Crowdsourcing Inference-Rule Evaluation

Naomi Zeichner
Bar-Ilan University
Ramat-Gan, Israel
zeichner.naomi@gmail.com

Jonathan Berant
Tel-Aviv University
Tel-Aviv, Israel
jonatha6@post.tau.ac.il

Ido Dagan
Bar-Ilan University
Ramat-Gan, Israel
dagan@cs.biu.ac.il

Abstract

The importance of inference rules to semantic applications has long been recognized and extensive work has been carried out to automatically acquire inference-rule resources. However, evaluating such resources has turned out to be a non-trivial task, slowing progress in the field. In this paper, we suggest a framework for evaluating inference-rule resources. Our framework simplifies a previously proposed "instance-based evaluation" method that involved substantial annotator training, making it suitable for crowdsourcing. We show that our method produces a large amount of annotations with high inter-annotator agreement for a low cost at a short period of time, without requiring training expert annotators.

1 Introduction

Inference rules are an important component in semantic applications, such as Question Answering (QA) (Ravichandran and Hovy, 2002) and Information Extraction (IE) (Shinyama and Sekine, 2006), describing a directional inference relation between two text patterns with variables. For example, to answer the question *'Where was Reagan raised?'* a QA system can use the rule *'X brought up in Y→X raised in Y'* to extract the answer from *'Reagan was brought up in Dixon'*. Similarly, an IE system can use the rule *'X work as Y→X hired as Y'* to extract the PERSON and ROLE entities in the "hiring" event from *'Bob worked as an analyst for Dell'*.

The significance of inference rules has led to substantial effort into developing algorithms that automatically learn inference rules (Lin and Pantel, 2001; Sekine, 2005; Schoenmackers et al., 2010), and generate knowledge resources for inference systems. However, despite their potential, utilization of inference rule resources is currently somewhat limited. This is largely due to the fact that these algorithms often produce invalid rules. Thus, evaluation is necessary both for resource developers as well as for inference system developers who want to asses the quality of each resource. Unfortunately, as evaluating inference rules is hard and costly, there is no clear evaluation standard, and this has become a slowing factor for progress in the field.

One option for evaluating inference rule resources is to measure their impact on an end task, as that is what ultimately interests an inference system developer. However, this is often problematic since inference systems have many components that address multiple phenomena, and thus it is hard to assess the effect of a single resource. An example is the Recognizing Textual Entailment (RTE) framework (Dagan et al., 2009), in which given a text T and a textual hypothesis H, a system determines whether H can be inferred from T. This type of evaluation was established in RTE challenges by ablation tests (see RTE ablation tests in ACLWiki) and showed that resources' impact can vary considerably from one system to another. These issues have also been noted by Sammons et al. (2010) and LoBue and Yates (2011). A complementary application-independent evaluation method is hence necessary.

Some attempts were made to let annotators judge rule correctness *directly*, that is by asking them to judge the correctness of a given rule (Shinyama et al., 2002; Sekine, 2005). However, Szpektor et al. (2007) observed that directly judging rules out of context often results in low inter-annotator agreement. To remedy that, Szpektor et al. (2007) and

Bhagat et al. (2007) proposed *"instance-based evaluation"*, in which annotators are presented with an *application* of a rule in a particular context and need to judge whether it results in a valid inference. This simulates the utility of rules in an application and yields high inter-annotator agreement. Unfortunately, their method requires lengthy guidelines and substantial annotator training effort, which are time consuming and costly. Thus, a simple, robust and replicable evaluation method is needed.

Recently, crowdsourcing services such as Amazon Mechanical Turk (AMT) and CrowdFlower (CF)[1] have been employed for semantic inference annotation (Snow et al., 2008; Wang and Callison-Burch, 2010; Mehdad et al., 2010; Negri et al., 2011). These works focused on generating and annotating RTE text-hypothesis pairs, but did not address annotation and evaluation of inference rules. In this paper, we propose a novel instance-based evaluation framework for inference rules that takes advantage of crowdsourcing. Our method substantially simplifies annotation of rule applications and avoids annotator training completely. The novelty in our framework is two-fold: (1) We simplify instance-based evaluation from a complex decision scenario to two independent binary decisions. (2) We apply methodological principles that efficiently communicate the definition of the "inference" relation to untrained crowdsourcing workers (*Turkers*).

As a case study, we applied our method to evaluate algorithms for learning inference rules between predicates. We show that we can produce many annotations cheaply, quickly, at good quality, while achieving high inter-annotator agreement.

2 Evaluating Rule Applications

As mentioned, in instance-based evaluation individual rule applications are judged rather than rules in isolation, and the quality of a rule-resource is then evaluated by the validity of a sample of applications of its rules. Rule application is performed by finding an instantiation of the rule *left-hand-side* in a corpus (termed *LHS extraction*) and then applying the rule on the extraction to produce an instantiation of the rule *right-hand-side* (termed *RHS instantiation*). For example, the rule *'X observe Y→X celebrate Y'* can be applied on the LHS extraction *'they observe holidays'* to produce the RHS instantiation *'they celebrate holidays'*.

The target of evaluation is to judge whether each rule application is valid or not. Following the standard RTE task definition, a rule application is considered valid if a human reading the *LHS extraction* is highly likely to infer that the *RHS instantiation* is true (Dagan et al., 2009). In the aforementioned example, the annotator is expected to judge that *'they observe holidays'* entails *'they celebrate holidays'*. In addition to this straightforward case, two more subtle situations may arise. The first is that the LHS extraction is meaningless. We regard a proposition as meaningful if a human can easily understand its meaning (despite some simple grammatical errors). A meaningless LHS extraction usually occurs due to a faulty extraction process (e.g., Table 1, Example 2) and was relatively rare in our case study (4% of output, see Section 4). Such rule applications can either be extracted from the sample so that the rule-base is not penalized (since the problem is in the extraction procedure), or can be used as examples of non-entailment, if we are interested in overall performance. A second situation is a meaningless RHS instantiation, usually caused by rule application in a wrong context. This case is tagged as non-entailment (for example, applying the rule *'X observe Y→X celebrate Y'* in the context of the extraction *'companies observe dress code'*).

Each rule application therefore requires an answer to the following three questions: 1) Is the LHS extraction meaningful? 2) Is the RHS instantiation meaningful? 3) If both are meaningful, does the LHS extraction entail the RHS instantiation?

3 Crowdsourcing

Previous works using crowdsourcing noted some principles to help get the most out of the service(Wang et al., 2012). In keeping with these findings we employ the following principles: **(a) Simple tasks**. The global task is split into simple sub-tasks, each dealing with a single aspect of the problem. **(b) Do not assume linguistic knowledge by annotators**. Task descriptions avoid linguistic terms such as "tense", which confuse workers. **(c) Gold standard validation**. Using CF's built-in methodology,

[1] https://www.mturk.com and http://crowdflower.com

Phrase	Meaningful	Comments
1) Doctors be treat Mary	Yes	Annotators are instructed to ignore simple inflectional errors
2) A player deposit an	No	Bad extraction for the rule LHS *'X deposit Y'*
3) humans bring in bed	No	Wrong context, result of applying *'X turn in Y→X bring in Y'* on *'humans turn in bed'*

Table 1: Examples of phrase "meaningfulness" (Note that the comments are not presented to Turkers).

gold standard (GS) examples are combined with actual annotations to continuously validate annotator reliability.

We split the annotation process into two tasks, the first to judge phrase meaningfulness (Questions 1 and 2 above) and the second to judge entailment (Question 3 above). In Task 1, the LHS extractions and RHS instantiations of all rule applications are separated and presented to different Turkers independently of one another. This task is simple, quick and cheap and allows Turkers to focus on the single aspect of judging phrase meaningfulness. Rule applications for which both the LHS extraction and RHS instantiation are judged as meaningful are passed to Task 2, where Turkers need to decide whether a given rule application is valid. If not for Task 1, Turkers would need to distinguish in Task 2 between non-entailment due to (1) an incorrect rule (2) a meaningless RHS instantiation (3) a meaningless LHS extraction. Thanks to Task 1, Turkers are presented in Task 2 with two meaningful phrases and need to decide only whether one entails the other.

To ensure high quality output, each example is evaluated by three Turkers. Similarly to Mehdad et al. (2010) we only use results for which the confidence value provided by CF is greater than 70%.

We now describe the details of both tasks. Our simplification contrasts with Szpektor et al. (2007), whose judgments for each rule application are similar to ours, but had to be performed simultaneously by annotators, which required substantial training.

Task 1: *Is the phrase meaningful?*
In keeping with the second principle above, the task description is made up of a short verbal explanation followed by positive and negative examples. The definition of "meaningfulness" is conveyed via examples pointing to properties of the automatic phrase extraction process, as seen in Table 1.

Task 2: *Judge if one phrase is true given another.*
As mentioned, rule applications for which both sides were judged as meaningful are evaluated for entailment. The challenge is to communicate the definition of "entailment" to Turkers. To that end the task description begins with a short explanation followed by "easy" and "hard" examples with explanations, covering a variety of positive and negative entailment "types" (Table 2).

Defining "entailment" is quite difficult when dealing with expert annotators and still more with non-experts, as was noted by Negri et al. (2011). We therefore employ several additional mechanisms to get the definition of entailment across to Turkers and increase agreement with the GS. We run an initial small test run and use its output to improve annotation in two ways: First, we take examples that were "confusing" for Turkers and add them to the GS with explanatory feedback presented when a Turker answers incorrectly. (E.g., the pair *('The owner be happy to help drivers', 'The owner assist drivers')* was judged as entailing in the test run but only achieved a confidence value of 0.53). Second, we add examples that were annotated unanimously by Turkers to the GS to increase its size, allowing CF to better estimate Turker's reliability (following CF recommendations, we aim to have around 10% GS examples in every run). In Section 4 we show that these mechanisms improved annotation quality.

4 Case Study

As a case study, we used our evaluation methodology to compare four methods for learning entailment rules between predicates: DIRT (Lin and Pantel, 2001), Cover (Weeds and Weir, 2003), BInc (Szpektor and Dagan, 2008) and Berant et al. (2010). To that end, we applied the methods on a set of one billion extractions (generously provided by Fader et al. (2011)) automatically extracted from the ClueWeb09 web crawl[2], where each extraction comprises a predicate and two arguments. This resulted in four learned inference rule resources.

[2] http://lemurproject.org/clueweb09.php/

Example	Entailed	Explanation given to Turkers
LHS: The lawyer sign the contract RHS: The lawyer read the contract	Yes	There is a chance the lawyer has not read the contract, but most likely that as he signed it, he must have read it.
LHS: John be related to Jerry RHS: John be a close relative of Jerry	No	The LHS can be understood from the RHS, but not the other way around as the LHS is more general.
LHS: Women be at increased risk of cancer RHS: Women die of cancer	No	Although the RHS is correct, it cannot be understood from the LHS.

Table 2: Examples given in the description of Task 2.

We randomly sampled 5,000 extractions, and for each one sampled four rules whose LHS matches the extraction from the union of the learned resources. We then applied the rules, which resulted in 20,000 rule applications. We annotated rule applications using our methodology and evaluated each learning method by comparing the rules learned by each method with the annotation generated by CF.

In Task 1, 281 rule applications were annotated as meaningless LHS extraction, and 1,012 were annotated as meaningful LHS extraction but meaningless RHS instantiation and so automatically annotated as non-entailment. 8,264 rule applications were passed on to Task 2, as both sides were judged meaningful (the remaining 10,443 discarded due to low CF confidence). In Task 2, 5,555 rule applications were judged with a high confidence and supplied as output, 2,447 of them as positive entailment and 3,108 as negative. Overall, 6,567 rule applications (dataset of this paper) were annotated for a total cost of $1000. The annotation process took about one week.

In tests run during development we experimented with Task 2 wording and GS examples, seeking to make the definition of entailment as clear as possible. To do so we randomly sampled and manually annotated 200 rule applications (from the initial 20,000), and had Turkers judge them. In our initial test, Turkers tended to answer "yes" comparing to our own annotation, with 0.79 agreement between their annotation and ours, corresponding to a kappa score of 0.54. After applying the mechanisms described in Section 3, false-positive rate was reduced from 18% to 6% while false-negative rate only increased from 4% to 5%, corresponding to a high agreement of 0.9 and kappa of 0.79.

In our test, 63% of the 200 rule applications were annotated unanimously by the Turkers. Importantly, all these examples were in perfect agreement with our own annotation, reflecting their high reliability. For the purpose of evaluating the resources learned by the algorithms we used annotations with CF confidence ≥ 0.7 for which kappa is 0.99.

Lastly, we computed the area under the recall-precision curve (AUC) for *DIRT*, *Cover*, *BInc* and *Berant et al.*'s method, resulting in an AUC of 0.4, 0.43, 0.44, and 0.52 respectively. We used the AUC curve, with number of recall-precision points in the order of thousands, to avoid tuning a threshold parameter. Overall, we demonstrated that our evaluation framework allowed us to compare four different learning methods in low costs and within one week.

5 Discussion

In this paper we have suggested a crowdsourcing framework for evaluating inference rules. We have shown that by simplifying the previously-proposed instance-based evaluation framework we are able to take advantage of crowdsourcing services to replace trained expert annotators, resulting in good quality large scale annotations, for reasonable time and cost. We have presented the methodological principles we developed to get the entailment decision across to Turkers, achieving very high agreement both with our annotations and between the annotators themselves. Using the CrowdFlower forms we provide with this paper, the proposed methodology can be beneficial for both resource developers evaluating their output as well as inference system developers wanting to assess the quality of existing resources.

Acknowledgments

This work was partially supported by the Israel Science Foundation grant 1112/08, the PASCAL-2 Network of Excellence of the European Community FP7-ICT-2007-1-216886, and the European Communitys Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 287923 (EXCITEMENT).

References

Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2010. Global learning of focused entailment graphs. In *Proceedings of the annual meeting of the Association for Computational Linguistics (ACL)*.

Rahul Bhagat, Patrick Pantel, and Eduard Hovy. 2007. LEDIR: An unsupervised algorithm for learning directionality of inference rules. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

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.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the Conference of Empirical Methods in Natural Language Processing (EMNLP '11)*.

Dekang Lin and Patrick Pantel. 2001. DIRT - discovery of inference rules from text. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Peter LoBue and Alexander Yates. 2011. Types of common-sense knowledge needed for recognizing textual entailment. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*.

Yashar Mehdad, Matteo Negri, and Marcello Federico. 2010. Towards cross-lingual textual entailment. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (ACL)*.

Matteo Negri, Luisa Bentivogli, Yashar Mehdad, Danilo Giampiccolo, and Alessandro Marchetti. 2011. Divide and conquer: Crowdsourcing the creation of cross-lingual textual entailment corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '11)*.

Deepak Ravichandran and Eduard Hovy. 2002. Learning surface text patterns for a question answering system. In *Proceedings of the annual meeting of the Association for Computational Linguistics (ACL)*.

Mark Sammons, V. G. Vinod Vydiswaran, and Dan Roth. 2010. "ask not what textual entailment can do for you...". In *Proceedings of the annual meeting of the Association for Computational Linguistics (ACL)*.

Stefan Schoenmackers, Oren Etzioni Jesse Davis, and Daniel S. Weld. 2010. Learning first-order horn clauses from web text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP '10)*.

Satoshi Sekine. 2005. Automatic paraphrase discovery based on context and keywords between ne pairs. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Yusuke Shinyama and Satoshi Sekine. 2006. Preemptive information extraction using unrestricted relation discovery. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL '06)*.

Yusuke Shinyama, Satoshi Sekine, and Kiyoshi Sudo. 2002. Automatic paraphrase acquisition from news articles. In *Proceedings of the second international conference on Human Language Technology Research (HLT '02)*.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. 2008. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '08)*.

Idan Szpektor and Ido Dagan. 2008. Learning entailment rules for unary templates. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*.

Idan Szpektor, Eyal Shnarch, and Ido Dagan. 2007. Instance-based evaluation of entailment rule acquisition. In *Proceedings of the annual meeting of the Association for Computational Linguistics (ACL)*.

Rui Wang and Chris Callison-Burch. 2010. Cheap facts and counter-facts. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*.

Aobo Wang, Cong Duy Vu Hoang, and Min-Yen Kan. 2012. Perspectives on crowdsourcing annotations for natural language processing. *Journal of Language Resources and Evaluation)*.

Julie Weeds and David Weir. 2003. A general framework for distributional similarity. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2003)*.