Crowdsourcing syntactic relatedness judgements for opinion mining in the study of information technology adoption

Asad B. Sayeed, Bryan Rusk, Martin Petrov,
Hieu C. Nguyen, Timothy J. Meyer
Department of Computer Science
University of Maryland
College Park, MD 20742 USA
asayeed@cs.umd.edu,brusk@umd.edu,
martin@martinpetrov.com,
{hcnguyen88,tmeyer88}@gmail.com

Amy Weinberg
Center for the Advanced
Study of Language
and Department of Linguistics
University of Maryland
College Park, MD 20742 USA
aweinberg@casl.umd.edu

Abstract

We present an end-to-end pipeline including a user interface for the production of word-level annotations for an opinion-mining task in the information technology (IT) domain. Our pre-annotation pipeline selects candidate sentences for annotation using results from a small amount of trained annotation to bias the random selection over a large corpus. Our user interface reduces the need for the user to understand the "meaning" of opinion in our domain context, which is related to community reaction. It acts as a preliminary buffer against low-quality annotators. Finally, our post-annotation pipeline aggregates responses and applies a more aggressive quality filter.

We present positive results using two different evaluation philosophies and discuss how our design decisions enabled the collection of high-quality annotations under subjective and fine-grained conditions.

1 Introduction

Crowdsourcing permits us to use a bank of anonymous workers with unknown skill levels to perform complex tasks given a simple breakdown of these tasks with user interface design that hides the full task complexity. Use of these techniques is growing in the areas of computational linguistics and information retrieval, particularly since these fields now rely on the collection of large datasets for use in machine learning. Considering the variety of applications, a variety of datasets is needed, but trained, known workers are an expense in principle that must be furnished for each one. Consequently, crowdsourcing offers a way to collect this data cheaply and quickly (Snow et al., 2008; Sayeed et al., 2010a).

We applied crowdsourcing to perform the fine-grained annotation of a domain-specific corpus. Our user interface design and our annotator quality control process allows these anonymous workers to perform a highly subjective task in a manner that correlates their collective understanding of the task to our own expert judgements about it. The path to success provides some illustration of the pitfalls inherent in opinion annotation. Our task is: domain and application-specific sentiment classification at the sub-sentence level—at the word level.

1.1 Opinions

For our purposes, we define opinion mining (sometimes known as sentiment analysis) to be the retrieval of a triple {*source, target, opinion*} (Sayeed et al., 2010b; Pang and Lee, 2008; Kim and Hovy, 2006) in which the *source* is the entity that originated the opinionated language, the *target* is a mention of the entity or concept that is the opinion's topic, and the *opinion* is a value (possibly a structure) that reflects some kind of emotional orientation expressed by the source towards the target.

In much of the recent literature on automatic opinion mining, *opinion* is at best a gradient between positive and negative or a binary classification thereof; further complexity affects the reliability of machine-learning techniques (Koppel and Schler, 2006).

We call opinion mining "fine-grained" when we are attempting to retrieve potentially many different

Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities, pages 69–77,
Portland, OR, USA, 24 June 2011. ©2011 Association for Computational Linguistics

{*source, target, opinion*} triples per document. This is particularly challenging when there are multiple triples even at a sentence level.

1.2 Corpus-based social science

Our work is part of a larger collaboration with social scientists to study the diffusion of information technology (IT) innovations through society by identifying opinion leaders and IT-relevant opinionated language (Rogers, 2003). A key hypothesis is that the language used by opinion leaders causes groups of others to encourage the spread of the given IT concept in the market.

Since the goal of our exercise is to ascertain the correlation between the source's behaviour and that of others, then it may be more appropriate to look at opinion analysis with the view that what we are attempting to discover are the views of an aggregate reader who may otherwise have an interest in the IT concept in question. We thus define an expression of opinion in the following manner:

> A expresses opinion about B if an interested third party C's actions towards B may be affected by A's textually recorded actions, in a context where actions have positive or negative weight.

This perspective runs counter to a widespread view (Ruppenhofer et al., 2008) which has assumed a treatment of opinionated language as an observation of a latent "private state" held by the source. This definition reflects the relationship of sentiment and opinion with the study of social impact and market prediction. We return to the question of how to define opinion in section 6.2.

1.3 Crowdsourcing in sentiment analysis

Paid crowdsourcing is a relatively new trend in computational linguistics. Work exists at the paragraph and document level, and it exists for the Twitter and blog genres (Hsueh et al., 2009).

A key problem in crowdsourcing sentiment analysis is the matter of quality control. A crowdsourced opinion mining task is an attempt to use untrained annotators over a task that is inherently very subjective. It is doubly difficult for specialized domains, since crowdsourcing platforms have no way of directly recruiting domain experts.

Hsueh et al. (2009) present results in quality control over snippets of political blog posts in a task classifying them by sentiment and political alignment. They find that they can use a measurement of annotator noise to eliminate low-quality annotations at this coarse level by reweighting snippet ambiguity scores with noise scores. We demonstrate that we can use a similar annotator quality measure alone to eliminate low-quality annotations on a much finer-grained task.

1.4 Syntactic relatedness

We have a downstream application for this annotation task which involves acquiring patterns in the distribution of opinion-bearing words and targets using machine learning (ML) techniques. In particular, we want to acquire the syntactic relationships between opinion-bearing words and within-sentence targets. Supervised ML techniques require gold standard data annotated in advance.

The Multi-Perspective Question-Answering (MPQA) newswire corpus (Wilson and Wiebe, 2005) and the J. D. Power & Associates (JDPA) automotive review blog post (Kessler et al., 2010) corpus are appropriate because both contain sub-sentence annotations of sentiment-bearing language as text spans. In some cases, they also include links to within-sentence targets. This is an example of an MPQA annotation:

> That was the moment at which the fabric of compassion tore, and worlds cracked apart; when **the contrast and conflict of civilisational values** became so great as to *remove any sense of common ground* - even on which to do battle.

The italicized portion is intended to reflect a negative sentiment about the bolded portion. However, while it is the case that the whole italicized phrase represents a negative sentiment, "remove" appears to represent far more of the negativity than "common" and "ground". While there are techniques that depend on access to entire phrases, our project is to identify sentiment spans at the length of a single word.

2 Data source

Our corpus for this task is a collection of articles from the IT professional magazine, *Information*

Week, from the years 1991 to 2008. This consists of 33K articles of varying lengths including news bulletins, full-length magazine features, and opinion columns. We obtained the articles via an institutional subscription, and reformatted them in XML[1].

Certain IT concepts are particularly significant in the context of the social science application. Our target list consists of 59 IT innovations and concepts. The list includes plurals, common variations, and abbreviations. Examples of IT concepts include "enterprise resource planning" and "customer relationship management". To avoid introducing confounding factors into our results, we only include explicit mentions and omit pronominal coreference.

3 User interface

Our user interface (figure 1) uses a drag-and-drop process through which workers make decisions about whether particular highlighted words within a given sentence reflect an opinion about a particular mentioned IT concept or innovation. The user is presented with a sentence from the corpus surrounded by some before and after context. Underneath the text are four boxes: "No effect on opinion" (none), "Affects opinion positively" (postive), "Affects opinion negatively" (negative), and "Can't tell" (ambiguous).

The worker must drag each highlighted word in the sentence into one of the boxes, as appropriate. If the worker cannot determine the appropriate box for a particular word, she is expected to drag this to the ambiguous box. The worker is presented with detailed instructions which also remind her that most of words in the sentence are not actually likely to be involved in the expression of an opinion about the relevant IT concept[2]. The worker is not permitted to submit the task without dragging all of the highlighted words to one of the boxes. When a word is dragged to a box, the word in context changes colour; the worker can change her mind by clicking an X next to the word in the box.

We used CrowdFlower to manage the task with Amazon Mechanical Turk as its distribution channel. We set CrowdFlower to present three sentences at a time to users. Only users with USA-based IP addresses were permitted to perform the final task.

4 Procedure

In this section, we discuss the data processing pipeline (figure 3) through which we select candidates for annotations and the crowdsourcing interface we present to the end user for classifying individual words into categories that reflect the effect of the word on the worker.

4.1 Data preparation

4.1.1 Initial annotation

Two social science undergraduate students were hired to do annotations on *Information Week* with the original intention of doing all the annotations this way. There was a training period where they annotated about 60 documents in sets of 20 in iterative consultation with one of the authors. Then they were given 142 documents to annotate simultaneously in order to assess their agreement after training.

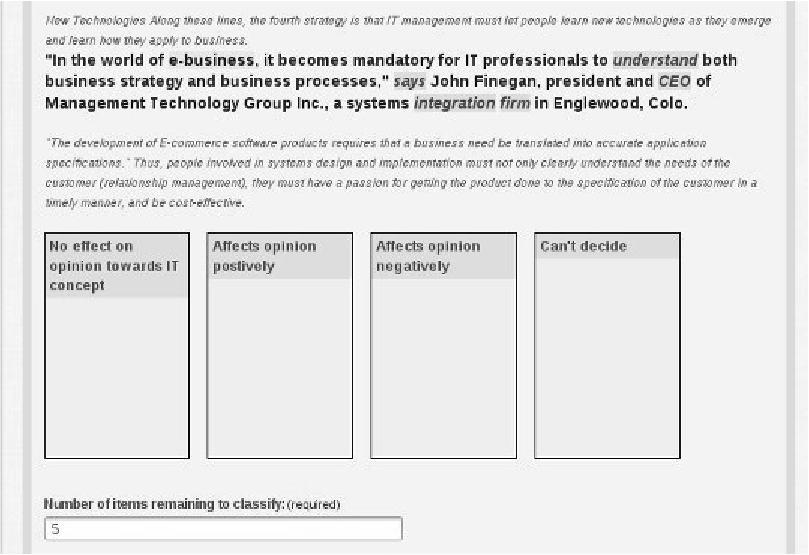
Annotation was performed in Atlas.ti, an annotation tool popular with social science researchers. It was chosen for its familiarity to the social scientists involved in our project and because of their stated preference for using tools that would allow them to share annotations with colleagues. Atlas.ti has limitations, including the inability to create hierarchical annotations. We overcame these limitations using a special notation to connect related annotations. An annotator highlights a sentence that she believes contains an opinion about a mentioned target on one of the lists. She then highlights the mention of the target and, furthermore, highlights the individual words that express the opinion about the target, using the notation to connect related highlights.

4.1.2 Candidate selection

While the use of trained annotators did not produce reliable results (section 6.2) in acceptable time frames, we decided to use the annotations in a process for selecting candidate sentences for crowdsourcing. All 219 sentences that the annotators selected as having opinions about within-sentence IT

[1] We will likely be able to provide a sample of sentence data annotated by our process as a resource once we work out documentation and distribution issues.

[2] We discovered when testing the interface that workers can feel obliged to find a opinion about the selected IT concept. We reduced it by explicitly reminding them that most words do not express a relevant opinion and by placing the none box first.



New Technologies Along these lines, the fourth strategy is that IT management must let people learn new technologies as they emerge and learn how they apply to business.

"In the world of e-business, it becomes mandatory for IT professionals to *understand* both business strategy and business processes," *says* John Finegan, president and *CEO* of Management Technology Group Inc., a systems *integration firm* in Englewood, Colo.

"The development of E-commerce software products requires that a business need be translated into accurate application specifications." Thus, people involved in systems design and implementation must not only clearly understand the needs of the customer (relationship management), they must have a passion for getting the product done to the specification of the customer in a timely manner, and be cost-effective.

No effect on opinion towards IT concept

Affects opinion postively

Affects opinion negatively

Can't decide

Number of items remaining to classify: (required)

5

Figure 1: A work unit presented in grayscale. "E-business" is the IT concept and would be highlighted in blue. The words in question are highlighted in gray background and turn red after they are dragged to the boxes.

concepts were concatenated into a single string and converted into a TFIDF unit vector.

We then selected all the sentences that contain IT concept mentions from the entire Information Week corpus using an OpenNLP 1.4.3 model as our sentence-splitter. This produced approximately 77K sentences. Every sentence was converted into a TFIDF unit vector, and we took the cosine similarity of each sentence with the TFIDF vector. We then ranked the sentences by cosine similarity.

4.1.3 Selecting highlighted words

We ran every sentence through the Stanford part-of-speech tagger. Words that belonged to open classes such as adjectives and verbs were selected along with certain closed-class words such as modals and negation words. These candidate words were highlighted in the worker interface.

We did not want to force workers to classify every single word in a sentence, because this would be too tedious. So we instead randomly grouped the highlighted words into non-overlapping sets of six. (Remainders less than five were dropped from the task.) We call these combinations of sentence, six words, and target IT concept a "highlight group" (figure 2).

Each highlight group represents a task unit which we present to the worker in our crowdsourcing application. We generated 1000 highlight groups from the top-ranked sentences.

The amount of industry attention *paid* to this *new class* of integration software *speaks* volumes about the *need* to extend the *reach* of **ERP** systems.

The *amount* of industry attention paid to this new class of integration *software* speaks *volumes* about the need to *extend* the reach of **ERP** *systems*.

Figure 2: Two highlight groups consisting of the same sentence and concept (ERP) but different non-overlapping sets of candidate words.

4.2 Crowdsourced annotation

4.2.1 Training gold

We used CrowdFlower partly because of its automated quality control process. The bedrock of this process is the annotation of a small amount of gold standard data by the task designers. CrowdFlower randomly selects gold-annotated tasks and presents them to workers amidst other unannotated tasks. Workers are evaluated by the percentage of gold-annotated tasks they perform correctly. The result of a worker performing a task unit is called a "judgement."

Workers are initially presented their gold-annotated tasks without knowing that they are answering a test question. If they get the question wrong, CrowdFlower presents the correct answer to them along with a reason why their answer was an error. They are permitted to write back to the task designer if they disagree with the gold judgement.

This process functions in a manner analogous to the training of a machine-learning system. Furthermore, it permits CrowdFlower to exclude or reject low-quality results. Judgements from a worker who slips below 65% correctness are rated as untrustworthy and not included in the CrowdFlower's results.

We created training gold in the manner recommended by CrowdFlower. We randomly selected 50 highlight groups from the 1000 mentioned in the previous section. We ran these examples through CrowdFlower using the interface we discuss in the next section. Then we used the CrowdFlower gold editor to select 30 highlight groups that contained clear classification decisions where it appeared that the workers were in relative consensus and where we agreed with their decision. Of these, we designated only the clearest-cut classifications as gold, leaving more ambiguous-seeming ones up to the users. For example, in the second highlight group in 2, we would designate *software* and *systems* as none and *extend* as positive in the training gold and the remainder as up to the workers. That would be a "minimum effort" to indicate that the worker understands the task the way we do.

Unfortunately, CrowdFlower has some limitations in the way it processes the responses to gold—it is not possible to define a minimum effort precisely. CrowdFlower's setting either allow us to pass workers based on getting at least one item in each class correct or by placing all items in their correct classes. The latter is too strict a criterion for an inherently subjective task. So we accepted the former. We instead applied our minimum effort criterion in some of our experiments as described in section 4.3.

4.2.2 Full run

We randomly selected another 200 highlight groups and posted them at 12 US cents for each set of three highlight groups, with at least three Mechanical Turk workers seeing each highlight group. The 30 training gold highlight groups were posted along with them. Including CrowdFlower and Amazon fees, the total cost was approximately 60 USD. We permitted only USA-based workers to access the task. Once initiated, the entire task took approximately 24 hours to complete.

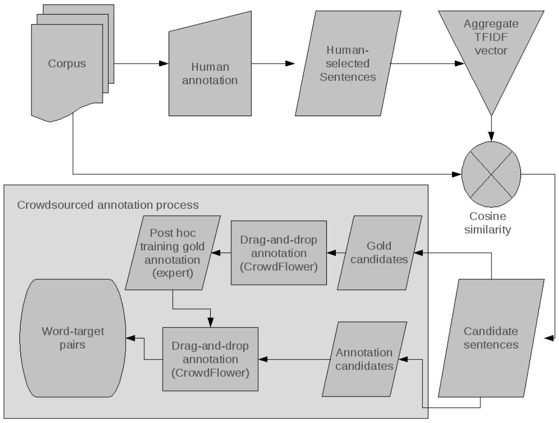


Figure 3: Schematic view of pipeline.

4.3 Post-processing

4.3.1 Aggregation

Each individual worker's ambiguous annotations are converted to none annotations, as the ambiguous box is intended as an outlet for a worker's uncertainty, but we choose to interpret anything that a worker considers too uncertain to be classified as positive or negative as something that is not strongly opinionated under our definitions.

Aggregation is performed by majority vote of the annotators on each word in each highlight group. If no classification obtains more than 50% for a given word, the word is dropped as too ambiguous to be accepted either way as a result. This aggregation has the effect of smoothing out individual annotator differences.

4.3.2 Extended quality control

While CrowdFlower provides a first-pass quality control system for selecting annotators who are doing the task in good faith and with some understanding of the instructions, we wanted particularly to select annotators who would be more likely to be consistent on the most obvious cases without overly constraining them. Even with the same general idea of our intentions, some amount of variation among the annotators is unavoidable; how do we then reject annotations from those workers who pass CrowdFlower's liberal criteria but still do not have an idea of annotation close enough to ours?

Our solution was to score the annotators *post hoc* by their accuracy on our minimum-effort training gold data. Then we progressively dropped the worst n annotators starting from $n = 0$ and measured the quality of the aggregated annotations as per the following section.

5 Results

This task can be interpreted in two different ways: as an annotation task and as a retrieval system. Annotator reliability is an issue insofar as it is important that the annotations themselves conform to a predetermined standard. However, for the machine learning task that is downstream in our processing pipeline, obtaining a consistent pattern is more important than conformance to an explicit definition. We can thus interpret the results as being the output of a system whose computational hardware happens to be a crowd of humans rather than silicon, considering that the time of the "run" is comparable to many automated systems; Amazon Mechanical Turk's slogan is "artificial artificial intelligence" for a reason.

Nevertheless, we evaluated our procedure under both interpretations by comparing against our own annotations in order to assess the quality of our collection, aggregation, and filtering process:

1. **As an annotation task**: we use Cohen's κ between the aggregated and filtered data vs. our annotations in the belief that higher above-chance agreement would imply that the aggregate annotation reflected collective understanding of our definition of sentiment. Considering the inherently subjective nature of this task and the interdependencies inherent in within-sentence judgements, Cohen's κ is not a definitive proof of success or failure.

2. **As a retrieval task**: Relative to our own annotations, we use the standard information retrieval measures of precision, recall, and F-measure (harmonic mean) as well as accuracy. We merge positive and negative annotations into a single opinion-bearing class and measure whether we can retrieve opinion-bearing words while minimizing words that are, in context, not opinion-bearing relative to the given target. (We do not merge the classes for agreement-based evaluation as there was not much overlap between positive and negative classifications.) The particular relative difference between precision and recall will suggest whether the workers had a consistent collective understanding of the task.

It should be noted that the MPQA and the JDPA do not report Cohen's κ for subjective text spans partly for the reason we suggest above: the difficulty of assessing objective agreement on a task in which subjectivity is inherent and desirable. There is also a large class imbalance problem. Both these efforts substitute retrieval-based measures into their assessment of agreement.

We annotated a randomly-selected 30 of the 200 highlight groups on our own. Those 30 had 169 annotated words of which 117 were annotated as none, 35 as positive, and 17 as negative. The results of our process are summarized in table 1.

In the 30 highlight groups, there were 155 total words for which a majority consensus (>50%) was reached. 48 words were determined by us in our own annotation to have opinion weight (positive or negative). There are only 22 annotators who passed CrowdFlower's quality control.

The stringent filter on workers based on their accuracy on our minimum-effort gold annotations has a remarkable effect on the results. As we exclude workers, the F-measure and the Cohen's κ appear to rise, up to a point. By definition, each exclusion raises the threshold score for acceptance. As we cross the 80% threshold, the performance of the system drops noticeably, as the smoothing effect of voting is lost. Opinion-bearing words also reduce in number as the threshold rises as some highlight groups simply have no one voting for them. We achieve our best result in terms of Cohen's κ on dropping the 7 lowest workers. We achieve our highest precision and accuracy after dropping the 10 lowest workers.

Between the 7th and 10th underperforming annotator, we find that precision starts to exceed recall, possibly due to the loss of retrievable words as some highlight groups lose all their annotators. Lost words can be recovered in another round of annotation.

Workers excluded	**No. of words lost (of 48)**	**Prec/Rec/F**	**Acc**	**Cohen's κ**	**Score threshold**
(prior polarity)	N/A	0.87 / 0.38 / 0.53	0.79	*-0.26*	N/A
0	0	0.64 / 0.71 / 0.67	0.79	0.48	0.333
1	0	0.64 / 0.71 / 0.67	0.79	0.48	0.476
3	0	0.66 / 0.73 / 0.69	0.80	0.51	0.560
5	0	0.69 / 0.73 / 0.71	0.81	0.53	0.674
7	2	0.81 / 0.76 / **0.79**	0.86	**0.65**	0.714
10	9	**0.85** / 0.74 / **0.79**	**0.88**	0.54	0.776
12	11	0.68 / 0.68 / 0.68	0.82	0.20	0.820

Table 1: Results by number of workers excluded from the task. The prior polarity baseline comes from a lexicon by Wilson et al. (2005) that is not specific to the IT domain.

6 Discussion

We have been able to show that crowdsourcing a very fine-grained, domain-specific sentiment analysis task with a nonstandard, application-specific definition of sentiment is possible with careful user interface design and mutliple layers of quality control. Our techniques succeed on two different interpretations of the evaluation measure, and we can reclaim any lost words by re-running the task. We used an elaborate processing pipeline before and after annotation in order to accomplish this. In this section, we discuss some aspects of the pipeline that led to the success of this technique.

6.1 Quality

There are three major aspects of our procedure that directly affect the quality of our results: the first-pass quality control in CrowdFlower, the majority-vote aggregation, and the stringent *post hoc* filtering of workers. These interact in particular ways.

The first-pass quality control interacts with the stringent filter in that even if it were possible to have run the stringent filter on CrowdFlower itself, it would probably not have been a good idea. Although we intended the stringent filter to be a minimum effort, it would have rejected workers too quickly. It is technically possible to implement the stringent filtering directly without the CrowdFlower built-in control, but that would have entailed spending an unpredictable amount more money paying for additional unwanted annotations from workers.

Furthermore, the majority-vote aggregation requires that there not be too few annotators; our results show that filtering the workers too aggressively harms the aggregation's smoothing effect. The lesson we take from this is that it can be beneficial to accept some amount of "bad" with the "good" in implementing a very subjective crowdsourcing task.

6.2 Design decisions

Our successful technique for identifying opinionated words was developed after multiple iterations using other approaches which did not succeed in themselves but produced outputs that were amenable to refinement, and so these techniques became part of a larger pipeline. However, the reasons why they did not succeed on their own are illustrative of some of the challenges in both fine-grained domain-specific opinion annotation and in annotation via crowdsourcing under highly subjective conditions.

6.2.1 Direct annotation

We originally intended to stop with the trained annotation we described in 4.1.1, but collecting opinionated sentences in this corpus turned out to be very slow. Despite repeated training rounds, the annotators had a tendency to miss a large number of sentences that the authors found to be relevant. On discussion with the annotators, it turned out that the variable length of the articles made it easy to miss relevant sentences, particularly in the long feature articles likely to contain opinionated language—a kind of "needle-in-a-haystack" problem.

Even worse, however, the annotators were variably conservative about what constituted an opinion. One annotator produced far fewer annotations than the other one—but the majority of her annotations were also annotated by the other one. Discussion with the annotators revealed that one of them simply had a tighter definition of what constituted an opinion. Attempts to define opinion explicitly for them still led to a situations in which one was far more conservative than the other.

6.2.2 Cascaded crowdsourcing technique

Insofar as we were looking for training data for use in downstream machine learning techniques, getting uniform sentence-by-sentence coverage of the corpus was not necessary. There are 77K sentences in this corpus which mention the relevant IT concepts; even if only a fraction of them mention the IT concepts with opinionated language, we would still have a potentially rich source of training data.

Nevertheless the direct annotation with trained annotators provided data for selecting candidate sentences for a more rapid annotation. We used the process in section 4.1.2 and chose the top-ranked sentences. Then we constructed a task design that divided the annotation into two phases. In the first phase, for each candidate sentence, we ask the annotator whether or not the sentence contains opinionated language about the mentioned IT concept. (We permit "unsure" answers.)

In the second phase, for each candidate sentence for which a majority vote of annotators decided that the sentence contained a relevant opinion, we run a second task asking whether particular words (selected as per section 4.1.3) were words directly involved in the expression of the opinion.

We tested this process with the 90 top-ranked sentences. Four individuals in our laboratory answered the "yes/no/unsure" question of the first phase. However, when we took their pairwise Cohen's κ score, no two got more than approximately 0.4. We also took majority votes of each subset of three annotators and found the Cohen's κ between them and the fourth. The highest score was 0.7, but the score was not stable, and we could not trust the results enough to move onto the second phase.

We also ran this first phase through Amazon Mechanical Turk. It turned out that it was far too easy to cheat on this yes/no question, and some workers simply answered "yes" or "no" all the time. Agreement scores of a Turker majority vote vs. one of the authors turned out to yield a Cohen's κ of 0.05—completely unacceptable.

Discussion with the in-laboratory annotators suggested the roots of the problem: it was the same problem as with the direct Atlas.ti annotation we reported in the previous section. It was very difficult for them to agree on what it meant for a sentence to contain an opinion expressed about a particular concept. Opinions about the nature of opinion ranged from very "conservative" to very "liberal." Even explicit definition with examples led annotators to reach very different conclusions. Furthermore, the longer the annotators thought about it, the more confused and uncertain they were about the criterion.

What is an opinion can itself be a matter of opinion. It became clear that without very tight review of annotation and careful task design, asking users an explicit yes/no question about whether a particular concept has a particular opinion mentioned in a particular sentence has the potential to induce overthinking by annotators, despite our variations on the task. The difficulty may also lead to a tendency to cheat. Crowdsourcing allows us to make use of non-expert labour on difficult tasks if we can break the tasks down into simple questions and aggregate non-expert responses, but we needed a somewhat more complex task design in order to eliminate the difficulty of the task and the tendency to cheat.

7 Future work

Foremost among the avenues for future work is experimentation with other vote aggregration and *post hoc* filtering schemes. For example, one type of experiment could be the reweighting of votes by annotator quality rather than the wholesale dropping of annotators. Another could involve the use of general-purpose sentiment analysis lexica to bias the vote aggregation in the manner of work in sentiment domain transfer (Tan et al., 2007).

This work also points to the potential for crowdsourcing in computational linguistics applications beyond opinion mining. Our task is a sentiment-specific instance of a large class of syntactic relatedness problems that may suitable for crowdsourcing. One practical application would be in obtaining training data for coreference detection. Another one may be in the establishment of empirical support for theories about syntactic structure.

Acknowledgements

This paper is based on work supported by the National Science Foundation under grant IIS-0729459.

References

Pei-Yun Hsueh, Prem Melville, and Vikas Sindhwani. 2009. Data quality from crowdsourcing: a study of annotation selection criteria. In *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, HLT '09, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jason S. Kessler, Miriam Eckert, Lyndsay Clark, and Nicolas Nicolov. 2010. The 2010 ICWSM JDPA sentment corpus for the automotive domain. In *4th Int'l AAAI Conference on Weblogs and Social Media Data Workshop Challenge (ICWSM-DWC 2010)*.

Soo-Min Kim and Eduard Hovy. 2006. Extracting opinions, opinion holders, and topics expressed in online news media text. In *SST '06: Proceedings of the Workshop on Sentiment and Subjectivity in Text*, pages 1–8, Morristown, NJ, USA. Association for Computational Linguistics.

Moshe Koppel and Jonathan Schler. 2006. The importance of neutral examples for learning sentiment. *Computational Intelligence*, 22(2).

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2).

Everett M. Rogers. 2003. *Diffusion of Innovations, 5th Edition*. Free Press.

Josef Ruppenhofer, Swapna Somasundaran, and Janyce Wiebe. 2008. Finding the sources and targets of subjective expressions. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odjik, Stelios Piperidis, and Daniel Tapias, editors, *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Asad B. Sayeed, Timothy J. Meyer, Hieu C. Nguyen, Olivia Buzek, and Amy Weinberg. 2010a. Crowdsourcing the evaluation of a domain-adapted named entity recognition system. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Asad B. Sayeed, Hieu C. Nguyen, Timothy J. Meyer, and Amy Weinberg. 2010b. Expresses-an-opinion-about: using corpus statistics in an information extraction approach to opinion mining. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10.

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*.

Songbo Tan, Gaowei Wu, Huifeng Tang, and Xueqi Cheng. 2007. A novel scheme for domain-transfer problem in the context of sentiment analysis. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, CIKM '07, New York, NY, USA.

Theresa Wilson and Janyce Wiebe. 2005. Annotating attributions and private states. In *CorpusAnno '05: Proceedings of the Workshop on Frontiers in Corpus Annotations II*, Morristown, NJ, USA. Association for Computational Linguistics.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *HLT/EMNLP*.