP w/ (*RelatedTo* → EK) and all WE=ELMo.

		Student essays			Debatepedia		
KPath selection	**K**	**P**	**R**	**F1**	**P**	**R**	**F1**
random KPaths	CN	56.73	57.80	57.16	60.50	60.16	60.33
SKP	CN	58.22	58.64	58.25	63.38	63.04	63.12
EK	CN	**59.58**	54.95	56.11	**63.90**	62.79	63.34
SKP* + EK	CN	59.13	58.68	**58.89**	63.64	**63.45**	**63.50**
LK	WN	57.13	56.26	56.69	63.00	62.70	62.85
SKP*+EK+LK	CWN	**63.43**	55.90	**59.43**	63.72	63.65	**63.69**

pp. in Student essays; +2.42 in Debatepedia). This finding not only applies to the global F1 metric, but also to macro Precision and Recall: we obtain considerable gains in Recall on Student essays of over 4 pp., i.e., a relative increase of more than 8%. Deeper analysis in §6 will show that knowledge helps especially for classifying rare *attack*-examples. We compare our knowledge representation and extraction method with the method in [14]. We empirically show that across two datasets and different embeddings we gain +4 F1 (on average) improvement. Knowledge also helps the linear SVM baseline (**SVM** vs. **SVM+CN/+CWN**). For both datasets we see gains. Adding only knowledge from ConceptNet improves over SVM by +1.22 pp. macro F1 in Student essays; +1.06 in Debatepedia. With access to the full knowledge we observe a more notable gain: +3.11 pp. macro F1 in Student essays; +1.85 pp. in Debatepedia (**SVM+CWN**). The fact that a linear classifier profits less from added knowledge compared to the neural system (**Bi-ATT** vs. **ARK**) is expected: the knowledge paths are sequential and thus easier to model with recurrent computations of the neural model. When computing path aggregates to make knowledge paths accessible for the SVM, we lose important structural information.

Ablation Study To gain better insight into the effects of different kinds of knowledge and selection methods, we run an ablation study over variants of ARK, where the number of paths is constant: In Table 1b row 1 we **randomly select** from the set of shortest knowledge paths between concepts appearing in AUs; row 2 uses knowledge selected using graph measures (**SKP**); row 3 shows model performance when using extended knowledge predicted on the fly with knowledge completion (**EK**); row 4 uses **both SKP* and EK**. Table 1b shows that using selected knowledge paths (SKP) improves F1 macro scores over models that use randomly selected knowledge paths. The effect is smaller for Student essays (+1.09 pp. F1), but considerable for Debatepedia (+2.79 pp. F1). Models that include automatically predicted knowledge for specific items (EK+SKP*) yield a further improvement of 0.64 and 0.38 pp. F1 macro scores for Student Essays and Debatepedia. This demonstrates that both knowledge selection and the instance-specific enrichment of the knowledge graph is important, and that EK complements SKP*.

Table 2. Example of knowledge paths used for prediction of argumentative relations.

Source	Relation	Predicted	Argumentative Unit 1	Argumentative Unit 2	Knowledge Paths
Essay	attack	attack	online classes have many advantages	traditional learning still has many benefits to the students	*benefit* RELATEDTO *advantage*; *online* ANTONYM *brick_and_mortar* SYNONYM *traditional*
Debate	support	support	Trans fats can be replaced w/o changing taste/price.	Trans fats should be banned.	*ban* ISA *action* RELATEDTO *change* RELATEDTO *replace*
Debate	support	attack	Instant replay call reviews should be implemented in baseball.	Instant replay makes game more about players, less about umps	*umps* FORM OF *ump* SYNONYM *umpire* RELATED TO *call*, *player* RELATEDTO *game* RELATEDTO *baseball* HASCONTEXT *ump* FORMOF *umps*

6. Analysis and Discussion

Minority Class Classification Data examples labeled with *attack* are scarce. This situation is extreme in Student Essays, where less than 10% of the data carry the *attack*-label. Therefore, systems usually struggle with this class (cf. [4]) and compensate for bad classification of *attack* examples with very good classification of *support* examples. Nevertheless, the minority class (*attack*) is at least as important as the *support* class. Thus, it is notable that our knowledge-enhanced systems **ARK+CN/+CWN** obtain a +53.8%/+93.3% relative increase in detecting the *attack* class (compared to the **Bi-ATT** baseline). WordNet, when used as sole source of knowledge (**ARK+WN**), leads to a lower but still remarkable improvement of +15.4%. A similar outcome is observed for the *linear model*: SVM+CN obtains a +66% increase for the *attack* class, while experiencing a -3.56% loss for *support*. To summarize, our selected paths greatly improve the results with respect to successfully predicting examples of the minority class (*attack*).

Knowledge Path Examples for Improved ARC To shed some light on how knowledge helps our ARC system, we analyze cases where the knowledge-enhanced neural model (**ARK**) corrects a mis-classification of the knowledge-agnostic model (**Bi-ATT**) with high probability. Some cases are displayed in Table 2. In the first case, a system lacking deeper knowledge can easily be fooled: both argument units contain phrases which are highly similar and carry positive sentiment (*advantages*; *benefits*) – yet, they are in an *attack* relation. A knowledgeable system, by contrast, would understand that 'online classes' and 'traditional learning' are opposites of each other. This valuable information is reflected in the retrieved two-hop path (right column): *online* ANTONYM brick-and-mortar SYNONYM *traditional*. To get from the *online*-concept to the *traditional*-concept, we have to traverse an ANTONYM-edge. This may signal to the system that despite the semantically highly similar content, the units are in fact attacking each other. In the second example, the system needs to understand that the word 'replace' in unit 1 has an implicit relation with 'banned' in unit 2 – again, this is captured by the selected path.

'Gay rights' or 'environment' – where does knowledge help? While our results indicate that knowledge is important for ARC, we found that the system needs more topic-specific common-sense knowledge. In the 3rd example (Table 2), although we extract and identify the relation between *players* and *umps* given the context, the missing knowledge is that in the sports domain, for replays *players* are more important than *umpires* – knowledge which we neither find in domain-specific KBs nor in commonsense KBs. We investigate the impact of knowledge infusion for different debate topics by clustering all topics in the dev set into 18 major areas (details will be released). '*Trans fats should be banned.*', e.g., appears in FOOD & NUTRITION; GAY RIGHTS includes debates such as *'Gay marriage should be legalized.'*. Figure 3 shows the comparative model perfor-

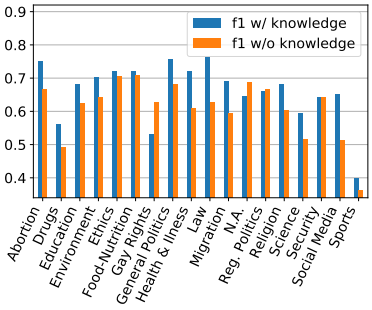


Figure 3. Macro F1 results of Bi-ATT vs. ARK+CWN model across 18 debate topic clusters (on DevSet).

mance over these topics. In 15 out of 18 topics injection of knowledge helps, especially in HEALTH, SOCIAL MEDIA and LAW, with great gains in macro F1 of more than 10 pp. By contrast, adding knowledge incurs a loss in GAY RIGHTS.

7. Conclusions

Determining relations between arguments requires knowledge beyond the text. In this work, we investigate ways of improving linear *and* neural systems by feeding knowledge paths that link concepts from two argumentative units. We extract the paths from background knowledge graphs and filter them with graph algorithms. Our experiments show that our method for incorporating commonsense knowledge is efficient for improving overall ARC results across two datasets. We show that extending the knowledge *on the fly* improves model performance – which further emphasizes the impact of knowledge for the task. An in-depth analysis shows that knowledge improves the performance across many topics, with very few exceptions. Finally, we provide an enhanced dataset for *support/attack* classification based on Debatepedia, which we will publicize.

References

[1] Cabrio E, Villata S. Natural language arguments: A combined approach. In: ECAI; 2012. p. 205–210.
[2] Stab C, Gurevych I. Annotating Argument Components and Relations in Persuasive Essays. In: COLING; 2014. p. 1501–1510.
[3] Stab C, Gurevych I. Identifying Argumentative Discourse Structures in Persuasive Essays. In: EMNLP; 2014. p. 46–56.
[4] Stab C, Gurevych I. Parsing Argumentation Structures in Persuasive Essays. Computational Linguistics. 2017;43(3):619–659.
[5] Reisert P, Inoue N, Okazaki N, Inui K. Deep Argumentative Structure Analysis as an Explanation to Argumentative Relations. In: ACL; 2017. p. 38–41.
[6] Speer R, Havasi C. Representing General Relational Knowledge in ConceptNet 5. In: LREC; 2012. p. 3679–3686.
[7] Walton D. Goal-based Reasoning for Argumentation. Cambridge University Press; 2015.
[8] Palau RM, Moens MF. Argumentation Mining: The Detection, Classification and Structure of Arguments in Text. In: ICAIL; 2009. p. 98–107.
[9] Nguyen H, Litman D. Context-aware Argumentative Relation Mining. In: ACL; 2016. p. 1127–1137.
[10] Peldszus A, Stede M. Joint Prediction in MST-style Discourse Parsing for Argumentation Mining. In: EMNLP; 2015. p. 938–948.

[11] Persing I, Ng V. End-to-End Argumentation Mining in Student Essays. In: NAACL; 2016. p. 1384–1394.

[12] Opitz J, Frank A. Dissecting Content and Context in Argumentative Relation Analysis. In: Workshop on Argument Mining; 2019. p. 25–34.

[13] Cocarascu O, Toni F. Identifying Attack and Support Argumentative Relations Using Deep Learning. In: EMNLP; 2017. p. 1374–1379.

[14] Kobbe J, Opitz J, Becker M, Hulpus I, Stuckenschmidt H, Frank A. Exploiting Background Knowledge for Argumentative Relation Classification. In: LDK; 2019. p. 1–8.

[15] Opitz J. Argumentative Relation Classification as Plausibility Ranking. In: KONVENS; 2019. p. 193–202.

[16] Moens MF. Argumentation Mining: How Can a Machine Acquire Common Sense and World Knowledge? Argument & Computation. 2018 01;9:1–14.

[17] Lawrence J, Reed C. Argument Mining: A Survey. Computational Linguistics. 2019;p. 1–55.

[18] Potash P, Bhattacharya R, Rumshisky A. Length, Interchangeability, and External Knowledge: Observations from Predicting Argument Convincingness. In: IJCNLP; 2017. p. 342–351.

[19] Choi H, Lee H. GIST at SemEval-2018 Task 12: A Network Transferring Inference Knowledge to Argument Reasoning Comprehension Task. In: SemEval; 2018. p. 773–777.

[20] Chen Z, Song W, Liu L. TRANSRW at SemEval-2018 Task 12: Transforming Semantic Representations for Argument Reasoning Comprehension. In: SemEval; 2018. p. 1142–1145.

[21] Saint-Dizier P. Knowledge-driven argument mining based on the qualia structure. Argument & Computation. 2017;8:193–210.

[22] Pustejovsky J. The generative lexicon. Computational linguistics. 1991;17(4):409–441.

[23] Botschen T, Sorokin D, Gurevych I. Frame- and Entity-Based Knowledge for Common-Sense Argumentative Reasoning. In: Workshop on Argument Mining; 2018. p. 90–96.

[24] Habernal I, Wachsmuth H, Gurevych I, Stein B. SemEval-2018 Task 12: The Argument Reasoning Comprehension Task. In: SemEval; 2018. p. 763–772.

[25] Bauer L, Wang Y, Bansal M. Commonsense for Generative Multi-Hop Question Answering Tasks. In: EMNLP; 2018. p. 4220–4230.

[26] Paul D, Frank A. Ranking and Selecting Multi-Hop Knowledge Paths to Better Predict Human Needs. In: NAACL; 2019. p. 3671–3681.

[27] Yang Z, Yang D, Dyer C, He X, Smola A, Hovy E. Hierarchical Attention Networks for Document Classification. In: NAACL; 2016. p. 1480–1489.

[28] Rajendran P, Bollegala D, Parsons S. Contextual Stance Classification of Opinions: A Step towards Enthymeme Reconstruction in Online Reviews. In: Workshop on Argument Mining; 2016. p. 31–39.

[29] Haveliwala TH. Topic-Sensitive PageRank. In: WWW; 2002. p. 517–526.

[30] Bavelas A. Communication Patterns in Task-Oriented Groups. Journal of the Acoustical Society of America. 1950;22(6):725–730.

[31] Li X, Taheri A, Tu L, Gimpel K. Commonsense Knowledge Base Completion. In: ACL; 2016. p. 1445–1455.

[32] Saito I, Nishida K, Asano H, Tomita J. Commonsense Knowledge Base Completion and Generation. In: CoNLL; 2018. p. 141–150.

[33] Bosselut A, Rashkin H, Sap M, Malaviya C, Asli C, Yejin C. COMET: Commonsense Transformers for Automatic Knowledge Graph Construction. In: ACL; 2019. p. 4762–4779.

[34] Becker M, Staniek M, Nastase V, Frank A. Assessing the Difficulty of Classifying ConceptNet Relations in a Multi-Label Classification Setting. In: RELATIONS - IWCS Workshop; 2019. p. 1–14.

[35] Miller GA. WordNet: A Lexical Database for English. Communications of the ACM. 1995;38(11):39.

[36] Pennington J, Socher R, Manning C. GloVe: Global Vectors for Word Representation. In: EMNLP; 2014. p. 1532–1543.

[37] Pradhan S, Hacioglu K, Krugler V, Ward W, Martin JH, Jurafsky D. Support Vector Learning for Semantic Argument Classification. Machine Learning. 2005;60(1–3):11–39.

[38] Kim Y. Convolutional Neural Networks for Sentence Classification. In: EMNLP; 2014. p. 1746–1751.

[39] Aker A, Sliwa A, Ma Y, Lui R, Borad N, Ziyaei S, et al. What Works and What does not: Classifier and Feature Analysis for Argument Mining. In: Workshop on Argument Mining; 2017. p. 91–96.

[40] Kingma DP, Ba JL. Adam: A Method for Stochastic Optimization. In: ICLR; 2014. p. 1–15.

[41] Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, et al. Deep Contextualized Word Representations. In: NAACL; 2018. p. 2227–2237.