A Hybrid Model for Annotating Named Entity Training Corpora

Robert Voyer, **Valerie Nygaard, Will Fitzgerald, and Hannah Copperman**
Microsoft
475 Brannan St. Suite 330
San Francisco, CA 94107, USA
{Robert.Voyer, Valerie.Nygaard, Will.Fitzgerald,
Hannah.Copperman}@microsoft.com

Abstract

In this paper, we present a two-phase, hybrid model for generating training data for Named Entity Recognition systems. In the first phase, a trained annotator labels all named entities in a text irrespective of type. In the second phase, naïve crowdsourcing workers complete binary judgment tasks to indicate the type(s) of each entity. Decomposing the data generation task in this way results in a flexible, reusable corpus that accommodates changes to entity type taxonomies. In addition, it makes efficient use of precious trained annotator resources by leveraging highly available and cost effective crowdsourcing worker pools in a way that does not sacrifice quality.

Keywords: annotation scheme design, annotation tools and systems, corpus annotation, annotation for machine learning

1 Background

The task of Named Entity Recognition (NER) is fundamental to many Natural Language Processing pipelines. Named entity recognizers are most commonly built as machine learned systems that require annotated training data. Manual annotation of named entities is an expensive process, and as a result, much recent work has been done to acquire training corpora automatically from the web. Automatic training corpus acquisition usually requires the existence of one or more first-pass classifiers to identify documents that correspond to a predetermined entity ontology. Using this sort of approach requires an additional set of training data for the initial classifier. More importantly, the quality of our training corpus is limited by the accuracy of any preliminary classifiers. Each automatic step in the process corresponds to increased error in the resulting system. It is not unusual for NE annotation schemas to change as the intended application of NER systems evolves over time – an issue that is rarely mentioned in the literature. Extending named entity ontologies when using an automated approach like the one outlined in (Nothman, 2008), for example, requires non-trivial modifications and extensions to an existing system and may render obsolete any previously collected data.

Our NER system serves a dual purpose; its primary function is to aid our deep natural language parser by identifying single and multiword named entities (NE) in Wikipedia articles. In addition to rendering these phrases as opaque units, the same classifier categorizes these entities as belonging to one of four classes: person, location, organization, and miscellaneous. These class labels serve as additional features that are passed downstream and facilitate parsing. Once identified and labeled, we then add corresponding entries to our semantic index for improved ranking and retrieval.

We scoped each type in the repertoire mentioned above in an attempt to most effectively support our parser and the end-to-end retrieval task. While this taxonomy resembles the one used in the 7th Message Understanding Conference (MUC-7) NER shared task (Chinchor, 1998), our specification is in fact slightly nuanced. For example, the organization and location classes used in our production system are much more limited, disallowing governmental committees, subcommittees, and other organizations that fall under the MUC-7 definition of organization. Indeed, the determination of types to tag and the definitions of these types is very much dependent upon the application for which a given NER system is being designed. Accurate

Proceedings of the Fourth Linguistic Annotation Workshop, ACL 2010, pages 243–246,
Uppsala, Sweden, 15-16 July 2010. ©2010 Association for Computational Linguistics

training and evaluation of NER systems therefore requires application-specific corpora.

Previously, we collected training documents for our system with a more automated two-pass system. In the first pass, we used a set of predefined heuristic rules – based on sequences of part-of-speech (POS) tags and common NE patterns – to identify overlapping candidate spans in the source data. These candidates were then uploaded as tasks to Amazon Mechanical Turk (AMT), in which users were asked to determine if the selected entity was one of 5 specified types. We used majority vote to choose the best decision. Candidates with no majority vote were resubmitted for additional Turker input.

There were a few drawbacks with this system. First and foremost, while the heuristics to identify candidate spans were designed to deliver high recall, it was impossible to have perfect coverage. This imposed an upper bound on the coverage of the system learning from this data. Recall would inevitably decline if we extended our NE taxonomy to include less formulaic types such as titles and band names, for example. One could imagine injecting additional layers of automatic candidate generators into the system to improve recall, each of which would incur additional overhead in judgment cost or complexity. The next issue was quality; many workers tried to scam the system, and others didn't quite understand the task, specifically when it came to differentiating types. The need to address these issues is what led us to our current annotation model.

2 Objective

As the search application supported by our NER system evolved, it became clear both that we would need to be able to support additional name types and that there was a demand for a lighter weight system to identify (especially multiword) NE spans without the need to specify the type. The underlying technology at the core of our existing NER software is well suited for such classification tasks. The central hurdle to extending our system in this way is acquiring a suitable training corpus. Consider the following list of potential classifiers:

1. A single type system capable of identifying product names
2. A targeted system for identifying only movie titles and person names
3. A generic NE span tagger for tagging all named entities
4. A generic-span tagger that tags all multiword named entities

Given that manual annotation is an extremely costly task, we consider optimization of our corpora for reuse while maintaining quality in all supported systems to be a primary goal. Secondly, although throughput is important – it is often said that quantity trumps quality in machine learned systems – the quality of the data is very highly correlated with the accuracy of the systems in question. At the scale of our typical training corpus – one to ten thousand documents – the quality of the data has a significant impact.

3 Methodology

In general, decomposing multifaceted annotation tasks into their fundamental decision points reduces the cognitive load on annotators, increasing annotator throughput while ultimately improving the quality of the marked-up data (Medero et al., 2006). Identifying named entities can be decomposed into two tasks: identifying the span of the entity and determining its type(s). Based on our experience, the first of these tasks requires much more training than the second. The corner cases that arise in determining if any arbitrary sequence of tokens is a named entity make this first task significantly more complex than determining if a given name is, for example, a person name. Decomposing the task into span identification and type judgment has two distinct advantages:

- The span-identification task can be given to more highly trained annotators who are following a specification, while the relatively simpler task can be distributed to naïve/crowdsource judges.
- The task given to the trained annotators goes much more quickly, increasing their throughput.

In a round of pilot tasks, our Corpus Development team performed dual-annotation and complete adjudication on a small sample of 100 documents. We used the output of these tasks to help identify areas of inconsistency in annotator behavior as well as vagueness in the specification. This initial round provided helpful feedback, which we used both to refine the task specification and to help inform the intuitions of our annotators.

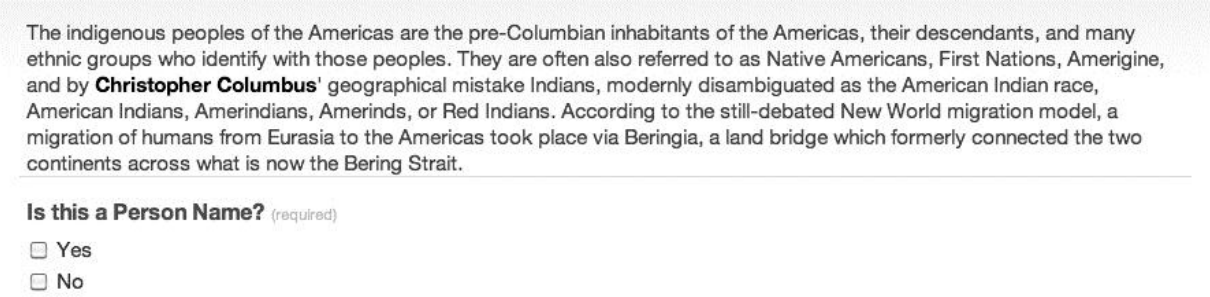

Figure 1: A NE type task in the Crowdflower interface

After these initial tasks, inter-annotator agreement was estimated at 91%, which can be taken to be a reasonable upper bound for our automated system.

In our current process, the data is first marked up by a trained annotator and then checked over by a second trained annotator, and finally undergoes automatic post-processing to catch common errors. Thus, our first step in addressing the issue of poor data quality is to remove the step of automated NE candidate generation and to shift part of the cognitive load of the task from untrained workers to expert annotators.

After span-tagged data has been published by our Corpus Development team, in order to get typed NE annotations for our existing system, we then submit candidate spans along with a two additional sentences of context to workers on AMT. Workers are presented with assignments that require simple binary decisions (Figure 1). Is the selected entity of type X – yes or no? Each unit is presented to at least 5 different workers. We follow this procedure for all labeled spans in our tagged corpus. This entire process can be completed for all of the types that we're interested in – person, location, organization, product, title, etc. Extending this system to cover arbitrary additional types requires simply that we create a new task template and instructions for workers.

Instead of putting these tasks directly onto AMT, we chose to leverage Crowdflower for its added quality control. Crowdflower is a crowdsourcing service built on top of AMT that associates a trust level with workers based on their performance on gold data and uses these trust levels to determine the correctness of worker responses. It provides functionality for retrieving aggregated reports, in which responses are aggregated not based on simple majority voting, but rather by users' trust levels. Our early experiments with this service indicate that it does in fact improve the quality of the output data. An added bonus of their technology is that we can associate confidence levels with the labels produced by workers in their system.

This entire process yields several different annotated versions of the same corpus: an un-typed named entity training corpus, along with an additional corpus for each named entity type. Ideally, each NE span submitted to workers will come back as belonging to zero or one classes. How do we reconcile the fact that our existing system requires a single label per token, when some tokens may in fact fall under multiple categories? Merging the type labels produced by Turkers (with the help of Crowdflower) is an interesting problem in itself. Ultimately, we arrived at a system that allows us to remove type labels that do not meet a confidence threshold, while also biasing certain types over others based on their difficulty. Interestingly, agreement rates among crowdsourcing workers can provide useful insight into the difficulty of labeling some types over others, potentially indicating which types are less precisely scoped. We consistently saw inter-judge agreement rates in the 92%–97% range for person names and locations, while agreement on the less well-defined category of organizations often yielded agreement rates closer to 85%.

4 Initial Results

As a first level comparison of how the new approach affects the overall accuracy of our system, we trained two named entity recognizers. The first system was trained on a subset of the training data collected using the old approach. System 2 was trained on a subset of documents collected using the new approach. Both systems are trained using only a single type – person names. For the former, we randomly selected 200 docs from our previous canonical training set, with the guiding principle that we should have roughly the same number of sentences as exist in our new training corpus (~7400 sentences). Both systems were evaluated against one of our standard, blind measurement sets, hand-

annotated with personal names. The results in table 1 indicate the strict phrase-level precision, recall, and F-score.

It bears mentioning that many NER systems report token-level accuracy or F-score using a flexible phrase-level metric that gives partial credit if either the type classification is correct or the span boundaries are correct. Naturally, these metrics result in higher accuracy numbers when compared to the strict phrase-level metric that we use. Our evaluation tool gives credit to instances where both boundaries and type are correct. Incorrect instances incur at least 2 penalties, counting as at least 1 false positive and 1 false negative, depending on the nature of the error. We optimize our system for high precision.

System	P	R	F-score
Old system	89.7	70.3	78.9
New system	91.6	72.1	80.7

Table 1: Strict phrase-level precision, recall, and F-score.

Our other target application is a generic entity tagger. For this experiment we trained on our complete set of 817 training documents (14,297 sentences) where documents are tagged for all named entities and types are not labeled. We evaluated the resulting system on a blind 100-document measurement set in which generic NE spans have been manually labeled by our Corpus Development team. These results are included in Table 2.

System	P	R	F-score
Generic span	80.3	85.7	82.9

Table 2: Strict phrase-level precision, recall and F-score for generic span tagging.

5 Conclusions

The results indicate that our new approach does indeed produce higher quality training data. An improvement of 1.8 F-score points is relatively significant, particularly given the size of the training set used in this experiment. It is worth noting that our previous canonical training set underwent a round of manual editing after it was discovered that there were significant quality issues. The system trained on the curated data showed marked improvement over previous versions. Given this, we could expect to see a greater disparity between the two systems if we used the output of our previous training data collection system as is.

The generic named entity tagger requires significantly fewer features than type-aware systems, allowing us to improve F-score while also improving runtime performance. We expect to be able to improve precision to acceptable production levels (>90%) while maintaining F-score with a bit more feature engineering, making this system comparable to other state-of-the-art systems.

To extend and improve these initial experiments, we would like to use identical documents for both single-type systems, compare performance on additional NE types, and analyze the learning curve of both systems as we increase the size of the training corpus.

References

Joohui An, Seungwoo Lee, and Gary Geunbae Lee. 2003. Automatic acquisition of named entity tagged corpus from world wide web. *The Companion Volume to the Proceedings of 41st Annual Meeting of the Association for Computational Linguistics*, pages 165-168.

Nancy Chinchor. 1998. Overview of MUC-7. *Proceedings of the 7th Message Understanding Conference.*

Julie Medero, Kazuaki Maeda, Stephanie Strassel, and Christopher Walker. 2006. An Efficient Approach to Gold-Standard Annotation: Decision Points for Complex Tasks. *Proceedings of the Fifth International Conference on Language Resources and Evaluation*.

Joel Nothman, Tara Murphy, and James R. Curran. 2009. Analyzing Wikipedia and Gold-Standard Corpora for NER Training. *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 612-620.

Joel Nothman, James R. Curran, and Tara Murphy. 2008. Transforming Wikipedia into named entity training data. *Proceedings of the Australian Language Technology Workshop*, pages 124–132.

Lee Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. *Proceedings of the Thirteenth Conference on Computational Natural Language Learning* (CoNLL-2009), pages 147-155.