Crowdsourcing the evaluation of a domain-adapted named entity recognition system

Asad B. Sayeed, Timothy J. Meyer, Hieu C. Nguyen, Olivia Buzek
Department of Computer Science
University of Maryland
College Park, MD 20742
asayeed@cs.umd.edu,
tmeyer1@umd.edu,
{hcnguyen88,olivia.buzek}
@gmail.com

Amy Weinberg
Department of Linguistics
University of Maryland
College Park, MD 20742
weinberg@umiacs.umd.edu

Abstract

Named entity recognition systems sometimes have difficulty when applied to data from domains that do not closely match the training data. We first use a simple rule-based technique for domain adaptation. Data for robust validation of the technique is then generated, and we use crowdsourcing techniques to show that this strategy produces reliable results even on data not seen by the rule designers. We show that it is possible to extract large improvements on the target data rapidly at low cost using these techniques.

1 Introduction

1.1 Named entities and errors

In this work, we use crowdsourcing to generate evaluation data to validate simple techniques designed to adapt a widely-used high-performing named entity recognition system to new domains. Specifically, we achieve a roughly 10% improvement in precision on text from the information technology (IT) business press via *post hoc* rule-based error reduction. We first tested the system on a small set of data that we annotated ourselves. Then we collected data from Amazon Mechanical Turk in order to demonstrate that the gain is stable. To our knowledge, there is no previous work on crowdsourcing as a rapid means of evaluating error mitigation in named entity recognizer development.

Named entity recognition (NER) is a well-known problem in NLP which feeds into many other related tasks such as information retrieval (IR) and machine translation (MT) and more recently social network discovery and opinion mining. Generally, errors in the underlying NER technology correlate with a steep price in performance in the NLP systems further along a processing pipeline, as incorrect entities propagate into incorrect translations or erroneous graphs of social networks.

Not all errors carry the same price. In some applications, omitting a named entity has the consequence of reducing the availability of training data, but including an incorrectly identified piece of text *as* as a named entity has the consequence of producing misleading results. Our application would be opinion mining; an omitted entity may prevent the system from attributing an opinion to a source, but an incorrect entity reveals non-existent opinion sources.

Machine learning is currently used extensively in building NER systems. One such system is BBN's Identifinder (Bikel et al., 1999). The IdentiFinder algorithm, based on Hidden Markov Models, has been shown to achieve F-measure scores above 90% when the training and testing data happen to be derived from Wall Street Journal text produced in the 1990s. We use IdentiFinder 3.3 as a starting point for performance improvement in this paper.

The use of machine learning in existing systems requires us to produce new and costly training data if we want to adapt these systems directly to other domains. Our *post hoc* error reduction strategy is therefore profoundly different: it relieves us of the burden of generating complete training examples. The data we generate are strictly corrections of the existing system's output. Our thus cheaper evaluation is therefore primarily on improvements to pre-

Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL, pages 345–348,
Los Angeles, California, June 2010.

cision, while minimizing damage to recall, unlike an evaluation based on retraining with new, fully-annotated text.

1.2 Crowdsourcing

Crowdsourcing is the use of the mass collaboration of Internet passers-by for large enterprises on the World Wide Web such as Wikipedia and survey companies. However, a generalized way to monetize the many small tasks that make up a larger task is relatively new. Crowdsourcing platforms like Amazon Mechanical Turk have allowed some NLP researchers to acquire data for small amounts of money from large, unspecified groups of Internet users (Snow et al., 2008; Callison-Burch, 2009).

The use of crowdsourcing for an NLP annotation task required careful definition of the specifics of the task. The individuals who perform these tasks have no specific training, and they are trying to get through as many tasks as they can, so each task must be specified very simply and clearly.

Part of our work was to define a named entity error detection task simply enough that the results would be consistent across anonymous annotators.

2 Methodology

2.1 Process overview

The overall process for running this experiment was as follows (figure 1).

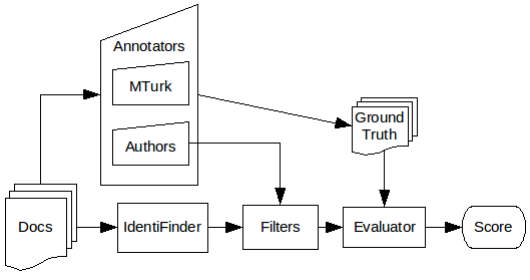

Figure 1: Diagram of data pipeline.

First, we performed an initial performance assessment of IdentiFinder on our domain. We selected 200 articles from an IT trade journal. IdentiFinder was used to tag persons and organizations in these documents. Domain experts (in this case, the authors of this paper) analyzed the entity tags produced by the NER system and annotated the erroneous tags. We built an error reduction system based on our error analysis. We then ran the IdentiFinder output through the error reduction system and evaluated its performance against our annotations.

Next, we constructed an Amazon Mechanical Turk-based interface for naïve web users or "Turkers" to annotate the IdentiFinder entities for errors. We measured the interannotator agreement between the Turkers and the domain experts, and we evaluated the IdentiFinder output and the repaired output against the expert-generated and Turker gold standards.

We selected a new batch of 800 articles and ran IdentiFinder and the filters on them, and we again ran our Mechanical Turk application on the IdentiFinder output. We measured the performance of IdentiFinder and filtered output against the Turker annotations.

2.2 Performance evaluation

Performance is evaluated in terms of standard precision and recall of entities. If the system output contains a person or organization labelled correctly as such, it considers this to be a hit. If it contains a person or organization that is mislabelled or otherwise incorrect in the gold standard annotation, it is a miss. We compute the F-measure as the harmonic mean of precision and recall.

As the IdentiFinder output is the baseline, and we ignore missed entities, by definition the baseline recall is 100%.

3 Experiments and results

Here we delve into further detail about the techniques we used and the results that they yielded. The results are summarized in table 1.

3.1 Baseline performance assessment

We randomly selected 200 documents from InformationWeek, a major weekly magazine in the IT business press. Running them through IdentiFinder produces NIST ACE-standard XML entity markup. We focused on the ENAMEX tags of person and organization type that IdentiFinder produces.

After we annotated the ENAMEX tags for errors, we found that closer inspection of the errors in the IdentiFinder output allowed us to classify the majority of them into three major categories:

Annotator	Collection	System	Precision	Recall	F-measure
Authors	200 document	IdentiFinder only	0.74	1	0.85
Authors	200 document	Filtered	0.86	0.98	0.92
MTurk	200 document	IdentiFinder only	0.69	1	0.82
MTurk	200 document	Filtered	0.79	0.97	0.87
MTurk	800 document	IdentiFinder only	0.67	1	0.80
MTurk	800 document	Filtered	0.77	0.95	0.85

Table 1: Results of evaluation of different document sets against ground truth source by annotation technique.

- IdentiFinder tags words that are simply not named entities.
- IdentiFinder assigns the wrong category (person or organization) to an entity.
- IdentiFinder includes extraneous words in an otherwise correct entity.

The second and third types of error are particularly challenging. An example of the second type is the following:

> **Yahoo** is a reasonably strong competitor to *Google*. It gets about half as much online revenue and search traffic as *Google*, ...

Google is marked twice incorrectly as being a person rather than an organization.

Finally, here is an example of the third error type:

> A San Diego bartender reported that *Bill Gates danced* the night away in his bar on Nov. 11.

IdentiFinder incorrectly marks "danced" as part of a person tag.

We were able to find the precision of IdentiFinder against our annotations: 0.74. This is poorer than the reported performance of IdentiFinder on Wall Street Journal text (Bikel et al., 1999).

3.2 Domain-specific error reduction

We wrote a series of rule-based filters to remove instances of the error types—of which there were many subtypes—described in the previous section. For instance, the third example above was eliminated via the use of a part-of-speech tagger; "danced" was labelled as a verb, and entities with tagged verbs were removed. In the second case, the mislabelling of Google as a person rather than an organization is identified by looking at IdentiFinder's majority labelling of Google throughout the corpus—as an organization. Simple rules about capitalization allow instances like the first example to be identified as errors.

This step increases the precision of the system output to 86%, while only sacrificing a tiny amount of recall. We see that this 10% increase is maintained even on the Mechanical Turk-generated annotations.

3.3 Mechanical Turk tasks

The basic unit of Mechanical Turk is the Human Intelligence Task (HIT). Turkers select HITs presented as web pages and perform the described task. Datacollectors create HITs and pay Amazon to disburse small amounts of money to Turkers who complete them.

We designed our Mechanical Turk process so that every HIT we create corresponds to an IdentiFinder-marked document. Within its corresponding HIT, each document is broken up into paragraphs. Following every paragraph is a table whose rows consist of every person/organization ENAMEX discovered by IdentiFinder and whose columns consist of one of the four categories: "Person," "Organization," "Neither," and "Don't Know." Then for each entity, the user selects exactly one of the four options.

Each HIT is assigned to three different Turkers. Every entity in that HIT is assigned a person or organization ENAMEX tag if two of the three Turkers agreed it was one of those (majority vote); otherwise, it is marked as an invalid entity.

We calculated the agreement between our annotations and those developed from the Turker majority

vote scheme. This yields a Cohen's κ of 0.68. We considered this to be substantial agreement.

After processing the same 200 document set from our own annotation, we found that the precision of IdentiFinder was 69%, but after error reduction, it increased to 79% with only a miniscule loss of known valid entities (recall).

We then took another 800 documents from InformationWeek and ran them through IdentiFinder. We did not annotate these documents ourselves, but instead turned them over to Turkers. IdentiFinder output alone has a 67% precision, but after error reduction, it rises to 77%, and recall is still minimally affected.

4 Discussion

4.1 Benefits

It appears that high-performing NER systems exhibit rather severe domain adaption problems. The performance of IdentiFinder is quite low on the IT business press. However, a simple rule-based system was able to gain 10% improvement in precision with little recall sacrificed. This is a particularly important improvement in applications with low tolerance for erroneous entities.

However, rule-based systems built by experts are known to be vulnerable to new data unseen by the experts. In order to apply this domain-specific error reduction reliably, it has to be tested on data gathered elsewhere. We used crowdsourced data to show that the rule-based system was robust when confronted with data that the designers did not see.

One danger in crowdsourcing is a potential lack of commitment on the part of the annotators, as they attempt to get through tasks as quickly as possible. It turns out that in an NER context, we can design a crowdsourced task that yields relatively reliable results across data sets by ensuring that for every data point, there were multiple annotators making only simple decisions about entity classification.

This method also provides us with a source of easily acquired supervised training data for testing more advanced techniques, if required.

4.2 Costs

It took not more than an estimated two person weeks to complete this work. This includes doing the expert annotations, designing the Mechanical Turk tasks, and building the domain-specific error reduction rules.

For each HIT, each annotator was paid 0.05 USD. For three annotators for 1000 documents, that is 150.00 USD (plus additional small Amazon surcharges and any taxes that apply).

5 Conclusions and Future Work

This work was done on a single publication in a single domain. One future experiment would be to see whether these results are reliable across other publications in the domain. Another set of experiments would be to determine the optimum number of annotators; we assumed three, but cross-domain results may be more stable with more annotators.

Retraining an NER system for a particular domain can be expensive if new annotations must be generated from scratch. While there is work on using advanced machine learning techniques for domain transfer (Guo et al., 2009), simply repairing the the errors *post hoc* via a rule-based system can have a low cost for high gains. This work shows a case where the results are reliable and the verification simple, in a context where reducing false positives is a high priority.

Acknowledgements

This paper is based upon work supported by the National Science Foundation under Grant IIS-0729459. This research was also supported in part by NSF award IIS-0838801.

References

Daniel M. Bikel, Richard Schwartz, and Ralph M. Weischedel. 1999. An algorithm that learns what's in a name. *Mach. Learn.*, 34(1-3).

Chris Callison-Burch. 2009. Fast, cheap, and creative: Evaluating translation quality using Amazon's Mechanical Turk. In *EMNLP 2009*, Singapore, August.

Honglei Guo, Huijia Zhu, Zhili Guo, Xiaoxun Zhang, Xian Wu, and Zhong Su. 2009. Domain adaptation with latent semantic association for named entity recognition. In *NAACL 2009*, Morristown, NJ, USA.

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 *EMNLP 2008*, Morristown, NJ, USA.