DS-GA-1012 Final Projet: Do URNNGs Learn Meaningful Constituency Grammar in English?

Bichen Kou bk2374@nyu.edu
Xiaocheng Li xl3119@nyu.edu
Jiayao Liu jl9875@nyu.edu
Jimin Tan jt3545@nyu.edu

Abstract

Understanding the behavior of unsupervised grammar induction models is crucial to how syntactic knowledge is extracted without annotations in training data. To inspect if URNNG discovers meaningful constituency grammars in English, we test URNNGs (Kim et al., 2019) on BLiMP benchmark and analyze its parsing performance. Besides, we evaluate the model consistency by comparing Corpus Self F1 and binary tree structures. We find that URNNGs learn constituency grammar only in certain tasks, with consistent performances.

1 Introduction

Unsupervised Recurrent Neural Network Grammars (URNNGs) (Kim et al., 2019) model sentence structures through a generative perspective without seeing an example of labeled data. Experiments on English and Chinese have shown that, URNNGs perform as well as their supervised counterparts (Kim et al., 2019) in language modelling. URNNGs are also competitive in unsupervised grammar induction to models like ON-LSTM (Yikang Shen and Courville, 2018). It suggests that URNNGs can possibly learn meaningful linguistic structures without seeing an example of a tree structure (Yikang Shen and Courville, 2018). To test whether URNNGs learns meaningful constituency grammars, we evaluate URNNGs' performance on BliMP, and find that URNNG has advantage over baseline on tasks that requires understanding of sentence structures.

Through analyzing self F-1 score, average tree depth and unique parse per sentence, we find behaviors of URNNGs are consistent. Visualization of parsing indicates that although URNNGs are bad at learning PennTreeBank parsing structures, it is consistent in branching direction and punctuation handling. [1]

[1] https://github.com/leehaausing/urnng_analysis

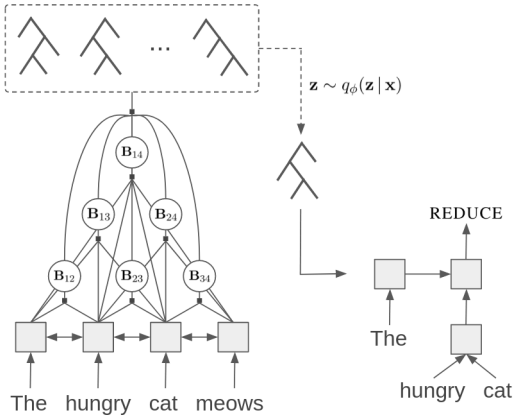

Figure 1: Overview of URNNG approach from Kim et al. (2019).

2 Related Work

URNNGs URNNGs (Kim et al., 2019) are unsupervised models that jointly implements grammar induction and language modeling. The architecture of URNNG consists of an inference network and a generative model, as showed in Fig. 1. The inference network generates a sequence of shift-reduce actions (Henderson, 2004), which guide the generative model to form a constituency tree. URNNGs are limited to generate the most simple version of a constituency tree. Other variants of unsupervised grammar induction models include ON-LSTM (Shen et al., 2018), PRPN (Yikang Shen and Courville, 2018), and DIORA (Drozdov et al., 2019).

Testing linguistic knowledge Testing language models with linguistic knowledge helps researchers understand learning outcomes of the model in various linguistic phenomena. Linzen et al. (2016) tests language models on subjective-verb agreement. Gulordava et al. (2018) expands the test to more agreement types of number. Marvin and

Linzen (2018) brings out a more general framework of designed pairs with minimal differences, containing phenomena such as subjective-verb agreement, objective relative clause agreement, and reflexive Anaphora. In our project, we will evaluate URNNG on BLiMP (Warstadt et al., 2019), a more comprehensive benchmark evaluating language models' learning on linguistic knowledge, covering 12 linguistic phenomena within 67 sub-datasets.

Testing model consistency Williams et al. (2018) analyze RL-SPINN (Bowman et al., 2016) and ST-Gumbel (Choi et al., 2018) trained on the Natural Language Inference (NLI) data (Bowman et al., 2015). They evaluate the mean and variance of model performance on NLI tasks, compute Self F1 scores, and visualize model outputs, i.e. parsed sentences. Self F1 scores is computed by taking outputs of one model as labels and computing the average F1 scores of other models.

3 Data

3.1 PennTreeBank

We primarily use Penn Treebank (Marcus et al., 1993) which contains a variety of corpus including 1989 Wall Street Journal (WSJ) articles with syntactic annotation. The gold annotations will be utilized for evaluation with regards to grammar induction and language modeling tasks.

3.2 BLiMP

BLiMP is a benchmark evaluating language models' learning on linguistic knowledge, covering 12 linguistic phenomena within 67 sub-datasets, each containing 1000 minimal pairs isolating specific contrasts in syntax, morphology, or semantics (Warstadt et al., 2019) . Data is automatically generated according to expert-crafted grammars, and the aggregate human agreement with the labels is 96.4 %.[2]

4 Methodology

4.1 Binary Tree Comparison

As shown in Table 1, URNNGs generate deeper trees than true parsing but shallower than Left-branching parsing, which implies that the binary tree given by URNNG is skewed to left or right.

[2]https://github.com/alexwarstadt/blimp

Type of Parse	Mean Depth	Max Depth
True Parse	7.896	20.271
Left Branching Parse	12.978	36.486
Random Parse	5.380	1.234
Random Balanced Parse	4.499	0.851
URNNG Parse	11.028	34.764

Table 1: Comparison of mean and maximum of tree depth across multiple baselines, true parsing and URNNG parsing.

4.2 Testing with Linguistics Knowledge

In our project, we will evaluate the URNNG model on BLiMP (Warstadt et al., 2019) which consists of minimally different sentence pairs. Each sentence pair contains a grammatical and an ungrammatical sentence. Here the acceptability of a sentence is computed as the log likelihood given by the URNNG model. It is considered as a correct prediction if the log likelihood of the grammatical sentence is higher than ungrammatical sentence. Besides, an RNNLM model is trained as a baseline model, the performance of which will be compared to URNNG on Blimp tasks. Following the experiment of (Kim et al., 2019) on syntactic evaluation with minimally different sentence pair, we filter out the sentence pair that contains <unk> token in a sentence pair, leaving 20550, about 30.67% valid sentence pairs in our project. Based on the results, we will analyze whether the model has learned specific syntactic knowledge to identify the grammatical contrast and how sensitive the model is to acceptability across different categories.

4.3 Model Consistency

To validate the consistency of trained URNNG models, we generate 10 models with 10 random seeds. Self F1, a metric to measure whether the models can reliably converge to the same parses as stated in (Williams et al., 2018), is calculated for these 10 models. Additionally, we divide 2514 sentences into 6 groups according to their length and visualize the distribution of mean depth per sentence and number of unique parses per sentence in each length group. To better understand the difference between URNNG parses and PennTreeBank(PTB) parses, we visualize confident but wrong predictions from URNNG. Such representative examples also expose whether there are any recognizable consistent or inconsistent patterns in the URNNG parsing results.

Phenomena	URNNG	RNNLM
Anaphor	0.6670	0.6660
Argument structure	0.6445	0.6359
Binding	0.6934	0.7191
Control/raising	0.5074	0.5052
Determiner-n agrmnt	0.7218	0.7571
Ellipsis	**0.4984**	0.4528
Filler-gap	0.4687	**0.5190**
Irregular forms	0.7720	**0.8242**
Island effects	0.4807	0.5013
NPI licensing	0.4776	0.4709
Quantifier	**0.6126**	0.5672
Subj-verb agrmnt	0.6897	0.6864

Table 2: URNNG and RNNLM performance on 12 liguistic phenomena (Accuracy)

5 Result

5.1 Testing with linguistic knowledge

We first present the performance of URNNG on different syntactic tasks. Table 2 shows the general performance of URNNG on 12 linguistic phenomena separately. The accuracy is weighted by the number of samples from each contributing tasks (67 in total). We also provide a comprehensive presentation of URNNG performance on the full 67 tasks separately as shown in Fig. 8.

Among the 12 linguistic phenomena, URNNG outperforms RNNLM in 7 of them, among which two are significant, i.e. Ellipsis and Quantifier. On the other hand, in Filler-gap and Irregular forms, URNNG perform worse than our RNNLM. These results are showing in more detail in Fig. 9. In conclusion, URNNG has minimal improvements over RNNLM on most tasks.

5.1.1 Parsing comparison

We analyze URNNG performance on these 12 linguistic phenomena in detail. We firstly analyze these performances in general, then visualize parses of two minimal pairs within Quantifier and Filler-gap tasks respectively.

Ellipsis means omitting part of the sentence and still keeping the sentence grammatical and meaningful. URNNGs perform better than RNNLM in this task. Similarly, Quantifier tests the limitations of using quantifiers, in which a good parse helps. However, in Filler-gap which deals with dependencies in phrasal movements, URNNG performs worse than our RNNLM baseline.

We then analyze URNNG parses in Quantifier and Filler-gap, examples are showed in Fig 2 and Fig 3. In Quantifer, URNNG generates a parse with reasonable syntax. However good and bad sentences generates same parses in Filler-Gap example. Moreover, they are the same type of parses generated by the bad sentences. We might come to the conclusion that URNNG can easily come to the same type of parses which does one left-branching parsing followed by right-branching.

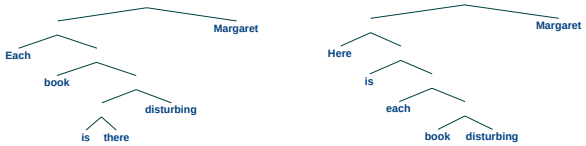

Figure 2: Quantifier example

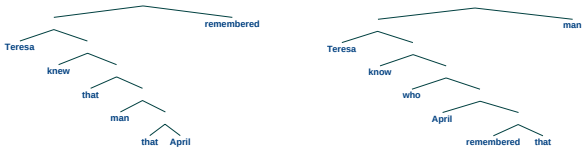

Figure 3: Filler-gap example

https://www.overleaf.com/project/5ebf154da8a9cc0001e891f7

5.2 Model Consistency

5.2.1 Corpus self F1 score

As shown in Fig.4, the mean Corpus self F1 score of 10 models are all above 80. A high Corpus self-corpus F1 score indicate high similarity across model predictions. Despite that models with seeds 1000, 1818 and 2345 have relatively larger score variance, the majority of models have high self Corpus F1 and small variance. Thus, we conclude that the URNNG models with different seeds are consistent when making predictions.

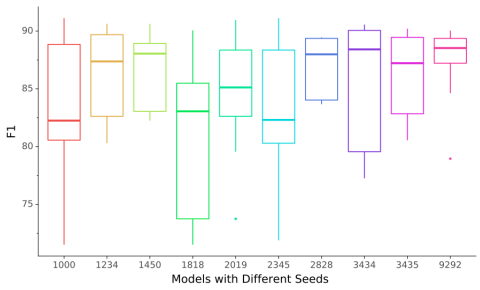

Figure 4: predictions of each model with a specific seed will be treated as true labels, for which we compute the corpus F1 scores of the other 9 models.

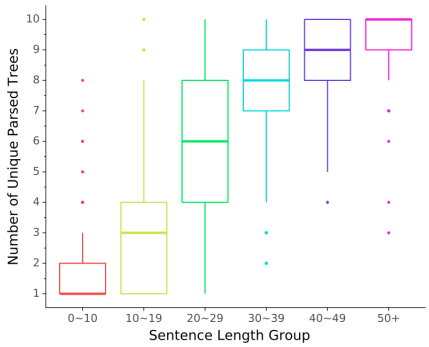

Figure 5: The distribution of unique sentence number in each sentence length group.

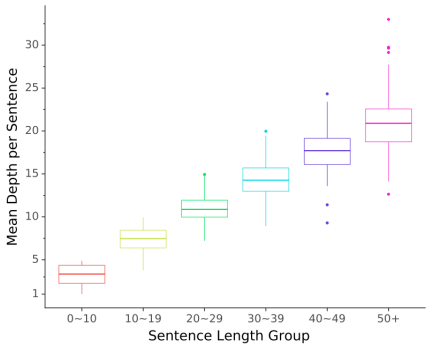

Figure 6: The distribution of mean depth per sentence in each sentence length group.

5.2.2 Binary tree parsing comparison

Intuitively, parsing variation increases as the sentence gets longer. This trend is observed in our data. As shown in Fig.5, it is unlikely for different URNNGs to generate a unique parsing for sentence with more than 30 words. As shown in Fig.6, the mean depth of predicted parses by 10 different URNNG modesl increases linearly as sentence gets longer. Different from number of unique trees, mean tree depth has a relatively consistent interquartile range and few outliers.

Parsed trees generated by URNNGs are deeper than PTB parsed tree, prone to right-branching, and more sensitive to punctuation. As shown in the Fig.7, URNNG trees tend to branch out the punctuation as a single leaf as early as possible, while the PTB gold trees usually keep punctuation for later branching. Out of 2514 sentences in the test set, URNNGs generate the right PTB parsing for only 32 sentences, indicating URNNG is bad at learning PTB structures. Moreover, there are 334 sentences with the same predicted structure from 10 different URNNG models, but 325 out of these 334 confident predictions are different from the PTB gold structures. In comparison, tendency of right-branching and handling of punctuation are the main causes of the difference. Although unable to learn the PTB structure, URNNG model is quite consistent in its branch direction and punctuation handling.

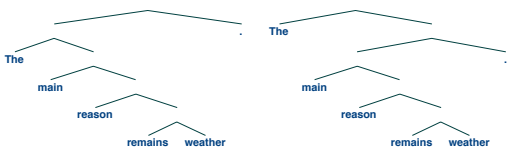

Figure 7: URNNG (left) vs Gold (right).URNNG are prone to right branching and tends to split punctuation marks as a single leaf as early as possible.

5.3 Conclusion

We conclude that URNNG has advantage and disadvantages over the baseline model. The performance difference are coming from the language modeling task that URNNG and RNNML adopted. In specific linguistic phenomena like Ellipsis and Quantifier, URNNG outperforms RNNLM by a large margin. Recall that URNNG is trained with shift-reduce which put more emphasis on modeling sentence structure than next word prediction task used in RNNLM. Since tasks like Ellipsis and Quantifier favors structure understanding, the superior performance by URNNG matches our expectation. URNNG perform worse in tasks that require more understanding of syntax. For example, the negative samples in filler gap tasks have relatively small modification to sentence structures and is therefore more difficult for URNNG to differentiate.

We discovered that URNNG generate consistence parsing trees from multiples models trained with different random seeds. The corpus self F1 score are high meaning outputs from a model are similar to the rest. URNNG generates deeper parsings than PennTreeBank (PTB) and parsed structures learned by URNNG is consistent in both punctuation handling and right-branching tendency.

The current parsing scheme is vulnerable to punctuation marks, and we need to further investigate methods to remove or integrate them without undermining syntactic structures. To extend on our current work, we can train a two-stage URNNG by pre-training RNNG and fine-tuning URNNG to gain better performance.

6 Contribution

Bichen Kou: train URNNG, implement Blimp evaluation and complete binary tree parsing comparison
Xiaocheng Li: train URNNG, RNNLM, implement Blimp evaluation and complete Blimp parsing analysis
Jiayao Liu: train URNNG, implement Blimp evaluation and complete Corpus self F1 analysis
Jimin Tan: train URNNG, implement Blimp evaluation and complete Blimp parsing analysis

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Samuel R. Bowman, Jon Gauthier, Abhinav Rastogi, Raghav Gupta, Christopher D. Manning, and Christopher Potts. 2016. A fast unified model for parsing and sentence understanding. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1466–1477, Berlin, Germany. Association for Computational Linguistics.

Jihun Choi, Kang Min Yoo, and Sang goo Lee. 2018. Learning to compose task-specific tree structures. In *AAAI*.

Andrew Drozdov, Patrick Verga, Mohit Yadav, Mohit Iyyer, and Andrew McCallum. 2019. Unsupervised latent tree induction with deep inside-outside recursive auto-encoders. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1129–1141, Minneapolis, Minnesota. Association for Computational Linguistics.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205, New Orleans, Louisiana. Association for Computational Linguistics.

James Henderson. 2004. Discriminative training of a neural network statistical parser. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 95–102, Barcelona, Spain.

Yoon Kim, Alexander Rush, Lei Yu, Adhiguna Kuncoro, Chris Dyer, and Gábor Melis. 2019. Unsupervised recurrent neural network grammars. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1105–1117, Minneapolis, Minnesota. Association for Computational Linguistics.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

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

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. 2018. Ordered neurons: Integrating tree structures into recurrent neural networks. *arXiv preprint arXiv:1810.09536*.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R Bowman. 2019. Blimp: A benchmark of linguistic minimal pairs for english. *arXiv preprint arXiv:1912.00582*.

Adina Williams, Andrew Drozdov, and Samuel R. Bowman. 2018. Do latent tree learning models identify meaningful structure in sentences? *Transactions of the Association for Computational Linguistics*, 6:253–267.

Chin-Wei Huang Yikang Shen, Zhouhan Lin and Aaron Courville. 2018. Neural language modeling by jointly learning syntax and lexicon. *arXiv:1711.02013*.

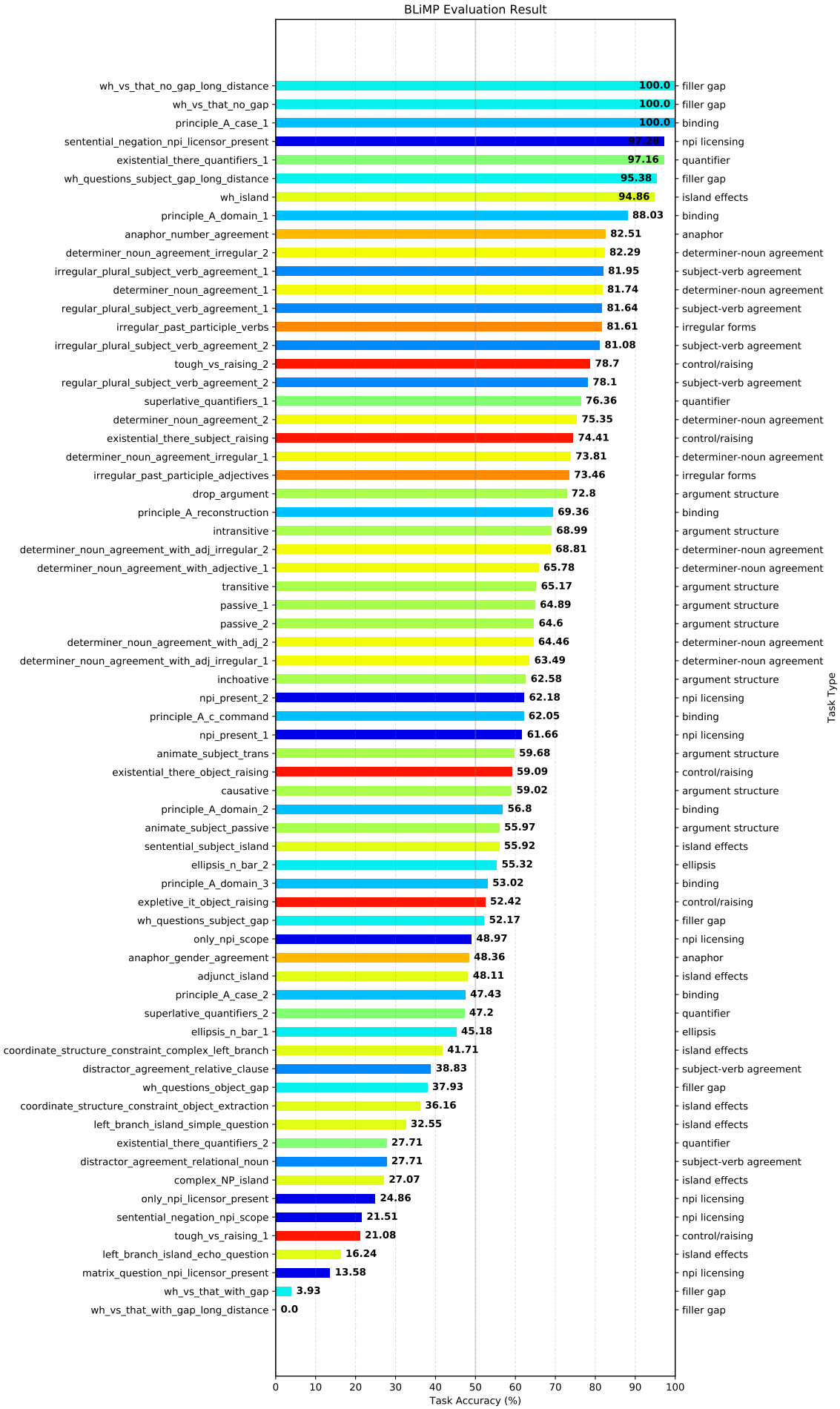

Figure 8: URNNG Performance on BLiMP

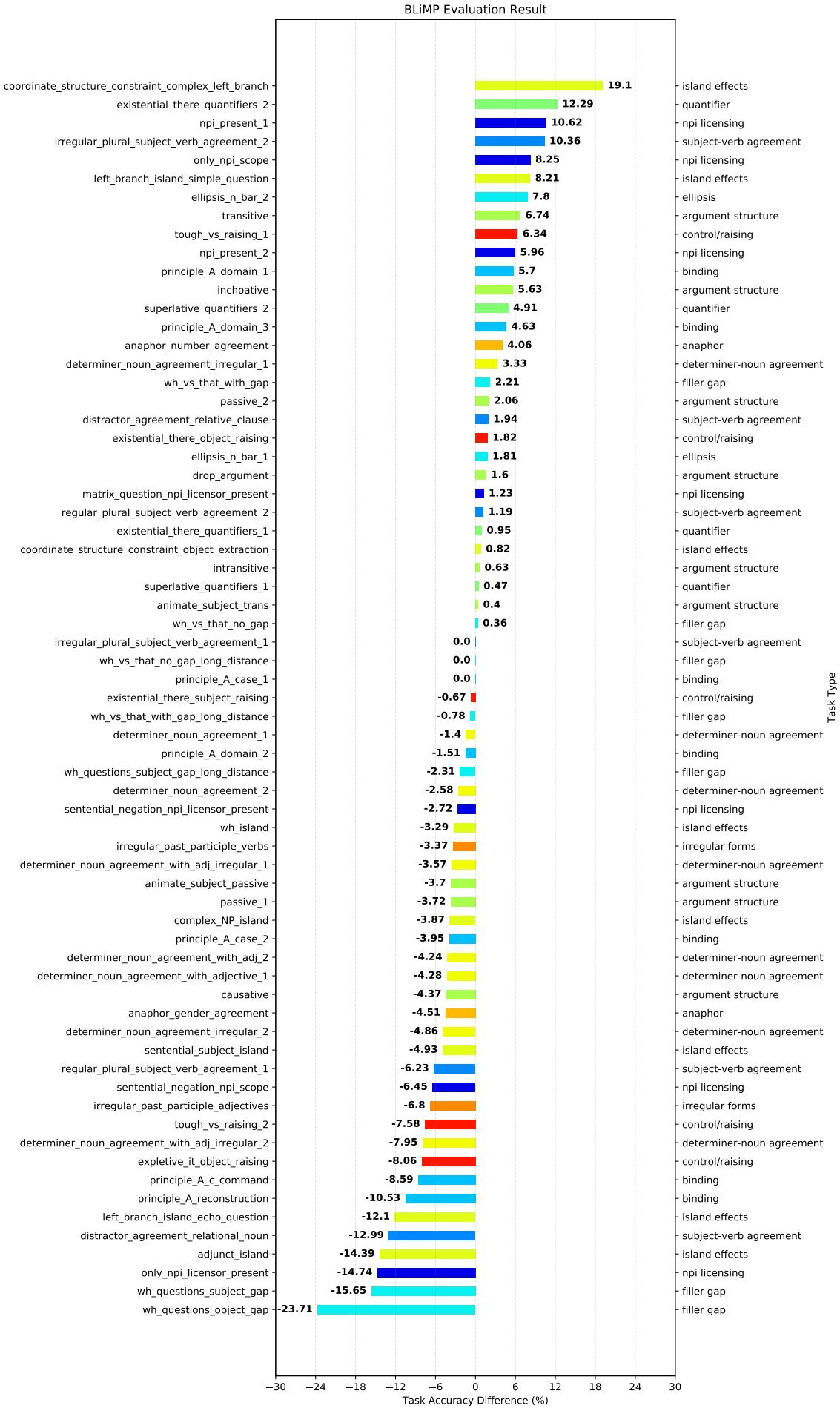

Figure 9: Difference in performance bewtween URNNG and RNNLM on BLiMP