Learning part-of-speech taggers with inter-annotator agreement loss

Barbara Plank, Dirk Hovy, Anders Søgaard
Center for Language Technology
University of Copenhagen, Denmark
Njalsgade 140, DK-2300 Copenhagen S
bplank@cst.dk,dirk@cst.dk,soegaard@hum.ku.dk

Abstract

In natural language processing (NLP) annotation projects, we use inter-annotator agreement measures and annotation guidelines to ensure consistent annotations. However, annotation guidelines often make linguistically debatable and even somewhat arbitrary decisions, and inter-annotator agreement is often less than perfect. While annotation projects usually specify how to deal with linguistically debatable phenomena, annotator disagreements typically still stem from these "hard" cases. This indicates that some errors are more debatable than others. In this paper, we use small samples of doubly-annotated part-of-speech (POS) data for Twitter to estimate annotation reliability and show how those metrics of *likely* inter-annotator agreement can be implemented in the loss functions of POS taggers. We find that these cost-sensitive algorithms perform better across annotation projects and, more surprisingly, even on data annotated according to the same guidelines. Finally, we show that POS tagging models sensitive to inter-annotator agreement perform better on the downstream task of chunking.

1 Introduction

POS-annotated corpora and treebanks are collections of sentences analyzed by linguists according to some linguistic theory. The specific choice of linguistic theory has dramatic effects on downstream performance in NLP tasks that rely on syntactic features (Elming et al., 2013). Variation across annotated corpora in linguistic theory also poses challenges to intrinsic evaluation (Schwartz et al., 2011; Tsarfaty et al., 2012), as well as for languages where available resources are mutually inconsistent (Johansson, 2013). Unfortunately, there is no grand unifying linguistic theory of how to analyze the structure of sentences. While linguists agree on certain things, there is still a wide range of unresolved questions. Consider the following sentence:

(1) @GaryMurphyDCU of @DemMattersIRL will take part in a panel discussion on October 10th re the aftermath of #seanref ...

While linguists will agree that *in* is a preposition, and *panel discussion* a compound noun, they are likely to disagree whether *will* is heading the main verb *take* or vice versa. Even at a more basic level of analysis, it is not completely clear how to assign POS tags to each word in this sentence: is *part* a particle or a noun; is *10th* a numeral or a noun?

Some linguistic controversies may be resolved by changing the vocabulary of linguistic theory, e.g., by leaving out numerals or introducing *ad hoc* parts of speech, e.g. for English *to* (Marcus et al., 1993) or words ending in *-ing* (Manning, 2011). However, standardized label sets have practical advantages in NLP (Zeman and Resnik, 2008; Zeman, 2010; Das and Petrov, 2011; Petrov et al., 2012; McDonald et al., 2013).

For these and other reasons, our annotators (even when they are trained linguists) often disagree on how to analyze sentences. The strategy in most previous work in NLP has been to monitor and later resolve disagreements, so that the final labels are assumed to be reliable when used as input to machine learning models.

Our approach

Instead of glossing over those annotation disagreements, we consider what happens if we embrace the uncertainty exhibited by human annotators

when learning predictive models from the annotated data.

To achieve this, we incorporate the uncertainty exhibited by annotators in the training of our model. We measure inter-annotator agreement on small samples of data, then incorporate this in the loss function of a structured learner to reflect the confidence we can put in the annotations. This provides us with cost-sensitive online learning algorithms for inducing models from annotated data that take inter-annotator agreement into consideration.

Specifically, we use online structured perceptron with drop-out, which has previously been applied to POS tagging and is known to be robust across samples and domains (Søgaard, 2013a). We incorporate the inter-annotator agreement in the loss function either as inter-annotator $F1$-scores or as the confusion probability between annotators (see Section 3 below for a more detailed description). We use a small amounts of doubly-annotated Twitter data to estimate $F1$-scores and confusion probabilities, and incorporate them during training via a modified loss function. Specifically, we use POS annotations made by two annotators on a set of 500 newly sampled tweets to estimate our agreement scores, and train models on existing Twitter data sets (described below). We evaluate the effect of our modified training by measuring intrinsic as well as downstream performance of the resulting models on two tasks, namely named entity recognition (NER) and chunking, which both use POS tags as input features.

2 POS-annotated Twitter data sets

The vast majority of POS-annotated resources across languages contain mostly newswire text. Some annotated Twitter data sets do exist for English. Ritter et al. (2011) present a manually annotated data set of 16 thousand tokens. They do not report inter-annotator agreement. Gimpel et al. (2011) annotated about 26 thousand tokens and report a raw agreement of 92%. Foster et al. (2011) annotated smaller portions of data for cross-domain evaluation purposes. We refer to the data as RITTER, GIMPEL and FOSTER below.

In our experiments, we use the RITTER splits provided by Derczynski et al. (2013), and the October splits of the GIMPEL data set, version 0.3. We train our models on the concatenation of RITTER-TRAIN and GIMPEL-TRAIN and evaluate them on the remaining data, the dev and test set provided by Foster et al. (2011) as well as an in-house annotated data set of 3k tokens (see below).

The three annotation efforts (Ritter et al., 2011; Gimpel et al., 2011; Foster et al., 2011) all used different tagsets, however, and they also differ in tokenization, as well as a wide range of linguistic decisions. We mapped all the three corpora to the universal tagset provided by Petrov et al. (2012) and used the same dummy symbols for numbers, URLs, etc., in all the data sets. Following (Foster et al., 2011), we consider URLs, usernames and hashtags as NOUN. We did not change the tokenization.

The data sets differ in how they analyze many of the linguistically hard cases. Consider, for example, the analysis of *will you come out to* in GIMPEL and RITTER (Figure 1, top). While Gimpel et al. (2011) tag *out* and *to* as adpositions, Ritter et al. (2011) consider them particles. What is the right analysis depends on the compositionality of the construction and the linguistic theory one subscribes to.

Other differences include the analysis of abbreviations (PRT in GIMPEL; X in RITTER and FOSTER), colon (X in GIMPEL; punctuation in RITTER and FOSTER), and emoticons, which can take multiple parts of speech in GIMPEL, but are always X in RITTER, while they are absent in FOSTER. GIMPEL-TRAIN and RITTER-TRAIN are also internally inconsistent. See the bottom of Figure 1 for examples and Hovy et al. (2014) for a more detailed discussion on differences between the data sets.

Since the mapping to universal tags could potentially introduce errors, we also annotated a data set directly using universal tags. We randomly selected 200 tweets collected over the span of one day, and had three annotators tag this set. We split the data in such a way that each annotator had 100 tweets: two annotators had disjoint sets, the third overlapped 50 items with each of the two others. In this way, we obtained an initial set of 100 doubly-annotated tweets. The annotators were *not* provided with annotation guidelines. After the first round of annotations, we achieved a raw agreement of 0.9, a Cohen's κ of 0.87, and a Krippendorff's α of 0.87. We did one pass over the data to adjudicate the cases where annotators disagreed,

		will	you	come	out	to	the	
GIMPEL	...	VERB	PRON	VERB	**ADP**	**ADP**	DET	...
RITTER		VERB	PRON	VERB	**PRT**	**PRT**	DET	

RITTER

	you/PRON	**come/VERB**	**out/PRT**	to/PRT	
...	it/PRON	**comes/VERB**	**out/ADP**	nov/NOUN	...

GIMPEL

	Advances/NOUN	and/CONJ	**Social/NOUN**	**Media/NOUN**	.../X	
...	Journalists/NOUN	and/CONJ	**Social/ADJ**	**Media/NOUN**	experts/NOUN	...

Figure 1: Annotation differences between (top) and within (bottom) two available Twitter POS data sets.

or where they had flagged their choice as debatable. The final data set (`lowlands.test`), referred below to as INHOUSE, contained 3,064 tokens (200 tweets) and is publicly available at `http://bitbucket.org/lowlands/costsensitive-data/`, along with the data used to compute inter-annotator agreement scores for learning cost-sensitive taggers, described in the next section.

3 Computing agreement scores

Gimpel et al. (2011) used 72 doubly-annotated tweets to estimate inter-annotator agreement, and we also use doubly-annotated data to compute agreement scores. We randomly sampled 500 tweets for this purpose. Each tweet was annotated by two annotators, again using the universal tag set (Petrov et al., 2012). All annotators were encouraged to use their own best judgment rather than following guidelines or discussing difficult cases with each other. This is in contrast to Gimpel et al. (2011), who used annotation guidelines. The average inter-annotator agreement was 0.88 for raw agreement, and 0.84 for Cohen's κ. Gimpel et al. (2011) report a raw agreement of 0.92.

We use two metrics to provide a more detailed picture of inter-annotator agreement, namely $F1$-*scores* between annotators on individual parts of speech, and *tag confusion probabilities*, which we derive from confusion matrices.

The $F1$-score relates to precision and recall in the usual way, i.e, as the harmonic mean between those two measure. In more detail, given two annotators A_1 and A_2, we say the precision

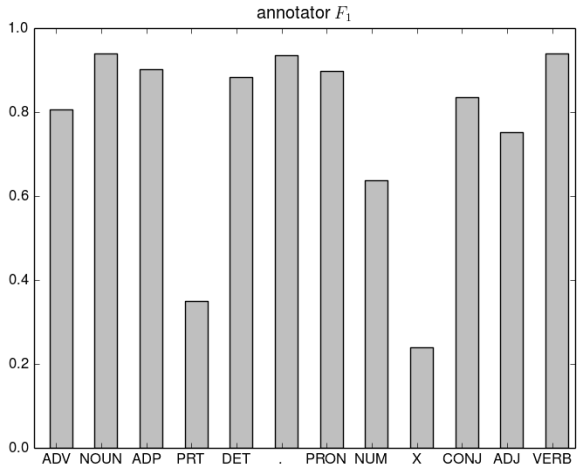

Figure 2: Inter-annotator $F1$-scores estimated from 500 tweets.

of A_1 relative to A_2 with respect to POS tag T in some data set X, denoted $Prec_T(A_1(X), A_2(X))$, is the number of tokens both A_1 and A_2 predict to be T over the number of times A_1 predicts a token to be T. Similarly, we define the recall with respect to some tag T, i.e., $Rec_T(A_1(X), A_2(X))$, as the number of tokens both A_1 and A_2 predict to be T over the number of times A_2 predicts a token to be T. The only difference with respect to standard precision and recall is that the gold standard is replaced by a second annotator, A_2. Note that $Prec_T(A_1(X), A_2(X)) = Rec_T(A_2(X), A_1(X))$. It follows from all of the above that the $F1$-score is symmetrical, i.e., $F1_T(A_1(X), A_2(X)) = F1_T(A_2(X), A_1(X))$.

The inter-annotator $F1$-scores over the 12 POS tags in the universal tagset are presented in Figure 2. It shows that there is a high agreement for nouns, verbs and punctuation, while the agree-

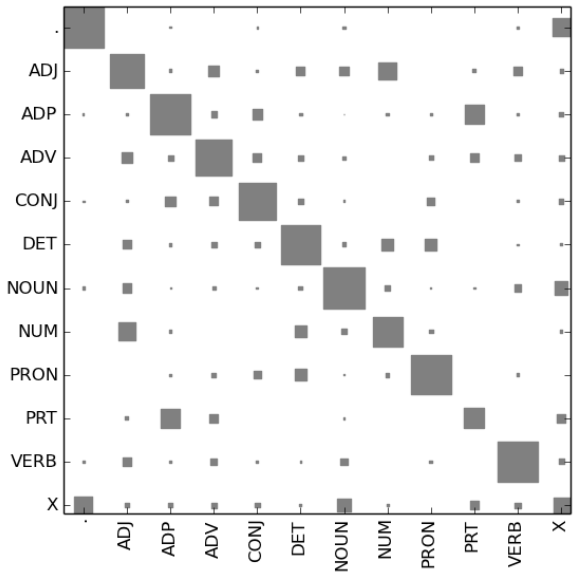

Figure 3: Confusion matrix of POS tags obtained from 500 doubly-annotated tweets.

ment is low, for instance, for particles, numerals and the X tag.

We compute tag confusion probabilities from a confusion matrix over POS tags like the one in Figure 3. From such a matrix, we compute the probability of confusing two tags t_1 and t_2 for some data point $\mathbf{x}$, i.e. $P(\{A_1(\mathbf{x}), A_2(\mathbf{x})\} = \{t_1, t_2\})$ as the mean of $P(A_1(\mathbf{x}) = t_1, A_2(\mathbf{x}) = t_2)$ and $P(A_1(\mathbf{x}) = t_2, A_2(\mathbf{x}) = t_1)$, e.g., the confusion probability of two tags is the mean of the probability that annotator A_1 assigns one tag and A_2 another, and vice versa.

We experiment with both agreement scores ($F1$ and confusion matrix probabilities) to augment the loss function in our learner. The next section describes this modification in detail.

4 Inter-annotator agreement loss

We briefly introduce the cost-sensitive perceptron classifier. Consider the weighted perceptron loss on our ith example $\langle \mathbf{x}_i, y_i \rangle$ (with learning rate $\alpha = 1$), $L_{\mathbf{w}}(\langle \mathbf{x}_i, y_i \rangle)$:

$$\gamma(\text{sign}(\mathbf{w} \cdot \mathbf{x}_i), y_i) \max(0, -y_i \mathbf{w} \cdot \mathbf{x}_i)$$

In a non-cost-sensitive classifier, the weight function $\gamma(y_j, y_i) = 1$ for $1 \leq i \leq N$. The

1: $X = \{\langle \mathbf{x}_i, y_i \rangle\}_{i=1}^{N}$ with $\mathbf{x}_i = \langle x_i^1, \ldots, x_i^m \rangle$
2: I iterations
3: $\mathbf{w} = \langle 0 \rangle^m$
4: **for** $iter \in I$ **do**
5: **for** $1 \leq i \leq N$ **do**
6: $\hat{y} = \arg\max_{y \in \mathcal{Y}} \mathbf{w} \cdot \Phi(\mathbf{x}_i, y)$
7: $\mathbf{w} \leftarrow \mathbf{w} + \gamma(\hat{y}, y_i)[\Phi(\mathbf{x}_i, y_i) - \Phi(\mathbf{x}_i, \hat{y})]$
8: $\mathbf{w}^* += \mathbf{w}$
9: **end for**
10: **end for**
11: **return** $\mathbf{w}^* / = (N \times I)$

Figure 4: Cost-sensitive structured perceptron (see Section 3 for weight functions γ).

two cost-sensitive systems proposed only differ in how we formulate $\gamma(\cdot, \cdot)$. In one model, the loss is weighted by the inter-annotator $F1$ of the gold tag in question. This boils down to

$$\gamma(y_j, y_i) = F1_{y_i}(A_1(X), A_2(X))$$

where X is the small sample of held-out data used to estimate inter-annotator agreement. Note that in this formulation, the predicted label is not taken into consideration.

The second model is slightly more expressive and takes *both* the gold and predicted tags into account. It basically weights the loss by how likely the gold and predicted tag are to be mistaken for each other, i.e., (the inverse of) their confusion probability:

$$\gamma(y_j, y_i)) = 1 - P(\{A_1(X), A_2(X)\} = \{y_j, y_i\})$$

In both loss functions, a lower gamma value means that the tags are more likely to be confused by a pair of annotators. In this case, the update is smaller. In contrast, the learner incurs greater loss when easy tags are confused.

It is straight-forward to extend these cost-sensitive loss functions to the structured perceptron (Collins, 2002). In Figure 4, we provide the pseudocode for the cost-sensitive structured online learning algorithm. We refer to the cost-sensitive structured learners as $F1$- and *CM-weighted* below.

5 Experiments

In our main experiments, we use structured perceptron (Collins, 2002) with random corruptions

using a drop-out rate of 0.1 for regularization, following Søgaard (2013a). We use the LXMLS toolkit implementation[1] with default parameters. We present learning curves across iterations, and only set parameters using held-out data for our downstream experiments.[2]

5.1 Results

Our results are presented in Figure 5. The top left graph plots accuracy on the training data per iteration. We see that CM-weighting does not hurt training data accuracy. The reason may be that the cost-sensitive learner does not try (as hard) to optimize performance on inconsistent annotations. The next two plots (upper mid and upper right) show accuracy over epochs on in-sample evaluation data, i.e., GIMPEL-DEV and RITTER-TEST. Again, the CM-weighted learner performs better than our baseline model, while the $F1$-weighted learner performs much worse.

The interesting results are the evaluations on out-of-sample evaluation data sets (FOSTER and IN-HOUSE) - lower part of Figure 5. Here, both our learners are competitive, but overall it is clear that the CM-weighted learner performs best. It consistently improves over the baseline and $F1$-weighting. The former is much more expressive as it takes confusion probabilities into account and does not only update based on gold-label uncertainty, as is the case with the $F1$-weighted learner.

5.2 Robustness across regularizers

Discriminative learning typically benefits from regularization to prevent overfitting. The simplest is the averaged perceptron, but various other methods have been suggested in the literature.

We use structured perceptron with drop-out, but results are relatively robust across other regularization methods. Drop-out works by randomly dropping a fraction of the active features in each iteration, thus preventing overfitting. Table 1 shows the results for using different regularizers, in particular, Zipfian corruptions (Søgaard, 2013b) and averaging. While there are minor differences across data sets and regularizers, we observe that the corresponding cell using the loss function suggested in this paper (CM) always performs better than the baseline method.

[1] https://github.com/gracaninja/lxmls-toolkit/

[2] In this case, we use FOSTER-DEV as our development data to avoid in-sample bias.

6 Downstream evaluation

We have seen that our POS tagging model improves over the baseline model on three out-of-sample test sets. The question remains whether training a POS tagger that takes inter-annotator agreement scores into consideration is also effective on downstream tasks. Therefore, we evaluate our best model, the CM-weighted learner, in two downstream tasks: shallow parsing—also known as chunking—and named entity recognition (NER).

For the downstream evaluation, we used the baseline and CM models trained over 13 epochs, as they performed best on FOSTER-DEV (cf. Figure 5). Thus, parameters were optimized only on POS tagging data, not on the downstream evaluation tasks. We use a publicly available implementation of conditional random fields (Lafferty et al., 2001)[3] for the chunking and NER experiments, and provide the POS tags from our CM learner as features.

6.1 Chunking

The set of features for chunking include information from tokens and POS tags, following Sha and Pereira (2003).

We train the chunker on Twitter data (Ritter et al., 2011), more specifically, the 70/30 train/test split provided by Derczynski et al. (2013) for POS tagging, as the original authors performed cross validation. We train on the 70% Twitter data (11k tokens) and evaluate on the remaining 30%, as well as on the test data from Foster et al. (2011). The FOSTER data was originally annotated for POS and constituency tree information. We converted it to chunks using publicly available conversion software.[4] Part-of-speech tags are the ones assigned by our cost-sensitive (CM) POS model trained on Twitter data, the concatenation of Gimpel and 70% Ritter training data. We did not include the CoNLL 2000 training data (newswire text), since adding it did not substantially improve chunking performance on tweets, as also shown in (Ritter et al., 2011).

The results for chunking are given in Table 2. They show that using the POS tagging model (CM) trained to be more sensitive to inter-annotator agreement improves performance over

[3] http://crfpp.googlecode.com

[4] http://ilk.uvt.nl/team/sabine/homepage/software.html

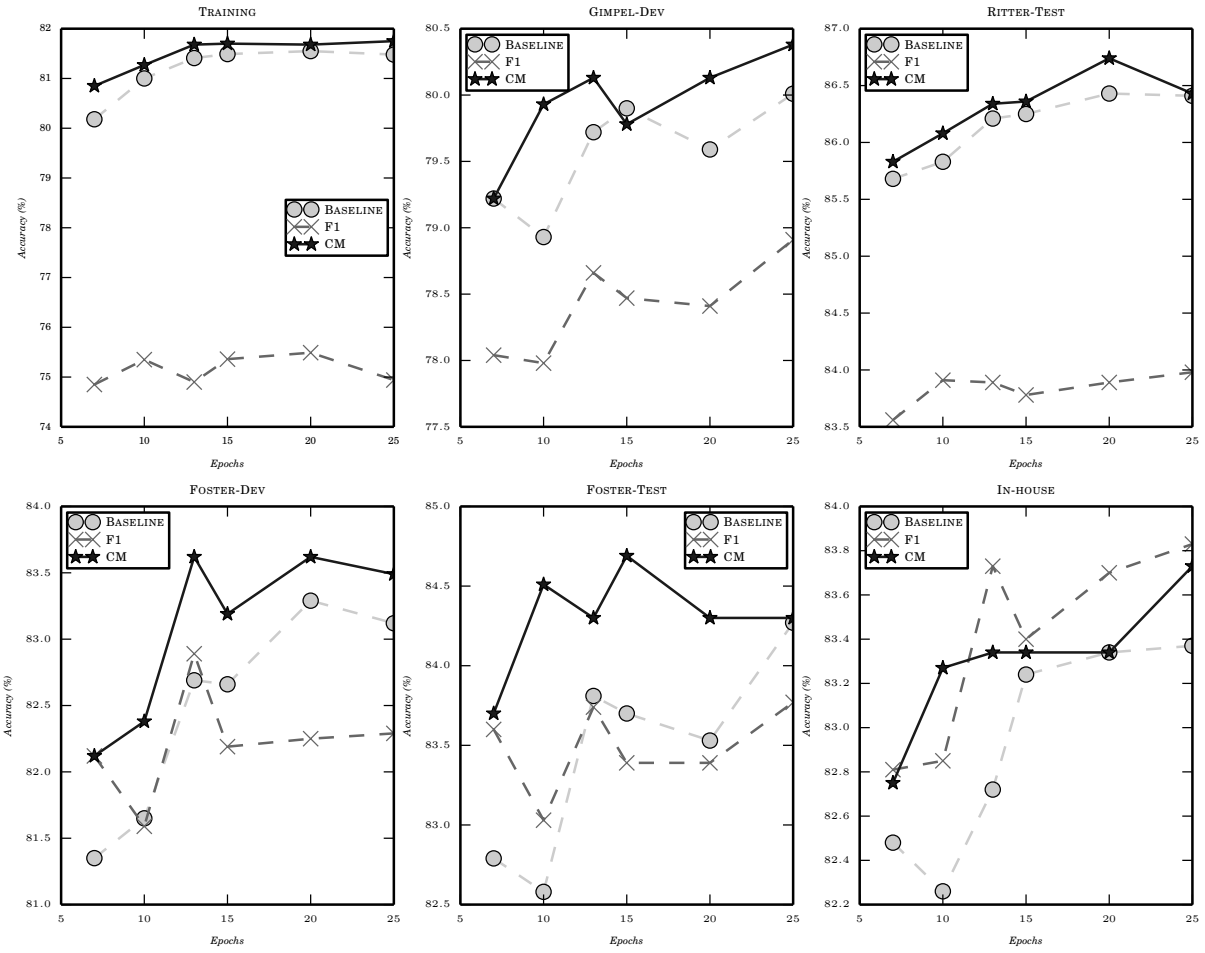

Figure 5: POS accuracy for the three models: baseline, confusion matrix loss (CM) and $F1$-weighted ($F1$) loss for increased number of training epochs. Top row: in-sample accuracy on training (left) and in-sample evaluation datasets (center, right). Bottom row: out-of-sample accuracy on various data sets. CM is robust on both in-sample and out-of-sample data.

	Ritter-Test			
F1:	All	NP	VP	PP
BL	76.20	78.61	74.25	86.79
CM	76.42	79.07	74.98	86.19
	Foster-Test			
F1:	All	NP	VP	PP
BL	68.49	70.73	60.56	86.50
CM	68.97	71.25	61.97	87.24

Table 2: Downstream results on chunking. Overall F1 score (All) as well as F1 for NP, VP and PP.

the baseline (BL) for the downstream task of chunking. Overall chunking F1 score improves. More importantly, we report on individual scores for NP, VP and PP chunks, where we see consistent improvements for NPs and VPs (since both nouns and verbs have high inter-annotator agreement), while results on PP are mixed. This is to be expected, since PP phrases involve adpositionals (ADP) that are often confused with particles (PRT), cf. Figure 3. Our tagger has been trained to deliberately abstract away from such uncertain cases. The results show that taking uncertainty in POS annotations into consideration during training has a positive effect in downstream results. It is thus better if we do not try to urge our models to make a firm decision on phenomena that neither

	BASELINE			CM		
Regularizer	FOSTER-DEV	FOSTER-TEST	IN-HOUSE	FOSTER-DEV	FOSTER-TEST	IN-HOUSE
Averaging	0.827	0.837	0.830	0.831	0.844	0.833
Drop-out	0.827	0.838	0.827	0.836	0.843	0.833
Zipfian	0.821	0.835	0.833	0.825	0.838	0.836

Table 1: Results across regularizers (after 13 epochs).

linguistic theories nor annotators do agree upon.

6.2 NER

In the previous section, we saw positive effects of cost-sensitive POS tagging for chunking, and here we evaluate it on another downstream task, NER.

For the named entity recognition setup, we use commonly used features, in particular features for word tokens, orthographic features like the presence of hyphens, digits, single quotes, upper/lowercase, 3 character prefix and suffix information. Moreover, we add Brown word cluster features that use 2,4,6,8,..,16 bitstring prefixes estimated from a large Twitter corpus (Owoputi et al., 2013).[5]

For NER, we do not have access to carefully annotated Twitter data for training, but rely on the crowdsourced annotations described in Finin et al. (2010). We use the concatenation of the CoNLL 2003 training split of annotated data from the Reuters corpus and the Finin data for training, as in this case training on the union resulted in a model that is substantially better than training on any of the individual data sets. For evaluation, we have three Twitter data set. We use the recently published data set from the MSM 2013 challenge (29k tokens)[6], the data set of Ritter et al. (2011) used also by Fromheide et al. (2014) (46k tokens), as well as an in-house annotated data set (20k tokens) (Fromheide et al., 2014).

F1:	RITTER	MSM	IN-HOUSE
BL	78.20	82.25	82.58
CM	78.30	82.00	82.77

Table 3: Downstream results for named entity recognition (F1 scores).

Table 3 shows the result of using our POS models in downstream NER evaluation. Here we observe mixed results. The cost-sensitive model is able to improve performance on two out of the three test sets, while being slightly below baseline performance on the MSM challenge data. Note that in contrast to chunking, POS tags are just one of the many features used for NER (albeit an important one), which might be part of the reason why the picture looks slightly different from what we observed above on chunking.

7 Related work

Cost-sensitive learning takes costs, such as misclassification cost, into consideration. That is, each instance that is not classified correctly during the learning process may contribute differently to the overall error. Geibel and Wysotzki (2003) introduce instance-dependent cost values for the perceptron algorithm and apply it to a set of binary classification problems. We focus here on structured problems and propose cost-sensitive learning for POS tagging using the structured perceptron algorithm. In a similar spirit, Higashiyama et al. (2013) applied cost-sensitive learning to the structured perceptron for an entity recognition task in the medical domain. They consider the distance between the predicted and true label sequence smoothed by a parameter that they estimate on a development set. This means that the entire sequence is scored at once, while we update on a per-label basis.

The work most related to ours is the recent study of Song et al. (2012). They suggest that some errors made by a POS tagger are more serious than others, especially for downstream tasks. They devise a hierarchy of POS tags for the Penn treebank tag set (e.g. the class NOUN contains NN, NNS, NNP, NNPS and CD) and use that in an SVM learner. They modify the Hinge loss that can take on three values: $0, \sigma, 1$. If an error occurred and the predicted tag is in the same class as the gold tag, a loss σ occurred, otherwise it counts as full cost. In contrast to our approach, they let the learner focus on the more difficult cases by occurring a bigger loss when the predicted POS tag

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

[6] http://oak.dcs.shef.ac.uk/msm2013/ie_challenge/

is in a different category. Their approach is thus suitable for a fine-grained tagging scheme and requires tuning of the cost parameter σ. We tackle the problem from a different angle by letting the learner abstract away from difficult, inconsistent cases as estimated from inter-annotator scores.

Our approach is also related to the literature on regularization, since our cost-sensitive loss functions are aimed at preventing over-fitting to low-confidence annotations. Søgaard (2013b; 2013a) presented two theories of linguistic variation and perceptron learning algorithms that regularize models to minimize loss under expected variation. Our work is related, but models variations in annotation rather than variations in input.

There is a large literature related to the issue of learning from annotator bias. Reidsma and op den Akker (2008) show that differences between annotators are not random slips of attention but rather different biases annotators might have, i.e. different mental conceptions. They show that a classifier trained on data from one annotator performed much better on in-sample (same annotator) data than on data of any other annotator. They propose two ways to address this problem: i) to identify subsets of the data that show higher inter-annotator agreement and use only that for training (e.g. for speaker address identification they restrict the data to instances where at least one person is in the focus of attention); ii) if available, to train separate models on data annotated by different annotators and combine them through voting. The latter comes at the cost of recall, because they deliberately chose the classifier to abstain in non-consensus cases.

In a similar vein, Klebanov and Beigman (2009) divide the instance space into easy and hard cases, i.e. easy cases are reliably annotated, whereas items that are hard show confusion and disagreement. Hard cases are assumed to be annotated by individual annotator's coin-flips, and thus cannot be assumed to be uniformly distributed (Klebanov and Beigman, 2009). They show that learning with annotator noise can have deteriorating effect at test time, and thus propose to remove hard cases, both at test time (Klebanov and Beigman, 2009) and training time (Beigman and Klebanov, 2009).

In general, it is important to analyze the data and check for label biases, as a machine learner is greatly affected by annotator noise that is not random but systematic (Reidsma and Carletta, 2008). However, rather than training on subsets of data or training separate models – which all implicitly assume that there is a large amount of training data available – we propose to integrate inter-annotator biases directly into the loss function.

Regarding measurements for agreements, several scores have been suggested in the literature. Apart from the simple agreement measure, which records how often annotators choose the same value for an item, there are several statistics that qualify this measure by adjusting for other factors, such as Cohen's κ (Cohen and others, 1960), the G-index score (Holley and Guilford, 1964), or Krippendorff's α (Krippendorf, 2004). However, most of these scores are sensitive to the label distribution, missing values, and other circumstances. The measure used in this paper is less affected by these factors, but manages to give us a good understanding of the agreement.

8 Conclusion

In NLP, we use a variety of measures to assess and control annotator disagreement to produce homogenous final annotations. This masks the fact that some annotations are more reliable than others, and which is thus not reflected in learned predictors. We incorporate the annotator uncertainty on certain labels by measuring annotator agreement and use it in the modified loss function of a structured perceptron. We show that this approach works well independent of regularization, both on in-sample and out-of-sample data. Moreover, when evaluating the models trained with our loss function on downstream tasks, we observe improvements on two different tasks. Our results suggest that we need to pay more attention to annotator confidence when training predictors.

Acknowledgements

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

References

Eyal Beigman and Beata Klebanov. 2009. Learning with annotation noise. In *ACL*.

Jacob Cohen et al. 1960. A coefficient of agreement

for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *EMNLP*.

Dipanjan Das and Slav Petrov. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *ACL*.

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

Jakob Elming, Anders Johannsen, Sigrid Klerke, Emanuele Lapponi, Hector Martinez, and Anders Søgaard. 2013. Down-stream effects of tree-to-dependency conversions. 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*.

Hege Fromheide, Dirk Hovy, and Anders Søgaard. 2014. Crowdsourcing and annotating NER for Twitter #drift. In *Proceedings of LREC 2014*.

Peter Geibel and Fritz Wysotzki. 2003. Perceptron based learning with example dependent and noisy costs. In *ICML*.

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*.

Shohei Higashiyama, Kazuhiro Seki, and Kuniaki Uehara. 2013. Clinical entity recognition using cost-sensitive structured perceptron for NTCIR-10 MedNLP. In *NTCIR*.

Jasper Wilson Holley and Joy Paul Guilford. 1964. A Note on the G-Index of Agreement. *Educational and Psychological Measurement*, 24(4):749.

Dirk Hovy, Barbara Plank, and Anders Søgaard. 2014. When POS datasets don't add up: Combatting sample bias. In *Proceedings of LREC 2014*.

Richard Johansson. 2013. Training parsers on incompatible treebanks. In *NAACL*.

Beata Klebanov and Eyal Beigman. 2009. From annotator agreement to noise models. *Computational Linguistics*, 35(4):495–503.

Klaus Krippendorf, 2004. *Content Analysis: An Introduction to Its Methodology, second edition*, chapter 11. Sage, Thousand Oaks, CA.

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

Christopher D Manning. 2011. Part-of-speech tagging from 97% to 100%: is it time for some linguistics? In *Computational Linguistics and Intelligent Text Processing*, pages 171–189. Springer.

Mitchell Marcus, Mary Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal dependency annotation for multilingual parsing. In *ACL*.

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*.

Dennis Reidsma and Jean Carletta. 2008. Reliability measurement without limits. *Computational Linguistics*, 34(3):319–326.

Dennis Reidsma and Rieks op den Akker. 2008. Exploiting 'subjective' annotations. In *Workshop on Human Judgements in Computational Linguistics, COLING*.

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

Roy Schwartz, Omri Abend, Roi Reichart, and Ari Rappoport. 2011. Neutralizing linguistically problematic annotations in unsupervised dependency parsing evaluation. In *ACL*.

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

Anders Søgaard. 2013a. Part-of-speech tagging with antagonistic adversaries. In *ACL*.

Anders Søgaard. 2013b. Zipfian corruptions for robust pos tagging. In *NAACL*.

Hyun-Je Song, Jeong-Woo Son, Tae-Gil Noh, Seong-Bae Park, and Sang-Jo Lee. 2012. A cost sensitive part-of-speech tagging: differentiating serious errors from minor errors. In *ACL*.

Reut Tsarfaty, Joakim Nivre, and Evelina Andersson. 2012. Cross-framework evaluation for statistical parsing. In *EACL*.

Daniel Zeman and Philip Resnik. 2008. Cross-language parser adaptation between related languages. In *IJCNLP*.

Daniel Zeman. 2010. Hard problems of tagset conversion. In *Proceedings of the Second International Conference on Global Interoperability for Language Resources*.