6-26-2012

When Is Word Sense Disambiguation Difficult? A Crowdsourcing Approach

Krishna N. Kaliannan
University of Pennsylvania

When Is Word Sense Disambiguation Difficult? A Crowdsourcing Approach

Abstract
We identified features that drive differential accuracy in word sense disambiguation (WSD) by building regression models using 10,000 coarse-grained WSD instances which were labeled on Mturk. Features predictive of accuracy include properties of the target word (word frequency, part of speech, and number of possible senses), the example context (length), and the Turker's engagement with our task. The resulting model gives insight into which words are difficult to disambiguate. We also show that having many Turkers label the same instance provides at least a partial substitute for more expensive annotation.

Disciplines
Business

When is Word Sense Disambiguation Difficult? A Crowdsourcing Approach

Adam Kapelner **Krishna Kaliannan** **Dean Foster**
The Wharton School of the University of Pennyslvania
Department of Statistics
3730 Walnut Street
Philadelphia, PA 19104
{kapelner, kkali, foster}@wharton.upenn.edu

Lyle Ungar
University of Pennyslvania
Department of Computer Science
200 S. 33rd St 504 Levine
Philadelphia, PA 19104
ungar@cis.upenn.edu

Abstract

We identified features that drive differential accuracy in word sense disambiguation (WSD) by building regression models using 10,000 coarse-grained WSD instances which were labeled on Mturk. Features predictive of accuracy include properties of the target word (word frequency, part of speech, and number of possible senses), the example context (length), and the Turker's engagement with our task. The resulting model gives insight into which words are difficult to disambiguate. We also show that having many Turkers label the same instance provides at least a partial substitute for more expensive annotation.

1 Introduction

Word sense disambiguation (WSD) is the process of identifying the meaning, or "sense," of a word in a written context. In his seminal survey, Navigli (2009) considers WSD an AI-complete problem — a task which is at least as hard as the most difficult problems in artificial intelligence.

There has been a flurry of interest in using pools of anonymous naive human labor, also known as "crowdsourcing," for WSD, especially in situations that are most difficult for algorithms. A thriving pool of crowdsourced labor is Amazon's Mechanical Turk (MTurk), an Internet-based microtask marketplace where the workers (called "Turkers") do simple, one-off tasks (called "human intelligence tasks" or "HITs"), for small payments. See Callison-Burch (2010) for MTurk's use in NLP and Chandler and Kapelner (2010) and Mason and Suri (2011) for further reading on the platform.

Following Akkaya et al. (2010), Parent (2010), and Passonneau et al. (2011), we perform a coarse-grained WSD study on MTurk; we had 1,000 disambiguation instances ("tasks") done by 10 unique Turkers each. We echo previous results that demonstrate Turkers are respectably accurate and that spam is virtually non-existent. We then use regression to identify a variety of factors that effect accuracy: frequency, length, part-of-speech and number of alternative senses of the target word, length of the contextual example, and number of words describing the correct sense. (See figure 1 for an illustration.)

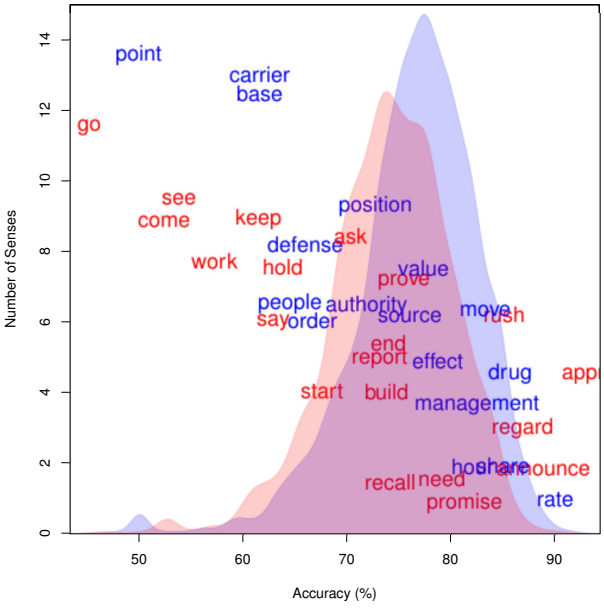

Figure 1: Number of senses vs. predicted accuracy for a sample of the words in our study. Nouns are blue; verbs are red. The densities are smoothed histograms of the noun and verb predicted accuracies.

2 Methods and Data Collection

We selected a subset of the OntoNotes data (Hovy et al., 2006), the SemEval-2007 coarse-grained English Lexical Sample WSD task training data (Pradhan et al., 2007). We picked 1,000 contextual examples ("snippets") at random from the full set of 22,281. Our sample consisted of 590 nouns and 410 verbs. For each snippet, ten WSD instances were completed by ten *unique* Turkers.

2.1 The WSD HIT

We designed a simple WSD task that rendered inside an MTurk HIT.[1] The Turker read one "snippet" with the target word emboldened, and then picked the best choice from among a set of coarse-grained senses (see Figure 2). We gave a blank text box for soliciting optional feedback and there was a submit button below. We term a completed WSD HIT a "disambiguation."

We employed anti-spam and survey bias minimizing tricks to obtain better data. We faded in each word in the snippet and the sense choices one-by-one at 300 words/min.[2] Additionally, we randomized the display order of the sense choices. This reduces "first response alternative bias" as explained in Krosnick (1991), but may decrease accuracy vis-a-vis displaying the senses in descending frequency order (Fellbaum et al., 1997). We also limited workers to be from the US to ensure fluency in English.

3 Results and Data Analysis

We recruited 595 Turkers to work on our tasks and we yielded an average accuracy of 73.4%, which is in line with previously reported experiments. We

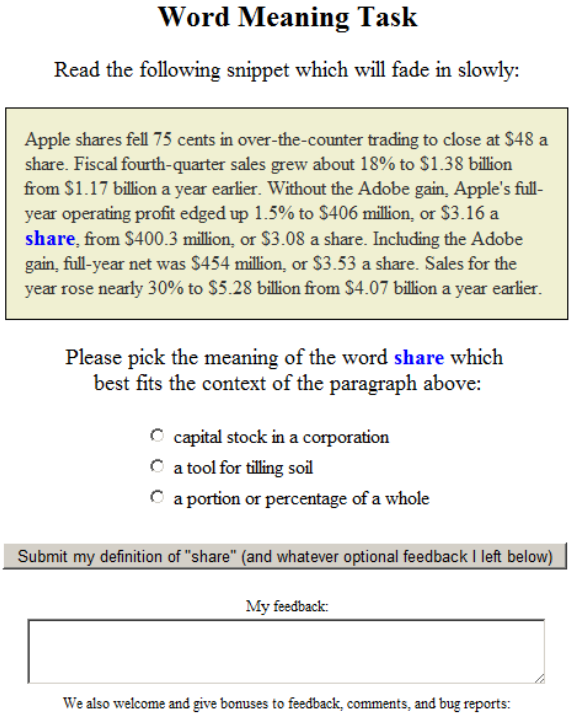

Figure 2: An example of the WSD task that appears inside an MTurk HIT. This was displayed piecewise as each word in the snippet and senses faded-in slowly.

measured inter-tagger agreement (ITA) using the alpha-reliability coefficient (Krippendorff, 1970) to be 0.664 which comports with Chklovski and Mihalcea (2003)'s *Open Mind Word Expert* system. However, this task was specially designed by Hovy et al. (2006) to have 90% ITA. Our measure is significantly less. Naive Turkers should not be expected to be experts.

Due to the high degree of variability in the responses, we were interested in (1) combining Turker responses to boost accuracy (2) evaluating heterogeneity in worker performance (3) investigating which features in the target word, the snippet text, and the text of the sense choices affect accuracy and (4) understanding which characteristics in the Turker's engagement of the task affect accuracy.

3.1 Combining Data to Optimize Prediction

We can combine the 10 unique disambiguation responses for each of the 1000 snippets to yield higher accuracy. Our algorithm is naive — we take the plurality vote and arbitrate ties randomly. This yields an accuracy of 85.7% which is in the ballpark of the

[1] The task was written in HTML and Javascript with a backend written in Ruby 1.9.2 on Rails 3.1 with a MySQL 5.0 database and RTurk 2.4.0. The backend was hosted on an optimized Linux setup by the experts at `engineyard.com`. The HIT was entitled "Tell us the best meaning of a word... do many and earn a lot! Really Easy!", the wage was $0.01, the time limit for each task was seven minutes, and the HITs expired after one hour. We posted batches of 750 new HITs to MTurk hourly upon expiration of the previous batch. Thus, the task was found readily on the homepage which drove the rapid completion.

[2] As Kapelner and Chandler (2010) found, this accomplishes three things: (1) Turkers who plan on cheating will be more likely to leave our task, (2) Turkers will spend more time on the task and, most importantly, (3) Turkers will more carefully read and concentrate on the meaing of the text.

best supervised statistical learning techniques which boast almost 90%.[3] We were also interested in determining the marginal accuracy of each Turker, so we simulated random subsets of two Turkers, three Turkers, etc and employed the same plurality vote. We also simulated the accuracy of the algorithm of collecting data until a plurality is reached. Table 1 illustrates these results.

3.2 Spammers, Superstars, Turker Equality, and Learning Effects

In order to compare our task to previous WSD systems, we investigate the presence of spammers, superstars, and learning effects by plotting the number of disambiguations correct by the number of disambiguations completed in figure 3. To test the null hypothesis that all workers are equal (and thus, average), each worker's *total contributions* are assumed to be drawn from independent Binomial random variables with probability of success $p = 73.4\%$. Does the worker's confidence interval (CI) contain p? Figure 3 reveals that every worker has approximately the same capacity for performing coarse-grained WSD (except for two superstars and two spammers). We echo Akkaya et al. (2010), Snow et al. (2008), and Singh et al. (2002) and conclude there is minimal spammer contribution. Further, we did not detect any learning effects since accuracy does not increase over time.

3.3 WSD Performance and Characteristics of Target Word, the Snippet, and Senses

What makes WSD difficult for naive Turkers? Are there too many senses to choose from? Is the snippet difficult to read? With 10,000 instances across 600 workers, we can attempt to answer these questions.

We first construct the features of interest. For the target word, we use the variables part-of-speech, length (in characters), and log frequency in American English from Davies (2008). For the snippet text, the number of characters. For the correct sense definition text, the number of characters and a feature that tallies the number of definition rephrasings.

[3]See table 3 in Pradhan et al. (2007) for a comparison of all algorithms in the SemEval conference. However, note that these supervised algorithms were given all the training data and then evaluated upon the test data while Turkers were *not* given any previous examples.

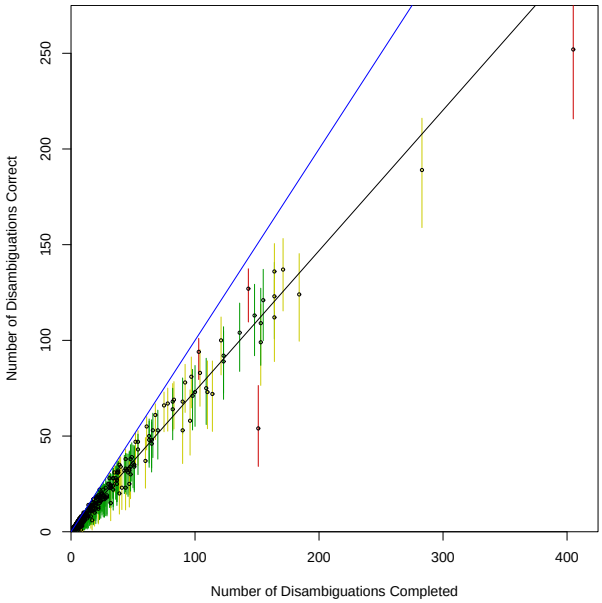

Figure 3: Accuracy of all 595 Turkers. The blue line indicates 100% accuracy and the black line is the average accuracy (73.4%). We plot the Bonferroni-corrected Binomial proportion CIs in green if they include p, red if otherwise, and yellow if they include the non-Bonferroni-corrected exact CIs do not include p.

For example, the word "allot" has a sense with definition text "let, make possible, give permission" which would be counted as three rephrasings. Then, a variable for the number of senses to choose from.

The tasks completed by the same worker are obviously correlated. Therefore, we add a fixed intercept for each of these workers. The result of a ordinary least squares (OLS) regression of correct (as binary) on the variables above is presented in table 2.[4]

We found that, controlling for all other variables, nouns have 8% higher disambiguation accuracy, most likely "because they commonly denote concrete, imagible referents" (Fellbaum et al., 1997). For each extra sense, accuracy suffers 3% which also is expected (ibid). The longer the target word, the snippet, or the correct sense, the more difficult the task. For each extra rephrasing of the definition of the target word, there is a gain of 3.5%. Thus, definitions which include multiple synonyms are easier to understand. As the word becomes more common in the English language, controlling for length of word

[4]We also ran a variety of fixed and random effects linear and logit models, all of which gave the same significance results.

# of Disambiguations	2	3	4	5	6	7	8	9	10	2.4 (1st plurality)
Accuracy	.734	.795	.808	.824	.830	.837	.840	.843	.857	.811

Table 1: Accuracy of the WSD task using plurality voting of different numbers of Turkers. The last column is the accuracy of the variable algorithm: starting with two workers and adding an additional worker until plurality.

	estimate	$\|t\|$
# senses to disambiguate	-2.9%	20.2 ***
# characters in correct sense	-0.065%	2.6 **
# rephrasings in correct sense	3.5%	5.7 ***
log target word frequency	-3.8%	7.8 ***
target word is noun?	8.2%	7.4 ***
# characters in target word	-1.0%	3.5 ***
# characters in snippet	-0.006%	2.6 **

Table 2: OLS regression of instance correctness on features of the target word, snippet, and senses. Fixed effects for each of the 595 Turkers are not shown. ** indicates significance at the $<$ .01 level, *** indicates significance at the $<$0.001 level.

and number of senses, accuracy still suffers. The more prevalent the word in our language, the more likely it will have overlapping senses.

3.4 Turker Characteristics

Are there any characteristics about the Turker's engagement with our task that impacts accuracy? We create the following predictors: time spent on task, the number of words in their optional feedback message, and the number of disambiguations that worker completed. To control for the difficulty of each task, we added 1,000 fixed intercepts — one for each unique task and to control for correlation among the workers, we added a fixed intercept for each worker.

Via OLS,[4] we found that leaving comments does not correspond to higher accuracy, contrary to Kapelner and Chandler (2010), and the number of tasks completed does not impact accuracy (this is as expected; see the discussion in section 3.2). Surprisingly, spending more time on the disambiguation task associates with a significant *reduction* in accuracy ($p < 0.001$).[5] Note that this is *after* we non-parametrically control for instance difficulty and worker ability. For every minute spent, a Turker is 3.6% less likely to answer correctly. We posit two theories: (1) taking breaks leads to loss of concentration (2) the "knee-jerk" response is best to retain (rumination should be discouraged). It is, of course, also possible that we fail to control for individual differences in instance difficulty – maybe some instances are hard for particular workers, as evidenced by their taking longer on them.

[5]We validated this linear approximation by regressing time spent as a polynomial and found the effect to be monotonically decreasing with a flat stretch in the middle.

4 Discussion

We ran a study where American MTurk participants disambiguated words among coarse senses in a sample of the OntoNotes data. Our conclusions about Turker ability are (1) they are as accurate as expected from naive raters but worse than experts (2) they are all roughly equal in ability (3) spam is negligible (4) they do not improve with experience (5) more than ten Turkers must be pooled if we wish to get accuracies that compete with the best machine algorithms. This study indicates that for under $20,000, one could build a system to accurately disambiguate 2-7 million words.

Furthermore, we now have insight into features that induce difficulty in WSD. One should expect worse results if the snippet or the correct sense definition are long, if the correct sense does not provide many synonym examples, if there are many senses to choose from, if the target is a common vocabulary word, or if the target is a verb. Further, it seems that time pressure may increase accuracy. A future experiment that proves this causally may be fruitful.

A Replication

The code, raw data, and analysis scripts are available under GPL2 at `github.com/anonymized`.

Acknowledgments

We thank anon, anon, anon for helpful comments and discussions. Anon thanks the National Science Foundation for the Graduate Research Fellowship that made this work possible.

References

C Akkaya, Alexander Conrad, and J Wiebe. 2010. Amazon mechanical turk for subjectivity word sense disambiguation. Number June, pages 195–203.

C Callison-Burch. 2010. Creating speech and language data with Amazon's Mechanical Turk.

Dana Chandler and Adam Kapelner. 2010. Breaking monotony with meaning: Motivation in crowdsourcing markets. *University of Chicago mimeo*.

Timothy Chklovski and R Mihalcea. 2003. Exploiting agreement and disagreement of human annotators for word sense disambiguation.

M. Davies, 2008. *The Corpus of Contemporary American English: 425 million words, 1990-present. Available online at http://corpus.byu.edu/coca/.*

Christiane Fellbaum, Joachim Grabowski, Shari Landes, and Shari L. 1997. Analysis of a hand-tagging task. In *In Proceedings of ANLP-97 Workshop on Tagging Text with Lexical Semantics*, pages 34–40.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: the 90 percent solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers on XX*, number June, pages 57–60. Association for Computational Linguistics.

Adam Kapelner and Dana Chandler. 2010. Preventing Satisficing in Online Surveys: A "Kapcha" to Ensure Higher Quality Data. In *CrowdConf ACM Proceedings*.

K. Krippendorff. 1970. Estimating the Reliability, Systematic Error and Random Error of Interval Data. *Educational and Psychological Measurement*, 30(1):61–70, April.

Jon A. Krosnick. 1991. Response strategies for coping with the cognitive demands of attitude measures in surveys. *Applied Cognitive Psychology*, 5(3):213–236, May.

Winter Mason and Siddharth Suri. 2011. Conducting behavioral research on Amazon's Mechanical Turk. *Behavior research methods*, June.

Roberto Navigli. 2009. Word sense disambiguation: A Survey. *ACM Computing Surveys*, 41(2):1–69, February.

Gabriel Parent. 2010. Clustering dictionary definitions using Amazon Mechanical Turk. *NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, (June):21–29.

Rebecca J. Passonneau, Vikas Bhardwaj, Ansaf Salleb-Aouissi, and Nancy Ide. 2011. Multiplicity andword sense: Evaluating and learning from multiply labeledword sense annotations.

S Pradhan, E Loper, and Dmitriy Dligach. 2007. Semeval-2007 task-17: English lexical sample, srl and all words. pages 87–92, June.

Push Singh, Thomas Lin, E. Mueller, Grace Lim, T. Perkins, and W. Li Zhu. 2002. Open Mind Common Sense: Knowledge acquisition from the general public. *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE*, pages 1223–1237.

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 '08: Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 254–263, Morristown, NJ, USA. Association for Computational Linguistics.