# Experiments with crowdsourced re-annotation of a POS tagging data set

**Dirk Hovy, Barbara Plank, and Anders Søgaard**
Center for Language Technology
University of Copenhagen
Njalsgade 140, 2300 Copenhagen
{dirk|bplank}@cst.dk, soegaard@hum.ku.dk

## Abstract

Crowdsourcing lets us collect multiple annotations for an item from several annotators. Typically, these are annotations for non-sequential classification tasks. While there has been some work on crowdsourcing named entity annotations, researchers have largely assumed that syntactic tasks such as part-of-speech (POS) tagging cannot be crowdsourced. This paper shows that workers *can* actually annotate sequential data almost as well as experts. Further, we show that the models learned from crowdsourced annotations fare as well as the models learned from expert annotations in downstream tasks.

## 1 Introduction

Training good predictive NLP models typically requires annotated data, but getting professional annotators to build useful data sets is often time-consuming and expensive. Snow et al. (2008) showed, however, that crowdsourced annotations can produce similar results to annotations made by experts. Crowdsourcing services such as Amazon's Mechanical Turk has since been successfully used for various annotation tasks in NLP (Jha et al., 2010; Callison-Burch and Dredze, 2010).

However, most applications of crowdsourcing in NLP have been concerned with classification problems, such as document classification and constructing lexica (Callison-Burch and Dredze, 2010). A large part of NLP problems, however, are structured prediction tasks. Typically, sequence labeling tasks employ a larger set of labels than classification problems, as well as complex interactions between the annotations. Disagreement among annotators is therefore potentially higher, and the task of annotating structured data thus harder.

Only a few recent studies have investigated crowdsourcing sequential tasks; specifically, named entity recognition (Finin et al., 2010; Rodrigues et al., 2013). Results for this are good. However, named entities typically use only few labels (LOC, ORG, and PER), and the data contains mostly non-entities, so the complexity is manageable. The question of whether a more linguistically involved structured task like part-of-speech (POS) tagging can be crowdsourced has remained largely unaddressed.[1]

In this paper, we investigate how well lay annotators can produce POS labels for Twitter data. In our setup, we present annotators with one word at a time, with a minimal surrounding context (two words to each side). Our choice of annotating Twitter data is not coincidental: with the short-lived nature of Twitter messages, models quickly lose predictive power (Eisenstein, 2013), and retraining models on new samples of more representative data becomes necessary. Expensive professional annotation may be prohibitive for keeping NLP models up-to-date with linguistic and topical changes on Twitter. We use a minimum of instructions and require few qualifications.

Obviously, lay annotation is generally less reliable than professional annotation. It is therefore common to aggregate over multiple annotations for the same item to get more robust annotations. In this paper we compare two aggregation schemes, namely majority voting (MV) and MACE (Hovy et al., 2013). We also show how we can use Wiktionary, a crowdsourced lexicon, to filter crowdsourced annotations. We evaluate the annotations in several ways: (a) by testing their accuracy with respect to a gold standard, (b) by evaluating the performance of POS models trained on

[1] One of the reviewers alerted us to an unpublished masters thesis, which uses pre-annotation to reduce tagging to fewer multiple-choice questions. See Related Work section for details.

*Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers)*, pages 377–382,
Baltimore, Maryland, USA, June 23-25 2014. ©2014 Association for Computational Linguistics

the annotations across several existing data sets, as well as (c) by applying our models in downstream tasks. We show that with minimal context and annotation effort, we can produce structured annotations of near-expert quality. We also show that these annotations lead to better POS tagging models than previous models learned from crowdsourced lexicons (Li et al., 2012). Finally, we show that models learned from these annotations are competitive with models learned from expert annotations on various downstream tasks.

## 2 Our Approach

We crowdsource the training section of the data from Gimpel et al. (2011)[2] with POS tags. We use Crowdflower,[3] to collect five annotations for each word, and then find the most likely label for each word among the possible annotations. See Figure 1 for an example. If the correct label is not among the annotations, we are unable to recover the correct answer. This was the case for 1497 instances in our data (cf. the token ":" in the example). We thus report on oracle score, i.e., the best label sequence that could possibly be found, which is correct except for the missing tokens. Note that while we report agreement between the crowdsourced annotations and the crowdsourced annotations, our main evaluations are based on models learned from expert vs. crowdsourced annotations and downstream applications thereof (chunking and NER). We take care in evaluating our models across different data sets to avoid biasing our evaluations to particular annotations. All the data sets used in our experiments are publicly available at `http://lowlands.ku.dk/results/`.

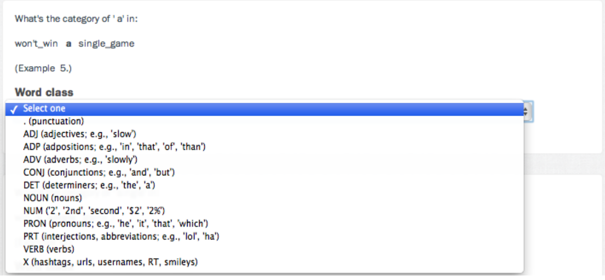
| x | Z | y |
|---|---|---|
| @USER | `NOUN,NOUN,X,NOUN,-,NOUN` | NOUN |
| : | `.,.,-,.,.,.` | X |
| I | `PRON,NOUN,PRON,NOUN,PRON,-` | PRON |
| owe | `VERB,VERB,-,VERB,VERB,VERB` | VERB |
| U | `PRON,X,-,NOUN,NOUN,PRON` | PRON |

$$\theta = 0.9, 0.4, 0.2, 0.8, 0.8, 0.9$$

Figure 1: Five annotations per token, supplied by 6 different annotators (− = missing annotation), gold label $\mathtt{y}$. $\theta$ = competence values for each annotator.

## 3 Crowdsourcing Sequential Annotation

In order to use the annotations to train models that can be applied across various data sets, i.e., making out-of-sample evaluation possible (see Section 5), we follow Hovy et al. (2014) in using the universal tag set (Petrov et al., 2012) with 12 labels.



What's the category of ' a' in:

won't_win **a** single_game

(Example 5.)

**Word class**

✓ Select one
. (punctuation)
ADJ (adjectives; e.g., 'slow')
ADP (adpositions; e.g., 'in', 'that', 'of', 'than')
ADV (adverbs; e.g., 'slowly')
CONJ (conjunctions; e.g., 'and', 'but')
DET (determiners; e.g., 'the', 'a')
NOUN (nouns)
NUM ('2', '2nd', 'second', '$2', '2%')
PRON (pronouns; e.g., 'he', 'it', 'that', 'which')
PRT (interjections, abbreviations; e.g., 'lol', 'ha')
VERB (verbs)
X (hashtags, urls, usernames, RT, smileys)

Figure 2: Screen shot of the annotation interface on Crowdflower

Annotators were given a bold-faced word with two words on either side and asked to select the most appropriate tag from a drop down menu. For each tag, we spell out the name of the syntactic category, and provide a few example words. See Figure 2 for a screenshot of the interface. Annotators were also told that words can belong to several classes, depending on the context. No additional guidelines were given.

Only trusted annotators (in Crowdflower: Bronze skills) that had answered correctly on 4 gold tokens (randomly chosen from a set of 20 gold tokens provided by the authors) were allowed to submit annotations. In total, 177 individual annotators supplied answers. We paid annotators a reward of $0.05 for 10 tokens. The full data set contains 14,619 tokens. Completion of the task took slightly less than 10 days. Contributors were very satisfied with the task (4.5 on a scale from 1 to 5). In particular, they felt instructions were clear (4.4/5), and that the pay was reasonable (4.1/5).

## 4 Label Aggregation

After collecting the annotations, we need to aggregate the annotations to derive a single answer for each token. In the simplest scheme, we choose the majority label, i.e., the label picked by most annotators. In case of ties, we select the final label at random. Since this is a stochastic process, we average results over 100 runs. We refer to this as MAJORITY VOTING (MV). Note that in MV we trust all annotators to the same degree. However, crowdsourcing attracts people with different mo-

[2] http://www.ark.cs.cmu.edu/TweetNLP/

[3] http://crowdflower.com

tives, and not all of them are equally reliable—even the ones with Bronze level. Ideally, we would like to factor this into our decision process.

We use MACE[4] (Hovy et al., 2013) as our second scheme to learn both the most likely answer and a competence estimate for each of the annotators. MACE treats annotator competence and the correct answer as hidden variables and estimates their parameters via EM (Dempster et al., 1977). We use MACE with default parameter settings to give us the weighted average for each annotated example.

Finally, we also tried applying the joint learning scheme in Rodrigues et al. (2013), but their scheme requires that entire sequences are annotated by the same annotators, which we don't have, and it expects BIO sequences, rather than POS tags.

**Dictionaries** Decoding tasks profit from the use of dictionaries (Merialdo, 1994; Johnson, 2007; Ravi and Knight, 2009) by restricting the number of tags that need to be considered for each word, also known as *type constraints* (Täckström et al., 2013). We follow Li et al. (2012) in including Wiktionary information as type constraints into our decoding: if a word is found in Wiktionary, we disregard all annotations that are not licensed by the dictionary entry. If the word is not found in Wiktionary, or if none of its annotations is licensed by Wiktionary, we keep the original annotations. Since we aggregate annotations independently (unlike Viterbi decoding), we basically use Wiktionary as a pre-filtering step, such that MV and MACE only operate on the reduced annotations.

## 5 Experiments

Each of the two aggregation schemes above produces a final label sequence $\hat{\mathbf{y}}$ for our training corpus. We evaluate the resulting annotated data in three ways.

1. We compare $\hat{\mathbf{y}}$ to the available expert annotation on the *training* data. This tells us how similar lay annotation is to professional annotation.

2. Ultimately, we want to use structured annotations for supervised training, where annotation quality influences model performance on held-out *test* data. To test this, we train a CRF model (Lafferty et al., 2001) with simple orthographic features and word clusters (Owoputi et al., 2013) on the annotated Twitter data described in Gimpel et al. (2011). Leaving out the dedicated test set to avoid in-sample bias, we evaluate our models across three data sets: RITTER (the 10% test split of the data in Ritter et al. (2011) used in Derczynski et al. (2013)), the test set from Foster et al. (2011), and the data set described in Hovy et al. (2014).

We will make the preprocessed data sets available to the public to facilitate comparison. In addition to a supervised model trained on expert annotations, we compare our tagging accuracy with that of a weakly supervised system (Li et al., 2012) re-trained on 400,000 unlabeled tweets to adapt to Twitter, but using a crowdsourced lexicon, namely Wiktionary, to constrain inference. We use parameter settings from Li et al. (2012), as well as their Wikipedia dump, available from their project website.[5]

3. POS tagging is often the first step for further analysis, such as chunking, parsing, etc. We test the downstream performance of the POS models from the previous step on chunking and NER. We use the models to annotate the training data portion of each task with POS tags, and use them as features in a chunking and NER model. For both tasks, we train a CRF model on the respective (POS-augmented) training set, and evaluate it on several held-out test sets. For chunking, we use the test sets from Foster et al. (2011) and Ritter et al. (2011) (with the splits from Derczynski et al. (2013)). For NER, we use data from Finin et al. (2010) and again Ritter et al. (2011). For chunking, we follow Sha and Pereira (2003) for the set of features, including token and POS information. For NER, we use standard features, including POS tags (from the previous experiments), indicators for hyphens, digits, single quotes, upper/lowercase, 3-character prefix and suffix information, and Brown word cluster features[6] with 2,4,8,16 bitstring prefixes estimated from a large Twitter corpus (Owoputi et al., 2013). We report macro-averages over all these data sets.

## 6 Results

**Agreement with expert annotators** Table 1 shows the accuracy of each aggregation compared to the gold labels. The crowdsourced annotations

[4] http://www.isi.edu/publications/licensed-sw/mace/

[5] https://code.google.com/p/wikily-supervised-pos-tagger/

[6] http://www.ark.cs.cmu.edu/TweetNLP/

| | |
|---|---|
| majority | 79.54 |
| MACE-EM | 79.89 |
| majority+Wiktionary | 80.58 |
| MACE-EM+Wiktionary | 80.75 |
| oracle | 89.63 |

Table 1: Accuracy (%) of different annotations wrt gold data

aggregated using MV agree with the expert annotations in 79.54% of the cases. If we pre-filter the data using Wiktionary, the agreement becomes 80.58%. MACE leads to higher agreement with expert annotations under both conditions (79.89 and 80.75). The small difference indicates that annotators are consistent and largely reliable, thus confirming the Bronze-level qualification we required. Both schemes cannot recover the correct answer for the 1497 cases where none of the crowdsourced labels matched the gold label, i.e. $y \notin \mathbf{Z}_i$. The best possible result either of them could achieve (the *oracle*) would be matching all but the missing labels, an agreement of 89.63%.

Most of the cases where the correct label was not among the annotations belong to a small set of confusions. The most frequent was mislabeling "*:*" and "...", both mapped to *X*. Annotators mostly decided to label these tokens as punctuation (.). They also predominantly labeled *your*, *my* and *this* as *PRON* (for the former two), and a variety of labels for the latter, when the gold label is *DET*.

| | RITTER | FOSTER | HOVY |
|---|---|---|---|
| Li et al. (2012) | 73.8 | 77.4 | 79.7 |
| MV | 80.5 | 81.6 | 83.7 |
| MACE | 80.4 | 81.7 | 82.6 |
| MV+Wik | 80.4 | 82.1 | 83.7 |
| MACE+Wik | 80.5 | 81.9 | 83.7 |
| Upper bounds | | | |
| oracle | 82.4 | 83.7 | 85.1 |
| gold | 82.6 | 84.7 | 86.8 |

Table 2: POS tagging accuracies (%).

**Effect on POS Tagging Accuracy** Usually, we don't want to match a gold standard, but we rather want to create new annotated training data. Crowdsourcing matches our gold standard to about 80%, but the question remains how useful this data is when training models on it. After all, inter-annotator agreement among professional annotators on this task is only around 90% (Gimpel et al., 2011; Hovy et al., 2014). In order to evaluate how much each aggregation scheme influences tagging performance of the resulting model, we train separate models on each scheme's annotations and test on the same four data sets. Table 2 shows the results. Note that the differences between the four schemes are insignificant. More importantly, however, POS tagging accuracy using crowdsourced annotations are on average *only 2.6% worse* than gold using professional annotations. On the other hand, performance is *much better* than the weakly supervised approach by Li et al. (2012), which only relies on a crowdsourced POS lexicon.

| POS model from | CHUNKING | NER |
|---|---|---|
| MV | 74.80 | 75.74 |
| MACE | 75.04 | 75.83 |
| MV+Wik | 75.86 | 76.08 |
| MACE+Wik | 75.86 | 76.15 |
| Upper bounds | | |
| oracle | 76.22 | 75.85 |
| gold | 79.97 | 75.81 |

Table 3: Downstream accuracy for chunking (l) and NER (r) of models using POS.

**Downstream Performance** Table 3 shows the accuracy when using the POS models trained in the previous evaluation step. Note that we present the average over the two data sets used for each task. Note also how the Wiktionary constraints lead to improvements in downstream performance. In chunking, we see that using the crowdsourced annotations leads to worse performance than using the professional annotations. For NER, however, we find that some of the POS taggers trained on aggregated data produce better NER performance than POS taggers trained on expert-annotated gold data. Since the only difference between models are the respective POS features, the results suggest that at least for some tasks, POS taggers learned from crowdsourced annotations may be *as good* as those learned from expert annotations.

## 7 Related Work

There is considerable work in the literature on modeling answer correctness and annotator competence as latent variables (Dawid and Skene, 1979; Smyth et al., 1995; Carpenter, 2008; Whitehill et al., 2009; Welinder et al., 2010; Yan et al., 2010; Raykar and Yu, 2012). Rodrigues et al. (2013) recently presented a sequential model for this. They estimate annotator competence as latent variables in a CRF model using EM. They evaluate their approach on synthetic and NER data annotated on Mechanical Turk, showing improvements over the MV baselines and the multi-label model by Dredze et al. (2009). The latter do not model annotator reliability but rather model label priors by integrating them into the CRF objective, and re-estimating them during learning. Both require annotators to supply a full sentence, while we use minimal context, which requires less annotator commitment and makes the task more flexible. Unfortunately, we could not run those models on our data due to label incompatibility and the fact that we typically do not have complete sequences annotated by the same annotators.

Mainzer (2011) actually presents an earlier paper on crowdsourcing POS tagging. However, it differs from our approach in several ways. It uses the Penn Treebank tag set to annotate Wikipedia data (which is much more canonical than Twitter) via a Java applet. The applet automatically labels certain categories, and only presents the users with a series of multiple choice questions for the remainder. This is highly effective, as it eliminates some sources of possible disagreement. In contrast, we do not pre-label any tokens, but always present the annotators with all labels.

## 8 Conclusion

We use crowdsourcing to collect POS annotations with minimal context (five-word windows). While the performance of POS models learned from this data is still slightly below that of models trained on expert annotations, models learned from aggregations approach oracle performance for POS tagging. In general, we find that the use of a dictionary tends to make aggregations more useful, irrespective of aggregation method. For some downstream tasks, models using the aggregated POS tags perform even better than models using expert-annotated tags.

## Acknowledgments

We would like to thank the anonymous reviewers for valuable comments and feedback. This research is funded by the ERC Starting Grant LOWLANDS No. 313695.

## References

Chris Callison-Burch and Mark Dredze. 2010. Creating Speech and Language Data With Amazon's Mechanical Turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*.

Bob Carpenter. 2008. Multilevel Bayesian models of categorical data annotation. Technical report, LingPipe.

A. Philip Dawid and Allan M. Skene. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied Statistics*, pages 20–28.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

Leon Derczynski, Alan Ritter, Sam Clark, and Kalina Bontcheva. 2013. Twitter part-of-speech tagging for all: overcoming sparse and noisy data. In *RANLP*.

Mark Dredze, Partha Pratim Talukdar, and Koby Crammer. 2009. Sequence learning from data with multiple labels. In *ECML/PKDD Workshop on Learning from Multi-Label Data*.

Jacob Eisenstein. 2013. What to do about bad language on the internet. In *NAACL*.

Tim Finin, Will Murnane, Anand Karandikar, Nicholas Keller, Justin Martineau, and Mark Dredze. 2010. Annotating named entities in Twitter data with crowdsourcing. In *NAACL-HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*.

Jennifer Foster, Ozlem Cetinoglu, Joachim Wagner, Josef Le Roux, Joakim Nivre, Deirde Hogan, and Josef van Genabith. 2011. From news to comments: Resources and benchmarks for parsing the language of Web 2.0. In *IJCNLP*.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. 2011. Part-of-Speech Tagging for Twitter: Annotation, Features, and Experiments. In *ACL*.

Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. 2013. Learning whom to trust with MACE. In *NAACL*.

Dirk Hovy, Barbara Plank, and Anders Søgaard. 2014. When pos datasets don t add up: Combatting sample bias. In *LREC*.

Mukund Jha, Jacob Andreas, Kapil Thadani, Sara Rosenthal, and Kathleen McKeown. 2010. Corpus creation for new genres: A crowdsourced approach to pp attachment. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*. Association for Computational Linguistics.

Mark Johnson. 2007. Why doesn't EM find good HMM POS-taggers. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *ICML*.

Shen Li, João Graça, and Ben Taskar. 2012. Wiki-ly supervised part-of-speech tagging. In *EMNLP*.

Jacob Emil Mainzer. 2011. Labeling parts of speech using untrained annotators on mechanical turk. Master's thesis, The Ohio State University.

Bernard Merialdo. 1994. Tagging English text with a probabilistic model. *Computational linguistics*, 20(2):155–171.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *NAACL*.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *LREC*.

Sujith Ravi and Kevin Knight. 2009. Minimized Models for Unsupervised Part-of-Speech Tagging. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. Association for Computational Linguistics.

Vikas C. Raykar and Shipeng Yu. 2012. Eliminating Spammers and Ranking Annotators for Crowdsourced Labeling Tasks. *Journal of Machine Learning Research*, 13:491–518.

Alan Ritter, Sam Clark, Oren Etzioni, et al. 2011. Named entity recognition in tweets: an experimental study. In *EMNLP*.

Filipe Rodrigues, Francisco Pereira, and Bernardete Ribeiro. 2013. Sequence labeling with multiple annotators. *Machine Learning*, pages 1–17.

Fei Sha and Fernando Pereira. 2003. Shallow parsing with conditional random fields. In *NAACL*.

Padhraic Smyth, Usama Fayyad, Mike Burl, Pietro Perona, and Pierre Baldi. 1995. Inferring ground truth from subjective labelling of Venus images. *Advances in neural information processing systems*, pages 1085–1092.

Rion Snow, Brendan O'Connor, Dan 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*. Association for Computational Linguistics.

Oscar Täckström, Dipanjan Das, Slav Petrov, Ryan McDonald, and Joakim Nivre. 2013. Token and type constraints for cross-lingual part-of-speech tagging. *TACL*, Mar(1):1–12.

Peter Welinder, Steve Branson, Serge Belongie, and Pietro Perona. 2010. The multidimensional wisdom of crowds. In *NIPS*.

Jacob Whitehill, Paul Ruvolo, Tingfan Wu, Jacob Bergsma, and Javier Movellan. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. *Advances in Neural Information Processing Systems*, 22:2035–2043.

Yan Yan, Rómer Rosales, Glenn Fung, Mark Schmidt, Gerardo Hermosillo, Luca Bogoni, Linda Moy, and Jennifer Dy. 2010. Modeling annotator expertise: Learning when everybody knows a bit of something. In *International Conference on Artificial Intelligence and Statistics*.