SemEval-2018 Task 12: The Argument Reasoning Comprehension Task

Ivan Habernal[†] **Henning Wachsmuth**[‡] **Iryna Gurevych**[†] **Benno Stein**[‡]

[†] Ubiquitous Knowledge Processing Lab (UKP) and Research Training Group AIPHES
Department of Computer Science, Technische Universität Darmstadt, Germany
www.ukp.tu-darmstadt.de www.aiphes.tu-darmstadt.de

[‡] Faculty of Media, Bauhaus-Universität Weimar, Germany
<firstname>.<lastname>@uni-weimar.de

Abstract

A natural language argument is composed of a claim as well as reasons given as premises for the claim. The *warrant* explaining the reasoning is usually left implicit, as it is clear from the context and common sense. This makes a comprehension of arguments easy for humans but hard for machines. This paper summarizes the first shared task on *argument reasoning comprehension*. Given a premise and a claim along with some topic information, the goal is to automatically identify the correct warrant among two candidates that are plausible and lexically close, but in fact imply opposite claims. We describe the dataset with 1970 instances that we built for the task, and we outline the 21 computational approaches that participated, most of which used neural networks. The results reveal the complexity of the task, with many approaches hardly improving over the random accuracy of ≈ 0.5. Still, the best observed accuracy (0.712) underlines the principle feasibility of identifying warrants. Our analysis indicates that an inclusion of external knowledge is key to reasoning comprehension.

1 Introduction

When we argue in natural language, we give reasons as premises for our claims. A fundamental pragmatic instrument in this regard is to leave those parts of an argument unstated that can be presupposed. This is particularly common for the reasoning between an argument's premises and its claim, called implicit *warrants* there (Toulmin, 1958). A warrant takes the role of an inference rule, i.e., the abstract structure of an argument is *reason* $\rightarrow$ (since) *warrant* $\rightarrow$ (therefore) *claim*. In principle, this structure applies to deductive arguments, which allows us to validate arguments properly formalized in propositional logic. However, most natural language arguments are in fact inductive (Govier, 2010) or defeasible (Walton, 2007).

Topic: Tax Break for Sports.

Additional Information: Should pro sports leagues enjoy nonprofit status?

Premise (Reason): Government is already struggling to pay for basic needs.

And since

- ✔ **Warrant 0:** the government isn't required to pay for all the country's needs
- ✗ **Warrant 1:** the government is required to pay for the country's needs

Claim: Sport leagues should not enjoy nonprofit.

Figure 1: Instance of the argument reasoning comprehension task. The correct warrant has to be classified.

Now, when we comprehend an argument, we reconstruct its warrant driven by the cognitive principle of relevance (Wilson and Sperber, 2004). What is easy for humans in many cases, however, turns out to be hard for machines, because reasoning usually depends on context and common sense. In (Habernal et al., 2018), we have thus introduced the *argument reasoning comprehension task* in order to study the construction and identification of implicit warrants for natural language arguments. It forms the basis of the shared task presented here:

Task Given an argument with a *reason* serving as a premise for a *claim*, along with the *topic* and some *additional information* of the discussion they occur in, identify the correct warrant among two opposing candidates, *warrant0* and *warrant1*.

With opposing, we here mean that the two candidate warrants actually imply contradicting claims, the correct one and its opposite. An instance of the task is shown in Figure 1. Being a binary classification task, the main evaluation measure of argument reasoning comprehension is accuracy.

To our knowledge, this is the first shared NLP task directly targeting argumentation; others tasks have only been sketched so far (Kiesel et al., 2015).

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 763–772
New Orleans, Louisiana, June 5–6, 2018.

A solution to our task will represent a substantial step towards automatic warrant reconstruction, which in turn is important for the general long-term goal of automatic argument evaluation. So far, most research on computational argumentation focused on mining claims and premises from text and assessing their properties. In contrast, filling the gap between claims and premises computationally remains an open issue, due to the inherent difficulty of reconstructing the world knowledge and reasoning patterns in arguments (Feng and Hirst, 2011; Green, 2014; Boltužić and Šnajder, 2016). Previous tasks have dealt with the textual entailment of a hypothesis from a proposition (Dagan et al., 2009) or with semantic inference (Bowman et al., 2015). While understanding semantics is important in the given task, argumentation also reasoning beyond what is understood, i.e., pragmatics.

As a basis for the shared task, we built a new dataset with 1970 instances based on authentic English arguments, whose concept and construction process is detailed in Section 2. We outline the systems that participated in the task in Section 3. Most systems implement a computational approach that employs one or more neural networks (often LSTMs, often with attention) based on different pre-trained embedding models. We then present the results of all systems on the test set of the shared task in Section 4 and analyze specific cases in Section 5, before we finally conclude (Section 6).

2 Dataset

This section presents the dataset with all instances used in the shared task. We summarize the main points from its construction process, which is described in detail in (Habernal et al., 2018).

2.1 Task Instances

Let R be the reason for the claim C in a natural language argument. Then there is a warrant W that explains why R supports C, but W is left implicit. For example, if C is "It should be illegal to declaw your cat" and R is "They need to use their claws for defense and instinct", then W could be specified as 'If cat needs claws for instincts, declawing would be against nature' or similar.

The question is how to find a warrant W for a given reason R and claim C. To obtain candidate warrants systematically for our dataset, we propose a trick. In particular, we first construct an *alternative warrant AW* that explains why R may serve as support for the opposite $\neg C$ of the claim C. For the example above, we invert C to "It should be *legal* to declaw your cat" ($\neg C$). $\neg C$ may be explained based on R quite plausibly with the alternative warrant "Most house cats don't face enemies" (AW). Analog to C and $\neg C$, we then invert AW to "Most house cats face enemies", which is a plausible warrant W for the original reason-claim pair (R, C).

Unit	Text
Reason	Cooperating with Russia on terrorism ignores Russia's overall objectives.
Claim	Russia cannot be a partner.
Warrant0	Russia has the same objectives of the US.
Warrant1	Russia has the opposite objectives of the US.
Reason	Economic growth needs innovation.
Claim	3-D printing will change the world.
Warrant0	There is no innovation in 3-d printing since it's unsustainable.
Warrant1	There is much innovation in 3-d printing and it is sustainable.
Reason	College students have the best chance of knowing history.
Claim	College students' votes do matter in an election.
Warrant0	Knowing history doesn't mean that we will repeat it.
Warrant1	Knowing history means that we won't repeat it.

Table 1: Three example task instances from the dataset. In all cases, *warrant1* is the alternative warrant. For brevity, we omit the topic and additional information.

Constructing a plausible alternative warrant is not always possible, as many reasons already convey the arguer's stance. If it is, however, W and AW usually capture the core of a reason's relevance and reveal the implicit presuppositions, due to the trick we performed for construction. For such as cases, we define an instance of our task as a 6-tuple:

Instance *(reason, claim, warrant0, warrant1, topic, additional information)*

The question to be answered is whether *warrant0* is W and *warrant1* is AW, or vice versa. As context, we provide a short *topic* specification and some *additional information* describing the topic. Figure 1 has already shown an example. Further are given in Table 1. They all result from the following process.

2.2 Data Acquisition and Annotation

To obtain a dataset with a permissive license, we decided to build a new dataset from scratch. As source data, we used user-generated web comments from the well-moderated *Room for Debate* of the New York Times, which covers arguments on a variety of contemporary controversial issues.[1]

[1] https://www.nytimes.com/roomfordebate

We manually selected 188 debates with polar questions in the title from a six-year span (2011–2017). We converted each question into a claim C (e.g., "It should be illegal to declaw your cat") and derived a directly opposing claim $\neg C$ ("It should be legal to declaw your cat"). Then, we crawled all comments from the debates and sampled about 11,000 high-ranked, root-level comments[2] from which 5,000 were selected randomly as a basis for the dataset construction. Each comment was split into elementary discourse units using SistaNLP (Surdeanu et al., 2015). To obtain task instances, we then performed the following eight-step crowdsourcing process using Amazon Mechanical Turk:

1. Stance Annotation. For each comment, the crowdworkers first classified what stance it takes, if it remains neutral, or if it does not take any stance.

2. Reason Span Annotation. In all 2,884 comments taking a stance, the workers then marked sequences of discourse units that give a reason for the claim.

3. Reason Gist Summarization. In this step, the workers rewrote all 5,119 marked reasons (2,026 within arguments), such that their gist remains the same but the clutter is removed. The result is a reason R for the claim C.

4. Reason Disambiguation. In order to ensure that *R implies C* really holds, the workers next decided whether C or $\neg C$ is more plausible for R, or whether both are similarly (im)plausible. We kept only those 1,955 reason-claim pairs where workers agreed that C is most plausible.

5. Alternative Warrant. This step was the trickiest. As in the example above, the workers had to specify a plausible alternative warrant AW, explaining why R implies $\neg C$, or declare that impossible.

6. Alternative Warrant Validation. Afterwards, other workers validated each of the 5,342 specified alternative warrants AW as to whether it actually relates to R, by identifying R among two alternatives: R itself and the lexically most similar reason from the same debate topic. For the 3,791 correctly validated cases, we let workers score how logical AW is (0–2) and only kept those 2,613 that had a mean score of at least 0.68. This threshold was chosen based on a manual examination of the scores.

[2]We removed 'noisy' candidates based on several indicators, such as the absence of quotations or URLs and certain lengths. We did not check any quality criteria of arguments, though, because this was not our focus; see, for instance, (Wachsmuth et al., 2017) for argumentation quality.

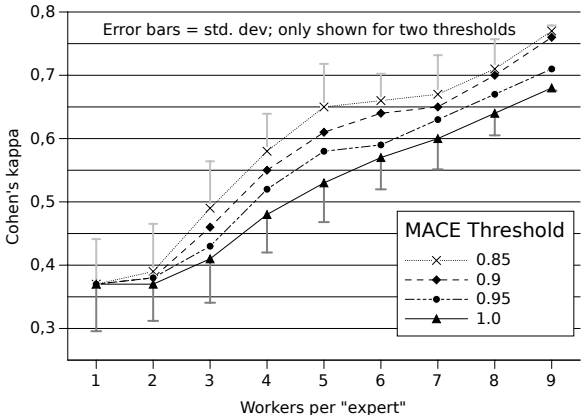

Figure 2: Cohen's κ agreement for stance annotation on 98 comments. As a trade-off between the number of kept instances and their reliability, we chose five annotators and a threshold of 0.95 for this task, which resulted in $\kappa = 0.58$ (moderate to substantial agreement).

7. Warrant For Original Claim. Given R and C, workers then should create a minimally modified version of each AW that may serve as an actual warrant W for C (as in the second half of the example above). They succeeded to do so in 2,447 cases.

8. Warrant Validation. To ensure that only W is correct for R and C, all tuples (R, C, W, AW) were validated again. Unclear cases were resolved by an expert. We obtained 1,970 instances of the argument reasoning comprehension task, so 1,970 pairs of *warrant0* and *warrant1* for a reason and a claim, along with a topic and the additional information.

2.3 Agreement

To assess quality in the crowdsourcing process, we relied on MACE (Hovy et al., 2013), which estimates gold labels for a set of workers, outperforming simple majority votes. Given the number of the different crowdsourcing tasks and their variety, here we only demonstrate the first step, namely stance annotation. We collected 18 assignments per item and split them into two groups (9+9) based on their submission time. We then considered each group as an independent experiment and estimated gold labels for each group. Having two independent "experts from the crowd" allowed us to compute standard agreement scores. We also varied the size of each group from 1 to 9 by repeated random sampling of assignments, and we tuned the MACE threshold for keeping only the most confident predictions. Figure 2 shows the Cohen's κ agreement for stance annotation with respect to the crowd size computed by our method. The decision what number of workers per task to take (five in case of

stance annotation) implies a trade-off between the number of instances and their reliability. We performed similar quality measures with reasonable agreement for the other crowdsourcing steps too. Details are given in (Habernal et al., 2018).

2.4 Datasets in the Shared Task

For the shared task, we split the 1,970 instances into three sets based on the year of the debate they were taken from: 2011–2015 became the training set (1,210 instances), 2016 the development set (316 instances), and 2017 the test set (444 instances). This follows the paradigm of learning on past data and predicting on new ones. In addition, it removes much lexical and topical overlap. The same split has been used by Habernal et al. (2018).

The shared task had two phases, trial and test. In the trial phase, the training and development set were given, both with gold labels stating the correct warrant for all instances. In the test phase, all three datasets were available. Naturally, no labels were given for the test set instances. All provided data is licensed under Creative Commons-family license.

3 Approaches

This section briefly summarizes the computational approaches of the systems that participated in the shared task as well as baselines. Intuitions and detailed explanations are given in the system description papers associated to the shared task.

3.1 Participating Systems

The following 20 systems participated in the shared task, sorted alphabetically. In addition, a 21st system called *Joker* took part, but the team behind did not provide any description. For many of the systems, many more details are given in the respective SemEval-2018 system description papers.

ArcNet uses GloVe embeddings and an LSTM encoder to get the semantic representation of each input (*reason*, *claim*, and both *warrants*). Then an attention mechanism aligns the *reason* and the *warrant* so that the reason-aware warrant representation is generated. Finally, a bilinear function matches the claim with the reason-aware warrant. The network is trained to minimize margin loss. The submission was based on an ensemble model of 10 training runs with the identical architecture.

ArgEns-GRU votes a majority on an ensemble of the following three systems: First, a shared GRU network that learns one representation of the *reason*, *claim*, and both *warrants* each, initialized with 100-dimensional GloVe embeddings. Its output is concatenated and passed through a softmax layer for the final predictions. Second, an extension of the GRU with an attention on the *reason*, *claim*, and both *warrants* each. And third, another GRU model extended with negation and polarity features.

ART uses a bi-directional LSTM with an attention mechanism on top, followed by a multi-layer perceptron network.

blcu_nlp not only pays attention to the consensual part between each warrant and other information, but also to the contradictory part between two warrants. On the model's input (GloVe embeddings), *warrant0*, *claim*, *reason*, and *debate info* are concatenated in order to put attention on *warrant1*. An analog structure is used for the attention on *warrant0*. After obtaining two vectors 'attented_w0' and 'attented_w1' — referring to the ESIM model (Chen et al., 2017) — the two warrants are aligned. A similarity matrix helps to highlight the consensual and the contradictory part. The decision is then drawn after passing through feed-forward layers. A majority voting strategy is used in the final ensemble, which is based on five models performing best on the development data.

Deepfinder shares one LSTM layer for *warrant0*, *warrant1*, *claim*, and *reason*, while the *topic* part uses one LSTM alone. All of them share the same word embedding layer before LSTM layers. After that, one individual dot product is computed for the output of the *warrant0* LSTM and each of the *claim*, *reason* and *claim* (the same is done for the *warrant1* LSTM). The resulting dot products are concatenated and fed into a softmax layer.

ECNU modifies the baseline intra-warrant attention (Habernal et al., 2018) by using a CNN and an LSTM for representing each sentence (*claim*, *reason*, *debate*, *warrant0*, and *warrant1*). Different parts of *warrant0* and *warrant1* are used as an attention vector to obtain representations of the warrants. Similarly, different parts of *claim* and the opposite *claim* serve as attention for the final representation. The final decision is then given by a vote from three networks.

GIST uses pretrained word2vec embeddings as well as the ESIM model (Chen et al., 2017), trained on the SNLI (Bowman et al., 2015) and MultiNLI

(Nangia et al., 2017) datasets. The parameters have been frozen afterwards. Then, pairs of sentences are fed into the the ESIM model. For *warrant0*, for example, these pairs are *(claim, warrant0)*, *(warrant0, reason)*, and *(warrant0, warrant1)*. Also, another bi-LSTM module encoding *claim*, *warrant*, and *reason* is added. The output vectors of each pair and the bi-LSTM are concatenated after averaging and max pooling, and the final prediction is made through feed-forward layers.

HHU encodes *reason*, *claim*, and *warrants* using a bi-directional LSTM. Next, *warrant0*, *reason*, and *claim* are fed into another LSTM; similarly, *warrant1*, *reason*, and *claim* to another LSTM in parallel. Both branches are followed by a dropout and two common dense layers. Embeddings have been pre-trained in four different flavors: fasttext-embeddings trained on the entire Wikipedia corpus, two embeddings trained on the task's dataset using the word2vec skip-gram model with different dimensionalities, and another word2vec model based on the tasks vocabulary but augmented with related articles from Wikipedia. For all embeddings, different parameter combinations and seeds were used to train an ensemble of 623 models in total.

ITNLP-ARC first encodes sentences (*warrant*, *reason*, *claim*) using LSTMs. Attention is used to merge the *reason* vector with the *claim* vector. A shared weight matrix then holds the relationship between the *warrant* and the attention vector, from which the maximum is chosen as the answer. An ensemble method is used for the final vote.

lyb3b encodes sentences using word2vec or GloVe embeddings and a bi-directional LSTM. The instances are treated as positive or negative, depending on the correct training *warrant*. The network then combines the *warrant* with the *reason*, *claim*, and *additional info*. Finally, a fully-connected layer is used to decide whether the instance is correct.

mingyan performs a word-by-word attention that is fused with the original representation then. Self-attention pooling produces a single vector fed into a sigmoid function, trained with cross-entropy loss.

NLITrans attempts to leverage the transfer of semantic knowledge from a bi-directional LSTM encoder with max pooling trained on the MultiNLI corpus (Nangia et al., 2017). This yields a small performance boost on the development set. All sentences (*claim*, *reason*, *warrant0*, and *warrant1*) are encoded with this a transferred encoder. Then, task-specific representations of two 'arguments', one for each warrant, are learned via fully-connected layers. A final linear layer generates an independent score representing the fit of each warrant to the argument. These are concatenated and passed through softmax to generate a probability distribution over the two warrants.

RW2C uses two neural networks. The first one classifies each *warrant* as true or false separately and chooses the one with higher confidence as the right one. The second model makes a decision given two warrant candidates. The final prediction is an ensemble over the previous predictions. Both models represent sentences using a CNN.

SNU_IDS decides whether a logic built on a set of given sentences (*claim*, *reason*, and *warrant*) is plausible. It accepts only one *warrant* at a time and outputs a score on the warrant's validity. The intuition is that the model can learn what has more meaningful semantics of natural language when it judges whether the logic of the given sequence is correct, instead of just selecting the more probable warrant among two candidates. The model consists of an encoding layer with GloVe embeddings (Pennington et al., 2014) and a CoVe sentence encoder (McCann et al., 2017), a 'localization' layer (a set of fully connected layers), and output layers that combine calculating several arithmetic measures over the input representation and compute a final score using a logistic layer on top.

TakeLab preprocesses sentences from the dataset, applies some arithmetic, converts them to Skip-Thought vectors, and feeds them into an SVM classifier with fine-tuned hyperparameters. The Skip-Thought vectors are sentence representation vectors whose encoder and decoder (with an identical structure to RNN encoder-decoders used for neural machine translation) are trained on a large corpus of books unbiased in domain (Kiros et al., 2015).

TRANSRW learns the semantic representation of sentences (*reason*, *warrants*, *claim*) using a convolutional neural network. The assumption behind is that a composition of the *reason* and the *warrant* is close to the representation of the *claim*.

UniMelb combines 3 stacked LSTMs, one for the *reason*, one for the *claim*, and one shared Siamese Network for the two *warrants* under investigation. It generates semantic feature vectors that

serve as input to a shared compressed feature space by using simple vector operations and semantic similarity classification to enforce the interrelationships between them. In doing so, the aim is to learn a form of "generative implication" through the semantic feature vectors. The vectors are able to correctly encode the interrelationships between a reason, a claim, and both the correct and incorrect warrants. The given data is augmented by utilizing WordNet synonym fuzzing.

YNU-HPCC uses a bi-directional LSTM with attention whose input is divided into three parts (*claim*, *reason*, and both *warrants*). To prevent overfitting, dropout is added before the final layer.

YNU_Deep combines the *reason* and the *claim* with a so-called 'story' feature. The story feature is merged with the *warrant*. The network is a bi-directional LSTM with attention and uses GloVe embeddings. Ensemble technology is put on top to mitigate the small size of the data.

ztangfdu first concatenates the *claim* and the *reason* as one sentence named 'sent1', and denotes the correct *warrant* as 'sent2' and the wrong *warrant* as 'sent3', respectively. The output of an LSTM layer with non-trained embeddings then represents each of the sentences. After applying mean pooling to transform the output matrices to vectors, two fully connected layers cater for obtaining the difference score between 'sent2' and 'sent3', whose minimization is the core of the loss function.

3.2 Baselines

For the official task, we provided only a simple naïve random baseline. The outcome (*warrant0* or *warrant1*) is drawn from a Bernoulli distribution ($\theta = 0.5$) resulting in a theoretical accuracy of 0.5. The reported baseline was a single random draw.

Further computational baseline approaches, such as a language model, are evaluated in (Habernal et al., 2018), but we did not consider them within the official competition. There, we also report human bounds for argument reasoning comprehension based on a crowdsourcing study, where each of 173 participants had to solve 10 instances. The mean accuracy was 0.798, but varied depending on the participants' prior knowledge of reasoning, logic, and argumentation. Those with extensive prior knowledge achieved 0.909, and 30 participants solved all instances correctly. We conclude that the task is reasonably solvable for humans.

Rank	System	Accuracy
1	GIST	0.712
2	blcu_nlp	0.606
3	ECNU	0.604
4	NLITrans	0.590
5	Joker*	0.586
6	YNU_Deep	0.583
7	mingyan	0.581
8	ArcNet	0.577
8	UniMelb	0.577
10	TRANSRW	0.570
11	lyb3b	0.568
12	SNU_IDS	0.565
13	ArgEns-GRU	0.556
14	ITNLP-ARC	0.552
15	YNU-HPCC	0.550
16	TakeLab	0.541
17	HHU	0.534
18	**Random baseline**	0.527
19	Deepfinder	0.525
20	ART	0.518
21	RW2C	0.500
22	ztangfdu	0.464

Table 2: Final results of the competition. For the star-denoted system, no description has been provided.

4 Results

The final accuracies of all participating systems are ranked in Table 2. Due to the limited size of the test set (444 instances) and the subtle accuracy differences of many systems, we also measured significance using the approximate randomization test, as described in (Riezler and Maxwell, 2005).[3] Table 3 shows p-values of all system pairs, including the random baseline. As p-values lower than 0.05 are usually considered statistically significant, only three systems outperform the random baseline. However, we would like to emphasize that drawing a strong conclusion about superiority of a particular neural-based system given only one benchmark value might be misleading, as Reimers and Gurevych (2017) showed for several NLP tasks.

We see that the winning system GIST significantly outperforms all other systems on this particular test data ($p \ll 0.05$). For future SemEval tasks, however, we encourage task organizers to solicit multiple submissions of the same system trained with different random initializations, and perform a proper Bayesian system comparison. The machine learning community has already abandoned the controversial p-value and replaced it with Bayesian methods that are easily interpretable and account well for uncertainty (Benavoli et al., 2017).

[3]The implementation of the complete task evaluation is available at https://github.com/habernal/semeval2018-task12-results.

	GIST	blcu_nlp	ECNU	NLITrans	YNU_Deep	mingyan	ArcNet	UniMelb	TRANSRW	lyb3b	SNU_IDS	ArgEns-GRU	ITNLP-ARC	YNU-HPCC	TakeLab	HHU	Random bsl.	Deepfinder	ART	RW2C	ztangfdu
GIST	**.71**																				
blcu_nlp	.00	**.61**																			
ECNU	.00	1.0	**.60**																		
NLITrans	.00	.59	.67	**.59**																	
YNU_Deep	.00	.42	.47	.85	**.58**																
mingyan	.00	.39	.45	.80	1.0	**.58**															
ArcNet	.00	.25	.34	.64	.84	.90	**.58**														
UniMelb	.00	.33	.37	.69	.87	.94	1.0	**.58**													
TRANSRW	.00	.25	.29	.55	.71	.76	.88	.87	**.57**												
lyb3b	.00	.12	.17	.38	.54	.62	.74	.80	1.0	**.57**											
SNU_IDS	.00	.13	.13	.37	.50	.60	.70	.74	.94	1.0	**.57**										
ArgEns-GRU	.00	.09	.08	.23	.31	.41	.47	.52	.71	.71	.78	**.56**									
ITNLP-ARC	.00	.03	.05	.11	.15	.21	.19	.40	.58	.47	.63	.92	**.55**								
YNU-HPCC	.00	.02	.03	.12	.17	.22	.24	.35	.54	.35	.54	.86	1.0	**.55**							
TakeLab	.00	.02	.03	.09	.16	.18	.21	.28	.37	.39	.42	.63	.73	.80	**.54**						
HHU	.00	.00	.01	.03	.03	.04	.04	.12	.23	.10	.18	.41	.39	.49	.87	**.53**					
Random bsl.	.00	.03	.03	.08	.11	.13	.16	.17	.23	.25	.30	.42	.50	.54	.74	.89	**.53**				
Deepfinder	.00	.00	.00	.02	.03	.04	.04	.07	.14	.06	.09	.25	.26	.27	.64	.75	1.0	**.52**			
ART	.00	.00	.00	.00	.01	.01	.01	.03	.07	.00	.04	.15	.10	.10	.47	.44	.84	.83	**.52**		
RW2C	.00	.00	.00	.00	.01	.01	.01	.02	.01	.02	.03	.07	.07	.10	.20	.24	.47	.45	.58	**.50**	
ztangfdu	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.01	.00	.07	.02	.02	.24	**.46**

Table 3: p-values obtained by running the approximate randomization test among all systems. For convenience, the diagonal (bold values) shows the accuracy of each system as in Table 2 but rounded to two decimal numbers. Only the three top systems (GIST, blcu_nlp, and ECNU) are significantly better than the random baseline (p-values < 0.05). The first system (GIST) also significantly outperforms the second system (blcu_nlp) (p-value $\ll 0.05$).

5 Analysis

First, we show a quantitative analysis of the results on the test instances. Figure 3 displays the distribution of all instances over the number of systems that classified each of them correctly. The shape of this rather bi-modal distribution reveals that there are both easy and hard cases. In particular, there are 13 instances completely unsolved and about 90 instances solved by fewer than five participating systems. On the other hand, 32 instances were solved by all systems.

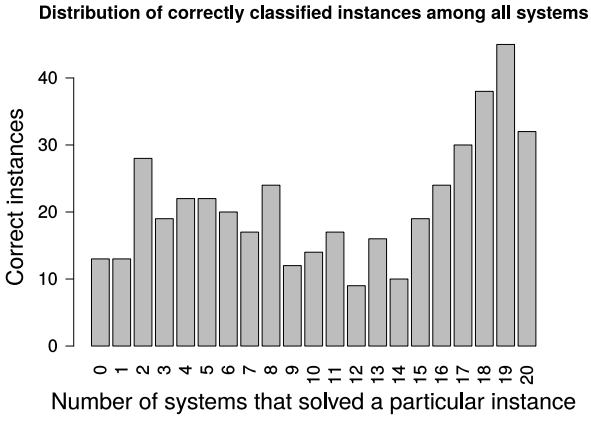

Figure 3: Despite many solvable instances (centered around the right mode), there are hard cases that most systems were not able to cope with (the left mode).

5.1 Easy instances

We qualitatively investigated instances that were classified correctly by all participating systems. It turned out that systems needed to learn only one single property common to all of them: *negation*. Correct warrants in these instances contain negating words ("not", "don't") or negated modals ("can't", "wouldn't"), as shown in Figure 4. This artifact originates from the process of intentionally creating the dichotomy between the alternative warrant and warrant (see Section 2) that in many cases consist of an assertion firstly created for the alternative warrant, and its negation for the correct warrant.

5.2 Difficult instances

A similar problem arises for the difficult instances, such as those not solved by any system. We manually analyzed them and found that the opposite of the easy instances caused misclassification here, namely *misleading negation*. In these instances, the correct warrant is a positive assertion while the alternative warrant is negated. It seems that the

Topic: Have Comment Sections Failed?

Additional Information: In recent years, many media companies have disabled them because of widespread abuse and obscenity.

Premise (Reason): Comment sections are just a propaganda device.

And since

✗ **Warrant 0:** propaganda is the grease of the democratic wheels

✔ **Warrant 1:** propaganda is not the grease of the democratic wheels

Claim: Comment sections have failed.

Topic: Does Turkey Still Belong in NATO?

Additional Information: Given President Erdogan's record on human rights and how his focus on the Kurdish minority has interfered with his fight against ISIS, is he a reliable ally?

Premise (Reason): Turkey does not have much in common with the rest of the countries in NATO.

And since

✔ **Warrant 0:** diversity wouldn't be good for NATO

✗ **Warrant 1:** diversity would be good for NATO

Claim: Turkey doesn't belong to NATO

Figure 4: Examples of "easy" instances from the test data solved by all systems, revealing that relying solely on the negation artifact in the correct warrant gives the right answer (IDs: 18247022_132_A104V8NZIQFN2F, 18068301_176_A3TKD7EJ6BM0M5).

Topic: Have Christians Created a Harmful Atmosphere for Gays?

Additional Information: Church-backed efforts to fight L.G.B.T. rights have been blamed for feeding a hateful atmosphere that accommodates attacks on gays.

Premise (Reason): The Bible is not consistent in it's treatment of sex and marriage.

And since

✔ **Warrant 0:** many Christians take the Bible literally

✗ **Warrant 1:** many Christians do not take the Bible literally

Claim: Christians have created a harmful atmosphere for gays

Topic: Is Google a Harmful Monopoly?

Additional Information: European regulators say the company's Android phone blocks rival services.

Premise (Reason): People can choose not to use Google.

And since

✔ **Warrant 0:** they can opt-out from being indexed by their search engine

✗ **Warrant 1:** they cannot opt-out from being indexed by their search engine

Claim: Google is not a harmful monopoly

Figure 5: Examples of "difficult" instances from the test data on which all systems failed. One possibly explanation is the misleading negation contained in these instances (IDs: 18865357_593_A1CF6U3GF7DZEJ, 18362833_247_A1CF6U3GF7DZEJ).

learned negation "feature" then makes the systems fall into the trap; see examples in Figure 5.

This data analysis clearly shows that it is possible to guess some answers right only given their surface or syntactic form, perhaps because such "features" are prevalent in the training data. However, they do not really help to find any underlying connections between the reasons, warrants, and claims. One solution to test for such cases would be to double the test set simply by adding to each instance another one with an opposite claim and switched warrants. From the reasoning perspective, such an instance still makes sense (which is actually a backbone principle of creating our data), but would clearly penalize systems relying on simple features, such as negation.

6 Conclusion

This paper has overviewed the first shared task on argument reasoning comprehension, one of the tasks at SemEval-2018. Being able to identify the correct warrant connecting an argument's reason to its claim automatically, which is the goal of the task, is the first step of understanding the argument's reasoning. We have outlined the dataset used in the task, the participating system, and the performance they achieved. The results have revealed how challenging the task is: Many systems improved only little over the random baseline. At the same time, the accuracy of GIST, the best system in the evaluation, suggests that it is possible in principle to identify warrants computationally.

Our analysis of the results showed that the participating systems were capable to solve cases with discriminative surface features, but failed where exactly these were misleading. The strongest systems relied on models trained on natural language inference corpora, which suggests that external knowledge may be key to argument reasoning comprehension. Still, more research needs to be done in the future to further investigate this hypothesis.

Acknowledgments

This work was supported by the German Research Foundation (DFG) within the ArguAna Project GU 798/20-1, and by the DFG-funded research training group "Adaptive Preparation of Information form Heterogeneous Sources" (AIPHES, GRK 1994/1).

References

Alessio Benavoli, Giorgio Corani, Janez Demsar, and Marco Zaffalon. 2017. Time for a Change: a Tutorial for Comparing Multiple Classifiers Through Bayesian Analysis. *Journal of Machine Learning Research*, 18:1–36.

Filip Boltužić and Jan Šnajder. 2016. Fill the Gap! Analyzing Implicit Premises between Claims from Online Debates. In *Proceedings of the Third Workshop on Argument Mining*, pages 124–133, Berlin, Germany. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for Natural Language Inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668, Vancouver, Canada. Association for Computational Linguistics.

Ido Dagan, Bill Dolan, Bernardo Magnini, and Dan Roth. 2009. Recognizing textual entailment: Rational, evaluation and approaches. *Natural Language Engineering*, 15(Special Issue 04):i–xvii.

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 - Volume 1*, HLT '11, pages 987–996, Portland, Oregon. Association for Computational Linguistics.

Trudy Govier. 2010. *A Practical Study of Argument*, 7th edition. Wadsworth, Cengage Learning.

Nancy L Green. 2014. Towards Creation of a Corpus for Argumentation Mining the Biomedical Genetics Research Literature. In *Proceedings of the First Workshop on Argumentation Mining*, pages 11–18, Baltimore, Maryland USA. Association for Computational Linguistics.

Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2018. The argument reasoning comprehension task: Identification and reconstruction of implicit warrants. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, page (to appear), New Orleans, LA, USA. Association for Computational Linguistics.

Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. 2013. Learning whom to trust with MACE. In *Proceedings of NAACL-HLT 2013*, pages 1120–1130, Atlanta, Georgia. Association for Computational Linguistics.

Johannes Kiesel, Khalid Al Khatib, Matthias Hagen, and Benno Stein. 2015. A shared task on argumentation mining in newspaper editorials. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 35–38. Association for Computational Linguistics.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-Thought Vectors. In *Advances in Neural Information Processing Systems 28*, pages 3276–3284, Montreal, CA. Curran Associates, Inc.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in Translation: Contextualized Word Vectors. In *Advances in Neural Information Processing Systems 30*, pages 6294–6305, Long Beach, CA, USA. Curran Associates, Inc.

Nikita Nangia, Adina Williams, Angeliki Lazaridou, and Samuel Bowman. 2017. The repeval 2017 shared task: Multi-genre natural language inference with sentence representations. In *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP*, pages 1–10, Copenhagen, Denmark. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2017. Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 338–348, Copenhagen, Denmark. Association for Computational Linguistics.

Stefan Riezler and John T. Maxwell. 2005. On some pitfalls in automatic evaluation and significance testing for MT. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 57–64, Ann Arbor, Michigan. Association for Computational Linguistics.

Mihai Surdeanu, Tom Hicks, and Marco Antonio Valenzuela-Escarcega. 2015. Two practical rhetorical structure theory parsers. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 1–5, Denver, Colorado. Association for Computational Linguistics.

Stephen E. Toulmin. 1958. *The Uses of Argument*. Cambridge University Press.

Henning Wachsmuth, Nona Naderi, Ivan Habernal, Yufang Hou, Graeme Hirst, Iryna Gurevych, and Benno Stein. 2017. 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.

Douglas Walton. 2007. *Media Argumentation: Dialect, Persuasion and Rhetoric*. Cambridge University Press.

Deirdre Wilson and Dan Sperber. 2004. Relevance Theory. In Laurence R. Horn and Gregory Ward, editors, *The Handbook of Pragmatics*, chapter 27, pages 607–632. Wiley-Blackwell, Oxford, UK.