Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 5055–5064
Marseille, 11–16 May 2020

Reproduction and Revival of the Argument Reasoning Comprehension Task

João Rodrigues, Ruben Branco, João Silva, António Branco
University of Lisbon
NLX—Natural Language and Speech Group, Department of Informatics
Faculdade de Ciências
Campo Grande, 1749-016 Lisboa, Portugal
{joao.rodrigues, ruben.branco, jsilva, antonio.branco}@di.fc.ul.pt

Abstract

Reproduction of scientific findings is essential for scientific development across all scientific disciplines and reproducing results of previous works is a basic requirement for validating the hypothesis and conclusions put forward by them. This paper reports on the scientific reproduction of several systems addressing the Argument Reasoning Comprehension Task of SemEval2018. Given a recent publication that pointed out spurious statistical cues in the data set used in the shared task, and that produced a revised version of it, we also evaluated the reproduced systems with this new data set.

The exercise reported here shows that, in general, the reproduction of these systems is successful with scores in line with those reported in SemEval2018. However, the performance scores are worst than those, and even below the random baseline, when the reproduced systems are run over the revised data set expunged from data artifacts. This demonstrates that this task is actually a much harder challenge than what could have been perceived from the inflated, close to human-level performance scores obtained with the data set used in SemEval2018. This calls for a revival of this task as there is much room for improvement until systems may come close to the upper bound provided by human performance.

Keywords: Reproduction, Replication, Argument Mining, Argument Reasoning, Argument Comprehension, ARCT, SemEval.

1. Introduction

The ability to repeat experiments and to reproduce their results is a cornerstone of scientific work and it is necessary to properly validate the findings that are published. Several examples of failed replication efforts, however, have been documented in different scientific fields (Branco et al., 2017). For Natural Language Processing (NLP), failure to reproduce results has been reported for WordNet similarity measures (Fokkens et al., 2013), sentiment analysis (Moore and Rayson, 2018), PoS tagging and named entity recognition (Reimers and Gurevych, 2017), among others.

The importance of open challenges that foster the reproduction of research results has been underlined (Fokkens et al., 2013; Branco, 2013) as part of the solution for the so-called *replication crisis* (Hutson, 2018). The community gathered around the science and technology of language has responded through initiatives such as the 4REAL workshops (Branco et al., 2017; Branco et al., 2018), and the forthcoming ReproLang cooperative shared task (ReproLang, 2019) of the LREC2020 conference.

The discussion regarding the reproduction of scientific experiments faces an additional, non-technical difficulty as there is no established consensus on the terminology that should be used, with the terms replicability, reproducibility and repeatability being used interchangeably in the literature. Here we adopt the definitions given by Stodden et al. (2014, p.vii) and followed by the Language Resources and Evaluation Journal (Branco et al., 2017): *Replication, the practice of independently implementing scientific experiments to validate specific findings, is the cornerstone of discovering scientific truth. Related to replication is reproducibility, which is the calculation of quantitative scientific results by independent scientists using the original data sets and methods.*

The goal of this paper is twofold. On the one hand, we report on the challenges faced and the insights gained from our endeavour in reproducing several systems submitted to the SemEval-2018 Task 12, the Argument Reasoning Comprehension Task (ARCT) (Habernal et al., 2018b), where the top 3 systems, with very good, close to human performance, were included. In a nutshell, the task consists of a binary decision among two input candidate sentences of which only one is fit to be a premise in the input argument. We were able to reproduce most systems, though with varying degrees of difficulty, and provide a detailed report for each case.

On the other hand, we take notice of a problem that was found by Niven and Kao (2019) in the original data set of ARCT when it was used with BERT (Devlin et al., 2019), and of spurious statistical cues that have been shown to bias the results obtained. As a cleaned ARCT data set that fixes this bias problem has been released, we also run the reproduced systems on this revised data set.

Given the sharp drop in performance that resulted — to very modest scores below to naive random choice baseline —, the present paper leads to a reassessment of the state-of-the-art for this task. While highlighting the importance of reproducing previous work, we believe it will help also to foster the revival of the ARCT task.

2. Related Work

Scientific reproduction. The challenges raised by attempts to reproduce scientific work have been increasingly documented in recent years (Raff, 2019). One could expect that Language Technology would face fewer challenges than other scientific fields, such as Biology or Physics, given that the works are by and large computationally based, but that does not seem to be the case, as several

computation specific challenges accrue to existing general challenges. Additionally, given that Machine Learning has been a prevalent approach in NLP for many years now, further challenges to a successful reproduction have their root also there, given a large number of approaches available, which nowadays are often based on neural architectures, and the large search space for hyper-parameters (Lucic et al., 2018).

Artificial Intelligence (AI). In a recent reproducibility analysis of research in Artificial Intelligence (Gundersen and Kjensmo, 2018), covering a total of 400 research articles in the top-level IJCAI[1] and AAAI[2] conferences, it was reported that only about 20% to 30% of the reproducibility indicators in the assessed articles were documented. The research articles were quantified as to their reproducibility based on a list of factors that are good indicators of reproducibility, namely the method, the data and the experiment. Using these indicators, Gundersen and Kjensmo (2018) reports that most of the research works were found to be irreproducible, although improvements were observed over time.

Another reproducibility study in the fields of Artificial Intelligence and Machine Learning was presented in (Raff, 2019). The author attempted to reproduce several Machine Learning algorithms without using the released code, following only the descriptions of the algorithms provided in the papers, and recorded a total of 26 attributes related to reproducibility for each paper. Raff (2019) categorizes these attributes into three levels of subjectivity, namely (i) objective attributes such as the number of authors, the number of pages and the year of publication; (ii) mildly subjective attributes such as the number of tables, equations, proofs, whether the hyper-parameters are specified and whether the data is made available; and (iii) subjective attributes such as the use of toy problems, paper readability and algorithm difficulty, among others. The attributes that showed a significant positive impact on raising the level of reproducibility were: existence of a formal proof, paper readability, algorithm difficulty, availability of pseudo-code, primary topic, whether the hyper-parameters are specified, amount of computation needed, whether the authors reply to questions, number of equations and number of tables.

Machine Learning. Upon assessing the comparability of different Machine Learning systems, Dodge et al. (2019) claim that simply reporting the score on the test set, as it is common practice, is insufficient to deem some systems better than others. In order to address this issue, a reproducibility checklist (based on the NeurIPS Machine Learning Reproducibility Checklist[3]) is presented along with a metric to measure the expected validation performance. Through this checklist, it is possible to arrive at recommendations to enhance the level of reproducibility and replicability. Regarding the reproducibility of experimental results, it is recommended to include the description of the computing infrastructure, average run-time for each approach, details of data splits, validation performance for each reported test result and the source code. Regarding the replicability of the experiments with hyper-parameter searches, it is recommended to include the search bounds, best hyper-parameters, number of search trials, method of choosing and selecting the hyper-parameters and the expected validation performance.

In the assessment reported below in the present paper, we use metrics based on these recommendations.

Language Technology. Regarding reproducibility in NLP, a cross-cutting analysis of papers presented at the EMNLP[4] 2018 conference (Dodge et al., 2019) showed a widespread, positive practice of reporting the best hyper-parameter settings (74%) and the data splits used (92%). However, only 8% of the papers describe the hyper-parameter search bounds, 14% the search strategy, and only 30% provide the source code. Given the importance of this information for the successful reproduction of a given work, it is safe to say that Language Technology is not exempt from its reproducibility bottlenecks.

Argument reasoning comprehension. Niven and Kao (2019) showed that some data artifacts in the ARCT data set (described in detail in Section 5.) have a deeper than expected impact on the accuracy of a BERT system trained over it. A state-of-the-art BERT (Devlin et al., 2019) model was trained on the original ARCT data set and on a revised version of that data set, without the spurious linguistic cues. Without the linguistic cues, BERT performs as well as a random guess, which casts doubts on the reliability of the results previously achieved on the ARCT shared task.

Like in (Niven and Kao, 2019), we resort here to the ARCT revised data set, expunged from the data artifacts that drift towards otherwise unjustified enhanced accuracy. And we perform a new assessment of the performance in solving this task, not with BERT, but using the ARCT systems reproduced.

3. Argument Reasoning Comprehension

The recent Argument Reasoning Comprehension Task (ARCT) was part (Task 12) of the SemEval-2018 Workshop on Semantic Evaluation (Habernal et al., 2018b), collocated with the ACL 2018 conference.

The ARCT can be considered a sub-task of argument mining, which is a broader task whose goal is to automatically identify argumentative structures within natural language expressions.

What constitutes an argumentative structure varies according to different argumentation theories, but argument mining techniques frequently structure arguments in two parts, viz. the *claim* and the *premise* (also referred to as reason). A claim is a concise statement that supports or contests a given topic, while a premise is a concise statement that provides evidence to support a claim. A third component may also be included, the *warrant* (also referred to as inference or argument). The warrant is not always explicitly present as it is often left as an implicit premise, and in informal logic it corresponds to the notion of enthymeme.

[1] International Joint Conference on Artificial Intelligence

[2] Association for the Advancement of Artificial Intelligence

[3] https://www.cs.mcgill.ca/~jpineau/ReproducibilityChecklist.pdf

[4] Empirical Methods in Natural Language Processing

In the ARCT task, an argument, composed by a claim and a premise, is provided together with two candidate warrants — a correct warrant and an alternative warrant that supports a claim that is contradictory to the claim in the input argument. The goal is to choose the correct warrant for the argument provided. One of the difficulties of this task is that, even when supplying candidate warrants, one may still need to rely on world knowledge to make a correct decision.

An example of input taken from the ARCT data set is the following: Given the claim *TFA*[5] *has not raised the status of teachers* and the premise *TFA has high turnover by design and is expensive*, choose the correct warrant among the two options (a) *TFA still enjoys a good reputation* and (b) *TFA does not enjoy a good reputation*. In this particular example, warrant (a) is annotated in the ARCT data set as being the correct one.

3.1. The data set

The creation of the ARCT data set is described in detail in (Habernal et al., 2018a). Since understanding this creation process is crucial to also understand how the spurious correlations present in the data set have arisen, here we repeat that description to a large extent. The presentation of the spurious data artifacts is left for Section 5.

The data set is based on authentic English arguments from debates extracted from formally written comments posted in the *Room for Debate* on-line debate section of the New York Times newspaper. Debates were manually selected under criteria that looked for polar topics and debates with room for argumentation. For each debate, (i) two opposing claims were manually created, (ii) premises for and against the claims were obtained by summarizing sampled comments, and finally (iii) warrants were manually created.

For example, given the debate topic *Should Foreign Language Classes Be Mandatory in College?*, accompanied by the contextual debate information *A Princeton University proposal would require students to study a language other than English, even if they are already proficient in a foreign language*, the following two opposing claims were manually created:

C_a *Foreign language classes should be mandatory in college*

C_b *Foreign language classes shouldn't be mandatory in college*

Premises were created from the comments in the debate. Comments, randomly sampled from the debate forum, were classified by crowdsourced annotators according to their stance regarding the provided claims — for or against — and summarized into concise statements (premises). Concerning the two claims above, examples of the corresponding premises could be the following:

P^a *Foreign language skills are rewarded in the job market*

P^b *It is hard for an adult to learn a second language*

where premise P^a supports claim C_a and premise P^b supports claim C_b.

[5]Teach For America (TFA) is a nonprofit organization for the education equality movement.

ARCT data set	original	revised
train	1,210	2,420
development	316	632
test	444	888

Table 1: Number of instances for the train, development and test splits regarding the original data set for ARCT and the subsequently revised data set.

Next, the implicit warrants for and against a claim were created. This process starts by creating what the authors call the alternative warrant, which is a plausible explanation as to why a premise supporting a claim can, in fact, be seen as supporting the opposite claim.

Given, for example, premise P^a *Foreign language skills are rewarded in the job market* from above and taking into account the opposing claim C_b *Foreign language classes shouldn't be mandatory in college*, the annotators wrote down alternative warrants that support claim C_b, such as W^b *this does not apply to every job and might not be important to some industries*. Put together into a full argument, these elements would come out as "R^a, but since W^b, we can claim that C_b", that is *Foreign language skills are rewarded in the job market*, but since *this does not apply to every job and might not be important to some industries*, we can claim that *foreign language classes shouldn't be mandatory in college*.

Each fabricated alternative warrant was validated by showing to other annotators the following: the fabricated alternative warrant, the claim it supports, and two alternative premises, the premise also supporting that claim and a distracting premise $P^?$ from the same debate topic:

$P^?$ *We should be able to speak other languages rather than expect everyone else to speak English*

P^b *It is hard for an adult to learn a second language*

W^b (and since) *this does not apply to every job and might not be important to some industries*

C_b *Foreign language classes shouldn't be mandatory in college*

The annotators were asked to choose between the correct premise P^b and the distracting premise $P^?$. If the correct premise was chosen then the warrant was validated.

Finally, the actual warrant for the original claim is obtained by performing minimal modifications to the fabricated alternative warrant. For example, given the alternative warrant *this does not apply to every job and might not be important to some industries*, the annotators could write down an actual warrant such as *this applies to every job and might be important to some industries too*.

The resulting data set has 1,970 instances based on 188 debates. Each instance consists of a claim, its supporting premise, and two warrants: the actual warrant and the alternative warrant.[6] The data set was split into 1,210 training instances, 316 development instances and 444 test instances, as presented in Table 1.

[6]Each instance includes some other data, such as an instance identifier, the debate title and the contextual information.

3.2. The participating systems

The shared task had two phases. In the first phase, the trial, the training and development sets with gold labels were made available. In the second phase, the test with the unlabeled test set instances was made available.

Competing with the baseline provided by a naive 50-50 random classifier, 21 systems participated.[7] The ranking and scores of the systems are presented in Table 2.

An upper bound for performance was found by having humans solving the task for 10 instances. A set of 173 crowdsourced participants set human accuracy at 0.798.[8]

The naive baseline of 50-50 random choice achieved an accuracy of 0.527. The top three systems, GIST (Choi and Lee, 2018), BLCU_NLP (Zhao et al., 2018) and ECNU (Tian et al., 2018), achieved an accuracy of 0.712, 0.606 and 0.604, respectively. These results lead the ARCT organizers to conclude that the identification of warrants for arguments is a feasible NLP task.

It is important to note that the organizers mentioned the existence of artifacts in the data set that could bias the classifiers, such as *negation cues*, and provided a solution to fix this problem on the existing corpus. However, these fixes were not implemented for ARCT, and their impact on the results was not fully realised until after the ARCT was concluded. We will come back to this issue in Section 5.

4. Reproduction exercise

In this section, we describe the essential points of the reproduction exercise we undertook. The ARCT participants were asked for a summary of their systems and could also optionally provide a system description paper. A FAQ[9] was provided giving information about what to include in a system description paper. Replicability is explicitly mentioned in the FAQ, where participants are encouraged to "present all details that will allow someone else to replicate your system".

4.1. Selecting what to reproduce

From the 21 participating systems, 13 were accompanied with a description paper, 7 gave only a summary of the system and 1 system (Joker) did not provide any description.

Only 5 participants made the source code available.

Overall, the participating systems show little variability regarding the machine learning methods resorted to, given that all the systems were based on neural networks, except one classifier that used support-vector machines. As for the underlying model, 14 systems used a long short-term memory (LSTM) model, 1 a gated recurrent unit (GRU) model, 2 a convolutional neural network (CNN), and 1 a support-vector machine (SVM) classifier. A total of 12 systems report using pre-trained distributional semantic models, the majority of the systems used the word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) word embeddings.

We confined the reproduction exercise to what appeared as reasonable and feasible given our goals, and taking into account that the objective was not an all-embracing reproduction effort, and that, ultimately, a system is reproducible as long as the authors share enough information about their work.

We undertook a best effort reproduction of a number of systems without contacting the authors of the respective description papers. We selected the three top scoring systems, namely GIST (Choi and Lee, 2018), BLCU_NLP (Zhao et al., 2018) and ECNU (Tian et al., 2018) — all of which fortunately have their source code available — plus the other systems that also made their source code available, namely NLITrans (Niven and Kao, 2018) and SNU_IDS (Kim et al., 2018). We also replicated TakeLab (Brassard et al., 2018), which, while not making its source code available, is the only one using a non-neural approach, namely a support-vector machine.

We also performed a system survey taking into account several reproduction metrics.

4.2. Indicators of reproducibility level

For the quantitative indicators to be used in the survey, we adopted several metrics from Dodge et al. (2019) and found relevant to include also the distribution metric mentioned in (Reimers and Gurevych, 2017). A description of the metrics used can be found below.

Infrastructure indicates whether the hardware infrastructure used for training and evaluating the models is reported in the paper. This allows assessing if it is feasible to reproduce the work with the existing computational resources (GPU, CPU and RAM), and will also help estimate the time necessary to run the experiment.

Empirical run-time indicates whether the time taken to create the models is given in the paper. As with the infrastructure metric, this allows assessing if it is feasible to reproduce the work within the available time frame.

Development accuracy indicates whether the accuracy of the development data, used to fine-tune the model, is given in the paper. Since non-explicit parameters may exist and given that machine learning models based on neural networks are also inherently non-deterministic, modeling the reproduction models accuracy using the validation data set avoids the creation of a model biased to the test set.

Score distribution indicates whether a score distribution is given, instead of a single value. Specifying a score distribution is important given that reporting the test score of a single run is often insufficient to truly assess the performance of a system in terms of generalization, especially for algorithms with some non-deterministic component (e.g. the random initialization of weights on a neu-

[7] GIST (Choi and Lee, 2018), BLCU_NLP (Zhao et al., 2018), ECNU (Tian et al., 2018), NLITrans (Niven and Kao, 2018), Joker, YNU Deep (Ding and Zhou, 2018), mingyan, ArcNet, UniMelb (Joshi et al., 2018), TRANSRW (Chen et al., 2018), lyb3b (Li and Zhou, 2018), SNU_iDS (Kim et al., 2018), ArgEnsGRU, ITNLP-ARC (Liu et al., 2018), YNU-HPCC (Zhang et al., 2018), TakeLab (Brassard et al., 2018), HHU (Liebeck et al., 2018), Deepfinder, ART, RW2C and ztangfdu. Systems that lack a citation did not provide a reference paper.

[8] If participants with extensive prior knowledge of the task are used, human accuracy jumps to 0.909, with 30 participants solving the task perfectly.

[9] http://alt.qcri.org/semeval2018/index.php?id=papers

ral network). Note that some systems that participated in ARCT were ensemble systems. For those, we consider that the result of a single run can nonetheless be taken as a distribution of scores since it involves multiple systems.

Hyper-parameter search bounds indicates whether the hyper-parameter search bounds are given in the paper. When tuning a machine learning algorithm, the impact of hyper-parameter choice on the results is a determinant factor to the overall score. Reporting the range of the search boundaries of each hyper-parameter greatly helps the reproduction process by delimiting the hyper-parameter space that has to be explored.

Hyper-parameter choice method indicates whether the hyper-parameter search method is described in the paper. This is closely tied to the previous metric, as knowing the process that drove the search is important to reproduce the work.

Hyper-parameters for best model indicates whether the hyper-parameter settings used to achieve the best score are given in the paper. Without the best hyper-parameter settings, an identical or even approximate reproduction of the results may not be possible to obtain, given the large hyper-parameter search space.

Hyper-parameter search trials indicates whether the number of trials for finding the best hyper-parameters is given. Given the computational power and time constraints, knowing the number of trials needed can be the difference between knowing if the reproduction is feasible or not given the available infrastructure.

A survey of the description papers and systems in ARCT and how they stand concerning these reproducibility indicators is presented in Table 2.[10] They ensure a suboptimal level of reproducibility — even in the case of the systems we were able to reproduce.

4.3. Systems reproduced

This section describes our reproduction endeavours. We note that all the systems chosen for reproduction presented a reference paper, so we provide only a very short overview of each model, and direct the interested reader to the corresponding paper for more information. Recall that all of the systems chosen for reproduction included source code, except for the TakeLab system which we re-implement based on the description in the paper.

For each reproduction attempt, we have created a report that, when necessary, go into very specific technical details (e.g. adjustments to command-line arguments, package versioning, bug fixing, etc) that are left out of the current paper. These reports may, however, be found at (NLX, 2019).

GIST (Choi and Lee, 2018) was the best system on the ARCT, with 0.712 accuracy, 0.175 over the baseline. The reference paper described the development score, the use of a distributed score and the best hyper-parameters. The system consists of LSTMs neural networks and makes use of transfer learning from the natural language inference corpora SNLI (Bowman et al., 2015) and MultiNLI (Williams et al., 2018). The source code was made available through a public code repository (GitHub). Although the supporting library requirements were described in the repository, the specific versions of the libraries were not reported. This made the reproduction difficult, due to conflicts with the Theano GPU library (Theano Development Team, 2016) which forced us to resort to a CPU-compatible library instead, with which we were able to run the system (though certainly taking much longer than it would have on a GPU). It took one person less than one working day to reproduce this system. Running the system with the default 10 epochs took around 64 minutes.

We obtained a score of 0.714, 0.002 points higher than that reported in ARCT. This is a tiny difference, which we attribute to non-deterministic steps in the process, such as weight initialization. Although the reference paper lacked a description of the infrastructure, the number of trials, and the hyper-parameter bounds and choice method, we consider the GIST system to be reproducible, mainly due to the availability of the source code.

BLCU_NLP (Zhao et al., 2018) was the second-best system on ARCT, with 0.606 accuracy, 0.106 points below the first system. The reference paper reported the development score, the score distribution and the best hyper-parameters. The system is an ensemble of ESIM (Chen et al., 2017) models, which are enhanced LSTM networks that incorporate syntactic information. Source code was made available through a public code repository (GitHub). Although the source code contained a minor problem in its instructions (pointing to a different script for execution) no other problems occur regarding running the source code. The reference paper mentions the use of the best five models for an ensemble majority vote, however, it does not mention the total number of models from which the five best models were selected. We decided to run the system ten times and choose the best five models to reproduce the ensemble. It took one person less than one working day to reproduce this system. Obtaining ten models took less than two hours.

We obtained a score of 0.642, 0.036 points higher than that reported in ARCT. We hypothesize that the number of total models, from which the best five models were selected, is the reason for the difference from the original score and the one we obtained. A possibility is that the authors picked five models from a larger set than ours and the models had higher overfitting to the development set than ours. Some details were missing from the reference paper but that did not prevent us from reproducing the system using the provided source code. As such, we consider the BLCU_NLP system to be reproducible.

ECNU (Tian et al., 2018) was the third-best system on ARCT, with 0.604 accuracy, only 0.002 behind the second-best system. The system is an ensemble of several neural networks, each encoding information using LSTM and LSTM+CNN. The reference paper included development score, score distribution and best hyper-parameters. The Python source code uses Tensorflow (Abadi et al., 2015) and Keras (Chollet and others, 2015), along with related

[10] Note that three metrics are not shown in the table, namely *Infrastructure*, *Empirical run-time* and *Hyper-parameter search trials*, as none of the systems reported them.

ARCT system			Reproduced scores		Replicability survey						
name	rank	acc.	dev. acc.	test acc.	paper	source	dev.	score	bounds	method	best
GIST	1	0.712	0.715	0.714	✓	✓	✓	✓			✓
BLCU_NLP	2	0.606	0.680	0.642	✓	✓	✓	✓			✓
ECNU	3	0.604	0.684	0.583	✓	✓	✓	✓			✓
NLITrans	4	0.590	0.660	0.623	✓	✓		✓	✓	✓	✓
Joker	5	0.586									
YNU Deep	6	0.583			✓		✓	✓			✓
mingyan	7	0.581									
ArcNet	8	0.577									
UniMelb	8	0.577			✓		✓		✓	✓	✓
TRANSRW	10	0.570			✓		✓	✓			✓
lyb3b	11	0.568			✓		✓	✓			
SNU_iDS	12	0.565	0.703	0.543	✓	✓	✓	✓			✓
ArgEns-GRU	13	0.556									
ITNLP-ARC	14	0.552			✓		✓	✓		✓	✓
YNU-HPCC	15	0.550			✓			✓	✓	✓	✓
TakeLab	16	0.541	0.516	0.541	✓		✓		✓		✓
HHU	17	0.534			✓		✓	✓	✓	✓	
baseline	18	0.527									
Deepfinder	19	0.525									
ART	20	0.518									
RW2C	21	0.500									
ztangfdu	22	0.464									

Table 2: The first three columns indicate the system names, their ranking and the test accuracy obtained in ARCT; the fourth and fifth columns report the development and test accuracy obtained in the reproduction exercise; for the remaining columns, a check mark (✓) indicates the presence of that indicator contributing to the reproducibility level. These indicator refer to the existence of: **paper**: a description paper; and in the description paper, the existence of: **source**: public source code; **dev.**: development accuracy score; **score**: score distribution; **bounds**: hyper-parameter search bounds; **method**: hyper-parameter choice method; and **best**: hyper-parameter settings for the best model.

dependencies such as numpy, but these dependencies, and their precise versions, had to be determined by trial and error as the paper and source code documentation did not specify that information and using the most recent versions yielded errors. After several version regressions, we settled for a working Tensorflow (1.0.0) and Keras (2.2.4) version. The Python version was also not reported, thus we used 3.6.9 (the most recent stable version at the time of running our experiment). The system makes use of the Stanford CoreNLP pipeline (Manning et al., 2014) to parse its input but, again, the precise version is not specified and had to be determined through inspection of the source code. The system relies on pre-trained Word2Vec embeddings, but their source was not described so we assumed them to be the standard GoogleNews pre-trained vectors (Mikolov et al., 2013). The source code implements several models, but since the documentation does not specify which are used in the ensemble, these had to be determined by inspecting the source code.[11] The experiments were ran on 2 Intel(R) Xeon(R) Gold 6152 CPU's. We obtained a score of 0.583, 0.021 points below than that reported in ARCT. We hypothesize that the reason for the score difference may lie in the ensemble models criterion. This criterion was not described in the paper, which could have potentially been an ensemble of the best models in the 3 runs. The criterion chosen for this work was to evaluate the ensemble system at each run and calculate the mean of the accuracy scores across the 3 runs. We consider the ECNU system hard to reproduce due to the lack of documentation, taking one person roughly two working days to fully reproduce it.

NLITrans (Niven and Kao, 2018) was the fourth-best system in ARCT, with 0.590 accuracy, 0.014 points behind the third-best system. Its reference paper has one of the most extensive descriptions of reproduction details. It reports a score distribution, the search method, bounds and best settings for the hyper-parameters. The system uses a neural network enhanced with transfer learning using the MultiNLI (Williams et al., 2018) natural language inference data set. The authors made available a reproduction script with a list of experimented models. Nevertheless, the system submitted to ARCT was not available given its large size. The best available pre-trained models were 512-sized (encoder) models, which the authors referred to as t512fwcomp.

A list of required packages and their respective versions were declared, which we installed, however, some needed packages were not included.

The experiments were run on a GeForce GTX 1080 Ti. The pre-trained embeddings used were 840B.300d Glove embeddings, all other hyper-parameters were kept as default, with a new seed being produced each run. As per the paper, the reported accuracy value is an average of 200 runs.

[11] The models used for the ensemble are intra_attention_ii, intra_attention_cnn and intra_attention_cnn_negclaim.

We obtained a score of 0.623, 0.033 points above that reported in ARCT. We hypothesize that the reason for this difference in scores is due to the fact that we did not have access to the NLI Model with a 2048 dimensional encoder size, used in the competition submission as an initialization model (Transfer). Having resorted to the available 512 dimension model, this could be the cause for the difference in scores reported in this work. It took one person two working days to reproduce this system.

SNU_IDS (Kim et al., 2018) took the 12th position in the ARCT with an accuracy of 0.565, which is 0.038 points above the baseline. The reference paper included development score, score distribution and best hyper-parameters. The system uses a neural network with GloVe word embeddings and a CoVe (McCann et al., 2017) sentence encoder and feed-forward layers. We encountered problems with PyTorch versioning, as we did with the NLITrans system reproduction. The experiments were run on a Dell R740 Server with 2 Xeon Gold 6152 CPUs and 256Gb of RAM. We used Python 3.6.2 to run the experiments, as reported in the paper. PyTorch version 0.3.0.post4 was used, along with the necessary packages and respective versions provided in the `requirements.txt` file. We encountered problems in downloading a required torch pretrained model. We ran the SECOVARC-last (w/ heuristics) model, which is the model submitted to the competition, with 840B.300d Glove pre-trained embeddings and all other hyper-parameters were kept as default. The final reported accuracy values are an average of 20 runs. It took one person one working day to reproduce this system.

TakeLab (Brassard et al., 2018) took the 16th place in ARCT with an accuracy of 0.541, which is 0.014 points above the baseline. Although no source code was provided, we decided to replicate this system given that is was the only one that used a non-neural approach, namely a SVM. The model was created by converting the data set to a sentence encoded vector using the Skip-thought vectors (Kiros et al., 2015) and training a SVM to predict the best warrant. We implemented the system from scratch given the descriptions provided in the reference paper, including the text normalization and feature extraction functionality.[12] We faced challenges using the original Skip-thought vectors which are provided by an independent library. Without knowing the specific version of Python used in the original Skip-thought implementation we tried running the library with several Python versions, however, unsuccessfully. As a solution we installed a version of Skip-thought[13] pre-trained models for Pytorch, which uses the original Skip-tough models. Given that no description was given for the number of words from each sentence that were converted to a vectorized representation, we experimented with the length of the maximum tokens and also with a 20 and 50 units length. The exact SVM library used was not specified, so we experimented with Weka (Witten et al., 2016) and the SVM provided in the Scikit-learn (Pedregosa et al., 2011) package, with the hyper-parameters described in the reference paper. The data set was normalized as described and the reported complement solution was implemented. We also used the uni-skip model and the bi-skip model, totaling a 9,600 unit vector as a feature.
It took one person working around two days to replicate this system. Since we obtained an accuracy score equal to the one reported on the reference paper, we consider this system to have been successfully replicated.

[12] We made available the source code for our implementation of the TakeLab system at (NLX, 2019).

[13] `https://pypi.org/project/skipthoughts/`

4.4. Summary of the reproduction exercise

It was possible to reproduce the five systems that made available the respective description paper, source code, development scores and the hyper-parameters for the best model, though some tinkering was necessary due to lack of proper documentation for the reproduction process. Fortunately, we had sufficient computer power and GPU to reproduce the implementation without computer power constraints, rendering the other indicators of the level of reproducibility surveyed secondary.
It was also possible to replicate the TakeLab system by successfully re-implementing from the description given in its paper.
Overall, based on the systems that had a reference paper, most systems used a score distribution approach and reported the development score and best hyper-parameters.
All systems that we tried to reproduce were fully reproduced. In only one case, for the SVM-based TakeLab, the same accuracy score reported in its ARCT paper was obtained. In the other five systems, the deltas between their accuracy scores of their reproduced version and the scores reported in ARCT are positive for: GIST (+0.002), ranked first, for blcu_nlp (+0.038), ranked second, for NLITrans (+0.033), rising one position in the ranking surpassing the ECNU with a negative delta (-0.021), the SNU_iDS was also reproduced with a negative delta (-0.022).
These differences do not affect the ranking of the systems determined at SemEval-2018 except for NLITrans, which would rank third rather than fourth.

5. A revival of the argument reasoning comprehension task

5.1. Revised data set

Niven and Kao (2019) reported that data artifacts known to exist in the ARCT data set have a very large biasing impact in terms of artificially inflating the accuracy of the systems trained on it.
These authors applied a state-of-the-art language model, viz. BERT (Devlin et al., 2019), over the ARCT data set and obtained a system with the performance accuracy of 0.77, setting a new top score for this task just 0.03 points below the untrained human performance upper bound. Although BERT has obtained state-of-the-art results in several NLP tasks, the near-human performance raised the question of what had BERT learned about argument comprehension that the previous systems failed to learn.
In order to provide a pair of two alternative warrants, the annotators typically introduced the word *not* to create the adversarial instance (from a spontaneously occurring warrant), thus originating a data artifact across the data set biasing to one of the two labels to be assigned. Additional

	ARCT SemEval-2018		ARCT Reproduced		ARCT Revised	
system	rank	test acc.	dev. acc.	test acc.	dev. acc.	test acc.
GIST	1	0.712	0.715	0.714	0.500	0.500
BLCU_NLP	2	0.606	0.680	0.642	0.498	0.466
ECNU	3	0.604	0.684	0.583	0.505	0.498
NLITrans	4	0.590	0.660	0.623	0.591	0.555
SNU_IDS	12	0.565	0.703	0.543	0.509	0.503
TakeLab	16	0.541	0.516	0.541	0.486	0.523

Table 3: The first three columns contain the systems name, ranking and accuracy obtained in the ARCT. The fourth and fifth columns report the development and test accuracy obtained in the reproduction exercise. The sixth and seventh columns report the accuracy obtained on the revised data set.

analysis of the task using unigrams and bigrams with different combinations of features, while measuring their productivity and coverage, also offered strong support to the hypothesis that data artifacts could have a strong impact on distorting the accuracy of systems trained on this data set.

To test this hypothesis, Niven and Kao (2019) developed a revised version of the ARCT data set expunged from these artifacts and run BERT over it. In contrast to the near to human performance accuracy attained before this correction, the performance of this BERT model dropped to close to the accuracy of the random choice baseline.

In (Niven and Kao, 2019) the statistical cues were distributed throughout all of the original data set warrants and respective claims. Given that the *premise* and the *alternative warrant* imply the *opposite claim*, adding the same argument with a claim negated and the label inverted resulted in the distribution of the statistical cues.

5.2. Reassessed systems

The huge drop in performance of BERT when run on the revised ARCT data set naturally raises the question of what could be the performance the systems that participated in the ARCT challenge if they are run on the revised ARCT expunged from the data artifacts that artificially makes the task much more easy than it is. Having gone through the reproduction exercise described in Section 4, we are in a position to be able to contribute to answer this question.

The results from this re-evaluation of the six systems reproduced are presented in Table 3.

5.3. Conclusions

In this paper, we report on a reproduction exercise of a number of systems submitted to the ARCT shared task of SemEval2018. We included in this exercise the systems whose source-code was distributed, plus another one that we replicated, in a total of 6, where the top performing 4 in SemEval2018 are included.

It was possible to reproduce all systems under scrutiny with performance scores that overall are in line with the ones reported in the respective papers. Another finding was that, taking into account a metric to assess reproducibility level, the description papers of the ARCT task score poorly in that metric, ensuring a sub-optimal level of reproducibility for the systems they describe — even in the case of the systems we were able to reproduce.

Another major finding is that ARCT is a challenge worth being readdressed. The performance results of the reproduced systems obtained with a revised data set, expunged from data artifacts, were lower than the random baseline. They demonstrate that the ARCT task is actually a much harder challenge than what could have been perceived from the inflated, close to human-level performance scores obtained with the data set version used in SemEval2018. This calls for a revival of this task as there is much room for improvement until systems may come close to the upper bound provided by human performance.

With the present exercise, we did not intend at all to antagonise the teams that entered and organized the ARCT competition. We were driven by the scientific purpose of getting an idea of the reproducibility level of the ARCT systems. This exercise eventually leads also to the positive outcome of reinforcing the need to renew the attention for this task given its non-trivial difficulty.

We did not reproduce all systems submitted to ARCT for which there is a description paper, with those systems whose source-code is not distributed being left out of this exercise. It remains a question to be answered empirically whether and how the performance of these systems is altered with the revised data set.

To support the reproduction of the presented paper, the relevant materials are available from (NLX, 2019).

Acknowledgments

The research reported here was partially supported by PORTULAN CLARIN—Research Infrastructure for the Science and Technology of Language, funded by Lisboa 2020, Alentejo 2020 and FCT—Fundação para a Ciência e Tecnologia under the grant PINFRA/22117/2016, and through the individual PhD grant ref. SFRH/BD/129824/2017.

Bibliographical References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Branco, A., Cohen, K. B., Vossen, P., Ide, N., and Calzolari, N. (2017). Replicability and reproducibility of research results for human language technology: introducing an lre special section. *Language Resources and Evaluation*, 51(1):1–5.

Branco, A., Calzolari, N., and Choukri, K. (2018). 4REAL 2018 workshop on replicability and reproducibility of research results in science and technology of language.

Branco, A. (2013). Reliability and meta-reliability of language resources: Ready to initiate the integrity debate? In *The Twelfth Workshop on Treebanks and Linguistic Theories (TLT12)*, page 27.

Brassard, A., Kuculo, T., Boltužić, F., and Šnajder, J. (2018). TakeLab at SemEval-2018 task12: Argument reasoning comprehension with skip-thought vectors. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 1133–1136, New Orleans, Louisiana, June. Association for Computational Linguistics.

Chen, Q., Zhu, X., Ling, Z.-H., Wei, S., Jiang, H., and Inkpen, D. (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.

Chen, Z., Song, W., and Liu, L. (2018). TRANSRW at SemEval-2018 task 12: Transforming semantic representations for argument reasoning comprehension. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 1142–1145, New Orleans, Louisiana, June. Association for Computational Linguistics.

Choi, H. and Lee, H. (2018). GIST at SemEval-2018 task 12: A network transferring inference knowledge to argument reasoning comprehension task. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 773–777, New Orleans, Louisiana, June. Association for Computational Linguistics.

Chollet, F. et al. (2015). Keras. `https://keras.io`.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Ding, P. and Zhou, X. (2018). YNU deep at SemEval-2018 task 12: A BiLSTM model with neural attention for argument reasoning comprehension. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 1120–1123, New Orleans, Louisiana, June. Association for Computational Linguistics.

Dodge, J., Gururangan, S., Card, D., Schwartz, R., and Smith, N. A. (2019). Show your work: Improved reporting of experimental results. 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 2185–2194.

Fokkens, A., Van Erp, M., Postma, M., Pedersen, T., Vossen, P., and Freire, N. (2013). Offspring from reproduction problems: What replication failure teaches us. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1691–1701.

Gundersen, O. E. and Kjensmo, S. (2018). State of the art: Reproducibility in artificial intelligence. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*.

Habernal, I., Wachsmuth, H., Gurevych, I., and Stein, B. (2018a). 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, Volume 1 (Long Papers)*, pages 1930–1940.

Habernal, I., Wachsmuth, H., Gurevych, I., and Stein, B. (2018b). SemEval-2018 task 12: The argument reasoning comprehension task. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 763–772.

Hutson, M. (2018). Artificial intelligence faces reproducibility crisis. *Science (New York, NY)*, 359(6377):725.

Joshi, A., Baldwin, T., Sinnott, R. O., and Paris, C. (2018). UniMelb at SemEval-2018 task 12: Generative implication using LSTMs, Siamese networks and semantic representations with synonym fuzzing. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 1124–1128, New Orleans, Louisiana, June. Association for Computational Linguistics.

Kim, T., Choi, J., and Lee, S.-g. (2018). SNU_IDS at SemEval-2018 task 12: Sentence encoder with contextualized vectors for argument reasoning comprehension. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 1083–1088, New Orleans, Louisiana, June. Association for Computational Linguistics.

Kiros, R., Zhu, Y., Salakhutdinov, R., Zemel, R. S., Torralba, A., Urtasun, R., and Fidler, S. (2015). Skip-thought vectors. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2*, pages 3294–3302. MIT Press.

Li, Y. and Zhou, X. (2018). Lyb3b at SemEval-2018 task 12: Ensemble-based deep learning models for argument reasoning comprehension task. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 1137–1141, New Orleans, Louisiana, June. Association for Computational Linguistics.

Liebeck, M., Funke, A., and Conrad, S. (2018). HHU at SemEval-2018 task 12: Analyzing an ensemble-based deep learning approach for the argument mining task of choosing the correct warrant. In *Proceedings of The 12th*

International Workshop on Semantic Evaluation, pages 1114–1119, New Orleans, Louisiana, June. Association for Computational Linguistics.

Liu, W., Sun, C., Lin, L., and Liu, B. (2018). ITNLP-ARC at SemEval-2018 task 12: Argument reasoning comprehension with attention. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 1089–1093, New Orleans, Louisiana, June. Association for Computational Linguistics.

Lucic, M., Kurach, K., Michalski, M., Gelly, S., and Bousquet, O. (2018). Are GANs created equal? a large-scale study. In *Advances in neural information processing systems*, pages 700–709.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

McCann, B., Bradbury, J., Xiong, C., and Socher, R. (2017). Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6294–6305.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Moore, A. and Rayson, P. (2018). Bringing replication and reproduction together with generalisability in NLP: Three reproduction studies for target dependent sentiment analysis. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1132–1144.

Niven, T. and Kao, H.-Y. (2018). NLITrans at SemEval-2018 task 12: Transfer of semantic knowledge for argument comprehension. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 1099–1103, New Orleans, Louisiana, June. Association for Computational Linguistics.

Niven, T. and Kao, H.-Y. (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.

NLX. (2019). NLX's GitLab repository for the ARCT replication experiment. `https://github.com/nlx-group/arct-rep-rev`.

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

Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Raff, E. (2019). A step toward quantifying independently reproducible machine learning research. In *Advances in Neural Information Processing Systems*, pages 5486–5496.

Reimers, N. and Gurevych, I. (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.

ReproLang. (2019). Reprolang 2020 shared task. `https://www.clarin.eu/event/2020/reprolang-2020`.

V. Stodden, et al., editors. (2014). *Implementing reproducible research*. CRC Press.

Theano Development Team. (2016). Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May.

Tian, J., Lan, M., and Wu, Y. (2018). ECNU at SemEval-2018 task 12: An end-to-end attention-based neural network for the argument reasoning comprehension task. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 1094–1098, New Orleans, Louisiana, June. Association for Computational Linguistics.

Williams, A., Nangia, N., and Bowman, S. (2018). A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Witten, I. H., Frank, E., Hall, M. A., and Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Zhang, Y., Wang, J., and Zhang, X. (2018). YNU-HPCC at SemEval-2018 task 1: BiLSTM with attention based sentiment analysis for affect in tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 273–278, New Orleans, Louisiana, June. Association for Computational Linguistics.

Zhao, M., Liu, C., Liu, L., Zhao, Y., and Yu, D. (2018). BLCU_NLP at SemEval-2018 task 12: An ensemble model for argument reasoning based on hierarchical attention. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 1104–1108, New Orleans, Louisiana, June. Association for Computational Linguistics.