Similarity-weighted Construction of Contextualized Commonsense Knowledge Graphs for Knowledge-intense Argumentation Tasks

Moritz Plenz[†] **Juri Opitz**[†] **Philipp Heinisch**[‡] **Philipp Cimiano**[‡] **Anette Frank**[†]

[†]Heidelberg University [‡]Bielefeld University
{plenz,opitz,frank}@cl.uni-heidelberg.de
{pheinisch,cimiano}@techfak.uni-bielefeld.de

Abstract

Arguments often do not make explicit how a conclusion follows from its premises. To compensate for this lack, we enrich arguments with structured background knowledge to support knowledge-intense argumentation tasks. We present a new *unsupervised* method for constructing *Contextualized Commonsense Knowledge Graphs* (CCKGs) that selects *contextually relevant* knowledge from large knowledge graphs (KGs) efficiently and at high quality. Our work goes beyond context-insensitive knowledge extraction heuristics by computing semantic similarity between KG triplets and textual arguments. Using these triplet similarities as weights, we extract *contextualized knowledge paths* that connect a conclusion to its premise, while maximizing similarity to the argument. We combine multiple paths into a CCKG that we optionally prune to reduce noise and raise precision. Intrinsic evaluation of the quality of our graphs shows that our method is effective for (re)constructing human explanation graphs. Manual evaluations in a large-scale knowledge selection setup confirm high recall and precision of implicit CSK in the CCKGs. Finally, we demonstrate the effectiveness of CCKGs in a knowledge-insensitive argument quality rating task, outperforming strong baselines and rivaling a GPT-3 based system.[1]

1 Introduction

Computational argumentation is a growing field with relevant applications, such as argument retrieval (Wachsmuth et al., 2017b; Bondarenko et al., 2021), argument analysis (Feng and Hirst, 2011; Janier et al., 2014; Wachsmuth et al., 2017a; Jo et al., 2020; Opitz et al., 2021) or generation (Schiller et al., 2021; Alshomary et al., 2021; Heinisch et al., 2022a). Argumentation requires deep understanding of argumentative statements

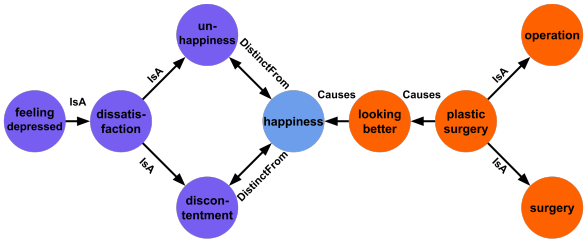


Figure 1: CCKG connecting the premise "*A person is unhappy if she is dissatisfied with her body.*" to the conclusion "*Plastic surgery raises patients' self esteem and allows them to lead normal happy lives.*" Concepts associated with premise and conclusion are colored in violet and orange, respectively. The graph makes explicit that *plastic surgery causes looking better*, which in turn *causes happiness*, which is *distinct from dissatisfaction*.

and how they relate to each other. Often, commonsense knowledge (CSK) is needed to understand how a premise connects to its conclusion, as these connections are often left implicit, as shown in Figure 1. While humans can easily infer implied knowledge, for machines extra mechanisms are needed to inject missing knowledge to better solve argumentative tasks (Moens, 2018; Becker et al., 2020; Lauscher et al., 2022; Singh et al., 2022).

Methods to inject such knowledge either rely on *parametric memory*, where CSK is stored in the parameters of large language models (LLMs), or *non-parametric memory*, where CSK is stored in external knowledge bases. In the LLM approach, latent CSK can be directly exploited in downstream tasks (Petroni et al., 2019; Li et al., 2021a) or the LLM is fine-tuned to generate the CSK in explicit form (Bosselut et al., 2019; Bansal et al., 2022). However, approaches based on parametric memory have drawbacks: they often are difficult to adapt to new domains (Liu et al., 2022a) or suffer from risk of hallucinations and unsafe generations (Levy et al., 2022) since they are not traceably grounded.

Explicit and structured CSK is available in commonsense knowledge graphs (KGs) (Vrandečić and

[1]Our code and data are available at https://github.com/Heidelberg-NLP/CCKG

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 6130–6158
July 9-14, 2023

Krötzsch, 2014; Speer et al., 2017; Hwang et al., 2021). But KGs are large and *not* contextualized, which makes selecting relevant knowledge difficult.

We can extract knowledge in the form of individual triplets (Liu et al., 2022b), but this does *not* allow for multi-hop reasoning over (potentially disconnected) triplets. Extracting *paths* consisting of multiple triplets allows multi-hop reasoning (Paul et al., 2020), but systems cannot exploit potential interactions between multiple paths. Our approach extends the idea of multi-hop path extraction by combining multiple such paths into a graph – our *Contextualized Commonsense Knowledge Graph*. The CCKGs are small and tailored to a specific argument, as shown in Figure 1, which makes them applicable in joint reasoning models (Yasunaga et al., 2022). Similar to *retrieval models* (Feldman and El-Yaniv, 2019) that extract relevant passages from text for knowledge extension, our approach extracts relevant subgraphs from structured KGs.

We can find connecting paths in large KGs by extracting *shortest paths* that link pairs of concepts. But the paths are not guaranteed to provide *relevant* knowledge for a given context, as intermediate triplets might be off-topic. To mitigate this problem, we compute *edge weights* to rate the semantic similarity of individual KG triplets to the argument at hand, and extract *weighted shortest paths* that are maximally similar to the argument. Combining the paths into a CCKG encapsulates relevant CSK. We compute the edge weights using SBERT without extra fine-tuning, and rely on graph algorithms for CCKG construction. Hence, our method is *unsupervised* and applicable in zero-shot settings.

Our main contributions are:

i.) We present an unsupervised *Contextualized Commonsense Knowledge Graph* (CCKG) construction method that enriches arguments with *relevant* CSK, by combining similarity-based contextualization with graph algorithms for subgraph extraction.

ii.) We evaluate the *quality* of CCKGs against manually created CSK graphs from an existing argumentation explainability task, where our method outperforms strong supervised baselines. Manual annotation shows that our CCKGs achieve high recall and precision for capturing implicit CSK.

iii.) We evaluate our CCKGs extrinsically in a knowledge-intense argumentative transfer task. We construct CCKGs to predict the *validity* and *novelty* of argument conclusions, using a lightweight classifier which combines graph and textual features. We achieve strong results, rivaling a SOTA GPT-3 system and outperforming other supervised systems, which – along with ablations – demonstrates the quality, effectiveness and transparency of CCKGs.

2 Background and Related Work

When humans debate a topic, they typically leverage a vast body of *background knowledge*, some already known to them and other knowledge subject to addition, e.g., by looking up a Wikipedia entry. Therefore, with the availability of large-scale KGs (Auer et al., 2007; Speer et al., 2017), and with the advent of LLMs that have been shown to learn knowledge during self-supervised training (Bosselut et al., 2019), we observe growing interest in incorporating knowledge into computational argumentation systems (Becker et al., 2020; Lauscher et al., 2022; Singh et al., 2022).

Of particular interest is the (re-)construction of implicit *commonsense knowledge* (CSK) (Moens, 2018; Lawrence and Reed, 2020; Becker et al., 2021) within or between arguments. Usually, the goal is to improve downstream-task performance of systems, e.g., improving argumentative relation classification by connecting concepts with paths found in KGs (Paul et al., 2020), or improving argument quality prediction by extracting KG distance features (Saadat-Yazdi et al., 2022). But the aim can also extend to *argumentative explanations*, propelled by an emergent need for more transparency of model predictions (Niven and Kao, 2019), which is crucial for argumentative decision making (Čyras et al., 2021). Therefore, Saha et al. (2021, 2022) manually created small CSK explanation graphs and developed fine-tuned language models to generate such graphs automatically.

Prior approaches for retrieving CSK suffer from several *issues*, e.g., Botschen et al. (2018) enrich single tokens but can't provide longer reasoning paths. By contrast, works that construct reasoning paths either do not exploit their interactions, are intransparent on which paths are used for prediction (Paul et al., 2020), employ thresholds that are hard to tailor to different tasks (Li et al., 2021b), or depend on knowledge generated from LLMs (Becker et al., 2021; Bansal et al., 2022; Saha et al., 2022), which may decrease trust in the provided knowledge due to hallucinations (Xiao and Wang, 2021; Hoover et al., 2021; Ji et al., 2022). In our work, we aim to unify the strengths of such approaches while mitigating their weaknesses: Our CCKG construc-

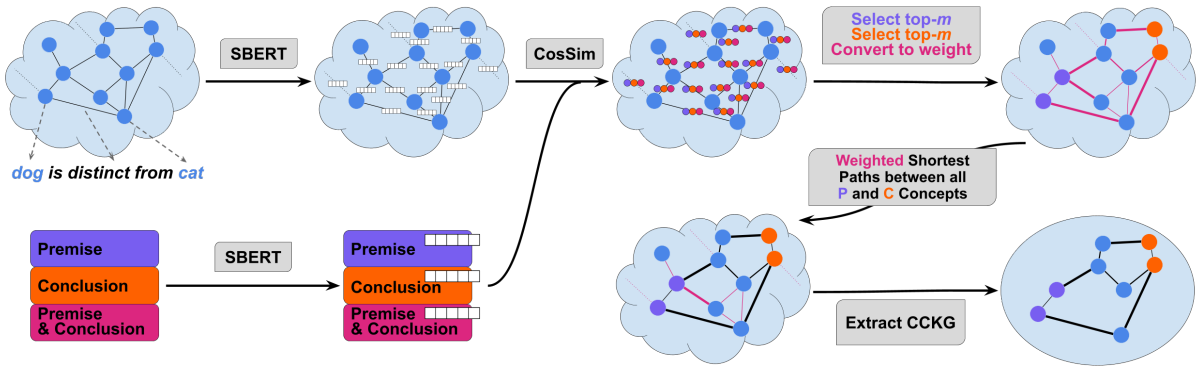


Figure 2: Overview of our method for Contextualized Commonsense Knowledge Graph (CCKG) construction.

tion method is i) context-sensitive without requiring threshold selection, and extracts CSK graphs that provide ii) accurate multi-hop reasoning structures and iii) are transparently grounded in a large KG, and hence, iv) yield strong task performance in a transfer setting.

3 CCKG Construction

Given an argument, we aim to enrich it with CSK that connects the argument's premise and conclusion via a *Contextualized Commonsense Knowledge Graph* (CCKG). Figure 2 shows an overview of our method. In a pre-processing step we compute a semantic embedding for each triplet of the KG. Using these embeddings we compute the semantic similarity of each triplet to the *premise*, to the *conclusion* and to the *argument* as a whole. By selecting the triplets with maximal similarity scores, we obtain relevant concepts associated with the premise and conclusion. Next we aim to connect these concepts with relevant reasoning paths, i.e., short paths containing triplets that match the argument. We thus convert the *argument* similarity scores to edge weights, and connect the extracted concepts with *weighted shortest paths* that maximize the semantic similarity between the path and the argument. Optionally, we further prune the CCKG.[2] Below we describe each step in detail.

Pre-processing We compute a sentence embedding for each triplet in the KG by first *verbalizing* the triplets using a designated template for each relation (see §B.1.2). We then use a *S(entence)BERT* (Reimers and Gurevych, 2019) encoder to compute an embedding for each verbalized triplet. Verbalization and triplet encoding is independent from the argument, so this step is executed only once.

[2]The pruning is not shown in Figure 2.

Similarity Computation Given an *argument* A consisting of *premise* P and *conclusion* C, we embed P, C and A = (P concatenated with C) using SBERT. For each embedding we compute its *cosine similarity* to all KG triplet embeddings. This gives us three similarity scores for each triplet: s_P, s_C and s_A. Note that all triplet similarity scores can be computed in one single matrix multiplication, which is cheap despite a usually large number of triplets in a KG.

Triplet Selection for Concept Extraction We select all concepts from the m triplets that achieve highest similarity to P and C, respectively, where m is a hyperparameter.[3] By using the semantic similarity of KG triplets to the textual argument as criterion for concept selection, we aim to extract concepts of higher *relevance* to the argument, compared to selection by string overlap. String overlap can only capture concepts that are explicitly mentioned, which can result in incomplete extractions in case only near-synonyms appear in the KG. Preliminary experiments (see §A.1) showed that computing similarity between individual concepts and the text results in concepts that are overly specific and not well connected in the KG. With limited connections, the shortest path search is restricted to very few paths, which can result in *non-* or *weakly-contextualized* paths. Thus, we extract P- and C-concepts from selected triplets, which yields more general and better connected concepts.

Similarity-weighted Shortest Paths We use Dijkstra's algorithm (Dijkstra, 1959) to find *weighted shortest paths* between all pairs of extracted concepts. The algorithm requires non-negative edge weights that represent the semantic *dissimilarity*

[3]This means that we extract up to $4m$ concepts from up to $2m$ triplets.

between a triplet and the argument. We therefore convert the argument similarity s_A of each triplet to an edge weight $w = (1-s_A)/2$. The weighted shortest paths minimize the sum of edge weights and thus, maximize semantic similarity between the argument and the path, i.e., its constituting triplets.

CCKG We combine all weighted shortest paths to yield the final CCKG. By construction it includes i) P-concepts and C-concepts representing the premise and conclusion, respectively, and ii) reasoning paths that provide CSK connections between P and C. Overall, the CCKG enriches the argument with CSK that makes the connections between premise and conclusion explicit.

Pruning By merging all weighted shortest paths, we obtain a graph with high coverage of potential reasoning paths, but it may also include noise. To eliminate noise and increase precision, we optionally prune CCKGs: We rank all concepts by their semantic similarity to the argument. Starting with the most dissimilar concept, we greedily delete concepts from the CCKG unless the concept is i) a P- or C-concept or ii) a separator, i.e., a concept that makes the CCKG disconnected if removed. These constraints ensure that the pruned CCKG still covers both premise and conclusion, and preserves their connection. Figure 1 shows a pruned CCKG.

4 Experiments

We evaluate our CCKG construction method *intrinsically* (§4.1) in an argumentative commonsense graph generation task and *extrinsically* (§4.2) in a knowledge-intense conclusion classification task.

Experimental setup We instantiate our KG using the English part of ConceptNet (CN) Version 5.7 (Speer et al., 2017), with or without the RelatedTo relation (see §B.2.1 for details). CN is directed, but for the shortest path search we consider all edges to be undirected. For SBERT similarity computation we verbalize the CN relations using templates shown in §B.1.2. We use the Huggingface implementation[4] of SBERT. For shortest path search we use Dijkstra's algorithm implemented in iGraph (Dijkstra, 1959; Csárdi and Nepusz, 2006).

Baselines Besides task-specific baselines we compare to two versions of our method that ablate the edge weights, i.e., the shortest path search is unweighted and hence *not* contextualized. We either i) randomly select one shortest path between each pair of concepts (**w/o $\mathbf{EW_O}$**), or ii) consider all shortest paths between two concepts (**w/o $\mathbf{EW_A}$**). The latter can result in large graphs[5] which increases computational costs in downstream tasks.

4.1 Intrinsic evaluation on ExplaGraphs

Our aim is to directly assess if the constructed CCKGs capture implicit CSK in arguments. This assessment is challenging, as gold data on implicit connections in argumentation is scarce and, as in any generation task, there is not only one single correct answer. To the best of our knowledge, only Saha et al. (2021) provide relevant data. They introduce *ExplaGraphs*, a *generative structured commonsense reasoning task* with a corresponding dataset. Given a *belief* and a *support* or *counter argument*, the task is to generate a *commonsense explanation graph* that is expected to explain the argument's *stance*.

We adapt their data to our setting of *premise-conclusion* pairs by considering the *argument* as a *premise* and the *belief* as *conclusion*, which yields plausible premise-conclusion pairs for *supports* (see §B.3.2). For example, the premise and conclusion in Figure 1 have been constructed this way. Similarly, we can turn their *counters* into premise-conclusion pairs. In this case, the belief does not form a plausible conclusion, but we can make their implicit knowledge connections explicit via the CCKG anyway.

Saha et al. (2021)'s gold graphs are manually constructed. Edge relations were chosen from the set of CN relations, with additional *negated* relations, such as *NotCapableOf*. Several constraints were enforced on the graphs to ensure better quality during data collection and to simplify evaluation. The graphs are connected directed acyclic graphs consisting of 3-8 triplets. Concepts have a maximum of three words and at least two concepts must stem from the premise and conclusion, respectively.

Our method does not necessarily fulfill these constraints by construction, and also cannot learn them, as it is unsupervised. Also, the imposed constraints are not necessarily beneficial for downstream tasks. We discuss appropriate metrics to compare our CCKGs against ExplaGraphs in §4.1.1.

Saha et al. (2021)'s data consists of 2368/ 398/

[4] sentence-transformers/all-mpnet-base-v2

[5] e.g., there are ∼100 shortest paths linking person to work in ConceptNet.

	Configuration	#nodes	#edges	C P ↑	C R ↑	C F1 ↑	T P ↑	T R ↑	T F1 ↑	GED ↓	G-BS ↑
CCKG	$m = 1$	4.0	3.0	**52.54**	37.94	**42.67**	**28.55**	19.78	**22.13**	**0.3435**	66.88
	$m = 2$	6.6	5.8	36.67	44.36	38.88	19.42	25.44	20.97	0.3745	**74.26**
	$m = 3$	9.2	8.5	29.25	**48.55**	35.49	15.51	**29.63**	19.56	0.4313	64.50
Supervised	RE-SP	5.9	4.9	42.19	46.17	42.94	1.17	1.34	1.20	0.3706	74.63
	T5	4.5	3.3	51.87	44.68	47.25	4.10	3.59	3.77	0.3320	76.26
	max-marg.	4.7	3.5	50.47	44.48	46.52	4.02	3.68	3.79	0.3315	**77.96**
	contrastive	4.6	3.4	**53.70**	**46.93**	**49.26**	**5.18**	**4.75**	**4.88**	**0.3314**	77.04

Table 1: Intrinsic evaluation of pruned CCKGs from ExplaKnow on the ExplaGraphs dev split. *C P*, *C R* and *C F1* are precision, recall and F1 scores of concepts. *T P*, *T R* and *T F1* are the same for triplets. *GED* is normalized Graph-Edit-Distance; *G-BS* is Graph-BERT-Score (see §B.2.2). All values are macro-averages across all 398 graphs.

400 text-graph pairs in the train, dev and test set. Since the test set is not public, we report results on the dev set. We do *not* need any data for hyperparameter tuning, as our method is unsupervised.

4.1.1 Evaluation against gold graphs

ExplaKnow Automatically assessing the semantic similarity of two graphs is challenging. Concepts in CN and ExplaGraphs are both in free-form, hence we find only few matching concepts in the two resources. To circumvent this problem for our intrinsic evaluation, we replace CN as our external KG with an artificially created *ExplaKnow* KG, which we form by combining all gold graphs from samples in the train and dev set into a single graph. The resulting KG has $\sim 1\%$ of CN's size, but with comparable density. Despite its smaller size, retrieving information from *ExplaKnow* is non-trivial as it includes many topics, with different perspectives and stances for each of them. We hence use *ExplaKnow* as a proxy to perform intrinsic quality estimation of our graph construction method against Saha et al. (2021)'s gold graphs. §B.2.1 shows detailed statistics of *ExplaKnow* vs. *CN*.

Metrics We aim to assess how *semantically* and *structurally* similar our CCKGs are to the gold graphs, using a selection of metrics that focus on different aspects of similarity. We measure *precision*, *recall* and *F1*-scores for *concepts* and *triplets*. *Concept F1-score* indicates the ratio of correct concepts in the constructed CCKGs, as a measure of topical overlap. By contrast, the triplets encode concrete CSK statements, and hence *triplet F1-score* provides a more rigid measure of the semantic overlap of a pair of graphs. Hence, we consider triplet F1-score as our main metric and report concept scores as additional information. We further include two graph structure metrics from Saha et al. (2021): normalized graph edit distance (*GED*) and G-BERTScore (*G-BS*). Please refer to §B.2.2 for further details on evaluation metrics applied in Saha et al. (2021).

Baselines We compare against supervised methods by Saha et al. (2021) (*RE-SP*) and Saha et al. (2022) (*T5*, *max-margin*, *contrastive*). Their models are all trained on gold graphs in a supervised manner. **RE-SP** predicts concepts with fine-tuned RoBERTa and BART models and edge probabilities between concepts are predicted with another fine-tuned RoBERTa model. The system finally combines the concepts and probability-weighted edges to a graph using integer linear programming. The other baselines predict a stance with a fine-tuned RoBERTa model, then a fine-tuned T5-large model predicts the graph in a linearized form conditioned on the belief, argument and predicted stance. **T5** is fine-tuned on the training data with a standard cross-entropy loss. **Max-margin** and **contrastive** extend *T5* to additionally learn from negative samples via a *max-margin loss*, and from positive and negative samples via a *contrastive loss*, respectively.

Automatic evaluation of CCKG on ExplaKnow

Table 1 shows results for pruned CCKGs. The supervised methods outperform CCKG by a small margin in *concept metrics*. By contrast, CCKG outperforms all supervised methods by $400\,\%$ and more in *triplet metrics*. This indicates that the supervised models tend to generate correct concepts, but struggle to connect them in meaningful relations. By contrast, our approach, being grounded in a KG, attracts contextually similar triplets.

The *GED* and *G-BS* metrics show better results for the supervised methods, differing by 1.2 pp. and 3.7 pp. for the best supervised systems, respectively.

However, our method matches or outperforms the RE-SP model that respects structural constraints by construction. Note that both metrics put high emphasis on the graph structure, which the supervised models are optimized to match. Our unsupervised method, by construction, does not necessarily fulfill the structural constraints that are imposed on the gold graphs, and cannot learn them. Hence, it is expected that the supervised models fit the structural constraints reflected in the train data much better. We thus consider the competitive performance of our unsupervised method as a strong result, which is confirmed by the very high triplet scores.

Increasing m ($\sim$ number of extracted concepts) increases the size of the CCKGs, which increases recall but lowers precision. The F1-scores are best for $m = 1$. For downstream tasks, m should be chosen according to the task, and depending on whether higher recall or higher precision is desired.

§B.2.3 reports further experiments which show that i) CCKGs outperform *uncontextualized baselines*, also when CCKGs are constructed from *ConceptNet*; ii) they achieve similar performance for support and counter instances; iii) verbalization of triplets has a small impact, but more *natural verbalizations* achieve better performance; iv) using more than one weighted shortest path increases recall but decreases precision; v) pruning driven by structural features achieves comparable quality to pruning by semantic similarity. In §4.2 we introduce a variation of the CCKG construction which extracts concepts from constituents of the argument. We also test this method on ExplaGraphs in §B.2.3.

4.1.2 Manual evaluation of CN subgraphs

Saha et al. (2021)'s graphs with *ExplaKnow* as underlying knowledge resource offer a concise evaluation target for our CCKG construction method. But ExplaKnow is small and its concepts have been tailored to the data during instance-level data creation. To obtain a quality estimate for CCKG in a more realistic setup, we additionally conduct a manual evaluation of CCKGs on the same data, but extracted from the large *ConceptNet* (CN) resource.

CCKGs from CN We construct CCKGs from CN, but exclude its unspecific RelatedTo edges. We set $m = 3$ since CN concepts are less specific compared to ExplaKnow, hence we expect that larger graphs are required to cover the targeted content. To counter-balance the larger graph size we apply pruning. In this setup, we cannot use Saha et al. (2021)'s gold graphs as evaluation targets and therefore perform manual quality assessment.

Annotation Two independent expert annotators[6] manually labeled all 199 *support instances* in the ExplaGraphs dev set. First, they assess if arguments are *plausible* and include an *implicit CSK connection* that links the conclusion to the premise. On the 115 instances that fulfilled both criteria unanimously, we **evaluate the quality of CCKGs**. To estimate **recall**, we inquire whether the CCKG expresses the implicit CSK that links the premise and the conclusion *completely*, *partially* or *not at all*. Such implicit CSK can be expressed, for example, by a chain of triplets as shown in Figure 1. To estimate fine-grained **precision**, the annotators had to label individual triplets as either *positive* (expresses implicit CSK), *neutral* (does not express implicit CSK, but matches the topic), *unrelated* (does not match the topic) or *negative* (contradicts implicit CSK or the conclusion)[7]. This allows us to assess the precision of triplets showing *implicit CSK* (positive triplets) and the precision of triplets being *in-topic* (positive, neutral or negative). See §B.3.1 for detailed annotation guidelines.

Results §B.3.2 Table 14 and 15 show detailed analysis of the annotation results. We report the main findings here. 29.57 % of CCKGs were unanimously judged to show the *implicit CSK connection completely*, i.e., the CCKG explains the argument fully. This result almost doubles to 59.13 % when considering graphs that at least one annotator labeled as complete. 88.70 % show the implicit CSK *partially*. Thus, CCKGs have **high recall** of implicit CSK and hence can help making implicit connections explicit. At the level of individual triplets, our annotation reveals that CCKGs have a **high macro triplet precision**, i.e., averaged over individual graphs, of 39.43 % and 73.87 % for *showing implicit CSK* when considering unanimously labeled triplets, and triplets labeled as positive by at least one annotator, respectively. Equivalent macro precision scores for *in-topic triplets* are 92.76 % and 99.20 %. This shows that a substantial amount of triplets reflects implicit CSK, and that almost all triplets are from the correct topic. Triplets from wrong topics are avoided due to strong contextualization in CCKG construction and pruning.

[6] Students with advanced/native competence of English.

[7] E.g., (human_cloning, IsA, considered_unethical) is an example of a negative triplet in a CCKG for an argument that *supports* cloning.

We also gained qualitative insights. **Missing knowledge**: We find cases of arguments on a topic that lacks coverage in CN, resulting in similar CCKGs for different arguments.[8] **Ambiguity**: CN concepts are not disambiguated. A path may thus run through concepts that take different senses, making the path meaningless.[9]

4.2 Extrinsic evaluation: Predicting Validity and Novelty of Arguments (VALNOV)

We now investigate the *effectiveness* of CCKGs – used to explicate implicit CSK in arguments – in the novel, knowledge-intense argumentation task VALNOV. We evaluate the *robustness* of our unsupervised method relying on non-parametric knowledge, compared to supervised graph generation systems applied out-of-domain, as well as SOTA VALNOV systems.

Task description Heinisch et al. (2022b) introduced a novel argument inference task VALNOV as a community shared task. Given a textual premise and conclusion, the task is to predict whether the conclusion is i) *valid* and ii) *novel* with respect to its premise. A conclusion is *valid* if it is *justified* by the premise. It is *novel* if it contains premise-related content that is not part of the premise, i.e. the conclusion *adds novel content* to the premise. Please refer to §B.4.1 for data statistics.

Systems are expected to report macro F1-scores for joint and individual prediction of validity and novelty. In joint modeling we distinguish 4 classes: i) *valid & novel*, ii) *non-valid & novel*, iii) *valid & non-novel*, iv) *non-valid & non-novel*. The training data is unbalanced with respect to these 4 classes.

Predicting Validity and Novelty from CCKGs We hypothesize that CCKGs show structural characteristics that correlate with validity and novelty: For instance, a *valid* conclusion should be well connected to its premise in the constructed CCKG, and a *novel* conclusion should result in a CCKG with long paths from the premise to its conclusion. To test these hypotheses we extract graph features from the CCKGs and combine them with textual features from the argument. We feed all features to shallow classifiers to predict the validity and novelty of conclusions. Note that interaction between the CCKG and the argument is limited in this approach, which allows us to isolate and investigate the expressiveness of our CCKGs.

CCKG details The VALNOV dataset contains arguments that are relatively long (76 tokens in avg.), often comprising more that one aspect/ perspective. This negatively effects the quality of triplet selection for concept extraction: the extracted concepts are semantically relevant, but often don't span the entire argument. Thus, we parse the text into constituents and select concepts from the top-m triplets for each constituent individually.

Pruning CCKGs completely bears the danger of removing relevant structural aspects of CCKGs. We therefore experiment with *partial pruning*, that only removes the most dissimilar prunable concepts. This enables a more fine-grained balance of recall and precision compared to complete pruning.

We obtain best performance using parsing, partial pruning (75 %), $m = 2$ and CN w/o RelatedTo. Please refer to §B.4.2 for further details on concept extraction with parsing and partial pruning.

Feature extraction: We extract 15 graph features from each CCKG: 5 characterizing its *size*, 6 its *connectivity* and 4 the *distance between premise and conclusion in the CCKG*. As textual features we use the *semantic similarity* of premise and conclusion, and predictions from a *NLI*-model. We obtain 19 features in total. See §B.4.3 for detailed description of the features.

Classifier We train Random Forests and SVMs in a multi-class setting, considering validity and novelty jointly. Following Saadat-Yazdi et al. (2022) we use upsampling to balance the training data. Results are averaged over 5 different runs. Please refer to §B.4.4 for hyperparameters and implementation details of the classifiers.

Baselines We compare to supervised ExplaGraphs generation systems by embedding their graphs into our classifier, and to systems participating in the VALNOV shared task: the two best-performing submissions, the System-Average (average of all submissions) and the ST baseline.

We evaluate against **supervised graph construction methods** (Saha et al., 2022) (see §4.1.1), to assess their performance in an out-of-domain setting, compared to our unsupervised CCKG construction method. We apply their trained graph generation

[8] 18 out of 22 instances on *entrapment* yield identical CCKGs, due to lack of coverage in CN.

[9] For example, the following chain of triplets (river_bank, IsA, bank, UsedFor, keeping_money_safe), is a path that connects the concepts river_bank and keeping_money_safe, and is established by the intermediary concept bank that takes a different meaning in the two constituting triplets.

models to VALNOV arguments and use the generated graphs exactly as we do for our CCKGs: we extract the same features to train the shallow classifier models, following our training protocol. Unlike our general-purpose CCKGs, these methods were trained to generate graphs for stance-classification tasks. Nevertheless, we can apply these methods to VALNOV as further baselines.

The shared task winning system **ST-1st** (van der Meer et al., 2022) prompted GPT-3 for *validity* and separately fine-tuned a RobERTa-based NLI model, further enhanced with contrastive learning, for *novelty*. The second-best shared task system **ST-2nd** (Saadat-Yazdi et al., 2022) is a FFNN trained with upsampling that combines diverse features from NLI predictions, semantic similarity, predictions of validity and novelty and structural knowledge extracted from WikiData. The shared task baseline **BL** consists of two RoBERTa models, fine-tuned for validity and novelty separately.

Our system resembles the *ST-2nd* approach, however, their system strongly emphasizes textual features, even leveraging a fine-tuned BERT predicting validity and novelty based on text alone, and considers only two structural features from uncontextualized WikiData paths. Our model, by contrast, relies on a minimal amount of textual features, leveraging standard pre-trained models without task-dependent fine-tuning. Hence, it strongly relies on graph features, building on the strong contextualization of CCKGs to the textual argument.

Results Table 2 shows the results on the VALNOV test set. Our system CCKG achieves the second best result in all metrics: *validity*, *novelty* and *joint* prediction. Best scores are achieved either by ST-1st with GPT-3 on *joint* and *validity* prediction or by Saha et al. (2022)'s T5 model for *novelty*. Yet our approach outperforms these systems in the respective complementary metrics: *novelty* for ST-1st and *validity* for T5. CCKG clearly outperforms T5 in joint F1 by 6.2 pp.

Heinisch et al. (2022b)'s analysis of the VALNOV results concludes that i) LLMs are powerful predictors of *validity*, due to the textual inference capabilities they acquire in pretraining on massive text sources. At the same time, ii) LLMs were shown to lag behind knowledge-based systems in *novelty* prediction. *Validity* was overall easier to solve than *novelty*, and systems that performed well for *novelty* had poor *validity* results,

	Systems and BLs	joint F1	Val F1	Nov F1
ST	1st (GPT-3)	**45.16**	**74.64**	61.75
	2nd (w/ KG)	43.27	69.80	62.43
	System Avg	35.94	62.74	52.97
	Baseline	23.90	59.96	36.12
EG w/ Ours	T5	37.71	67.07	**63.53**
	max-margin	36.22	67.61	63.27
	contrastive	37.82	64.77	59.96
	CCKG (Ours)	**43.91**	**70.69**	**63.30**
Ablation	w/o Graph feats.	-11.65	-3.40	-5.12
	w/o Text feats.	-20.65	-20.74	-17.69
	w/o EW_O	-6.51	-3.80	-1.69
	w/o EW_A	-3.25	-4.45	1.76
	string matching	-6.71	-3.23	0.55
	w/o connectivity feats.	-5.60	-4.01	-0.60
	w/o PC-distance feats.	-2.27	-0.27	-3.73

Table 2: Results on VALNOV: *joint*, *validity* and *novelty* F1-scores. We compare against Shared Task (*ST*) results and ExplaGraphs generation models, integrated in our VALNOV classifier (*EG w/ Ours*). Ablated scores are relative to *CCKG (Ours)*.

and vice versa.[10] It is therefore no surprise that our system cannot compete with GPT-3 for *validity*. However, it achieves 2nd best performance on *validity* at a high level of 70.69 % F1 without sacrificing *novelty*. Leveraging structural knowledge, T5 achieves highest scores for novelty, but performs poorly in validity, and hence, only ranks 5th in the joint ranking. CCKGs perform well in both, validity and novelty, with one unified approach, unlike ST-1st. Our strong joint score of 43.72 % only gets surpassed by ST-1st, which leverages two independent systems for validity and novelty. Thus, simple feature extraction from CCKGs achieves interpretable and yet compatible scores. Our ablation will show that this is possible due to strong contextualization in the graph construction.

Ablation Removing graph or text features from CCKG (ours) reduces performance by 11.65 pp. and 20.65 pp., respectively. The text is more important for *validity*, while the graph has a larger impact on *novelty*. Yet, both metrics benefit from both modalities. This indicates that text and CCKG contain complementary information and should be considered jointly in future work.

Ablating all edge weights incurs considerable performance losses for *validity* and *joint* F1. *Novelty* is less affected, which shows that *contextualization* is more relevant for validity. We can also em-

[10] For example, prompting GPT-3 for novelty resulted in only 46.07 % F1 score.

poverish contextualization by extracting concepts via string matching. This decreases performance by 6.71 pp., again with a larger decrease for validity.

Feature ablation confirms that connectivity features are most relevant for validity, while premise-conclusion distance in the CCKG is most relevant for novelty. Further ablations are shown in §B.4.5.

5 Conclusion

In this work we proposed an unsupervised method to construct *Contextualized Commonsense Knowledge Graphs* (CCKGs). Our extensive evaluations show that CCKGs are of high quality, outperform context-insensitive baselines and rival strong supervised graph construction methods on diverse argumentation tasks, while offering increased robustness. Being grounded in a KG, the information captured in our CCKGs is traceable and hence interpretable. Future work could explore incorporation of more specific KGs to address particular domains. Using our compact, high-quality CCKGs in stronger interaction with LLMs is another step to address in future work.

Limitations

In principle our method is applicable in many domains, for example, one could use a biomedical knowledge graph instead of ConceptNet in a relevant domain. However, in this paper we only evaluate the quality of our approach in argumentative tasks which require commonsense knowledge. Our approach is unsupervised, but its performance depends on the quality of the used knowledge graph and SBERT model.

Similarly, we only evaluate CCKGs for English data, although our approach is not limited to English if one uses multilingual SBERT models (Reimers and Gurevych, 2020) or a multilingual knowledge graph.

Finally, our approach is purely extractive and hence, is limited by the coverage and quality of knowledge graphs. However, improving knowledge graphs is an active field of research and hence, high-quality and high-coverage knowledge graphs are to be expected. Furthermore, our extracted CCKGs could be augmented with generative models if coverage in the knowledge graph is not sufficient. However, that would reduce the interpretability that our approach provides.

Ethical Considerations

Our method extracts subgraphs from knowledge graphs. Hence, any potential biases present in the knowledge graph can propagate to our CCKGs. While this can be problematic, our approach allows to trace biases back to their origin. This is comparable to manual information extraction, as all knowledge sources can contain biases – for example political tendencies in newspapers. Strategies to automatically avoid biases (Mehrabi et al., 2021) could also be incorporated in future work. However, as our approach is a pure extraction, it can not generated new potentially harmful information. Thus, CCKGs are perhaps more reliable for sensitive application than knowledge representations generated without grounding.

Acknowledgements

We want to thank Swarnadeep Saha for generating the supervised graphs (T5, max-margin and contrastive) which we compare to in §4.2. We also thank our annotators for their support.

This work was funded by DFG, the German Research Foundation, within the project ACCEPT, as part of the priority program "Robust Argumentation Machines" (RATIO, SPP-1999).

References

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

Milad Alshomary, Wei-Fan Chen, Timon Gurcke, and Henning Wachsmuth. 2021. Belief-based generation of argumentative claims. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 224–233, Online. Association for Computational Linguistics.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The Semantic Web*, pages 722–735, Berlin, Heidelberg. Springer Berlin Heidelberg.

Rachit Bansal, Milan Aggarwal, Sumit Bhatia, Jivat Kaur, and Balaji Krishnamurthy. 2022. CoSe-co: Text conditioned generative CommonSense contextualizer. In *Proceedings of the 2022 Conference of*

the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1128–1143, Seattle, United States. Association for Computational Linguistics.

Maria Becker, Ioana Hulpus, Debjit Paul, Juri Opitz, Jonathan Kobbe, Heiner Stuckenschmidt, and Anette Frank. 2020. Explaining Arguments with Background Knowledge – Towards Knowledge-based Argumentation Analysis . *Datenbank Spektrum (Special Issue: Argumentation Intelligence)*, 20:131–141.

Maria Becker, Siting Liang, and Anette Frank. 2021. Reconstructing implicit knowledge with language models. In *Proceedings of Deep Learning Inside Out (DeeLIO): The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 11–24, Online. Association for Computational Linguistics.

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.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

Teresa Botschen, Daniil Sorokin, and Iryna Gurevych. 2018. Frame- and entity-based knowledge for common-sense argumentative reasoning. In *Proceedings of the 5th Workshop on Argument Mining*, pages 90–96, Brussels, Belgium. Association for Computational Linguistics.

Gábor Csárdi and Tamás Nepusz. 2006. The igraph software package for complex network research. In *InterJournal Complex Systems*.

Edsger W. Dijkstra. 1959. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271.

Yair Feldman and Ran El-Yaniv. 2019. Multi-hop paragraph retrieval for open-domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2296–2309, Florence, Italy. Association for Computational Linguistics.

Vanessa Wei Feng and Graeme Hirst. 2011. Classifying arguments by scheme. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 987–996, Portland, Oregon, USA. Association for Computational Linguistics.

Philipp Heinisch, Anette Frank, Juri Opitz, and Philipp Cimiano. 2022a. Strategies for framing argumentative conclusion generation. In *Findings of the Association for Computational Linguistics: ACL-INLG 2022*. Association for Computational Linguistics.

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

Philipp Heinisch, Moritz Plenz, Juri Opitz, Anette Frank, and Philipp Cimiano. 2022c. Data augmentation for improving the prediction of validity and novelty of argumentative conclusions. In *Proceedings of the 9th Workshop on Argument Mining (ArgMining)*, Online and in Gyeongju, Republic of Korea. International Conference on Computational Linguistics.

Jacob Louis Hoover, Wenyu Du, Alessandro Sordoni, and Timothy J. O'Donnell. 2021. Linguistic dependencies and statistical dependence. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2941–2963, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jena D. Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. 2021. Comet-atomic 2020: On symbolic and neural commonsense knowledge graphs. In *AAAI*.

Mathilde Janier, John Lawrence, and Chris Reed. 2014. Ova+: an argument analysis interface. In *Computational Models of Argument*, Frontiers in artificial intelligence and applications, pages 463–464, Netherlands. IOS Press. Fifth International Conference on Computational Models of Argument, COMMA 2014 ; Conference date: 09-09-2014 Through 12-09-2014.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2022. Survey of hallucination in natural language generation. *ACM Comput. Surv.* Just Accepted.

Yohan Jo, Seojin Bang, Emaad Manzoor, Eduard Hovy, and Chris Reed. 2020. Detecting attackable sentences in arguments. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1–23, Online. Association for Computational Linguistics.

Anne Lauscher, Henning Wachsmuth, Iryna Gurevych, and Goran Glavaš. 2022. Scientia Potentia Est—On the Role of Knowledge in Computational Argumentation. *Transactions of the Association for Computational Linguistics*, 10:1392–1422.

John Lawrence and Chris Reed. 2020. Argument Mining: A Survey. *Computational Linguistics*, 45(4):765–818.

Sharon Levy, Emily Allaway, Melanie Subbiah, Lydia Chilton, Desmond Patton, Kathleen McKeown, and William Yang Wang. 2022. Safetext: A benchmark for exploring physical safety in language models. In *EMNLP*.

Belinda Z. Li, Maxwell Nye, and Jacob Andreas. 2021a. Implicit representations of meaning in neural language models. 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 1813–1827, Online. Association for Computational Linguistics.

Weichen Li, Patrick Abels, Zahra Ahmadi, Sophie Burkhardt, Benjamin Schiller, Iryna Gurevych, and Stefan Kramer. 2021b. Topic-guided knowledge graph construction for argument mining. In *2021 IEEE International Conference on Big Knowledge (ICBK)*, pages 315–322.

Linqing Liu, Patrick Lewis, Sebastian Riedel, and Pontus Stenetorp. 2022a. Challenges in generalization in open domain question answering. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2014–2029, Seattle, United States. Association for Computational Linguistics.

Qi Liu, Dani Yogatama, and Phil Blunsom. 2022b. Relational memory-augmented language models. *Transactions of the Association for Computational Linguistics*, 10:555–572.

Ninareh Mehrabi, Pei Zhou, Fred Morstatter, Jay Pujara, Xiang Ren, and Aram Galstyan. 2021. Lawyers are dishonest? quantifying representational harms in commonsense knowledge resources. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5033, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Marie-Francine Moens. 2018. Argumentation mining: How can a machine acquire common sense and world knowledge? *Argument & Computation*, 9(1):1–14.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664, Florence, Italy. Association for Computational Linguistics.

Juri Opitz, Philipp Heinisch, Philipp Wiesenbach, Philipp Cimiano, and Anette Frank. 2021. Explainable unsupervised argument similarity rating with Abstract Meaning Representation and conclusion generation. In *Proceedings of the 8th Workshop on Argument Mining*, pages 24–35, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Debjit Paul, Juri Opitz, Maria Becker, Jonathan Kobbe, Graeme Hirst, and Anette Frank. 2020. Argumentative Relation Classification with Background Knowledge. In *Proceedings of the 8th International Conference on Computational Models of Argument (COMMA 2020)*, volume 326 of *Frontiers in Artificial Intelligence and Applications*, pages 319–330. Computational Models of Argument.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

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.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

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

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

Ameer Saadat-Yazdi, Xue Li, Sandrine Chausson, Vaishak Belle, Björn Ross, Jeff Z. Pan, and Nadin Kökciyan. 2022. Kevin: A knowledge enhanced validity and novelty classifier for arguments. In *Proceedings of the 9th Workshop on Argument Mining*, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Swarnadeep Saha, Prateek Yadav, and Mohit Bansal. 2022. Explanation graph generation via pre-trained language models: An empirical study with contrastive learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1190–1208, Dublin, Ireland. Association for Computational Linguistics.

Swarnadeep Saha, Prateek Yadav, Lisa Bauer, and Mohit Bansal. 2021. ExplaGraphs: An explanation graph

generation task for structured commonsense reasoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7716–7740, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Keshav Singh, Naoya Inoue, Farjana Sultana Mim, Shoichi Naito, and Kentaro Inui. 2022. IRAC: A domain-specific annotated corpus of implicit reasoning in arguments. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4674–4683, Marseille, France. European Language Resources Association.

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

Michael van der Meer, Myrthe Reuver, Urja Khurana, Lea Krause, and Selene Báez Santamaría. 2022. Will it blend? mixing training paradigms & prompting for argument quality prediction. In *Proceedings of the 9th Workshop on Argument Mining*, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Henning Wachsmuth, Nona Naderi, Ivan Habernal, Yufang Hou, Graeme Hirst, Iryna Gurevych, and Benno Stein. 2017a. Argumentation quality assessment: Theory vs. practice. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 250–255, Vancouver, Canada. Association for Computational Linguistics.

Henning Wachsmuth, Martin Potthast, Khalid Al-Khatib, Yamen Ajjour, Jana Puschmann, Jiani Qu, Jonas Dorsch, Viorel Morari, Janek Bevendorff, and Benno Stein. 2017b. Building an argument search engine for the web. In *Proceedings of the 4th Workshop on Argument Mining*, pages 49–59, Copenhagen, Denmark. Association for Computational Linguistics.

Yijun Xiao and William Yang Wang. 2021. On hallucination and predictive uncertainty in conditional language generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2734–2744, Online. Association for Computational Linguistics.

Michihiro Yasunaga, Antoine Bosselut, Hongyu Ren, Xikun Zhang, Christopher D. Manning, Percy Liang, and Jure Leskovec. 2022. Deep bidirectional language-knowledge graph pretraining. In *Neural Information Processing Systems (NeurIPS)*.

Biao Zhang, Ivan Titov, and Rico Sennrich. 2020a. Fast interleaved bidirectional sequence generation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 503–515, Online. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

Kristijonas Čyras, Antonio Rago, Emanuele Albini, Pietro Baroni, and Francesca Toni. 2021. Argumentative xai: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4392–4399. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

A Method

A.1 Preliminary experiments on concept extraction

As a preliminary experiment, we test how to extract concepts that are well-connected in the KG. Concepts which are not well-connected have limited options to be connected to each other, which hinders contextualization in the shortest path search. Hence, we require well-connected concepts which are not overly specific. We estimate the connectivity and specificity of concepts by their degree and number of words, respectively.

We experiment with i) extracting concepts that are most similar to the text, and ii) extracting all concepts from the triplets that are most similar to the text. In each case we measure similarity between the concept / triplet and the text with the same SBERT model. As KG we use ConceptNet (CN) without RelatedTo triplets (please refer to §B.1.1 for further context on the choice of the KG).

Table 3 shows the macro averages over the development split of ExplaGraphs (Saha et al., 2021) for $m = 1$. Varying m only has a small impact on the results. Extracting concepts via ranking triplets results in shorter concepts with high degrees, i.e.

metric	concept	triplet
number of words	2.42	1.83
degree	4.21	103.39

Table 3: Comparison of direct concept extraction and concept extraction via triplet ranking. Values are averages over extracted concept from dev set of ExplaGraphs for $m = 1$.

general and well-connected concepts. Thus, we extract concepts by first ranking triplets and then selecting all concepts in the top-m triplets.

B Experiments

B.1 Experimental setup

B.1.1 Discussion of RelatedTo in CN

More than half of all triplets in CN have the relation RelatedTo (see Table 6). This is a very general relation and thus might cause a high degree of semantically vacuous connections. Hence, paths constructed from CN without RelatedTo are potentially longer, but more explicit and therefore also more expressive. On the other hand, RelatedTo might be necessary to make certain connections in CN. Thus we experiment with two different versions of CN: one with RelatedTo and one without RelatedTo.

To create a graph from CN excluding the RelatedTo relation, we first remove all triplets with this relation and then all concepts with degree 0. Table 6 shows statistics of CN with and without RelatedTo.

B.1.2 Triplet verbalization

SBERT was pre-trained on natural language sentences, and thus is not ideal for capturing semantics of triplets. We could fine-tune SBERT to learn triplet-representations, but that might reduce the generalizability of our model. Therefore we prefer to convert the triplets to natural language, which can be processed by SBERT without any fine-tuning.

To translate triplets to natural language we designed *natural* templates that preserve the relation's meaning, but are more natural. To analyze the impact of the verbalization templates we also created *static* templates, which are closer to the original relations. Our templates are shown in Table 4 and Table 5 for CN and ExplaKnow.

Note that these templates can propagate grammatical errors from the triplets, e.g. (humans, Desires, freedom) would get verbalized to *humans desires freedom* instead of the grammatically correct *humans desire freedom*. In principle, automatically correcting these errors could be included in the pre-processing step of our method, but for simplicity we refrained from doing so.

Relation	Natural	Static
RelatedTo	is related to	is related to
IsA	is a	is a
FormOf	is a form of	is a form of
CapableOf	is capable of	is capable of
MotivatedByGoal	is motivated by	is motivated by the goal
HasContext	has context	has the context
HasPrerequisite	has prerequisite	has the prerequisite
Synonym	is a synonym of	is a synonym of
Antonym	is an antonym of	is an antonym of
AtLocation	is in	is at the location
Desires	desires	desires
UsedFor	is used for	is used for
HasSubevent	has subevent	has the subevent
HasProperty	is	has the property
PartOf	is a part of	is a part of
DefinedAs	is defined as	is defined as
HasA	has	has a
MannerOf	is a manner of	is a manner of
Causes	causes	causes
HasFirstSubevent	starts with	has the first subevent
HasLastSubevent	ends with	has the last subevent
ReceivesAction	$\star$ can be done to	receives the action
InstanceOf	is an instance of	is an instance of
NotCapableOf	is not capable of	is not capable of
CausesDesire	causes desire	causes the desire
DistinctFrom	is distinct from	is distinct from
NotDesires	does not desire	does not desire
MadeOf	is made of	is made of
Entails	entails	entails
CreatedBy	is created by	is created by
NotHasProperty	is not	does not have the property
LocatedNear	is near	is located near
SymbolOf	is a symbol of	is a symbol of

Table 4: Verbalization templates for all relations in CN. $\star$: the order of concepts is inverted in the verbalization, i.e. (A, ReceivesAction, B) is verbalized as *B can be done to A*.

B.2 ExplaGraphs automatic evaluation

B.2.1 Knowledge Graph statistics

For the statistics in Table 6 we consider the KGs as multi-graphs, i.e. two triplets which differ only by their relation are considered as two separate edges. The table shows statistics for ConceptNet with and without the RelatedTo Relation (see §B.1.1) and for ExplaKnow, the artificial KG constructed from ExplaGraphs. The average number of words is the average across all concepts in the graph.

The table shows that ExplaKnow is smaller than CN, but has a comparable average degree. However, concepts in ExplaKnow have more words than

Relation	Natural	Static
IsA	is a	is a
IsNotA	is not a	is not a
CapableOf	is capable of	is capable of
NotCapableOf	is not capable of	is not capable of
HasContext	has context	has context
NotHasContext	does not have context	does not have context
SynonymOf	is a synonym of	is a synonym of
AntonymOf	is an antonym of	is an antonym of
AtLocation	is in	is at the location
NotAtLocation	is not in	is not at the location
Desires	desires	desires
NotDesires	does not desire	does not desire
UsedFor	is used for	is used for
NotUsedFor	is not used for	is not used for
HasSubevent	has subevent	has subevent
NotHasSubevent	does not have subevent	does not have subevent
HasProperty	is	has the property
NotHasProperty	is not	does not have the property
PartOf	is a part of	is a part of
NotPartOf	is not a part of	is not a part of
Causes	causes	causes
NotCauses	does not cause	does not cause
ReceivesAction	⋆can be done to	receives the action
NotReceivesAction	⋆can not be done to	does not receive the action
MadeOf	is made of	is made of
NotMadeOf	is not made of	is not made of
CreatedBy	is created by	is created by
NotCreatedBy	is not created by	is not created by

Table 5: Verbalization templates for all relations in ExplaKnow.
⋆: the order of concepts is inverted in the verbalization, e.g. (A, NotReceivesAction, B) is verbalized as *B can not be done to A*.

CN's concepts on average. The intersection scores show that only 35 % of concepts in ExplaKnow are contained in CN, and less than 1 % of ExplaKnow triplets are in CN.

B.2.2 Metrics proposed in Saha et al. (2021)

Saha et al. (2021) propose evaluation of constructed graphs in three steps, where the first two steps evaluate if the *stance* is correctly predicted, and if the graph is *structurally* correct, i.e. if it fulfills the structural constraints imposed by Saha et al. (2021). Graphs are only evaluated in the third step, if the stance-prediction and the structure are correct. In this work, we do not focus on stance prediction and also do not aim at fulfilling the artificial structural constraints. Hence, we skip the first two stages and evaluate our metrics on all graphs, independent of the predicted stance and structural constraints.

In their third evaluation stage, Saha et al. (2021) consider four metrics. However, two of them are automatically assessed by fine-tuned LLMs. These LLMs were fine-tuned on graphs which fulfill the structural constraints, and hence, we would have to use the LLMs out-of-domain if we were to apply them to our CCKGs. Thus, we can not rely on these automatic metrics for our graphs. However, we do adopt the other two proposed metrics from stage three: Graph edit distance (**GED**) measures the minimal number of *edits* to make two graphs isomorph. Edits are local changes, i.e. relabeling, adding or removing a concept or an edge. For increased consistency the GED is normalized to range from 0 to 1. G-BERTScore (**G-BS**) is an extension of BERTScore (Zhang et al., 2020b) to graphs. Triplets are considered as sentences, and BERTScore is used as an alignment score between each pair of triplets. G-BS is computed from the best alignment between the two graphs given the alignment scores.

B.2.3 Additional experiments

This section shows experiments that are slight variations to the setting presented in Table 1. Hence, unless stated otherwise, all CCKGs are pruned CCKGs constructed from ExplaKnow.

Uncontextualized CCKG baselines Table 7 shows the CCKGs and pruned CCKGs compared to the uncontextualized baselines. The results show that CCKGs outperform the baselines without edge weights in concept and triplet precision and F1, as well as in GED and G-BS. Pruning by SBERT similarity introduces contextualization to the baselines, which allows w/o EW_O (i.e. only one randomly chosen unweighted shortest path between two concepts) to achieve comparable performances to the pruned CCKGs. In triplet F1 score the pruned baseline achieves the best result, but it is only outperforming the pruned CCKGs by insignificant 0.09 pp.

The baselines achieve increased recall compared to CCKGs, but the baselines also produce larger graphs which explains the improvements.

CN as KG Table 8 shows the results when using ConceptNet (CN) as KG instead of ExplaKnow. Scores have an upper bound due to the small overlap between CN and the gold graphs (see §B.2.1). Especially for triplets only very low scores are possible.

However, the results show that CCKGs outperform the baselines without edge weights in concept and triplet precision and F1, as well as in GED and G-BS. The performance gap is especially prominent when comparing the unpruned versions. This is likely because the pruning by SBERT similarity introduces contextualization into the otherwise uncontextualized baselines.

The w/o EW^A baselines (i.e. all unweighted shortest paths between two concepts) outperforms

Knowledge Graph	# concepts	# triplets	avg. degree	avg. # words	∩ concepts	∩ triplets
ExplaKnow	7,267	11,437	3.1	2.1	0.35	0.00
CN w/o RelatedTo	939,836	1,313,890	2.8	1.6	1.00	1.00
CN w/ RelatedTo	1,134,506	3,017,472	5.3	1.6	1.00	1.00

Table 6: Knowledge graph statistics. *avg. # words* is the average number of words per concept; ∩ *concepts* and ∩ *triplets* are the number of concepts and triplets respectively in the intersection between the KG and *CN w/ RelatedTo* normalized by the number of concepts / triplets in the respective KG.

Configuration			#nodes	#edges	C P ↑	C R ↑	C F1 ↑	T P ↑	T R ↑	T F1 ↑	GED ↓	G-BS ↑
CCKG		$m=1$	4.1	3.2	**52.10**	38.28	**42.58**	**28.12**	20.19	**22.02**	**0.3458**	66.41
		$m=2$	7.1	6.6	35.90	45.40	38.53	18.68	26.60	20.55	0.3872	**71.39**
		$m=3$	10.1	10.3	28.26	**49.96**	34.81	14.43	**31.11**	18.61	0.4524	60.53
	pruned	$m=1$	4.0	3.0	**52.54**	37.94	**42.67**	**28.55**	19.78	**22.13**	**0.3435**	66.88
	pruned	$m=2$	6.6	5.8	36.67	44.36	38.88	19.42	25.44	20.97	0.3745	**74.26**
	pruned	$m=3$	9.2	8.5	29.25	**48.55**	35.49	15.51	**29.63**	19.56	0.4313	64.50
w/o EW_A		$m=1$	5.5	6.1	**47.97**	40.22	**40.91**	**24.95**	22.85	**20.88**	**0.3805**	**61.47**
		$m=2$	11.4	16.2	29.76	49.34	34.27	14.02	31.96	16.94	0.4811	54.21
		$m=3$	18.2	28.5	21.49	**55.49**	28.76	9.70	**39.11**	13.88	0.5829	38.96
	pruned	$m=1$	4.0	3.1	**52.36**	37.63	**42.46**	**28.16**	19.80	**22.01**	**0.3455**	67.61
	pruned	$m=2$	6.7	6.2	36.48	44.33	38.72	18.73	26.06	20.64	0.3799	**72.54**
	pruned	$m=3$	9.3	9.5	28.95	**48.69**	35.24	14.74	**30.71**	19.05	0.4406	61.00
w/o EW_O		$m=1$	4.6	4.3	**49.77**	39.21	**41.77**	**26.14**	21.91	**21.56**	**0.3600**	**64.98**
		$m=2$	8.9	10.7	32.48	47.81	36.45	15.50	29.90	18.31	0.4393	60.42
		$m=3$	13.7	18.3	24.35	**53.71**	31.75	11.30	**36.70**	15.89	0.5286	45.63
	pruned	$m=1$	3.9	3.1	**52.51**	37.78	**42.61**	**28.38**	19.98	**22.22**	**0.3441**	67.75
	pruned	$m=2$	6.6	6.2	36.70	44.55	38.95	18.92	26.19	20.83	0.3786	**72.71**
	pruned	$m=3$	9.2	9.3	29.05	**48.60**	35.33	14.80	**30.61**	19.11	0.4390	61.28

Table 7: Intrinsic evaluation of pruned CCKGs constructed from ExplaKnow. *w/o EW_A* and *w/o EW_O* are the baselines with unweighted shortest paths described in §4.

CCKGs in terms of recall, but the baseline graphs are also many times larger which greatly harms the precision.

This confirms that CCKGs perform well in the intrinsic evaluation, also when they are constructed from CN.

Support vs. counter instances Table 9 shows the intrinsic evaluation for *support* and *counter* instances separately, i.e. we split the dev set according to the gold stance label. Overall, the results are similar for support and counter instances, except for the concept precision where the supports are more than 4 pp. better. Hence, we do not explicitly show the difference between support and counter in the rest of this paper.

Verbalization Table 10 shows the intrinsic evaluation for *natural* and *static* verbalization templates. The verbalization has a small impact on the results, but the natural verbalization yields better results overall.

In our extrinsic evaluation the verbalization has a larger impact. This could be due to the fact that we evaluate our method extrinsically on CN instead of ExplaKnow. Due to the increased number of triplets in CN a more precise differentiation by the natural verbalization could be more important in CN than in ExplaKnow.

Multiple shortest paths There can be potentially many contextually relevant reasoning paths between each pair of concepts. Hence, considering only the single weighted shortest path between each concept-pair might be too restrictive in the CCKG construction.

Using Yen's algorithm we can compute the k weighted shortest paths between two concepts, where k is another hyperparameter. Dijkstra's algorithm can be seen as the special case of Yen's algorithm with $k = 1$. However, using Yen's algorithm comes at increased costs for us, since Yen's algorithm only computes paths between two specific concepts, while Dijkstra's algorithm computes the shortest paths from one concept to all other concepts in one go. Thus, Yen's algorithm has to be run $m(m-1)$ times, while Dijkstra's algorithm only has to be run $m-1$ times, where m is the number of initially extracted concepts. Furthermore, the time-complexity of Yen's algorithm is kn times Dijkstra's algorithm's time-complexity, where n is the number of concepts in the KG ($n \sim 1,000,000$ for CN). Hence, the path extraction for CCKGs with k shortest paths takes mkn times longer compared to our normal approach.[11]

Table 11 shows the results for pruned CCKGs with $k = 1$ and $k = 3$. Without pruning, the CCKGs with $k = 3$ are larger, leading to a higher recall but lower precision. Overall, the F1 score decreases as the decreased precision outweighs the increased recall. When applying pruning, k only has small effects on F1 scores, with $k = 1$ achieving the best performance. Hence, higher value of k lead to increased computational costs without increasing performance.

Different pruning methods We prune by ranking concepts according to their semantic similarity to the argument, as measured by SBERT. This reduces noise, as contextually irrelevant (i.e. dissimilar) concepts are removed. We expect that to some extent this similarity should also be reflected in the graph structure, and central concepts should be more relevant. Thus, we also try pruning by ranking concepts according to their PageRank. We recompute PageRank after each concept-deletion to ease pruning of chains of concepts.

Table 12 shows that the two pruning methods perform similarly; both increasing precision at the expense of a lower recall. However, pruning by SBERT shows comparable or better performance as pruning by PageRank in all metrics. Thus, we rely on SBERT for pruning.

Constituent parser for concept extraction In the extrinsic evaluation (§4.2), we face the problem that arguments consist of long premises and conclusions. Extracting concepts with our usual approach yields concepts that match the premise and argument, but often they do *not* cover all aspects of the text. Hence, we first parse the texts into *constituents*, and then extract concepts for each constituent individually. Please refer to §B.4.2 for more details.

Table 13 shows the results when relying on constituents for concept extraction. Using the constituents more than doubles the CCKGs in size, but also increases concept and triplet recall by more than 30 pp. The precision on the other hand decreases due to the increased graph size. Overall the concept F1 scores decrease and the triplet F1 scores increase slightly. However, the structural similarity to the gold graphs, as measured by GED and G-

[11] Code for Yen's algorithm adapted from https://gist.github.com/ALenfant/5491853

Configuration			#nodes	#edges	C P ↑	C R ↑	C F1 ↑	T P ↑	T R ↑	T F1 ↑	GED ↓	G-BS ↑
CCKG	w/o RT	$m=1$	4.4	3.4	**20.03**	14.03	**15.40**	**0.30**	0.22	0.24	**0.4393**	57.59
		$m=2$	8.5	8.3	12.91	17.13	13.63	0.24	**0.38**	**0.27**	0.4980	**59.51**
		$m=3$	12.9	14.0	9.79	**19.52**	12.11	0.19	**0.38**	0.23	0.5762	49.75
	pruned	$m=1$	4.2	3.0	**20.54**	14.03	**15.73**	**0.30**	0.22	0.24	**0.4314**	59.27
		$m=2$	7.5	6.6	13.75	16.95	14.32	0.25	**0.32**	**0.27**	0.4737	**64.35**
		$m=3$	10.8	10.4	10.80	**19.04**	13.05	0.19	**0.32**	0.22	0.5389	56.35
	w/ RT	$m=1$	4.2	3.4	**22.27**	15.27	**17.09**	**0.21**	0.18	**0.19**	**0.4373**	58.86
		$m=2$	7.9	8.0	14.41	18.22	15.08	0.11	0.21	0.15	0.4910	**61.65**
		$m=3$	11.6	13.4	10.78	**20.05**	13.21	0.13	**0.36**	0.18	0.5632	51.69
	pruned	$m=1$	3.9	2.8	**22.81**	14.98	**17.30**	**0.23**	0.14	0.17	**0.4282**	60.88
		$m=2$	6.9	6.1	15.40	17.87	15.80	0.11	0.14	0.12	0.4633	**67.48**
		$m=3$	9.8	9.8	11.97	**19.76**	14.30	0.14	**0.32**	**0.19**	0.5272	58.15
w/o EW_A	w/o RT	$m=1$	11.8	18.2	**17.45**	14.46	**13.21**	**0.19**	0.22	**0.19**	**0.5193**	**45.41**
		$m=2$	36.7	66.2	8.36	18.33	8.99	0.13	**0.44**	0.16	0.6663	36.46
		$m=3$	79.1	153.7	4.22	**21.08**	5.81	0.08	**0.44**	0.11	0.8078	19.56
	pruned	$m=1$	4.2	3.2	**20.49**	13.94	**15.66**	**0.30**	0.22	**0.24**	**0.4340**	**59.04**
		$m=2$	13.2	18.3	13.02	17.11	13.54	0.19	**0.32**	0.22	0.5142	58.31
		$m=3$	32.5	56.2	9.69	**19.38**	11.62	0.14	**0.32**	0.18	0.6215	44.55
	w/ RT	$m=1$	15.5	28.4	**18.47**	15.78	**14.14**	**0.14**	0.26	**0.14**	**0.5342**	**45.22**
		$m=2$	46.4	95.7	8.11	19.18	8.90	0.04	0.29	0.07	0.6948	32.81
		$m=3$	91.8	201.6	4.28	**21.75**	5.95	0.04	**0.42**	0.07	0.8223	17.52
	pruned	$m=1$	3.9	3.0	**22.70**	14.97	**17.24**	**0.20**	0.14	**0.16**	**0.4320**	**61.29**
		$m=2$	14.9	24.2	14.41	17.93	14.80	0.09	0.14	0.10	0.5079	59.98
		$m=3$	33.0	63.2	10.81	**20.06**	12.88	0.10	**0.39**	0.15	0.6091	45.91
w/o EW_O	w/o RT	$m=1$	5.3	5.0	**18.85**	14.03	**14.59**	**0.27**	0.22	**0.22**	**0.4662**	**52.38**
		$m=2$	12.2	15.0	10.64	17.05	11.58	0.20	**0.38**	**0.22**	0.5686	48.14
		$m=3$	21.1	29.0	7.09	**19.26**	9.28	0.13	**0.38**	0.17	0.6681	35.71
	pruned	$m=1$	4.1	3.1	**20.53**	13.90	**15.67**	**0.30**	0.22	0.24	**0.4326**	59.21
		$m=2$	8.1	7.9	13.35	16.75	13.91	0.23	**0.32**	**0.25**	0.4895	**61.44**
		$m=3$	13.0	14.7	10.22	**18.77**	12.34	0.17	**0.32**	0.21	0.5661	50.95
	w/ RT	$m=1$	5.1	5.0	**20.66**	15.05	**16.06**	**0.18**	0.18	**0.17**	**0.4625**	**54.94**
		$m=2$	11.6	15.4	11.57	17.98	12.65	0.08	0.21	0.12	0.5651	48.40
		$m=3$	19.3	29.3	7.72	**20.10**	10.15	0.09	**0.42**	0.14	0.6656	35.32
	pruned	$m=1$	3.9	3.0	**22.57**	14.74	**17.07**	**0.22**	0.14	0.17	**0.4315**	61.30
		$m=2$	7.4	7.6	14.93	17.62	15.34	0.13	0.17	0.14	0.4792	**63.78**
		$m=3$	11.3	13.7	11.38	**19.58**	13.64	0.14	**0.42**	**0.19**	0.5544	52.13

Table 8: Intrinsic evaluation of pruned CCKGs constructed from CN. *w/o EW_A* and *w/o EW_O* are the baselines with unweighted shortest paths described in §4

Configuration		#nodes	#edges	C P ↑	C R ↑	C F1 ↑	T P ↑	T R ↑	T F1 ↑	GED ↓	G-BS ↑
all	$m=1$	4.0	3.0	**52.54**	37.94	**42.67**	**28.55**	19.78	**22.13**	**0.3435**	66.88
	$m=2$	6.6	5.8	36.67	44.36	38.88	19.42	25.44	20.97	0.3745	**74.26**
	$m=3$	9.2	8.5	29.25	**48.55**	35.49	15.51	**29.63**	19.56	0.4313	64.50
support	$m=1$	3.8	2.8	**54.76**	37.58	**43.19**	**28.40**	18.55	**21.27**	**0.3511**	64.83
	$m=2$	6.3	5.5	38.22	44.66	39.97	19.78	25.03	21.11	0.3744	**74.70**
	$m=3$	8.7	8.1	30.95	**49.26**	36.97	16.24	**29.56**	20.08	0.4219	66.20
counter	$m=1$	4.2	3.2	**50.32**	38.30	**42.15**	**28.69**	21.02	**22.99**	**0.3359**	68.93
	$m=2$	6.9	6.0	35.13	44.05	37.78	19.07	25.84	20.82	0.3746	**73.82**
	$m=3$	9.6	8.9	27.55	**47.83**	34.01	14.79	**29.71**	19.04	0.4407	62.80

Table 9: Intrinsic evaluation of pruned CCKGs constructed from ExplaKnow on the ExplaGraphs dev split. Results are shown on i) all 398 instances, ii) the 199 *support* instances and iii) the 199 *counter* instances.

Configuration		#nodes	#edges	C P ↑	C R ↑	C F1 ↑	T P ↑	T R ↑	T F1 ↑	GED ↓	G-BS ↑
natural	$m = 1$	4.0	3.0	**52.54**	37.94	**42.67**	**28.55**	19.78	**22.13**	**0.3435**	66.88
	$m = 2$	6.6	5.8	36.67	44.36	38.88	19.42	25.44	20.97	0.3745	**74.26**
	$m = 3$	9.2	8.5	29.25	**48.55**	35.49	15.51	**29.63**	19.56	0.4313	64.50
static	$m = 1$	3.8	2.8	**52.41**	37.07	**42.03**	**28.06**	19.15	**21.51**	**0.3478**	65.71
	$m = 2$	6.4	5.5	36.52	42.94	38.24	19.25	24.47	20.52	0.3760	**74.50**
	$m = 3$	8.8	8.1	28.90	**46.73**	34.69	14.94	**27.95**	18.68	0.4320	65.92

Table 10: Intrinsic evaluation of pruned CCKGs constructed from ExplaKnow with *natural* and *static* verbalization.

k		m	#nodes	#edges	C P ↑	C R ↑	C F1 ↑	T P ↑	T R ↑	T F1 ↑	GED ↓	G-BS ↑
1		1	4.1	3.2	**52.10**	38.28	**42.58**	**28.12**	20.19	**22.02**	**0.3458**	66.41
		2	7.1	6.6	35.90	45.40	38.53	18.68	26.60	20.55	0.3872	**71.39**
		3	10.1	10.3	28.26	**49.96**	34.81	14.43	**31.11**	18.61	0.4524	60.53
	pruned	1	4.0	3.0	**52.54**	37.94	**42.67**	**28.55**	19.78	**22.13**	**0.3435**	66.88
		2	6.6	5.8	36.67	44.36	38.88	19.42	25.44	20.97	0.3745	**74.26**
		3	9.2	8.5	29.25	**48.55**	35.49	15.51	**29.63**	19.56	0.4313	64.50
3		1	9.7	12.4	**31.95**	51.48	**37.33**	**15.75**	35.90	**19.78**	**0.4545**	**51.03**
		2	16.6	23.0	20.53	58.25	29.26	9.20	42.63	14.46	0.5924	33.46
		3	17.8	24.9	20.23	**60.87**	29.36	9.19	**45.74**	14.75	0.6055	31.68
	pruned	1	4.0	3.1	**52.38**	38.21	**42.68**	**28.40**	19.96	**22.12**	**0.3445**	66.79
		2	6.9	6.2	36.29	44.79	38.65	18.96	26.04	20.71	0.3802	**72.83**
		3	7.7	7.3	34.76	**47.92**	38.84	18.37	**29.42**	21.41	0.3934	69.34

Table 11: Intrinsic evaluation of CCKGs constructed from ExplaKnow. The number of shortest paths between each pair of extracted concepts is $k = 1$ and $k = 3$.

Configuration		#nodes	#edges	C P ↑	C R ↑	C F1 ↑	T P ↑	T R ↑	T F1 ↑	GED ↓	G-BS ↑
None	$m = 1$	4.1	3.2	**52.10**	38.28	**42.58**	**28.12**	20.19	**22.02**	**0.3458**	66.41
	$m = 2$	7.1	6.6	35.90	45.40	38.53	18.68	26.60	20.55	0.3872	**71.39**
	$m = 3$	10.1	10.3	28.26	**49.96**	34.81	14.43	**31.11**	18.61	0.4524	60.53
SB	$m = 1$	4.0	3.0	**52.54**	37.94	**42.67**	**28.55**	19.78	**22.13**	**0.3435**	66.88
	$m = 2$	6.6	5.8	36.67	44.36	38.88	19.42	25.44	20.97	0.3745	**74.26**
	$m = 3$	9.2	8.5	29.25	**48.55**	35.49	15.51	**29.63**	19.56	0.4313	64.50
PR	$m = 1$	4.0	3.0	**52.26**	37.58	**42.36**	**28.14**	19.30	**21.70**	**0.3445**	66.83
	$m = 2$	6.6	5.7	36.38	43.71	38.48	18.93	24.47	20.34	0.3763	**74.23**
	$m = 3$	9.1	8.5	29.04	**48.03**	35.20	15.17	**28.86**	19.09	0.4327	64.46

Table 12: Intrinsic evaluation of different pruning methods on ExplaKnow. Pruning is ranked by *None*: no pruning; *SB*: SBERT; *PR*: PageRank.

BS, decreases as a result of the larger graph sizes. Thus, not using constituents for concept extraction achieves better scores overall in the intrinsic evaluation. We expect that this would change in an evaluation with longer sentences and larger gold graphs.

B.3 ExplaGraphs manual evaluation

B.3.1 Annotation description

For each instance, we asked a series of questions to annotators for which they had to select one answer or say that they can not make a decision. The first set of questions revolved around the argument as such, without considering the graph. In **Q1** annotators selected the correct of 9 predefined *topics*. Next, in **Q2**, we asked whether the conclusion is *plausible* given the premise. We asked this to assess i) quality of Saha et al. (2021)'s arguments, and ii) whether we obtain plausible premise-conclusion pairs from the belief-argument pairs. If an argument was labeled as plausible, then in **Q3** annotators had to decide if they can identify an *implicit CSK connection* that links the conclusion to the premise. If so, we also ask the annotators to formulate and write down the perceived CSK connection in plain language. This serves to familiarize the annotators with the argument, and provides them with a reference to their own interpretation in the later graph quality assessment steps.

The second set of questions were only presented for plausible arguments with a perceived CSK connection, to assess the quality of the provided CCKGs. **Q4**: To estimate the *recall* we asked if the graph shows the implicit connection i) *completely* ii) *partially* or iii) *not at all*. Then, to estimate *precision* at a fine-grained analysis level, each individual triplet had to be labeled in **Q5** as i) *positive* (expresses implicit CSK) ii) *neutral* (does not express implicit CSK, but matches the topic) iii) *unrelated* (does not match the topic) or iv) *negative* (contradicts implicit CSK or the conclusion, but the topic is appropriate). An example of a *negative* triplet would be (human_cloning, IsA, considered_unethical) in a CCKG for an argument with a pro-cloning conclusion. For *negative* triplets, we further asked (**Q6**) if its *negation* expresses relevant implicit CSK, and (**Q7**) if the graph extended with the negated triplet(s) shows the CSK connection. However, *negative* triplets were rare in our CCKGs, such that we could not perform analysis of Q6 and Q7.

Please refer to our official annotation guidelines at https://github.com/Heidelberg-NLP/CCKG/blob/main/annotation_guidlines.pdf for more details on each question, as well as illustrative examples.

B.3.2 Annotation results

For each question, Table 14 reports the *support*, i.e. the number of instances that were annotated by both annotators. Note that the support decreases in Q3 and Q4, since annotation instances that were labeled with *no* in Q2 (*plausible argument?*) or Q3 (*implicit CSK in argument?*) were not further annotated by the individual annotators. We only report values for which both annotators provided labels. Q5 has a support of 1,169 triplets that come from the same 115 graphs as annotated in Q4.

To measure **inter-annotator agreement**, we report the counts of the assigned labels per class and annotator (*A1*, *A2*), and compute agreement scores using a) *Cohen's Kappa* κ, where we compute κ of individual labels in a one-vs-all setting, i.e. by considering all other labels as the same label. This we complement by b) counts and percentages of the *overlap of label assignments* (A1 $\wedge$ A2) by the two annotators per class.[12] We also report the percentage of labels assigned by both annotators unanimously or by at least one annotator.

We now investigate the annotation results on Q1 to Q5.

Q1 (*Topic*): The arguments are uniformly distributed across topics. The topics are quite distinct such that the annotators could assign them to the correct classes with ease, with only minimal divergences, yielding a high inter-annotator agreement ($\kappa = 0.916$).

Q2 (*plausible?*): A large majority of instances (79.90 %) were unanimously labeled as plausible, which shows that Saha et al. (2021)'s support *belief-argument* pairs can indeed be interpreted as *premise-conclusion* pairs.

However, κ is low, as one annotator considered all but 3 arguments as plausible, while the other considered 38 of the 199 arguments, i.e., 19 %, as implausible. On deeper inspection we found that these 19 % suffered from various deficiencies: multiple negations made interpretation very difficult and did often not yield a valid supporting argument; in other cases the pairs were presented in

[12] The percentage is computed relative to the average of A1 and A2.

Configuration			#nodes	#edges	C P ↑	C R ↑	C F1 ↑	T P ↑	T R ↑	T F1 ↑	GED ↓	G-BS ↑
w/o constituents		$m=1$	4.1	3.2	**52.10**	38.28	**42.58**	**28.12**	20.19	**22.02**	**0.3458**	66.41
		$m=2$	7.1	6.6	35.90	45.40	38.53	18.68	26.60	20.55	0.3872	**71.39**
		$m=3$	10.1	10.3	28.26	49.96	34.81	14.43	31.11	18.61	0.4524	60.53
		$m=4$	13.0	14.1	23.26	52.37	31.11	11.59	33.45	16.30	0.5158	50.47
		$m=5$	15.9	17.8	19.94	**54.41**	28.28	9.88	**36.08**	14.81	0.5637	42.98
	pruned	$m=1$	4.0	3.0	**52.54**	37.94	**42.67**	**28.55**	19.78	**22.13**	**0.3435**	66.88
		$m=2$	6.6	5.8	36.67	44.36	38.88	19.42	25.44	20.97	0.3745	**74.26**
		$m=3$	9.2	8.5	29.25	48.55	35.49	15.51	29.63	19.56	0.4313	64.50
		$m=4$	11.5	11.2	24.43	50.94	32.17	12.77	31.88	17.58	0.4890	54.90
		$m=5$	13.8	13.8	21.17	**52.87**	29.54	10.96	**34.14**	16.11	0.5337	47.59
w/ constituents		$m=1$	19.5	24.4	**26.47**	71.13	**36.30**	**16.13**	55.71	**23.00**	**0.5489**	**39.86**
		$m=2$	36.5	51.9	15.97	78.63	25.32	8.91	64.71	14.84	0.7038	22.40
		$m=3$	52.9	80.0	11.25	82.79	19.16	5.83	69.72	10.40	0.7843	15.01
		$m=4$	68.8	109.0	8.91	85.12	15.67	4.51	72.88	8.25	0.8286	11.42
		$m=5$	85.2	139.7	7.31	**87.23**	13.17	3.60	**75.70**	6.71	0.8600	8.99
	pruned	$m=1$	13.8	13.7	**30.53**	66.47	**40.18**	**19.63**	48.57	**26.51**	**0.4690**	**51.51**
		$m=2$	23.4	24.9	20.29	74.41	30.91	12.62	57.35	19.99	0.6076	33.72
		$m=3$	32.9	37.2	15.27	78.90	24.97	9.30	62.98	15.79	0.6895	24.89
		$m=4$	42.2	50.3	12.43	81.44	21.10	7.49	66.41	13.15	0.7406	19.94
		$m=5$	51.4	63.1	10.56	**83.91**	18.40	6.34	**69.96**	11.41	0.7777	16.45

Table 13: Intrinsic evaluation of different concept extractions for pruned CCKGs constructed from ExplaKnow.

the wrong direction to count as an argument. One of our annotators considered the arguments with great care and we could validate his judgements in almost all cases. We are therefore confident that the vast majority of such cases could be captured in our annotation.

Q3 (*implicit CSK in argument?)*: Only $6.29\,\%$ of arguments were unanimously judged as not being linked through implicit CSK, which confirms that Saha et al. (2021)'s data collection successfully resulted in *belief-argument* pairs that require explanations. In $72.33\,\%$ of cases both annotators agreed that there is implicit CSK (115 instances). On these 115 instances we evaluate the performance of our CCKGs.

Q4 (*CSK covered in CCKG?*): Here the annotators evaluated whether the presented CCKG covered the implicit knowledge, by referring to what they had written down in Q3, but they could also accept another valid interpretation expressed by the graph. $29.57\,\%$ of CCKGs were unanimously annotated to cover the implicit CSK *completely*, i.e. the argument could be fully understood based on knowledge shown in the CCKG. When considering CCKGs annotated by at least one annotator as complete, the score doubles to $59.13\,\%$. $88.70\,\%$ were unanimously judged to cover the implicit CSK at least *partially*, which corresponds to a *high recall of implicit CSK* in the constructed CCKGs. I.e., most CCKGs make the connection between conclusion and premise more explicit, and hence, they can be expected to support computational systems in knowledge-intense argumentation tasks. With 0.413, Cohen's κ is higher for *completely* than for *partially*, indicating that partial coverage is more subjective to decide.

Q5 (*Triplet rating*): The remaining 115 CCKGs contain 1,169 triplets in total. Out of these, $39.44\,\%$ were unanimously labeled as *positive*, i.e., the triplet *reflects implicit CSK* that links the conclusion to the premise (again, annotators are asked to compare the CCKG to their answer to Q3, but are free to accept other valid connections in the CCKG), and for $74.68\,\%$ at least one annotator rated the triplet as positive. This shows that a substantial amount of triplets reflect implicit CSK, while the judgement may be subjective, depending on the annotator's own interpretation. Also, it is often difficult to decide what the exact implicit CSK is.

$13.94\,\%$ of all triplets were unanimously labeled *neutral*, i.e. they express knowledge pertaining to the topic of the argument. As such, they contribute additional knowledge or context for the argument, but no CSK that is required to support the conclusion.

Only $1.71\,\%$ of triples were unanimously labeled as *unrelated*, i.e. as not matching the ar-

gument because they no not match the topic. These triplets represent noise in the CCKG, and are mostly avoided by the strong contextualization during graph construction. Only a small number remains after pruning.

1.07 % of all triplets are unanimously labeled *negative*, i.e. they contradict the conclusion or the implicit CSK. These triplets are from the correct topic, but often show the issue from a different perspective and do not support the conclusion.

In the first block of Table 15, we also report macro averages over the triplet precision measured in Q5 (triplet rating) for individual graphs. We report the score for triplets showing implicit CSK (i.e. *positive* triplets) and triplets being from the correct topic (i.e. *positive*, *neutral* or *negative* triplets). Again, we report the *support* and the values for each individual annotator *A1* and *A2*. We derive a joint rating from both annotators by either i) A1 $\wedge$ A2: A triplet is only considered as positive / in-topic if both annotators labeled it as such, or ii) A1 $\vee$ A2: A triplet is considered as positive / in-topic if at least one annotator labeled it as such.

The unanimous macro precision is 39.43 % for triplets showing implicit CSK and 73.87 % when considering triplets rated as positive by at least one annotator. This matches our observation from the micro scores. Our CCKGs show high in-topic macro precision with 92.76 % in the unanimous setting and exceeding 99 % when considering triplets rated by at least one annotator as in-topic.

Table 15 also shows the macro precision for graphs which were unanimously judged to reflect the implicit CSK in the argument *completely* and *partially* in Q4. The precision of unanimous positive triplets increases by more than 15 pp. when considering only CCKGs that reflect the implicit CSK completely. On the other hand, the precision of in-topic triplets increases more when considering CCKGs that reflect the implicit CSK only partially. This indicates that CCKGs that fail to reflect implicit CSK completely still reflect CSK from the correct topic.

Overall, the manual annotation shows strong performance of the CCKGs in terms of implicit CSK recall, implicit CSK precision, and in-topic precision.

B.4 VALNOV

B.4.1 Data statistics

Heinisch et al. (2022b) collect arguments from diverse topics, where the conclusions are partially automatically generated. The binary labels for validity and novelty are manually created by multiple annotators. The data for the VALNOV Shared Task[13] has been constructed from arguments from an argumentative dataset (Ajjour et al., 2019), and has been extended by conclusions automatically generated with T5 (Heinisch et al., 2022a; Raffel et al., 2020), producing instances of paired premise-conclusion pairs. All instances were manually assigned binary labels for *validity* and *novelty*.

The VALNOV train/ dev/ test sets consist of 750/ 202/ 520 instances. However, 48 of the train instances are *defeasible*, i.e. instances with no annotator majority for validity or novelty. We remove these instances, leaving us with 702 training items.

The train set is unbalanced, with only 2 % of the train data being from the *non-valid and novel* class.

Heinisch et al. (2022c) extend the dataset by integrating datasets from different tasks as well as synthetic data. In this work we only use the original dataset proposed by Heinisch et al. (2022b).

B.4.2 Model variations

Concept extraction with constituents The arguments in the VALNOV dataset are relatively long (76 tokens in avg.), often containing more that one aspect / perspective. This negatively effects the quality of triplet selection for concept extraction: the extracted concepts are semantically relevant, but often do not span the entire argument. We thus split the text into constituents using a SOTA parser (Zhang et al., 2020a), and select concepts for each constituent separately. The hyperparameter m now controls the number of extracted triplets for each constituent. We use their `crf-con-roberta-en` model at `www.github.com/yzhangcs/parser`. Leaf nodes often consist of only one or two concepts, which limits contextualization for these constituents. Hence, we disregard the leaf nodes to reduce noise in concept extraction.

Partial pruning Pruning CCKGs completely bears the risk of removing relevant structure. However, not pruning at all leaves the CCKGs in a potentially noisy state. To allow for a more fine-grained balance, we apply *partial pruning*. I.e., we rank concepts and prune the CCKG accordingly, but instead of pruning all possible concepts we

[13] The task was organized as part of the ArgMining workshop 2022.

Question	Label	Counts [#]				Agreement		Quality [%]	
		Support	A1	A2	A1 ∧ A2	κ	A1 ∧ A2 [%]	A1 ∧ A2	A1 ∨ A2
Q1	all labels	199			184	0.916			
which	abandon marriage		24	26	24	0.954	96.00	12.06	13.07
topic?	ban cosmetic surgery		22	20	20	0.947	95.24	10.05	11.06
	adopt an austerity regime		22	20	19	0.894	90.48	9.55	11.56
	fight urbanization		22	22	22	1.000	100.00	11.06	11.06
	subsidize embryonic stem cell research		19	18	17	0.911	91.89	8.54	10.05
	legalize entrapment		22	22	22	1.000	100.00	11.06	11.06
	ban human cloning		21	21	19	0.894	90.48	9.55	11.56
	close Guantanamo Bay detention camp		21	21	21	1.000	100.00	10.55	10.55
	adopt atheism		22	25	19	0.783	80.85	9.55	14.07
	×		4	4	1	0.235	25.00	0.50	3.52
Q2	all labels	199			160	0.021			
plausible	yes		196	161	159	0.021	89.08	79.90	99.50
argument?	no		3	38	1	0.021	4.88	0.50	20.10
Q3	all labels	159			125	0.298			
implicit CSK	yes		149	115	115	0.298	87.12	72.33	93.71
in argument?	no		10	44	10	0.298	37.04	6.29	27.67
Q4	all labels	115			68	0.268			
CSK in	completely		43	59	34	0.413	66.67	29.57	59.13
CCKG?	partially		59	56	34	0.183	59.13	29.57	70.43
	completely or partially		102	115	102	0.000	94.01	88.70	100.00
	not at all		13	0	0	0.000	0.00	0.00	11.30
Q5 (micro)	all labels	⋆1169			656	0.230			
triplet rating	positive		556	778	461	0.306	69.12	39.44	74.68
	neutral		465	321	163	0.133	41.48	13.94	53.29
	unrelated		100	54	20	0.212	25.97	1.71	11.46
	negative		48	16	12	0.362	37.50	1.03	4.45
	positive or neutral		1021	1099	985	0.251	92.92	84.26	97.09
	in-topic (i.e. all but unrelated)		1069	1115	1035	0.212	94.78	88.54	98.29

Table 14: Results of manual annotation. *A1* and *A2* show the label counts for each individual annotator. *A1* ∧ *A2* shows the counts (#) and percentages (%) of instances labeled unanimously by *A1* and *A2*; *A1* ∨ *A2* shows the percentage of instance labels assigned by at least one of the annotators. × means that the annotator could not decide. Topic labels for Q0 are posed as "We should ..." .
⋆: the supporting 1,169 triplets are from the 115 supporting CCKG graphs from Q4.

CSK shown in CCKG	Label	support	A1	A2	A1 ∧ A2	A1 ∨ A2
All	Implicit CSK	115	48.36	64.95	39.43	73.87
	Topic	115	94.94	97.01	92.76	99.20
Completely	Implicit CSK	34	66.36	74.90	56.37	84.89
	Topic	34	95.94	97.16	93.57	99.53
Partial	Implicit CSK	34	48.02	58.39	36.77	69.64
	Topic	34	99.18	97.79	97.42	99.55

Table 15: Macro precision scores of manual annotation on Q5 (triplet rating) in %. *A1* and *A2* are the macro averages for each individual annotator, $A1 \wedge A2$ is the macro average when only considering unanimous decisions and $A1 \vee A2$ is the macro average when considering triplets which at least one annotator judged as positive / in-topic.

only remove the first 25 %, 50 % or 75 %, which corresponds to removing only the most dissimilar concepts.

B.4.3 Feature extraction

Structural features We extract 5 features describing the **size** of CCKGs (number of concepts, number of triplets, number of premise-concepts, number of conclusion-concepts, number of concepts shared by premise and conclusion), 6 features describing the **connectivity** of CCKGs (number of cluster with and without edge weights and the corresponding modularity, density, transitivity), and 4 features describing the **distance** between premise and conclusion in the CCKG (weighted and unweighted MinCut between premise-concepts and conclusion-concepts, average and maximal weighted length between premise-concepts and conclusion concepts). This yields 15 graph features in total.

Textual features We consider the **semantic similarity** between premise and conclusion (measured by SBERT), and the **NLI** probabilities that the premise *entails*, is *neutral* or *contradicts* the conclusion. We compute the NLI predictions from a RoBERTa-large (Zhuang et al., 2021) model which was fine-tuned on NLI data.[14] This yields 4 text features in total.

B.4.4 Classifier

We use scikit-learn (Pedregosa et al., 2011)'s *RandomForest* and *SVM*. For the SVM we test linear and RBF kernels.

Our RandomForests consist of 1000 trees with Gini impurity and 4 features considered at each split. Data is sampled with bootstrapping. For regularization we use Minimal Cost-Complexity Pruning with the hyperpaprameter α. We choose the best value for α on the dev split from $\{0, 1e-4, 5e-4, 1e-3, 5e-3, 1e-2, 5e-2, 1e-1, 5e-1\}$.

For the SMVs we apply a shrinking heuristic and choose the regularization parameter C on the dev split from $\{1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 0.5, 1, 2, 5, 10\}$. For the RBF kernel we set γ to *scale* or *auto*, also determined on the dev split.

The best setting for CCKGs was RandomForest with $\alpha = 0.01$. For methods from Saha et al. (2022) the best methods were T5: RandomForest with $\alpha = 0.05$; max-margin: RandomForest with $\alpha = 0.05$; and contrastive: SVM with RBF with $\gamma = auto$ and $C = 5$.

[14] We applied `roberta.large.mnli` from https://github.com/facebookresearch/fairseq/blob/main/examples/roberta/README.md

	configuration	joint F1	Val F1	Nov F1
	CCKG	43.91	70.69	63.30
CCKG	w/o EW_O	**-6.51**	-3.80	-1.69
	w/o EW_A	-3.25	**-4.45**	1.76
	w/ static verbalization	-4.32	**-6.34**	2.44
	w/ RelatedTo	-3.66	**-6.86**	2.20
	w/o pruning	**-8.24**	-2.81	-0.14
	w/ full pruning	**-5.28**	-3.39	-1.53
Concept Extract.	$m = 1$ w/ const.	**-5.39**	-5.27	-1.43
	$m = 3$ w/ const.	**-7.57**	-3.03	0.01
	$m = 1$ w/o const.	-8.11	-3.53	**-11.08**
	$m = 2$ w/o const.	-3.36	**-5.86**	0.79
	$m = 3$ w/o const.	-3.88	**-5.83**	0.41
	string matching	**-6.71**	-3.23	0.55
Features	w/o Text feats.	-20.65	**-20.74**	-17.69
	w/o Graph feats.	**-11.65**	-3.40	-5.12
	w/o connectivity feats.	**-5.60**	-4.01	-0.60
	w/o size	-2.60	**-2.93**	0.80
	w/o PC-distance feats.	-2.27	-0.27	**-3.73**
	w/o upsampling	**-4.11**	-2.43	-3.10

Table 16: System ablations: values show performance differences to our full system results. 1st block: different CCKG constructions; 2nd block: configurations for concept extraction; 3rd block: CCKGs with different features / upsampling turned off.

B.4.5 Ablation

Our white-box feature-based system allows for a thorough ablation study (see Table 16). We first explore variations in **CCKG construction**. Ablating all *edge weights* incurs considerable performance losses for the joint and validity scores. Considering only one random path between each pair of concepts (w/o EW_O) additionally has reduced performance for novelty. However, considering all unweighted shortest path (w/o EW_A) increases the novelty score by 1.76 pp. This indicates that contextualization is more relevant for validity, perhaps because without edge weights the model can not distinguish between valid and non-valid connections. The static *verbalization* (see §B.1.2) reduces the quality of edge weights, and hence decreases validity score by 6.34 pp. On the other hand, it increases the novelty score but not enough to compensate for the reduction in validity. Unspecific RelatedTo edges have a strong negative impact for validity but improve novelty, by attracting more knowledge. No *pruning* fails to distinguish valid from non-valid conclusions due to too many noisy connections. Too much pruning on the other hand removes structural diversity and hence decreases

the predictive power of CCKGs. The results suggest that contextualized graph construction has a strong impact on *validity* and the *joint* score, which intuitively makes sense as the contextualization promotes valid connections. At the same time, the fluctuating effects for *novelty* indicate that novelty and validity are difficult to calibrate, but at a relatively low impact level.

The impact of **concept extraction** can be best observed when comparing $m = 1$ with $m = 2, 3$ without the constituent parser. Choosing $m = 1$ results in small graphs, which can not cover all aspects of the argument. Hence, the resulting graphs are not suitable for predicting novelty. Increasing m alleviates this problem, but decreases validity. We found $m = 2$ with constituent parsing to yield best results.

Feature ablation shows that both, *text and graph features*, are necessary to achieve good performance. The textual features have a stronger impact on validity, while the graph features are more impactful for novelty prediction. Yet, both metrics benefit from both modalities. This indicates that text and CCKG contain complementary information and should be considered jointly in future work. Finally, we remove selected graph features from the classifier, i.e. all *size*, *connectivity* or *premise-conclusion distance* features, at a time. This induces losses of 5.60 pp. / 4.01 pp. joint / validity score, for connectivity features, and strong losses of 3.73 pp. for novelty when removing PC-distance features. This supports our hypothesis that validity correlates with the connectivity, and novelty with the distance between premise-concepts and conclusion-concepts in the CCKGs.

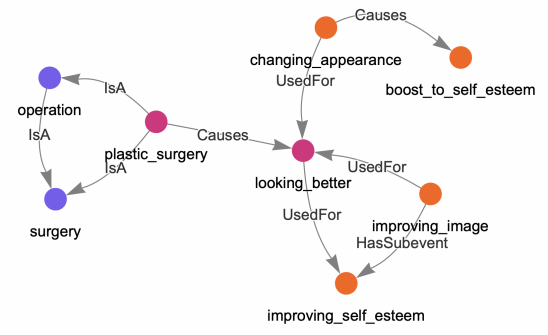
Table 17 shows feature ablations when constructing graphs with the **supervised methods** from Saha et al. (2022). The graph contributes more to novelty prediction in all three methods. This is consistent with previous findings, as the models leverage structural data which was found to be important for novelty. However, the effect of ablating features varies for each method and no clear trend is apparent.

configuration	joint F1	Val F1	Nov F1
T5	37.71	67.07	63.52
w/o Text feats.	-10.97	**-16.35**	-12.84
w/o Graph feats.	**-5.45**	0.21	-5.34
w/o connectivity feats.	**-0.63**	0.26	-0.57
w/o size	0.87	-0.66	**-1.72**
w/o PC-distance feats.	**0.64**	0.55	-0.08
w/o upsampling	-0.10	0.49	**-4.33**
max-margin	36.22	67.61	63.27
w/o Text feats.	-15.22	-18.59	**-19.90**
w/o Graph feats.	-3.96	-0.33	**-5.08**
w/o connectivity feats.	**0.67**	0.04	0.52
w/o size	**0.69**	0.04	0.55
w/o PC-distance feats.	**5.32**	-1.07	0.37
w/o upsampling	-4.05	0.43	**-16.79**
contrastive	37.82	64.77	59.96
w/o Text feats.	-5.35	**-10.27**	-2.70
w/o Graph feats.	**-5.56**	2.51	-1.77
w/o connectivity feats.	-0.75	**2.85**	-0.76
w/o size	**-5.85**	2.51	-1.91
w/o PC-distance feats.	-1.00	**4.28**	-3.49
w/o upsampling	0.39	**-0.44**	0.27

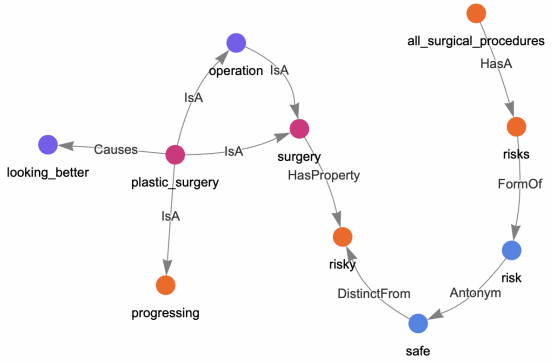
Table 17: Feature ablations for Saha et al. (2022)' graphs with our feature extraction and classification. Ablated scores show performance distance to respective base approach.

C Example CCKGs

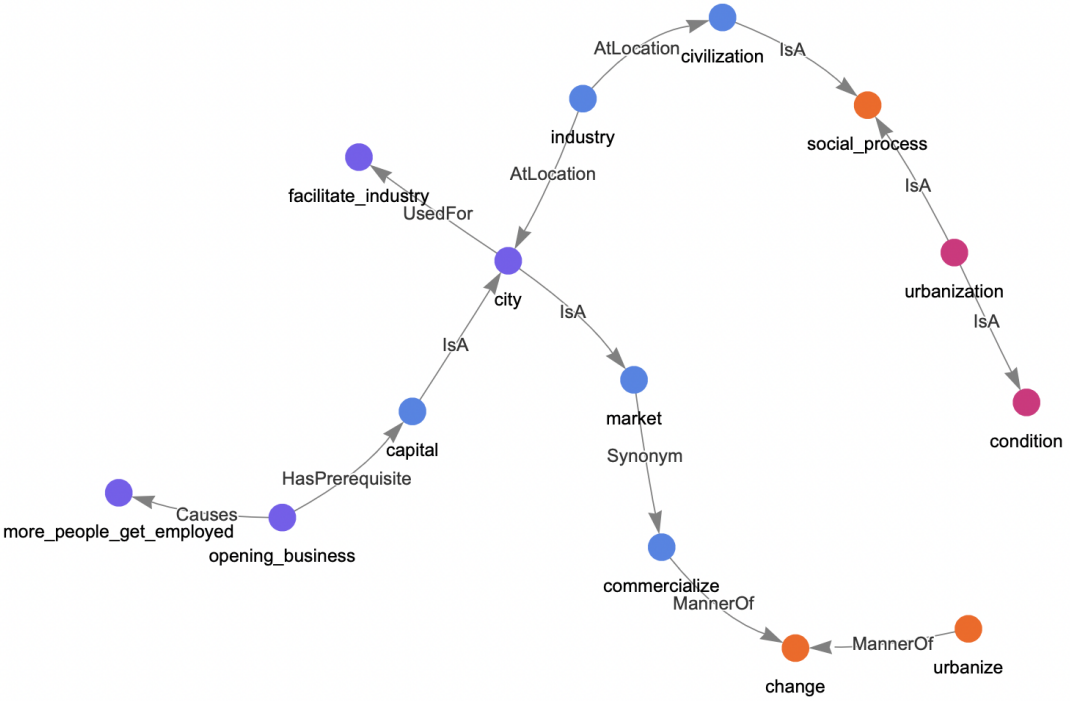
The graphs in Figures 3, 4 and 5 show extracted premise concepts in violet, conclusion concepts in orange and intermediate concepts in blue. Concepts which are extracted for both, the premise and the conclusion, are shown in pink. Visualizations were done with *pyvis* (`www.github.com/WestHealth/pyvis`).



(a) Premise: *Cosmetic surgery makes people feel whole again.*
Conclusion: *Cosmetic surgery improves self esteem.*

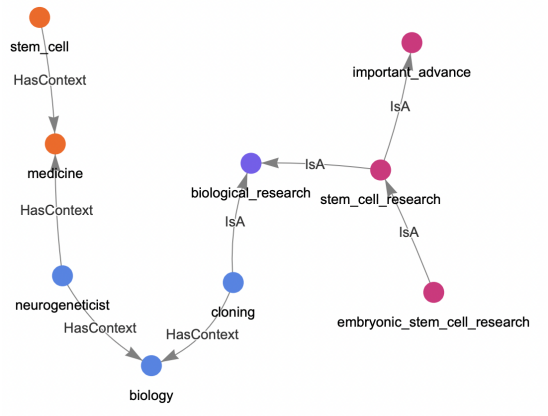


(b) Premise: *Cosmetic surgery can cause defects.*
Conclusion: *Cosmetic surgery can be dangerous.*

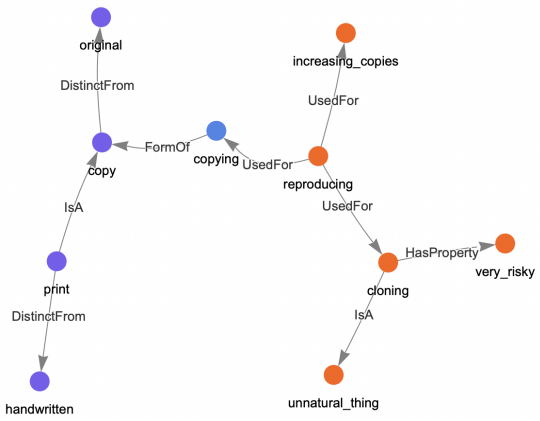


(c) Premise: *Urbanization increases employment for many.* Conclusion: *Urbanization is a positive for society.*

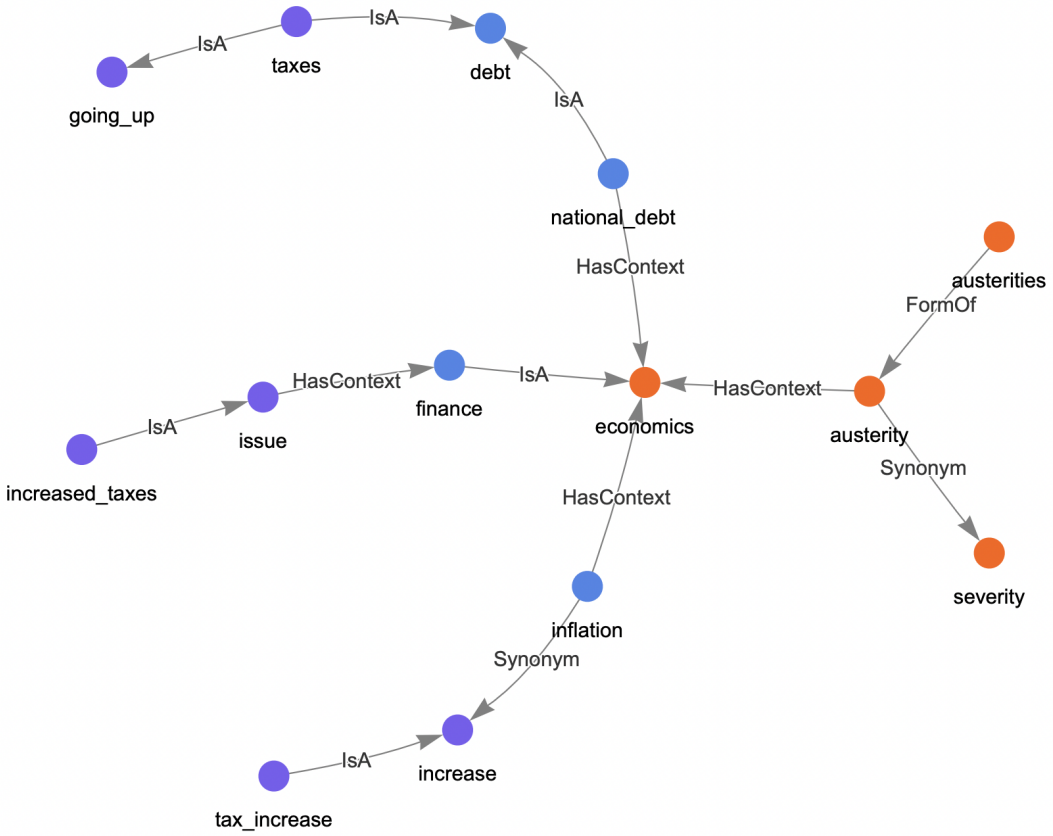
Figure 3: Example CCKGs for arguments from ExplaGraphs dev set. Graphs are pruned CCKGs extracted from CN without RelatedTo with $m = 3$. Figure 3c has the disambiguity problem: capital is once used as city, and once as financial asset.



(a) Premise: *Embryonic stem cell research is a no brainer.* Conclusion: *Embryonic stem cell research is very important to medicine.*

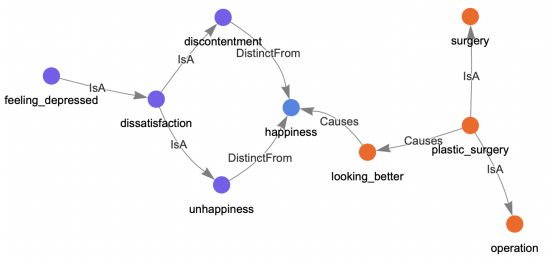


(b) Premise: *Getting your original out of the copier and putting it against the copy always shows differences.* Conclusion: *Cloning is inherently decreasing quality.*

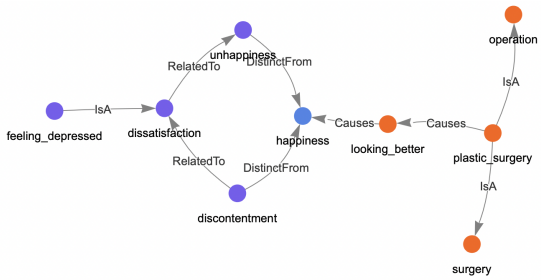


(c) Premise: *Austerity raises taxes on citizens.* Conclusion: *Austerity would cripple the population.*

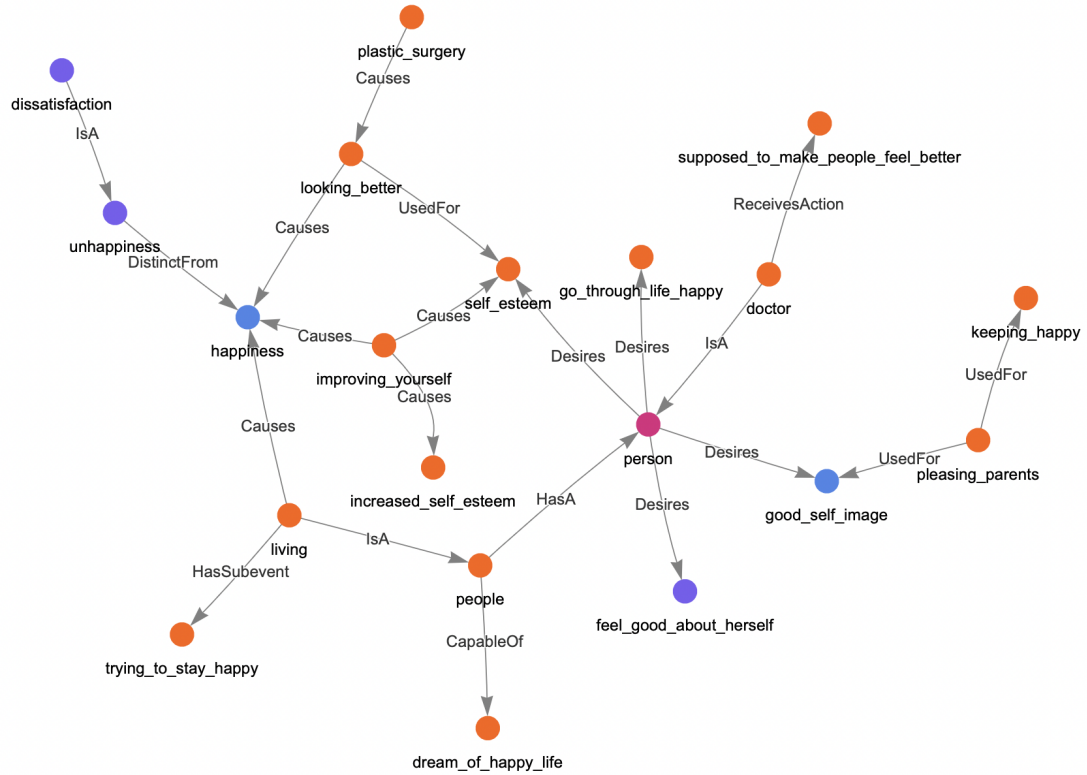
Figure 4: Randomly selected example CCKGs for arguments from ExplaGraphs dev set. Graphs are pruned CCKGs extracted from CN without $\mathrm{RelatedTo}$ with $m = 3$.



(a) $m = 3$, without RelatedTo, without constituent parser



(b) $m = 3$, with RelatedTo, without constituent parser



(c) $m = 1$, without RelatedTo, with constituent parser

Figure 5: Example CCKGs for premise "*A person is unhappy if she is dissatisfied with her body.*" and conclusion "*Plastic surgery raises patients' self esteem and allows them to lead normal happy lives.*"

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
4.1.2, 4.2, 5, Limitation

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

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

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

B ☑ Did you use or create scientific artifacts?

Footnote 1

☑ B1. Did you cite the creators of artifacts you used?
4 (experimental setup), 4.1 & 4.1.1 (Dataset and baselines), 4.2 (Dataset and baselines), B4.3, B4.4

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
All used artifacts are publically available and free to use. We will make our artefacts public on acceptance.

☑ 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)?
4.1 (repurposing of dataset), 4.2 (using models out of domain), 5 (our approach is only tested on ConceptNet)

☒ 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?
Collecting or using data was not the focus of this work and hence, checking it was beyond the scope of our work. That being said, we did produce novel data in our annotation study, but the annotators are anonymous.

☒ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Again, this was not the main scope of this work. However, we did provide insights, when necessary, for example the structural constraints in 4.1 or the length of arguments in 4.2.

☑ 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.
4.1, 4.1.2, B3.2, B4.1

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

C ☑ Did you run computational experiments?

4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Our model is unsupervised and hence has 0 learnable parameters. The complexity of our approach is partially described in section3. We did not discuss the complexity of components that are i) part of preprocessing, ii) computationally trivial (e.g. combining paths to a graph) or iii) part of previous work (e.g. running SBERT).

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
4, 4.1.1, 4.1.2, 4.2, A.1, B.1.2, B.4.4

☑ 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?
4.2, B3.2

☑ 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.)?
4, B4.4, C

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

4.1.2

☑ 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.?
https://github.com/Heidelberg-NLP/CCKG/blob/main/annotation_guidlines.pdf

☑ 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)?
In 4.1.2. we say that they are students. However, we do not explicitely say that we paid them adequately (although we did).

☑ 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?
The annotation guidelines include it.

☒ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Our data collection was only on rating existing textual data and subsets of existing knowledge graphs. Thus, we did not work with sensitive data and do not think that explicit approval was necessary.

☒ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
We only had two annotators and releasing such information might remove anonymity of annotators. However, we do say that both annotators are CL students with strong English skills.