Learning to parse with IAA-weighted loss

Héctor Martínez Alonso[†] **Barbara Plank**[†] **Arne Skjærholt**[‡] **Anders Søgaard**[†]
[†]Njalsgade 140, Copenhagen (Denmark), University of Copenhagen
[‡]Gaustadalléen 23B, Oslo (Norway), University of Oslo
alonso@hum.ku.dk,bplank@cst.dk,arnskj@ifi.uio.no,soegaard@hum.ku.dk

Abstract

Natural language processing (NLP) annotation projects employ guidelines to maximize inter-annotator agreement (IAA), and models are estimated assuming that there is one single ground truth. However, not all disagreement is noise, and in fact some of it may contain valuable linguistic information. We integrate such information in the training of a cost-sensitive dependency parser. We introduce five different factorizations of IAA and the corresponding loss functions, and evaluate these across six different languages. We obtain robust improvements across the board using a factorization that considers dependency labels and directionality. The best method-dataset combination reaches an average overall error reduction of 6.4% in labeled attachment score.

1 Introduction

Typically, NLP annotation projects employ guidelines to maximize inter-annotator agreement. Possible inconsistencies are resolved by adjudication, and models are induced assuming there is one single ground truth. However, there exist linguistically hard cases where there is no clear answer (Zeman, 2010; Manning, 2011), and incorporating such disagreements into the training of a model has proven helpful for POS tagging (Plank et al., 2014a; Plank et al., 2014b).

Inter-annotator agreement (IAA) is straightforward to calculate for POS, but not for dependency trees. There is no well-established standard for computing agreement on trees (Skjærholt, 2014). For a dependency tree, annotators can disagree in attachment, labeling, or both. We implement different strategies, i.e., *factorizations* (§2), to capture disagreement on specific syntactic phenomena.

Our hypothesis is that a dependency parser can be informed of disagreements to regularize over annotators' biases. Testing our hypothesis requires the availability of doubly-annotated data, and involves two steps: i) how to *factorize* attachment or labeling disagreements; and ii) how to *inform* the parser of them during learning (§3).

2 Factorizations

Assume a sample of sentences annotated by annotators A_1 and A_2. With such a sample we can estimate probabilities of the two annotators' disagreeing on the annotation of a word or span, relative to some dependency tree factorization. We factorize disagreement on dependency tree annotations relative to four properties of the annotated dependency edges: the POS of the dependent, the POS of the head, the label of the edge and the direction (left or right) of the head with regards to the dependent. This section describes the different factorizations.

We present five factorizations, depicted in Figure 1. With artificial root notes, all words in a dependency tree have one incoming edge. This means that in our sample, any word w_i has two $\langle headId, label\rangle$ annotations, i.e., $\langle h_1, l_1\rangle$ and $\langle h_2, l_2\rangle$ given by A_1 and A_2, respectively, with POS$(\cdot)$ being a function from word indices to POS. The five factorizations are as follows:

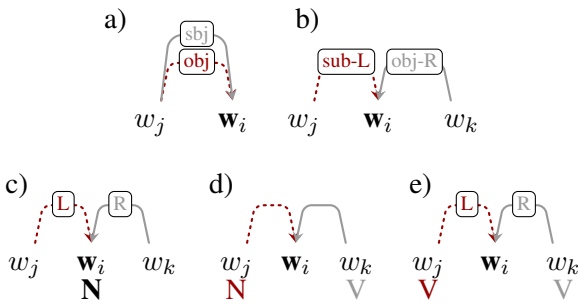

Figure 1: Factorizations: a) LABEL, b) LABELD; c) CHILDPOSD, d) HEADPOS and e) HEADPOSD. Red and green depict different choices by annotators A_1 and A_2.

a) LABEL: disagreement over label pairs, regardless of attachment (h_1,h_2). That is, $\langle h_1, l_1\rangle$ and $\langle h_2, l_2\rangle$ count as disagreement, iff $l_1 \neq l_2$.

b) LABELD, same as LABEL, but incorporating edge direction. That is, $\langle h_1, l_1\rangle$ and $\langle h_2, l_2\rangle$ count as disagreement, for any $j, k \in h_1, h_2$, iff $h_j < i < h_k$ *or* $l_1 \neq l_2$.

c) CHILDPOSD, i.e., disagreement on attachment direction given POS(i). That is, for POS(i), $\langle h_1, l_1\rangle$ and $\langle h_2, l_2\rangle$ count as disagreement, iff $h_j < i < h_k$.

d) HEADPOS: disagreement on head POS. That is, $\langle h_1, l_1\rangle$ and $\langle h_2, l_2\rangle$ count as disagreement, iff POS(h_1)≠POS(h_2).

e) HEADPOSD, i.e., HEADPOS, plus direction. That is, $\langle h_1, l_1\rangle$ and $\langle h_2, l_2\rangle$ count as disagreement, iff POS(h_1)≠POS(h_2) *or* $h_j < i < h_k$.

Each factorization yields a symmetric confusion matrix. In our Norwegian data (§4), for instance, for LABEL there are 834 words that have been labeled as ATR (attribute) by both annotators, while there are 44 cases where one annotator has given the ATR label and the other has given the ADV (adverbial) label. For LABELD, there are 968 words that have been labeled as ADV where both annotators agree on the head being on the left side of the word, whereas there are 9 cases where the annotators agree on ADV label but not on the direction of the head. These 9 cases count as disagreements for LABELD but not for LABEL.

lang	train	test	*l*	*p*
NO	13.7k/209k	5.8k/96.7k	29	19
EN	3.6k/70k	†1.0k/20.3k	30	44
DA	4.2k/74k	†1.2k/23.4k	31	25
CA	3.9k/73k	1.7k/34.4k	27	11
HR	3.1k/79k	1.3k/35.5k	26	27
FI	9.1k/123k	3.9k/54.4k	45	12

Table 1: Data statistics: number of sentences/tokens, dependency labels *l*, POS tags *p* for NO (Norwegian), EN (English), DA (Danish), CA (Catalan), Croatian (HR) and Finnish (FI); †=canonical test split available.

3 Cost-sensitive updates

We use the cost-sensitive perceptron classifier, following Plank et al. (2014a), but extend it to transition-based dependency parsing, where the predicted values are transitions (Goldberg and Nivre, 2012). Given a gold y_i and predicted label $\hat{y}_i$ (POS tags or transitions), the loss is weighted by $\gamma(\hat{y}_i, y_i)$:

$$L_{\mathbf{w}}(\hat{y}_i, y_i) = \gamma(\hat{y}_i, y_i) \max(0, -y_i \mathbf{w} \cdot \mathbf{x}_i)$$

Whenever a transition has been wrongly predicted, we retrieve the predicted edge and compare it to the gold dependency to calculate γ. $\gamma(y_i, y_j)$ is then the inverse of the confusion probability estimated from our sample of doubly-annotated data. For example, using the factorization LABEL, if the parser predicts w_i to be SUBJECT and the gold annotation is OBJECT, the confusion probability is the number of times one annotator said SUBJECT while the other said OBJECT out of the times one annotator said one of them. In LABELD, A_1 and A_2 can disagree even if both say the grammatical function of some word w_i is SUBJECT, namely if one says the subject is left of w_i, and the other says it is right of w_i. The confusion probability is then the count of disagreements over the total number of cases where both annotators said a word was SUBJECT.

In our baseline model, $\gamma(\hat{y}_i, y_i) = 1$. The values for our cost-sensitive systems (LABEL, LABELD, CHILDPOSD, HEADPOS, HEADPOSD) are never above 1, which means that we are selectively underfitting the parser for specific syntactic phenomena. In other words, we use the doubly-annotated data to regularize our model, hopefully preventing overfitting to annotators' biases.

4 Data

We use six treebanks (Buch-Kromann et al., 2003; Buch-Kromann et al., 2007; Arias et al., 2014; Solberg et al., 2014; Agić and Merkler, 2013; Haverinen et al., 2010) for which we could get a sample of doubly-annotated data. All these treebanks are directly developed as dependency treebanks, instead of being converted from constituent treebanks. Table 1 gives overview statistics of the treebanks, Table 2 lists the sizes of the doubly-annotated samples, as well as F1 scores between annotators and α values (Skjærholt, 2014). The doubly-annotated samples are solely used to estimate confusion probabilities, and not for training or testing. When a treebank had no canonical train/test split, we took the final 30% for testing.

			between annotator:			
lang	**sents**	**tokens**	**LAS**	**UAS**	**LA**	α plain
NO	400	5.3k	94.70	96.47	96.62	0.984
EN	264	5.5k	88.44	93.83	91.95	0.925
DA	162	2.4k	90.43	96.12	92.40	0.957
CA	63	1.3k	94.48	98.26	95.64	0.978
HR	100	2.4k	78.89	89.16	84.07	0.939
FI	400	5.1k	83.45	88.77	89.83	0.950

Table 2: Statistics of the doubly-annotated data.

5 Experiments

In our experiments, we use `redshift`,[1] a transition-based arc-eager dependency parser that implements the dynamic oracle (Goldberg and Nivre, 2012) with averaged perceptron training. We modified the parser[2] to read confusion matrices and weigh the updates with the respective γ. We compare the five (§2) factorized systems to a baseline system that does not take confusion probabilities into account, i.e., standard `redshift`. Throughout the experiments, we fix the number of iterations to 5, and we use pseudo-projectivization (Nivre and Nilsson, 2005).[3] The parser does not include morphological features, which lowers performance for morphological rich languages like FI. We report labeled attachment scores (LAS) incl. punctuation.

[1] `https://github.com/syllog1sm/redshift`

[2] The modified code, as well as the confusion matrices for all factorizations, is available at `https://bitbucket.org/lowlands/iaa-parsing`

[3] 15–33% of the sentences contain non-projectivities.

We use bootstrap sampling in all our experiments in order to get more reliable results. This method allows abstracting away from biases—in sampling and annotation—of training and test splits. We use two complementary evaluation methods: cross-validation within the training data, and learning curves against the test set. We calculate significance using the approximate randomization test (Noreen, 1989) with 10k iterations.

Cross-validation In this setup, we perform 50 runs of 5-fold cross validation on bootstrap-based samples of the training data. This allows us to gauge the effect of our factorization without committing to a certain test set. We report on the average of the total of 250 runs.

Learning curve To calculate the learning curves, we train the parser on increasing amounts of training data, bootstrap-sampled in steps of 10%, and evaluate against the test set. Each 10% increment is repeated $k = 50$ times. We finally report average overall error reduction over the baseline.

6 Results

Cross-validation The results for cross-validation are shown in Table 3. For 5 out of the 6 languages we get significant improvements over the baseline with some factorization. We obtain improvements on all treebanks using LABELD, and on five out of six using CHILDPOSD. For CA, with the smallest doubly-annotated sample, results are not as consistent across the two evaluation methods.

Learning curve Table 4 summarizes the overall average error reduction over the 10-step bootstrap-based learning curve (with 50 runs at each step). We get consistent improvements for languages for which we have a sample of 100+ sentences (Table 2). Again, the most robust factorization is LABELD. Figure 2 shows the learning curves for the system with the highest error reduction (NO with CHILDPOSD).

Additional studies In order to evaluate whether our results are meaningful and not just artifacts of random regularization, we performed a sanity check for the best performing system and factorization (i.e., NO with CHILDPOSD factorization). We

	BASELINE	CHILDPOSD	LABEL	LABELD	HEADPOS	HEADPOSD
NO	90.98	**92.67**∗	91.16	91.34	92.08∗	90.48
EN	81.72	83.48∗	80.35	83.05∗	85.89∗	**85.91**∗
DA	80.56	83.67∗	82.90∗	82.47∗	83.23∗	**84.11**∗
CA	83.78	83.26	**84.21**∗	83.79	82.84	82.61
HR	76.94	78.07	78.22	77.52	**79.49**∗	78.71∗
FI	66.19	66.74	64.88	**67.18**	65.63	65.27

Table 3: Crossvalidation results (in LAS incl. punctuation). Gray: below baseline. Best factorization per language in boldface. Significance at $p < 0.01$ (computed over runs and wrt baseline) is indicated by $*$.

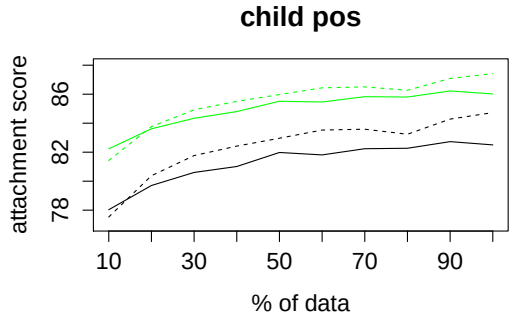

Figure 2: Bootstrap learning curve (k=50) for NO with CHILDPOSD. Black: LAS, green: UAS; solid line: baseline; dashed line: IAA-weighted model.

	CHILDPOSD	LABEL	LABELD	HEADPOS	HEADPOSD
NO	**6.4%**	0.6%	0.7%	3.3%	1.2%
EN	2.0%	2.6%	2.9%	**5.3%**	3.8%
DA	0.7%	1.6%	1.0%	**2.0%**	1.0%
CA	-2.0%	-0.1%	-0.1%	-2.9%	-2.8%
HR	-0.2%	0.3%	**0.7%**	0.1%	0.1%
FI	**0.4%**	-0.4%	0.1%	-0.1%	-0.70%

Table 4: Overall avg. error red. across learning curves.

shuffled the confusion matrix and ran the bootstrap learning curve with $k = 50$ repetitions, for five different shufflings. The mean over the five runs for the overall average error reductions is negative (-0.38%, compared to the 2.4% mean for the original, non-shuffled version). We thus conclude that our factorizations capture linguistically plausible information rather than random noise.

7 Related Work

Plank et al. (2014a) propose IAA-weighted cost-sensitive learning for POS tagging. We extend their line of work to dependency parsing.

A single sentence can have more than one plausible dependency annotation. Some researchers have proposed evaluation metrics that do not penalize disagreements (Schwartz et al., 2011; Tsarfaty et al., 2011), while others have argued that we should instead ensure the consistency of treebanks (Dickinson, 2010; Manning, 2011; McDonald et al., 2013). Others have claimed that because of these ambiguities, only downstream evaluations are meaningful (Elming et al., 2013).

Syntactic annotation disagreement has typically been studied in the context of treebank development. Haverinen et al. (2012), for example, analyze annotator disagreement for Finnish dependency syntax, and compare it against parser performance. Skjærholt (2014) use doubly-annotated data to evaluate various agreement metrics. Our paper differs from both lines of research in that we leverage disagreements from doubly-annotated data to obtain more robust models. While we agree that evaluation metrics should probably reflect disagreements, we show that our learning algorithms can indeed benefit from information about disagreement, also using standard performance metrics.

8 Conclusions

We have evaluated five different factorizations on six treebanks to evaluate the impact of IAA-weighted learning for dependency parsing, obtaining promising results. The findings support our hypothesis that annotator disagreement is informative for parsing. The LABELD factorization—which takes both labeling and word order into account—is the overall most robust factorization across all languages. However, the best factorization for each language varies. This variation can be a result of the morphosyntax of the language, but also of the dependency annotation formalisms, annotation method, training corpus and size of the doubly-annotated sample.

Acknowledgements

We thank Jorge Vivaldi, Filip Ginter and Željko Agić for providing doubly-annotated data. This research is partially funded by the ERC Starting Grant LOWLANDS No. 313695.

References

Željko Agić and Danijela Merkler. 2013. Three syntactic formalisms for data-driven dependency parsing of croatian. In *Text, Speech, and Dialogue*. Springer.

Blanca Arias, Nuria Bel, Mercè Lorente, Montserrat Marimón, Alba Milà, Jorge Vivaldi, Muntsa Padró, Marina Fomicheva, and Imanol Larrea. 2014. Boosting the creation of a treebank. In *LREC*.

Matthias Buch-Kromann, Line Mikkelsen, and Stine Kern Lynge. 2003. Danish dependency treebank. In *TLT*.

Matthias Buch-Kromann, Jürgen Wedekind, and Jakob Elming. 2007. The Copenhagen Danish-English Dependency Treebank v.2.0. *http://buch-kromann.dk/matthias/cdt2.0*.

Markus Dickinson. 2010. Detecting errors in automatically-parsed dependency relations. In *ACL*.

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

Yoav Goldberg and Joakim Nivre. 2012. A dynamic oracle for arc-eager dependency parsing. In *COLING*.

Katri Haverinen, Timo Viljanen, Veronika Laippala, Samuel Kohonen, Filip Ginter, and Tapio Salakoski. 2010. Treebanking finnish. In *TLT*.

Katri Haverinen, Filip Ginter, Samuel Kohonen, Timo Viljanen, Jenna Nyblom, and Tapio Salakoski. 2012. A dependency-based analysis of treebank annotation errors. In *Computational Dependency Theory*. IOS Press.

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*. Springer.

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*.

Joakim Nivre and Jens Nilsson. 2005. Pseudo-projective dependency parsing. In *ACL*.

Eriw W. Noreen. 1989. *Computer-intensive methods for testing hypotheses: an introduction*. Wiley.

Barbara Plank, Dirk Hovy, and Anders Søgaard. 2014a. Learning part-of-speech taggers with inter-annotator agreement loss. In *EACL*.

Barbara Plank, Dirk Hovy, and Anders Søgaard. 2014b. Linguistically debatable or just plain wrong? In *ACL*.

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

Arne Skjærholt. 2014. A chance-corrected measure of inter-annotator agreement for syntax. In *ACL*.

Per Erik Solberg, Arne Skjærholt, Lilja Øvrelid, Kristin Hagen, and Janne Bondi Johannessen. 2014. The Norwegian Dependency Treebank. In *LREC*.

Reut Tsarfaty, Joakim Nivre, and Evelina Ndersson. 2011. Evaluating dependency parsing: robust and heuristics-free cross-nnotation evaluation. In *EMNLP*.

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