Max-Margin Incremental CCG Parsing

Miloš Stanojević
School of Informatics
University of Edinburgh
m.stanojevic@ed.ac.uk

Mark Steedman
School of Informatics
University of Edinburgh
steedman@inf.ed.ac.uk

Abstract

Incremental syntactic parsing has been an active research area both for cognitive scientists trying to model human sentence processing and for NLP researchers attempting to combine incremental parsing with language modelling for ASR and MT. Most effort has been directed at designing the right transition mechanism, but less has been done to answer the question of what a probabilistic model for those transition parsers should look like.

A very incremental transition mechanism of a recently proposed CCG parser when trained in straightforward locally normalised discriminative fashion produces very bad results on English CCGbank. We identify three biases as the causes of this problem: label bias, exposure bias and imbalanced probabilities bias.

While known techniques for tackling these biases improve results, they still do not make the parser state of the art. Instead, we tackle all of these three biases at the same time using an improved version of beam search optimisation that minimises all beam search violations instead of minimising only the biggest violation. The new incremental parser gives better results than all previously published incremental CCG parsers, and outperforms even some widely used non-incremental CCG parsers.

1 Introduction

It has been known for a long time that human sentence processing is highly incremental (Marslen-Wilson, 1973), with early formation of semantic representations. A parser that is able to form representation early must have some notion of *partial structure* such as "S missing an object NP". Also, such parser needs to be able to combine partial structures into bigger partial structures. These two properties are at the core of Combinatory Categorial Grammar (CCG) (Ades and Steedman, 1982; Steedman, 2000). CCG represents partial constituents using complex categories. For example *S/NP* is the category of a transitive sentential prefix such as *I like* or *I think I like* requiring an object *NP* on its right. Such prefix categories are constructed using combinatory rules such as function composition. In this way we can form (mostly) left-branching derivation trees that can be parsed incrementally even with simple transition mechanisms such as shift-reduce parsers.

Still, left branching structures are not sufficient to solve all the problems of incremental sentence processing. Right adjuncts are particularly problematic. They appear on the right of the head that they modify which means that they need to be predicted, but at the same time they are optional which makes it impossible to predict them with confidence. Stanojević and Steedman (2019) tackle this issue by using *incremental tree-rotation* and *revealing* operations that allow adjuncts not to be predicted, but still be easy to attach to the head in case they appear. They show great improvement in the incrementality of this approach as measured by connectedness (the average stack size).

However, Stanojević and Steedman (2019) parser is not fully incremental because its *oracle* (the function that decides which transition to take in case of non-determinism)[1] is a probabilistic model that looks at the whole sentence. It does so using bi-directional ELMo embeddings with the addition of bi-directional LSTMs. The present paper describes a *fully incremental* version of Stanojević and Steedman (2019) parser using an incremental oracle that does not look at the words that are not yet processed.

We should note that by a *fully incremental* parsing model we do not mean a parser that has all the partial trees on the stack fully connected at ev-

[1] Note that this sense of a psycholinguistic term *oracle* is not the same as the one used in dependency parsing literature.

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 4111–4122

ery point in time. This is a property of extremely predictive top-down parsers, while the parser that we use is a CCG bottom-up parser. This choice is intentional—even though there is clear evidence that human sentence processing is highly incremental, we argue below that there is no unequivocal evidence that it is *more* incremental than would be allowed under the *Strict Competence Hypothesis* (SCH) which states that the parser cannot construct any structure that is not licensed by the competence grammar, given CCG's generalized notion of constituency (Steedman, 1989).

Most research in incremental parsing has been directed at finding the right parsing algorithm (Abney and Johnson, 1991; Resnik, 1992; Hale, 2014; Stanojević and Stabler, 2018) or grammar formalism (Steedman, 1989; Stabler, 1991; Sturt and Lombardo, 2005; Demberg et al., 2013; Stanojević et al., 2020), but not much has been done in addressing the issue of finding the right oracle. Early approaches to this problem were late-closure and minimal-attachment heuristics (Frazier, 1979; Pereira, 1985) which do not appear to be language universal (Cuetos and Mitchell, 1988). Altmann and Steedman (1988) have shown that these heuristics are overruled by human parser if the context gives evidence for a particular interpretation, in itself further evidence for processing incrementality at all levels. It seems natural to model the non-deterministic decision by using a probabilistic model which will condition on words and possibly on the context. Oracles of the modern broad coverage incremental parsers are without exception statistical in nature.

The most typical statistical oracle is a locally normalised generative model ether in the form of simple PCFG (Stolcke, 1995; Hale, 2001), feature based (Roark and Johnson, 1999; Roark, 2001) or neural model (Dyer et al., 2016; Hale et al., 2018). RNNG (Dyer et al., 2016) is the main contemporary representative of this approach. RNNG is a top-down parser which in its first version used a non-incremental discriminative locally-normalised model. To make the parser fully incremental Dyer et al. (2016) exchanged the discriminative model for a generative one. This was not enough to get a working single-model incremental parser. Stern et al. (2017) added a couple more modifications to the search, namely word-synchronous beams with a very large number of hypotheses, that gave good results.

Could we just apply these same techniques to the CCG parser of Stanojević and Steedman (2019) and replace non-incremental probabilistic model with an incremental one? The short answer is no. As it will be shown later, a straightforward adaptation of the beam search and switching to a generative model does indeed improve accuracy over the model that does not do that, but not enough to make the incremental parser competitive. We provide an explanation for these results and offer an alternative approach.

We identify the problem for building incremental parsing models in terms of three biases: (1) label-bias, (2) exposure-bias and (3) imbalanced probability search bias. These biases are well known from the machine learning literature in structured prediction, but they do not usually have the extreme effect that is seen in the case of incremental parsing. The techniques used in RNNG address some of these biases individually but none of the techniques addresses all three together. Instead of using a collection of techniques for each bias, we replace them all with a single solution in the form of a global unnormalized model trained with beam-search optimization that minimises all margin violations in the beam simultaneously. This single technique addresses all of the mentioned biases and gives results that outperform all previous incremental parsing models even with a relatively small beam. This is not to say that *all* unwanted biases are removed—for instance, beam search is still a biased search. However, the biases that remain do not have the drastic effect on performance of the three identified above.

2 Baseline model

The parser of Stanojević and Steedman (2019) already offers a fully incremental transition system with a non-incremental probabilistic model that gives state of the art accuracy in recovering predicate-argument dependencies. The parser encodes words using ELMo (Peters et al., 2018) and BiLSTM (Graves et al., 2005), sub-trees with tree encoders and the stack with Stack-LSTM (Dyer et al., 2015). This provides the encoding of the whole configuration together with the buffer, because the buffer is implicitly encoded via ELMo and Bi-LSTM, which look at the whole sentence. Given the hidden vector representation of the configuration, the parser uses a feed-forward network to determine the probability of the next action.

There are three main types of transitions:

- Parsing actions: shift and reduce(X) where X is a unary or binary combinatory rule;
- Supertagging actions: tag(X) where X is one of the lexical supertags from English CCGbank (Hockenmaier and Steedman, 2007);
- Right-adjunction actions: adjoin(X) where X is one of the nodes to which the adjunct can be adjoined.

We refer the reader to (Stanojević and Steedman, 2019) for more detail on the original neural model and transition system, which are not of particular relevance here. What matters is only that (1) the number of tagging actions is much bigger than the number of possible parsing actions and (2) that the buffer is implicitly encoded with ELMo and Bi-LSTM. To make the parsing model fully incremental first we modify ELMo embeddings: instead of using full ELMo embeddings we use only the forward LSTM part of ELMo. This decreases performance by only two points on the dev set F1 score from 89.5 to 87.5. Finally, we replace Bi-LSTM with normal LSTM (Hochreiter and Schmidhuber, 1997). This causes a significant drop in performance to 60.9.

We take the fully incremental model with 60.9 F1 as our baseline, and show how it can be improved, to come as close as possible to the non-incremental version that uses the same embeddings, which has accuracy 87.5 F1, changing only the method of training, keeping the network architecture and embeddings the same.

3 Three sources of bias

3.1 Label bias

Label bias is a frequent bias present in some types of locally normalised models. It was first recognised by Bottou (1991), but became more widely known with the publication of CRFs (Lafferty et al., 2001). Here we give an explanation of label-bias in incremental parsing context. For a more formal treatment see Andor et al. (2016).

In a general non-incremental setting, a discriminative parsing model assigns a probability to the whole transition sequence as $p(\mathbf{y}|\mathbf{x})$ where $\mathbf{y} = [y_0, y_1, \ldots, y_m]$ is sequence of parsing actions and $\mathbf{x} = [x_0, x_1, \ldots, x_n]$ is a sequence of words. Since the model is locally normalised we can express this conditional probability as the product of conditional probabilities of each parsing action: $p(\mathbf{y}|\mathbf{x}) = \prod_i p(y_i|\mathbf{y}_{<i}, \mathbf{x})$. In the non-incremental version of the parser there are no independence assumptions, so every parsing action can condition on the whole sequence of words $\mathbf{x}$. However, in the incremental version we can condition only on the $k(i)$ words that have been observed (have shifted from the buffer to the stack) in first i transitions. This makes the new model of the whole transition sequence be $p(\mathbf{y}|\mathbf{x}) = \prod_i p(y_i|\mathbf{y}_{<i}, \mathbf{x}_{<k(i)})$.

This small change has big consequences on parsing. Imagine the situation in which the incremental parser has processed a prefix $\mathbf{x}_{<k(i)}$. This prefix may be genuinely ambiguous making the parser have two derivations in the beam, one in state *A* and the other in state *B*, both equally good up until that point in the sentence. After processing some more words, the parser might find a word that resolves the ambiguity and provides evidence that the state *A* was correct. A good incremental parser would then give a higher score to all derivations that originated in state *A* and a lower score to derivations that originated in state *B*. However, the locally-normalised model cannot do that. Because the model is locally normalised, the probability of all transitions leaving any state must sum to 1, so even if all transitions are bad (they come from a bad state) they cannot all be penalised.

What this means is that parser cannot recover from garden-paths *even with an unboundedly large beam*.[2] This is a deficiency of the probabilistic model because of the introduced independence assumption that the parsing action depends only on the processed prefix. This makes the model effectively ignore its input in some situations.

When we are parsing with greedy search the label-bias will have no influence, because there will be no two states that compete with each other while having a different history. Label-bias is harmful only in the case of beam search.

3.2 Exposure bias

The usual way of training any sequence prediction model is to train the prediction of the next action based on the gold history in the data. But at test time the model will have a predicted history rather than a gold history. On occasions when that predicted history is wrong, the model may not assign good probabilities to the future actions because it has not been exposed to this erroneous history in

[2] We use the term *garden-path* in a more general sense than in psycholinguistic literature to mean taking any transition path that may end up being wrong.

its training data. This problem is often referred to as exposure bias.

This is again specifically relevant for incremental parsing. Let's say that the parser did enter into a garden-path, and that there are still some words left in the suffix. There will still be many transition sequences that the parser could choose from, before it finishes parsing the whole sentence. Even though they are all bad, because we are in a garden-path, they are not all equally bad. We want the parser to choose the transition sequence that would make the most out of this bad situation.

The exposure-bias, unlike the label-bias, influences greedy search too. In fact, exposure-bias is particularly important for greedy models because they are more likely to fall into a garden-path.

3.3 Imbalanced probability search bias

Incremental parsing models that condition on the whole history cannot carry out exact search, and have to use approximate methods like beam search. Beam search is a biased search because it searches only in the local neighbourhoods of the locally most probable derivations. This locality is proportional to the size of the beam. If the beam were unbounded then search would be exact, but often we use a small beam that is only a small relaxation of greedy search.

The fact that the beam search is biased is well known and often accepted as a necessary evil, but it has been recognised by Stern et al. (2017) that for some parsing models the issue is particularly bad because of *imbalanced probability bias*. In their case, an incremental RNNG model had actions for parsing and actions for generation of words. The number of parsing actions was many orders of magnitude smaller than the number of word generation actions. This made the probability of word generation very small. The expensive action of word generation happens in all derivations an equal number of times but it happens in different time steps. Beam search may accordingly discard a good hypothesis too early because that hypothesis has used expensive actions early.

The imbalanced probability bias implicitly prefers states with low entropy. Bias for the low entropy states is often associated with *label-bias*, however the reasons and situations when this happens are different from *imbalanced probability search bias*. Label-bias is a deficiency of the probabilistic model, while imbalanced probability is a deficiency of the search method. This is visible in the context of search with an unboundedly large beam: the model with label-bias would still prefer states with low entropy while imbalanced probability bias would not be present—search would be exact so it would not matter at which point in time expensive actions were applied.

4 Eliminating biases

Some of these biases are well known in the literature of structure prediction and various proposals have been made for reducing their effect. However, most of these techniques usually address only one of the biases, and the combination of these techniques is not always straightforward.

As mentioned before, *label-bias* is caused by model being (i) discriminative, (ii) locally normalised and (iii) having independence assumptions about future input not influencing current actions. We could remove label-bias by removing one of these properties from the model. Clearly, we cannot remove property (iii) because we want an incremental model. The simplest solution is to change property (i) and make the model generative. The generative model would give us probability $p(\mathbf{x}, \mathbf{y})$ instead of $p(\mathbf{y}|\mathbf{x})$. This is done by having an additional action for generation of a word following a shift action. Here the model cannot ignore the input because it is forced to generate it. It can also recognise garden-paths: if we are in a state that cannot generate the following word that means we are in a garden-path and will punish all transitions from that state. However, this solution introduces imbalanced probability search bias because we will introduce word-generation actions that have much higher entropy.

Lafferty et al. (2001) advocated dropping the property (ii) by making the model globally normalised. This would allow transitions to have local *weights* instead of local probabilities. If all transitions from some state are bad, the model is able to give low weight to all of them because weights do not have to sum to one. Lafferty et al. (2001) advocated using conditional random fields (CRF), which are globally normalised probabilistic models, but margin-based alternatives like Max-Margin Markov Networks (M^3N) (Taskar et al., 2004) and Structured SVM (Tsochantaridis et al., 2004) could be used in the same way. These particular solutions are not applicable here because they require (implicitly) enu-

merating all possible derivations which is not possible with a model like ours that makes only few independence assumptions.

Exposure bias happens because model is not exposed to its errors during training time. With *dynamic oracle* (Goldberg and Nivre, 2012) parser is trained on its own prediced history instead of the gold sequence of actions (static oracle). Whenever the model is in some sampled state (which is not necessarily a good state), we train the model to pick the transition that is a beginning of a path that would lead the parser to the ending state with the highest achievable metric score from that state. Finding such a transition is not trivial for all systems and all metrics (Cross and Huang, 2016). To this date there have been no proposals for a dynamic oracle for CCG parsing with F1 metric over CCG dependency structures and it is not even clear if there is a polynomial solution to this problem. Therefore this is not an option that we can use.

An alternative is to use a reinforcement learning algorithm REINFORCE (Williams, 1992). REINFORCE samples derivations for training just like dynamic-oracle, but does not require design of a task-specific oracle extraction algorithm. Instead, it implicitly minimises the expected error of the desired metric. Fried and Klein (2018) have shown that in some circumstances REINFORCE can give results almost as good as dynamic oracle, but it requires using additional techniques to compensate for high variance of the training method. The method of applying REINFORCE to the discriminative parser is straightforward because sampling trees from the discriminative parser is easy. However, that is not the case for the generative model from which we have to sample both trees and sentences at the same time. That is why we will apply REINFORCE only to the discriminative model.

Imbalanced probability causes a search bias so the way it was addressed by Stern et al. (2017) is to modify the search itself. Stern et al. (2017) introduced a *word synchronous beam search* (WordSync) in which all the hypotheses that are competing with each other are guaranteed to have the same number of expensive actions.

Most of these methods are either not applicable (exact CRF, exact M^3N, dynamic oracle), or they solve only some subset of the previously mentioned biases. However, we can resort to some approximate methods to global models. For instance, instead of enumerating all hypotheses to compute normalization we could use a beam search as an approximation. This was done for CRF objective in (Zhou et al., 2015; Andor et al., 2016) and for (single-violation) M^3N objective in (Wiseman and Rush, 2016). They all need to compare in some way the gold hypothesis to the rest of the beam, but the issue arises when the gold hypothesis falls out of the beam. For that situation they use different heuristics. CRF approximation of Zhou et al. (2015) and Andor et al. (2016) uses Early update of Collins and Roark (2004). During training with Early update, the beam search is stopped when the gold hypothesis falls out of the beam and the parameter update is performed. In the *Beam-Search Optimization* (BSO) method of Wiseman and Rush (2016) an alternative heuristic is used from Daumé III and Marcu (2005) called LaSO. LaSO does the update at the same point as Early but, unlike Early, it continues decoding by removing all elements of the beam except for the gold one. This will potentially result in another update for the same training instance.

We have implemented most of these methods in attempt to improve incremental CCG parsing. However, even though many of them gave some improvements over the baseline, none of them was good enough to give a reasonably good parser. To further improve the model we propose two novel approaches: Gen-Rescaling and BSO-*-All where * stands for both Early and LaSO heuristics.

4.1 New method I: Rescaling

Word Synchronous beam search did solve the imbalanced probabilities issue for RNNG models, but its improvements do not transfer to CCG. Here we take a different approach: instead of adapting the search to the model, we adapt the model to the search. Since probabilities are imbalanced a possible way to solve that issue is to balance them by exponentiating them with some weight. We use the Beam Search Optimisation (BSO) LaSO method from the previous section to train only 3 new parameters: one for supertagging actions, one for word generation actions and one for reduce actions. These three numbers will be used to exponentiate the probability of the respective actions and by that put them on the same scale. This method is applied to a generative model and therefore addresses label-bias and imbalanced probability bias, but it does not address exposure-bias.

After training the rescaled generative model

scores every new transition sequence with: $p(a)^{2.17}p(t)^{1.08}p(w)^{1.00}$ where a, t and w are parsing, tagging and word generation actions respectively, while the numbers are the three learned parameters that put probabilities in the same scale.

4.2 New method II: BSO-*-All

To address all biases together using only a single techniques we modified margin approaches to minimize all margin violations in the beam instead of just the single one. When gold hypothesis falls out of the beam BSO-Early and BSO-LaSO use only the most violated hypothesis to update the parameters. However, there is no good reason not to update against all violations present in the beam. LeCun et al. (2006) argue that the good property of CRF models is that they simultaneously decrease weight of all bad hypotheses simultaneously. Our BSO-*-All approach can be seen as an approximation of this idea using a beam. This small modification does not slow down training in any significant manner (we already have a forward pass for all the additional hypotheses because they are in the beam) and it gives significant improvements in parsing accuracy.

5 Experiments

We have conducted experiments on English CCGbank (Hockenmaier and Steedman, 2007). For evaluation we use F1 score over labelled-directed and unlabelled-undirected dependencies. The parser is implemented in Scala and uses DyNet (Neubig et al., 2017) for the neural computation. The code is available on github.[3]

There are two dependency types often used in CCG parsing research: first one from (Clark et al., 2002) which is much closer to the typical CCG notion of dependencies and the second one from (Clark and Curran, 2007) which is more formalism-neutral but less expressive. The only implementation of the second method is the one in C&C parser and is not able to handle all the categories that come from CCGbank. This is the reason why most previous work on incremental CCG parsing has used the dependencies of Clark et al. (2002). In order to be able to compare to them we use the same dependencies.

[3] https://github.com/stanojevic/Rotating-CCG/tree/incremental_max_margin

5.1 Models tested

We have tested the following methods:

Disc Incremental discriminative model (the baseline).

Disc-REINFORCE Discriminative model trained using REINFORCE to maximise the expected reward (F1 score of CCG dependencies).

Gen Generative model that additionally has word generation transitions.

Gen-WordSync Same generative model but decoded with word-synchronous beam with main beam size 100, word-beam size 10 and no fast-tracking.

Gen-Rescaled Generative model that uses additional three weights to put the probabilities of all actions on the same scale.

BSO-Early-Single and BSO-LaSO-Single Un-normalised model trained with Early and LaSO updates but only with single violation per update as proposed in Wiseman and Rush (2016).

BSO-Early-All and BSO-LaSO-All Same as above but with minimizing all violations present in the beam. We refer to them together as BSO-*-All.

CRF-Early Globally normalized model with Early update as proposed in (Zhou et al., 2015; Andor et al., 2016).

CRF-LaSO Same as above but modified to use LaSO instead of Early update.

All beam approximation methods used beam of size 32. The number of samples in REINFORCE is 32 and it includes a gold hypothesis for stability as suggested by Fried and Klein (2018).

CRF-Early achieved accuracy of 36.9%, BSO-Early-Single of 51.7% and Gen-WordSync of 58.1% which are all way below the baseline. CRF-Early and BSO-Early-Single update methods probably gave bad results because the training is too unstable with Early heuristic that often does not get to learn from the whole transition sequence. We are not sure why Gen-WordSync gave bad results. It could be that word-synchronous decoding while addressing the imbalanced probability search bias introduces some other search bias that is even more harmful. Another reason could

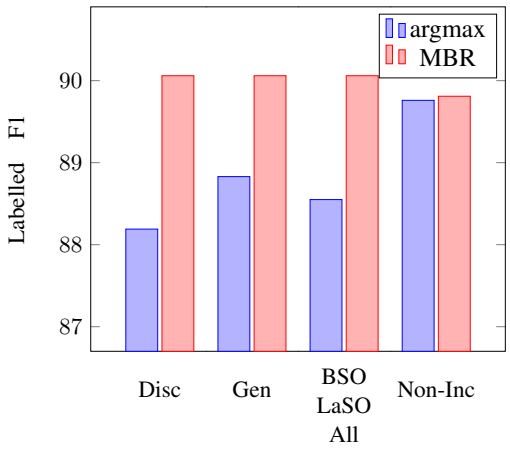

Figure 1: Reranking 100 samples of dev set sentences generated by discriminative non-incremental model.

be that, unlike RNNG, we have introduced an additional bottleneck of supertagging transitions that would require additional modifications. We will not consider these methods in the rest of the paper.

5.2 Results: Incremental Beam Search

Figure 2 shows the results for all the other methods with different beam sizes. REINFORCE training does improve the robustness of the discriminative model. It improved greedy decoding by 10% more than any other method, but due to label-bias it cannot exploit the benefits of a larger beam.

The generative model addresses the label bias which is evident from relatively good results with a bigger beam. When on top of the generative model we add Rescaling parameters the model gets even more benefit as the beam gets larger.

The BSO-LaSO-Single model that addresses all three biases at the same time gets very good results and is outperformed by Gen-Rescaled model only in the context of a very large beam. Gen-Rescaled and BSO-LaSO-Single get close to 80% but do not go above it. Our BSO-*-All modification to beam search optimisations gives significantly better results already with a very small beam. With beam of size 8 BSO-LaSO-All crosses the border of 80% and it improves all the way to 82.7%. This is only 4.8% lower than the upper bound set by the non-incremental model. BSO-LaSO-All is a small modification over BSO-LaSO-Single but is responsible for more than 5% of improvement over it. The importance of updating for all violations is particularly striking with the case of BSO-Early where the accuracy increases by 29%.

CRF-Early already has the property of updating against all bad hypotheses in the beam but it differs from our best method in the type of loss (logistic vs max-margin) and the update heuristic (Early vs LaSO). We have also tried modifying the CRF method to use LaSO (CRF-LaSO) which made the model significantly better than the original CRF-Early but still much lower than BSO-*-All.

5.3 Results: Reranking

Is the gap between non-incremental models and incremental models due to the imperfect search or to the imperfect prediction models? To test that we have conducted an experiment where the models need only to rerank a list of 100 derivations sampled from non-incremental model for each sentence in the development set. This puts beam search out of the equation and tests only how good are the models as discriminators between good and bad trees. The samples have trees of mixed quality: the worst score a parser could get by reranking the trees is 67.8 F1 while the best is 95.8 F1.

The results in Figure 1 show that the gap between incremental and non-incremental models is around one point of F1-score. This is much smaller than the results with beam search would lead us to expect. Also here the generative model outperforms BSO-LaSO-All. This means that the primary reason for success of BSO-LaSO-All over Gen in beam search is probably due to its incremental scoring (a property that was also noticed by Goyal et al. (2019) for similar models) and/or lack of imbalanced probability bias.

We have also conducted reranking using Minimum-Bayes Risk (MBR) method (Kumar and Byrne, 2004) which finds the hypothesis that would minimise the expected loss under some metric. In the parsing context that means finding the tree with the best expected F1-score (Goodman, 1996; Titov and Henderson, 2006; Stanojević and Sima'an, 2015). MBR is defined only for probabilistic models, but as Titov and Henderson (2006) show it could also be adapted and applied to non-probabilistic models, such as our BSO-LaSO-All model.

Figure 1 shows that while MBR does not make any significant difference for the non-incremental model, it makes a huge difference for the incremental models. With MBR they all manage to outperform the non-incremental model. However, we should not credit this right away to the quality of the incremental models. As Fried et al. (2017)

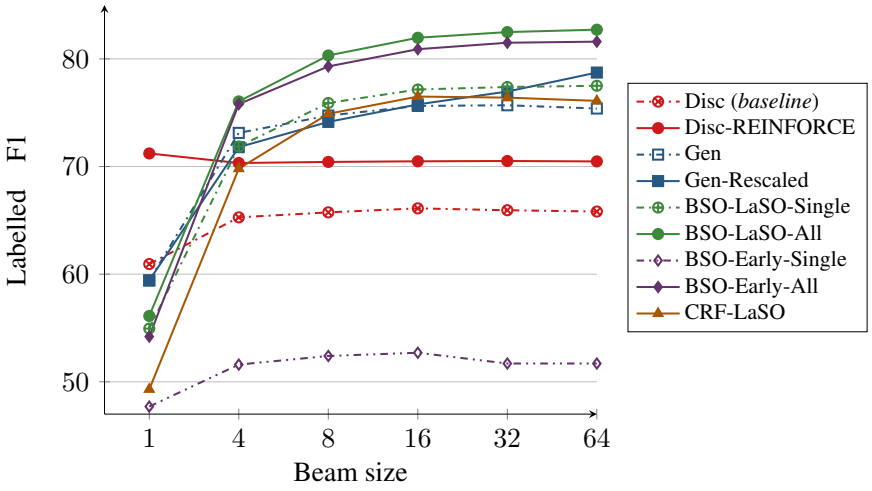

Figure 2: Influence of beam size on the dev results.

point out, improvements in reranking with a different model could be a result of model ensembling.

5.4 Results: Test set performance

Table 1 compares our strongest method on the test set against all the previously published incremental CCG models. The results show that it outperforms all the previous incremental models when using beams of the same size. The improvement is even bigger with the larger beam. Even thought our primary goal is not to compete with non-incremental parsers, our incremental model outperforms some widely used non-incremental CCG parsers such as EasyCCG (Lewis and Steedman, 2014). The result is particularly good for unlabelled dependencies.

We also report the results of applying MBR reranking using incremental model over the samples generated by the non-incremental model. This model outperforms other incremental and non-incremental models on all metrics.

			Tag	UF	LF
Incremental	Hassan et al. (2008)	beam= 1	—	59.0	—
	Ambati (2016)	beam= 1	74.6	67.5	57.5
	this work	beam= 1	78.8	69.9	55.8
	Goyal et al. (2019)	beam= 5	85.5	—	—
	this work	beam= 5	90.1	92.2	82.1
	Ambati (2016)	beam=16	90.8	88.3	80.8
	this work	beam=16	91.4	91.5	82.3
	this work	beam=64	**92.0**	**92.3**	**83.4**
Non-Incremental	Lewis and Steedman (2014)		93.0	88.6	81.3
	Ambati et al. (2015)		91.2	89.0	81.4
	Hockenmaier (2003)		92.2	92.0	84.4
	Zhang and Clark (2011)		93.1	—	85.5
	Clark and Curran (2007)		94.3	93.0	87.6
	Stanojević and Steedman (2019)		95.4	95.8	90.2
	this work	MBR reranking	**95.6**	**95.9**	**90.6**

Table 1: Results on the test set. The results of Non-Incremental parsers are shown only as a reference.

6 Other relevant work

The incremental CCG parser of Ambati (2016) uses the linear model trained with a structured perceptron objective and the early update heuristic. Given the simplicity of that model, it performs surprisingly well. The reason is the fact that the structured perceptron addresses all the biases identified in our paper. Our work has been an attempt to bring these benefits to more modern neural models.

Another interesting approach to tackle label-bias while keeping the probabilistic interpretation is the error-states model of Vaswani and Sagae (2016). This model in its original formulation would not be computationally efficient in our setting because there are too many instances of error-states to be trained on in CCG parsing caused by large number of transitions. Possibly some modification based on sampling could remedy this.

There has also been some recent work on reducing the imbalanced probability bias. Mabona et al. (2019) propose an algorithmic solution for organising beam search into buckets that have the same number of expensive transitions. Crabbé et al. (2019) propose a sampling based approach with the same motivation of controlling which hypotheses are being compared.

Of relevance for the CCG incrementality are Sturt and Lombardo (2005) and Demberg et al. (2013) who claimed that human sentence processing is more incremental than CCG allows under

SCH for sentences like:

The pilot embarrassed Mary and put herself in a very awkward situation.

Here a male gender-biased interpretation of the antecedent "the pilot" conflicts with a feminine bound reflexive "herself". The eye-movements show processing difficulty as soon as "put herself" is read, rather than being delayed until the completion of the VP by the PP. This allows subject binding to be established by VP coordination.

Stanojević et al. (2020) argue that Sturt and Lombardo's result is explained by the fact that the category for "put" is predictive of a future PP, allowing establishment of binding in advance of parsing without strict incrementality or compromising SCH.

7 Conclusion and Future work

The methods discussed here have been applied to the task of incremental CCG parsing, but they are not limited to CCG or even to parsing as a task. In principle, they could be applied to any task involving sequential structure prediction. We see this as the most interesting use case not only for the BSO-*-All training method but also for having an incremental CCG parser. Such parsers can potentially make much more informed decisions about the next word, compared to the models based on mere sequence of words prefix, by including semantic and referential meaning (Altmann and Steedman, 1988), as well as syntax.

Acknowledgments

This work was supported by ERC H2020 Advanced Fellowship GA 742137 SEMANTAX grant.

References

Steven P. Abney and Mark Johnson. 1991. Memory requirements and local ambiguities of parsing strategies. *Journal of Psycholinguistic Research*, 20:233–249.

Anthony Ades and Mark Steedman. 1982. On the order of words. *Linguistics and Philosophy*, 4:517–558.

Gerry Altmann and Mark Steedman. 1988. Interaction with context during human sentence processing. *Cognition*, 30(3):191–238.

Bharat Ram Ambati. 2016. *Transition-based combinatory categorial grammar parsing for English and Hindi*. Ph.D. thesis, University of Edinburgh.

Bharat Ram Ambati, Tejaswini Deoskar, Mark Johnson, and Mark Steedman. 2015. An Incremental Algorithm for Transition-based CCG Parsing. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 53–63. Association for Computational Linguistics.

Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. 2016. Globally normalized transition-based neural networks. *arXiv preprint arXiv:1603.06042*.

L. Bottou. 1991. *Une Approche Theorique de l'Apprentissage Connexionniste: Applications a la Reconnaissance de la Parole*. Ph.D. thesis.

Stephen Clark and James R Curran. 2007. Wide-coverage efficient statistical parsing with CCG and log-linear models. *Computational Linguistics*, 33(4):493–552.

Stephen Clark, Julia Hockenmaier, and Mark Steedman. 2002. Building deep dependency structures with a wide-coverage CCG parser. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 327–334. Association for Computational Linguistics.

Michael Collins and Brian Roark. 2004. Incremental parsing with the perceptron algorithm. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 111. Association for Computational Linguistics.

Benoit Crabbé, Murielle Fabre, and Christophe Pallier. 2019. Variable beam search for generative neural parsing and its relevance for the analysis of neuro-imaging signal. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1150–1160, Hong Kong, China. Association for Computational Linguistics.

James Cross and Liang Huang. 2016. Span-based constituency parsing with a structure-label system and provably optimal dynamic oracles. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1–11.

Fernando Cuetos and Don C. Mitchell. 1988. Cross-linguistic differences in parsing: Restrictions on the use of the late closure strategy in spanish. *Cognition*, 30(1):73 – 105.

Hal Daumé III and Daniel Marcu. 2005. Learning as search optimization: Approximate large margin methods for structured prediction. In *Proceedings of the 22nd international conference on Machine learning*, pages 169–176. ACM.

Vera Demberg, Frank Keller, and Alexander Koller. 2013. Incremental, predictive parsing with psycholinguistically motivated tree-adjoining grammar. *Computational Linguistics*, 39:1025–1066.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 334–343, Beijing, China. Association for Computational Linguistics.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A Smith. 2016. Recurrent Neural Network Grammars. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 199–209.

Lyn Frazier. 1979. *On comprehending sentences: Syntactic parsing strategies*. Ph.D. thesis, University of Massachusetts, Amherst.

Daniel Fried and Dan Klein. 2018. Policy gradient as a proxy for dynamic oracles in constituency parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 469–476.

Daniel Fried, Mitchell Stern, and Dan Klein. 2017. Improving Neural Parsing by Disentangling Model Combination and Reranking Effects. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 161–166.

Yoav Goldberg and Joakim Nivre. 2012. A dynamic oracle for arc-eager dependency parsing. *Proceedings of COLING 2012*, pages 959–976.

Joshua Goodman. 1996. Parsing Algorithms and Metrics. In *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*, ACL '96, pages 177–183, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kartik Goyal, Chris Dyer, and Taylor Berg-Kirkpatrick. 2019. An empirical investigation of global and local normalization for recurrent neural sequence models using a continuous relaxation to beam search. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1724–1733.

Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. 2005. Bidirectional LSTM Networks for Improved Phoneme Classification and Recognition. In *Proceedings of the 15th International Conference on Artificial Neural Networks: Formal Models and Their Applications - Volume Part II*, ICANN'05, pages 799–804, Berlin, Heidelberg. Springer-Verlag.

John Hale. 2001. A probabilistic Earley parser as a psycholinguistic model. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pages 1–8. Association for Computational Linguistics.

John Hale, Chris Dyer, Adhiguna Kuncoro, and Jonathan Brennan. 2018. Finding syntax in human encephalography with beam search. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2727–2736. Association for Computational Linguistics.

John T. Hale. 2014. *Automaton Theories of Human Sentence Comprehension*. CSLI, Stanford.

Hany Hassan, Khalil Sima'an, and Andy Way. 2008. A syntactic language model based on incremental CCG parsing. In *SLT*, pages 205–208. IEEE.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8).

Julia Hockenmaier. 2003. *Data and models for statistical parsing with Combinatory Categorial Grammar*. Ph.D. thesis, University of Edinburgh. College of Science and Engineering. School of Informatics.

Julia Hockenmaier and Mark Steedman. 2007. CCGbank: a corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computational Linguistics*, 33(3):355–396.

Shankar Kumar and William Byrne. 2004. Minimum Bayes-Risk Decoding for Statistical Machine Translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Yann LeCun, Sumit Chopra, Raia Hadsell, Marc Aurelio Ranzato, and Fu Jie Huang. 2006. A tutorial on energy-based learning. In *Predicting structured data*. MIT Press.

Mike Lewis and Mark Steedman. 2014. A* CCG parsing with a supertag-factored model. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 990–1000.

Amandla Mabona, Laura Rimell, Stephen Clark, and Andreas Vlachos. 2019. Neural generative rhetorical structure parsing. *arXiv preprint arXiv:1909.11049*.

William Marslen-Wilson. 1973. Linguistic structure and speech shadowing at very short latencies. *Nature*, 244:522–523.

Graham Neubig, Chris Dyer, Yoav Goldberg, Austin Matthews, Waleed Ammar, Antonios Anastasopoulos, Miguel Ballesteros, David Chiang, Daniel Clothiaux, Trevor Cohn, Kevin Duh, Manaal Faruqui, Cynthia Gan, Dan Garrette, Yangfeng Ji, Lingpeng Kong, Adhiguna Kuncoro, Gaurav Kumar, Chaitanya Malaviya, Paul Michel, Yusuke Oda, Matthew Richardson, Naomi Saphra, Swabha Swayamdipta, and Pengcheng Yin. 2017. Dynet: The dynamic neural network toolkit. *arXiv preprint arXiv:1701.03980*.

Fernando C. Pereira. 1985. A new characterization of attachment preferences. In David R. Dowty, Lauri Kartunnen, and Arnold M. Zwicky, editors, *Natural language parsing: Psychological, Computational, and Theoretical perspectives*, pages 307–319.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Philip Resnik. 1992. Left-corner parsing and psychological plausibility. In *Proceedings of the 14th International Conference on Computational Linguistics, COLING 92*, pages 191–197.

Brian Roark. 2001. Probabilistic top-down parsing and language modeling. *Computational linguistics*, 27(2):249–276.

Brian Roark and Mark Johnson. 1999. Efficient probabilistic top-down and left-corner parsing. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 421–428. Association for Computational Linguistics.

Edward P Stabler. 1991. Avoid the pedestrian's paradox. In *Principle-based parsing*, pages 199–237. Springer.

Miloš Stanojević and Edward Stabler. 2018. A Sound and Complete Left-Corner Parsing for Minimalist Grammars. In *Proceedings of the Eight Workshop on Cognitive Aspects of Computational Language Learning and Processing*, pages 65–74. Association for Computational Linguistics.

Miloš Stanojević, John Hale, and Mark Steedman. 2020. Predictive Processing of Coordination in CCG. In *Proceedings of the 33rd Annual CUNY Conference on Human Sentence Processing*, Amherst, Massachusetts. University of Massachusetts.

Miloš Stanojević and Khalil Sima'an. 2015. Reordering Grammar Induction. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 44–54, Lisbon, Portugal. Association for Computational Linguistics.

Miloš Stanojević and Mark Steedman. 2019. CCG Parsing Algorithm with Incremental Tree Rotation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics, Volume 1 (Long Papers)*. Association for Computational Linguistics.

Mark Steedman. 1989. Grammar, interpretation, and processing from the lexicon. In William Marslen-Wilson, editor, *Lexical Representation and Process*, pages 463–504. MIT Press, Cambridge, MA.

Mark Steedman. 2000. *The Syntactic Process*. MIT Press, Cambridge, MA.

Mitchell Stern, Daniel Fried, and Dan Klein. 2017. Effective inference for generative neural parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1695–1700.

Andreas Stolcke. 1995. An efficient probabilistic context-free parsing algorithm that computes prefix probabilities. *Computational linguistics*, 21(2):165–201.

Patrick Sturt and Vincenzo Lombardo. 2005. Processing coordinated structures: Incrementality and connectedness. *Cognitive Science*, 29:291–305.

Ben Taskar, Carlos Guestrin, and Daphne Koller. 2004. Max-Margin Markov Networks. In *Advances in neural information processing systems*, pages 25–32.

Ivan Titov and James Henderson. 2006. Loss Minimization in Parse Reranking. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 560–567, Sydney, Australia. Association for Computational Linguistics.

I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. 2004. Support Vector Machine Learning for Interdependent and Structured Output Spaces. In *International Conference on Machine Learning (ICML)*, pages 104–112.

Ashish Vaswani and Kenji Sagae. 2016. Efficient structured inference for transition-based parsing with neural networks and error states. *Transactions of the Association for Computational Linguistics*, 4:183–196.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.

Sam Wiseman and Alexander M Rush. 2016. Sequence-to-Sequence Learning as Beam-Search Optimization. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1296–1306.

Yue Zhang and Stephen Clark. 2011. Shift-reduce CCG parsing. In *Proceedings of the 49th Annual*

Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, pages 683–692. Association for Computational Linguistics.

Hao Zhou, Yue Zhang, Shujian Huang, and Jiajun Chen. 2015. A neural probabilistic structured-prediction model for transition-based dependency parsing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1213–1222.