An Annotation Protocol for Collecting User-Generated Counter-Arguments using Crowdsourcing

Paul Reisert[1,3], Gisela Vallejo[2], Naoya Inoue[3,1], Iryna Gurevych[2], and Kentaro Inui[3,1]

[1] RIKEN Center for Advanced Intelligence Project (AIP)
paul.reisert@riken.jp
[2] Ubiquitous Knowledge Processing Lab (UKP)
Department of Computer Science, Technische Universität Darmstadt
{vallejo,gurevych}@ukp.informatik.tu-darmstadt.de
[3] Tohoku University
{naoya-i,inui}@ecei.tohoku.ac.jp

Abstract. Constructive feedback is important for improving critical thinking skills. However, little work has been done to automatically generate such feedback for an argument. In this work, we experiment with an annotation protocol for collecting user-generated counter-arguments via crowdsourcing. We conduct two parallel crowdsourcing experiments, where workers are instructed to produce i) a counter-argument, and ii) a counter-argument after identifying a fallacy. Our analysis indicates that we can collect counter-arguments that are useful as constructive feedback, especially when workers are first asked to identify a fallacy type.

Keywords: Critical Thinking · Counter-Argument · Fallacy · Crowdsourcing · Annotation Study · Constructive Feedback

1 Introduction

Automatic essay scoring is the task of automatically evaluating a wide-range of essay criteria in a pedagogical context, such as organization [10], self-directed learning [7], thesis clarity [11] and author stance [12]. Several works have also integrated argumentative features [13, 2, 8] for evaluation. Applications such as Grammarly[4] and eRater[5] have received wide attention for automatically assess the contents of an essay.

An example of the usefulness of constructive feedback is shown in Figure 1. In response to the *topic*, T_1, the argument A_1 extracted from a student's essay. In response to A_1, a teacher would provide constructive feedback to the student for improving their argument (e.g., CA_1 & CA_2). Afterwards, a student could revise their argument to produce a stronger one (i.e., R_1) and improve their critical thinking skills for future essays.

[4] https://www.grammarly.com/
[5] https://www.ets.org/erater

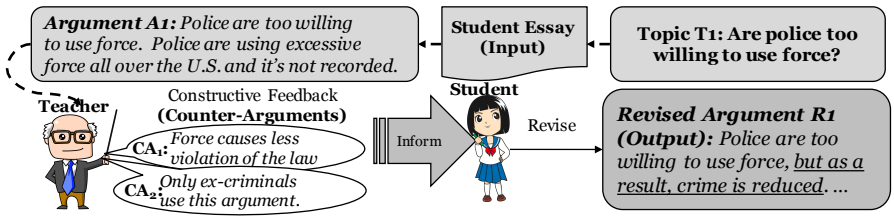

Fig. 1: Example of argument revision via constructive feedback.

We aim to create a method for improving automatic constructive feedback generation, which can help reduce time for graders and allow writers to instantly learn their mistakes. Towards this goal, fallacy detection and counter-arguments have been shown to be useful. Habernal et al. [3] created a game which allowed users to identify fallacies. In the pedagogical context, several studies have identified common fallacies in student essays [9, 6, 1]. For counter-arguments, Wachsmuth et al. [14] created a task for retrieving the best counter-argument for a given argument, and Hua and Wang [5] generated counter-arguments by extracting external evidence. However, it still remains an open issue as to how to create a corpus useful for modeling constructive feedback.

In this work, we conduct two parallel crowdsourcing experiments in order to determine if a large-scale, high-quality corpus of user-generated counter-arguments required for modeling constructive feedback can be created. We instruct non-expert workers i) to produce counter-arguments simply given an argument, and ii) to produce a counter-argument after identifying a specified fallacy type. We then conduct an analysis on the collected counter-arguments for determining their usefulness. Our results suggest that workers can produce useful counter-arguments, especially when instructed to identify a fallacy type.

2 Collecting user-generated counter-arguments

2.1 Data and crowdsourcing

We conduct our experiments on top of the Argument Reasoning Comprehension (ARC) corpus [4]. ARC contains 2,477 context-independent, micro-level (i.e., single claim and premise) arguments with 172 diverse topics, making the corpus ideal for modeling constructive feedback.

We use the crowdsourcing platform Figure Eight[6] for quickly collecting counter-arguments. We assume that a large-scale corpus of counter-arguments can be produced by non-expert crowdworkers with appropriate guidelines.

Counter-argument generation without fallacy identification (CAG) We first conduct trial experiments on Figure Eight for calibrating appropriate

[6] `http://www.figure-eight.com`

Fallacy Type	Yes	No	Unsure	Cohen's κ
Appeal to Common Practice	⟨13,17⟩	⟨5,1⟩	⟨2,2⟩	0.44
Begging the Question	⟨14,18⟩	⟨6,2⟩	⟨0,0⟩	0.41
Hasty Generalization	⟨15,15⟩	⟨4,5⟩	⟨1,0⟩	0.68
Questionable Cause	⟨15,14⟩	⟨4,4⟩	⟨1,2⟩	0.46
Red Herring	⟨15,17⟩	⟨4,2⟩	⟨1,1⟩	0.49
Agreed instances	64	10	0	

Table 1: CAG-F distribution and inner-annotator agreement between annotators ⟨**A**,**B**⟩.

interface, guidelines, and settings. Per given *topic*, the worker is shown a *claim* and *premise* and instructed to produce a sentence-long counter-argument that attacks one or both of them. We use the following settings for CAG: 10 second *minimum time per instance*, level 3 annotators (i.e., high-quality), and $0.10 per answer.

Counter-argument generation with fallacy identification (CAG-F) We conduct a parallel experiment in which crowdworkers were asked if a pre-specified fallacy type exists in the original argument. We randomly select 5 fallacy types and their examples from SoftSchools[7]: *appeal to common practice*, *begging the question*, *hasty generalization*, *questionable cause*, and *red herring*. We create separate crowdsourcing jobs for each fallacy type. Workers are instructed to answer if the fallacy type exists, and if so, they are asked to produce a counter-argument. We use the same settings as CAG. However, annotators are not required to write a counter-argument if they select *no* or *unsure*, so we award each answer with $0.05 and offer workers a bonus if they produce a *good* counter-argument.

2.2 Annotation Statistics

For CAG, we collect 100 user-generated counter-arguments for 100 arguments. The time to complete the experiment was roughly 2.5 hours. For each of the 5 jobs in CAG-F, we employed 5 crowdworkers per argument (100 arguments total). The average time to complete each experiment was roughly 1.3 hours.

3 Analysis and discussion

We conduct a qualitative analysis using two annotators specializing in the field of argumentation. One annotator created the crowdsourcing guidelines and conducted the experiments. We asked both annotators to judge the quality of CAG and CAG-F counter-arguments by the following: *Is the counter-argument attacking the claim, premise, or both?*, and *Using the counter-argument, could you make the original argument better?*. If one answer was *no*, the counter-argument was labeled as *no*. For CAG, we have both annotators answer the above questions for the 100 counter-arguments. For CAG-F, for each of the 5 fallacy types,

[7] `http://www.softschools.com/examples/fallacies/`

Claim	Premise	CAG	CAG-F
Unpaid internship exploit college students	Interns are replacing employees.	*unpaid internship offer students chance of getting experience and therefore do not exploit them*	*its too hasty to say that all Interns are replacing employees* (**Hasty Generalization**)
Home schoolers deserve a tax break	Home schooled children should get the same state financial backing given to public school attendees [...]	*most of the time they may not get equal education facilities of public attendees*	*no tax relief is needed because there are no real costs for such learning.* (**Begging the Question**)

Table 2: Examples of CAG and CAG-F Counter-Arguments agreed as *yes*.

we randomly select 20 arguments with a unique topic to the fallacy type, where some arguments are shared across different fallacy types.

3.1 Results

Table 1 shows the distribution of answers and the inner-annotator agreement for CAG-F, and Table Table 2 shows examples from both stages. For CAG, the Cohen's kappa[8] (κ) between both annotators is 0.29, which is slightly lower than CAG-F (0.37). In total, 74 (64 *yes* and 10 *no*) instances were agreed upon, indicating a slight improvement (20%) over CAG.

Disagreements For CAG, we observed all but one instance of the 21 instances labeled as *no* by one annotator (**B**) were labeled as *no* by the other (**A**). When observing the 20 remaining instances labeled as *no* by **A**, we found that most were labeled as a *simple contradiction*, *unrelated*, or *incomprehensible/ungrammatical*. We believe this attributes to the fact that **A** created the guidelines and experiments and was more critical of the quality. For CAG-F, we observed that **A** labeled *no* 3 times when **B** said *yes*. We discovered that the reasons are *agreeing stance*, *irrelevant*, and *untrue* (e.g., "*Cyclists have nothing to do with bike lanes*"). **B** said *no* 11 times when **A** answered *yes* with the following reasons: *non-counter-argument*, *untrue*, and *unclear*.

4 Conclusion

Towards automatically generating constructive feedback, in this work, we experimented with constructing an annotation protocol for collecting user-generated counter-arguments via crowdsourcing. We conducted two parallel crowdsourcing experiments where, given an argument, workers were instructed to i) produce a counter-argument, and ii) first identify a fallacy type and then produce a counter-argument. Our results indicate that we can collect counter-arguments useful as constructive feedback in both settings, especially when workers were instructed to first identify a fallacy in the original argument.

[8] We calculate the Cohen's kappa after filtering out *unsure* instances.

References

1. El Khoiri, N., Widiati, U.: Logical fallacies in indonesian efl learners' argumentative writing: Students' perspectives. Dinamika Ilmu **17**(1), 71–81 (2017)
2. Ghosh, D., Khanam, A., Han, Y., Muresan, S.: Coarse-grained argumentation features for scoring persuasive essays. In: Proceedings of the 54th Annual Meeting of ACL (Volume 2: Short Papers). pp. 549–554 (2016)
3. Habernal, I., Pauli, P., Gurevych, I.: Adapting Serious Game for Fallacious Argumentation to German: Pitfalls, Insights, and Best Practices. In: Proceedings of the Eleventh International Conference on LREC. pp. 3329–3335 (2018)
4. Habernal, I., Wachsmuth, H., Gurevych, I., Stein, B.: The argument reasoning comprehension task: Identification and reconstruction of implicit warrants. In: Proceedings of the 2018 Conference of NAACL: HLT, Volume 1 (Long Papers). pp. 1930–1940. Association for Computational Linguistics (2018)
5. Hua, X., Wang, L.: Neural argument generation augmented with externally retrieved evidence. In: Proceedings of the 56th Annual Meeting of ACL (Volume 1: Long Papers). pp. 219–230 (2018)
6. Indah, R.N., Kusuma, A.W.: Fallacies in english department students claims: A rhetorical analysis of critical thinking. Jurnal Pendidikan Humaniora **3**(4), 295–304 (2015)
7. Lucas, C., Gibson, A., Buckingham Shum, S.: Utilization of a novel online reflective learning tool for immediate formative feedback to assist pharmacy students reflective writing skills. American Journal of Pharmaceutical Education (2018)
8. Nguyen, H.V., Litman, D.J.: Argument mining for improving the automated scoring of persuasive essays. In: The Thirty-Second AAAI Conference on Artificial Intelligence. pp. 5892–5899 (2018)
9. Oktavia, W., Yasin, A., et al.: An analysis of studentsargumentative elements and fallacies in studentsdiscussion essays. English Language Teaching **2**(3) (2014)
10. Persing, I., Davis, A., Ng, V.: Modeling organization in student essays. In: Proceedings of the 2010 Conference on EMNLP. pp. 229–239. Association for Computational Linguistics (2010)
11. Persing, I., Ng, V.: Modeling thesis clarity in student essays. In: Proceedings of the 51st Annual Meeting of ACL(Volume 1: Long Papers). vol. 1, pp. 260–269 (2013)
12. Persing, I., Ng, V.: Modeling stance in student essays. In: Proceedings of the 54th Annual Meeting of ACL (Volume 1: Long Papers). vol. 1, pp. 2174–2184 (2016)
13. Wachsmuth, H., Al-Khatib, K., Stein, B.: Using Argument Mining to Assess the Argumentation Quality of Essays. In: Proceedings of the 26th International Conference on COLING. pp. 1680–1692 (2016)
14. Wachsmuth, H., Syed, S., Stein, B.: Retrieval of the best counterargument without prior topic knowledge. In: Proceedings of the 56th Annual Meeting of ACL (Volume 1: Long Papers). vol. 1, pp. 241–251 (2018)