Tree Transformer's Disambiguation Ability of Prepositional Phrase Attachment and Garden Path Effects

Lingling Zhou and **Suzan Verberne** and **Gijs Wijnholds**
Leiden Institute of Advanced Computer Science, Leiden University
l.zhou.5@umail.leidenuniv.nl
{s.verberne,g.j.wijnholds}@liacs.leidenuniv.nl

Abstract

This work studies two types of ambiguity in natural language: prepositional phrase (PP) attachment ambiguity, and garden path constructions. Due to the different nature of these ambiguities – one being structural, the other incremental in nature – we pretrain and evaluate the Tree Transformer of Wang et al. (2019), an unsupervised Transformer model that induces tree representations internally. To assess PP attachment ambiguity we inspect the model's induced parse trees against a newly prepared dataset derived from the PP attachment corpus (Ratnaparkhi et al., 1994). Measuring garden path effects is done by considering surprisal rates of the underlying language model on a number of dedicated test suites, following Futrell et al. (2019). For comparison we evaluate a pretrained supervised BiLSTM-based model trained on constituency parsing as sequence labelling (Gómez-Rodríguez and Vilares, 2018). Results show that the unsupervised Tree Transformer does exhibit garden path effects, but its parsing ability is far inferior to the supervised BiLSTM, and it is not as sensitive to lexical cues as other large LSTM models, suggesting that supervised parsers based on a pre-Transformer architecture may be the better choice in the presence of ambiguity.

1 Introduction

One of the core challenges for Natural Language Understanding (NLU) systems is that they must advertently deal with ambiguity; depending on the type of ambiguity, systems may or may not incorporate mechanisms for handling ambiguous input.

A typical example of *structural ambiguity*, i.e. an ambiguity that is not resolved without external context, is that of prepositional phrase attachment. An often used example, with its two possible interpretations, is given below:

(a) **I saw the man with the telescope**
(b) *I saw the man through the telescope*
(c) *I saw the man that had the telescope*

Another form of syntactic ambiguity is given by garden path effects (Bever, 1970), where a comprehender is guided toward a locally plausible but ultimately incorrect parse, exemplified in

(d) **The horse raced past the barn fell.**

Here, the reader initially interprets the horse to be the subject of 'raced', only to revise this assumption after reading the final verb 'fell'.

In this paper, we investigate to what extent Transformer-based models are sensitive to these ambiguities.To that end, we rely on the language modelling and parsing components of a model; the language modelling component allows for measuring surprisal, and thus measuring the presence and sensitivity of garden path effects (Futrell et al., 2019). The parsing component is required in order to measure a model's capacity to disambiguate cases of PP attachment. Rather than assessing whether models can identify ambiguity, we investigate *to what extent* models can disambiguate by means of their parsing components.

Thus, we focus on the Tree Transformer model (Wang et al., 2019), which adds a constrained attention layer to the base Transformer architecture to do unsupervised parsing while training on a language modelling objective. In doing so, the model allows us to evaluate the parser as well as the language model on the two different evaluation tasks. For comparison, we evaluate the *parsing as sequence labelling* approach of Gómez-Rodríguez and Vilares (2018), where a BiLSTM-based language model is trained in a supervised fashion. As an extra baseline, we compare against an end to end setup with LLM prompting (Tunstall et al., 2023).

In terms of evaluation data, we rely on two sources: for measuring PP attachment ambiguities, we modernize the existing prepositional phrase attachment corpus (Ratnaparkhi, 1998), using a standard BERT language model, to be more naturalistic and closer to the kind of data our parsers will have observed during training. Second, for measuring

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 12291–12301

garden path effects, we employ existing datasets from previous work (Futrell et al., 2019).

The results from our comparative evaluation not only assess the Tree Transformer's proficiency in parsing and language modeling but also provide a deeper understanding of its intrinsic mechanisms when handling complex grammar structures and ambiguity. We summarize our main contributions as follows:

- We convert the PP attachment corpus into naturalistic sentences using BERT.
- We train a Tree Transformer from scratch on the Penn Treebank, and evaluate it on our novel data to quantify the Tree Transformer's performance.
- We evaluate pre-trained BiLSTM on our novel data to quantify the model's performance.
- We quantify the garden path effect size of Tree Transformer and its sensitivity to subtle lexical cues, compared with models from other work.

All our code and data is available online.[1]

2 Background

PP Attachment ambiguity Due to the widespread presence of prepositional phrase attachment ambiguity in natural language, it has garnered significant attention from scholars. In early research, structure-based methods have been explored, such as the Right Association (Kimball, 1973) and Minimal Attachment methods (FRAZIER, 1978), which despite their popularity due to simplicity, exhibit notable shortcomings and perform suboptimally in practical applications. Additionally, statistics-based methods have been employed. Hindle and Rooth (1993) introduced the first corpus-based co-occurrence statistical method, known as "lexical association". To address sparsity issues in these methods, Collins and Brooks (1995) proposed a back-off model and other work has applied WordNet (Fellbaum, 1998) classes (Stetina and Nagao, 1997; Toutanova et al., 2004). However, these methods are limited due to their specificity, making them cumbersome for practical applications. Especially in the context of neural language modelling, the exploration of PP attachment ambiguity is limited.

Garden Path Effects On the other hand, garden path effects have been studied in the context of neural language models: van Schijndel and Linzen (2018a,b) demonstrate garden path effects in LSTM models by simulating human reading times. Surprisal theory (Hale, 2001; Levy, 2008) proposed that observed slowdowns are a result of the unpredictability of each word appearing in a sentence. Subsequently, the surprisal introduced by grammar-based language models has been shown to be correlated with reading time by Demberg and Keller (2008). Other work has shown that the surprisal in RNNs is a powerful predictor of human reading times (Frank and Bod, 2011; Goodkind and Bicknell, 2018). van Schijndel and Linzen (2018a) demonstrated the ability of RNNs to make reading time predictions comparable to grammar-based language models. In addition to validating the surprisal theory across different models, Futrell et al. (2019) tested multiple LSTMs and RNNG models to determine if they exhibit the garden path effect and observed their levels of surprisal in disambiguating sentences. Their results serve as a baseline in our work.

(Un)supervised parsing and language models Much previous research that attempts to incorporate tree structures into neural networks has focused on supervised syntactic parsing, relying on annotated parse trees. Through clever encoding of the parse tree this process can be simplified to sequence labelling, as proposed by Gómez-Rodríguez and Vilares (2018) for a BiLSTM-based model.

On the side of unsupervised parsing, researchers have explored various techniques, hoping that models could learn latent tree structures from unlabeled data without explicit syntactic annotations. Among them, Yogatama et al. (2016) depicted the problem as a reinforcement learning task. In addition, there have been attempts based on recurrent neural networks, such as PRPN (Shen et al., 2017), On-LSTM (Shen et al., 2018), and Tree-LSTMs (Tai et al., 2015), an variations like URNNG (Kim et al., 2019) and DIORA (Drozdov et al., 2019). More recently, adaptations of the base Transformer architecture (Vaswani et al., 2017) have been proposed for unsupervised parsing, like Transformer Grammars (Sartran et al., 2022), and the Tree Transformer Wang et al. (2019) which is our focus.

3 Datasets

In our experiments we use four datasets for evaluation, which we discuss in order below.

[1] https://github.com/L-innnng/Tree-Transformer-analysis

Category	Word
Verb v	provide
Noun n_1	services
Preposition p	for
Noun n_2	customers
Label l	V

Table 1: An original example from the prepositional phrase attachment corpus (Ratnaparkhi et al., 1994).

3.1 Prepositional Phrase Attachment

The prepositional phrase attachment corpus (Ratnaparkhi et al., 1994) was originally extracted the from the Penn Treebank (PTB) (Marcus et al., 1993) and was used to study PP attachment ambiguity. In this corpus, each example comes as a quadruple $\{v, n_1, p, n_2\}$ and a label l to indicate the actual attachment of the PP (p, n_2), which can be attached to the noun n_1 or the verb v, resulting in different syntactic structures and meanings. Table 1 displays an example from the original corpus.

Generating naturalistic sentences In order to have naturalistic example sentences that are still naturally ambiguous, we convert the examples from the PP attachment corpus into longer phrases in a structured way.

We only consider cases from the original corpus that were extracted from the WSJ-test set, because for these cases we can be sure that they will not occur in the PTB train set used for training the parser later on. We use a pretrained BERT model (Devlin et al., 2018) [2] on this set to expand the quadruple to 7-tuples $\{n_0, v, m_1, n_1, p, m_2, n2\}$ to form a complete sentence, leaving the original attachment labels unchanged. This is done as follows.

Prior to using the language model, data cleaning is necessary. Gerunds or present participle, past participles, and the verbs like "be" that do not conform to the required forms are filtered out. Symbols and numbers that fail to meet the requirements are removed. Subsequently, the process of generating the sentence begins.

We then mask the subject position, i.e. "[MASK] v n_1 p n_2", and take the personal pronoun with the highest score as the subject n_0. The following step is to mask the modifier of noun n_1 like "n_0 v [MASK] n_1 p n_2", and the noun, adjective, determiner and possessive pronoun with the highest score is used as m_1. The modifier of the second noun m_2 is obtained in the same way. Throughout this process we use the POS tagger from NLTK to identify words of the correct type. As a result we generate 1424 sentences used for evaluation. An example conversion of the item example in Table 1 is *They provide various services for their customers*.

[2] We specifically use Huggingface's `bert-base-uncased` implementation.

3.2 Garden Path Effects

In order to investigate models' sensitivity to garden path effects, we rely on three datasets introduced by Futrell et al. (2019): Main-verb/Reduced-relative (MV/RR), NP/Z (Overt Object) and NP/Z (Verb Transitivity).

MV/RR Ambiguity This dataset contains 28 examples where each example consists of 4 sentences. The first verb of the sentences is ambiguous. It can be considered both as the main verb of the sentence and as a word introducing a reduced relative clause. This ambiguity can persist for an extensive stretch of the following context until the disambiguator appears:

(1) a. The women brought the sandwich from the kitchen **fell** in the dining room

b. The women given the sandwich from the kitchen **fell** in the dining room

c. The women who was brought the sandwich from the kitchen **fell** in the dining room

d. The women who was given the sandwich from the kitchen **fell** in the dining room

In sentence 1a, the verb "brought" is initially analyzed as part of the main verb phrase, but upon the appearance of the disambiguator "fell", readers' comprehension is severely disrupted and the verb "brought" had to be reanalyzed as part of a relative clause. In contrast, the garden path effect theoretically should be reduced in sentences like 1b. Compared to "gave", "given" is more explicit, indicating that it should not be the main verb of the sentence. Furthermore, the garden path effect should be eliminated in sentences 1c, where the presence of the words "who was" makes it clear to the reader.

NP/Z Ambiguity This ambiguity arises from a noun phrase which can act as either the direct object of the main verb in the subordinate clause or as the subject of the main clause. There are two datasets with a similar configuration using overt

objects and intransitive verbs to disambiguate, respectively. Additionally, using a comma to mark the end of a clause makes the sentence easier at the disambiguator. Both of them contains 24 examples, each with 4 sentences.

Overt Object As shown in sentence 2a, prior to encountering “burst", “shot" is naturally interpreted as a transitive verb with “the woman" as its direct object. However, upon the appearance of “burst", readers realize that “shot" should be considered as an intransitive verb with “the woman" as the subject of the main clause. In sentence 2b, introducing the explicit object “the gun" to the transitive verb effectively reduces or eliminates the ambiguity before the disambiguator appears.

(2) a. As the gangster shot the woman **burst** into hysterics
b. As the gangster shot his gun the woman **burst** into hysterics
c. As the gangster shot, the woman **burst** into hysterics
d. As the gangster shot his gun, the woman **burst** into hysterics

Verb Transitivity As shown in sentence 3b, it replaces the transitive verb “shot" with the intransitive verb “laughed" to reduce the garden path effect by making readers believe that “the women" is not its object. However, this lexical information about syntactic structure is so subtle that it is not known whether humans are as sensitive to it as theory.

(3) a. As the gangster shot the woman **burst** into hysterics
b. As the gangster laughed the woman **burst** into hysterics
c. As the gangster shot, the woman **burst** into hysterics
d. As the gangster laughed, the woman **burst** into hysterics

4 Experimental set-up

4.1 Tree Transformer Model

For training and validation of the Tree Transformer we use the standard splits of the Penn Treebank: sections 2 to 21 for training (WSJ-train) and 23 for testing (WSJ-test). Our trained model is a faithful replication of the model by Wang et al. (2019) except for differences in the preprocessing of the training set. Where Wang et al. (2019) simplified the training data to being composed of words with certain POS tags, we require punctuation to be included in the training data and so we retain it during training.

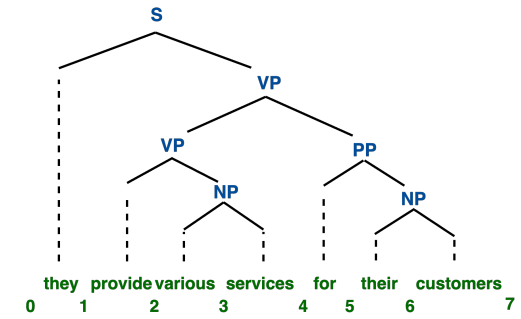


Figure 1: A constituency parse tree with its span representations.

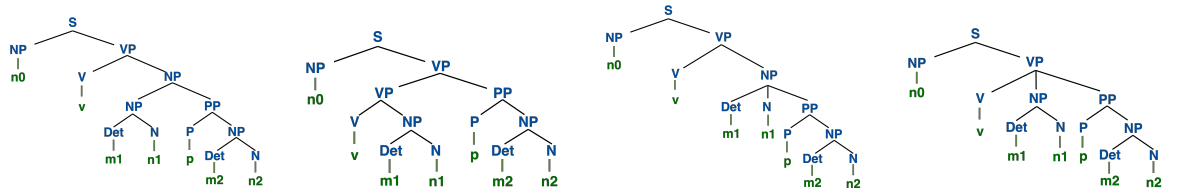
As for validation, following the evaluation setting established in prior research, the performance of the Tree Transformer is assessed by calculating F1 scores on the WSJ-test, which is processed in the same way as WSJ-train. We report macro-average F1 score over all the predicted trees against the ground-truth parse trees in the WSJ-test set, using unlabeled bracketing representations. For illustration, we provide the parse tree and bracketed span representation for a converted sentence in Figure 1.

4.2 BiLSTM Model

In order to study the difference between the Tree Transformer and a supervised parser, we include the pretrained BiLSTM model, i.e. $BILSTM^{\Phi'}_{m=2,e,ch}$, from Gómez-Rodríguez and Vilares (2018) to make additional comparison. This model utilizes NCRFpp, a sequence labeling framework grounded on bidirectional short-term memory networks (Yang and Zhang, 2018). It has been trained over the original WSJ-train. For comparison with Tree Transformer, the preprocessed WSJ-test is used in the test process. The predicted sequence labels will be decoded into a parse tree, which will then be converted into span representation to calculate F1 score in the same way as for the Tree Transformer.

4.3 LLM prompting

For a comparison against generative models, we include a prompting baseline with Zephyr 7B β (Tunstall et al., 2023). The prompt forces the model to predict verb or noun attachment, following the prompt included in the Appendix.



(a) Noun Attachment (binary) (b) Verb Attachment (binary) (c) Noun Attachment (ternary) (d) Verb Attachment (ternary)

Figure 2: Two binary ground-truth parse trees where the PP attaches to the noun (a) or to the verb (b), and two ternary ground-truth parse trees where the PP attaches to the noun (c) or to the verb (d).

4.4 Parsing PP Attachment

We consider two binary ground-truth parse trees for PP attachment, depending on whether the prepositional phrase attaches to the noun (Figure 2(a)) or to the verb (Figure 2(b)). For the BiLSTM model, the generated parsing tree is not necessarily a binary tree, thus we allow for ternary ground-truth parse trees as well (Figure 2(c) and 2(d)).

We calculate classification accuracy by considering whether the model returns a parse tree that either corresponds to the ground-truth tree, or is incomplete but still has attached the prepositional phrase to the noun or to the verb in a subtree. In the next section we give a more in-depth analysis of the different cases the models produced, as well as a breakdown of accuracy with respect to different verbs and prepositions.

4.5 Measuring Garden Path Effects

Similar to the previous work by Futrell et al. (2019), we investigate the behavior of the Tree Transformer and to what extent it reflects incremental representations of syntactic states. This is done by calculating *surprisal* of a word w_i as its inverse log-probability given a prior hidden state h_{i-1} of the model:

$$S(w_i) = -\log p(w_i \mid h_{i-1})$$

The measure of the garden path effect, for an example like sentence 1 can be quantified by subtracting the surprisal at the disambiguator in sentence 1c from the surprisal at the disambiguator in 1a and offsetting against the same difference in the anticipated less surprising case of 1b minus 1d.

We calculate both differences, and compare them with the garden path effects of LSTMs and RNNG in the study by Futrell et al. (2019) on datasets MV/RR as well as NP/Z (verb transitivity).

5 Results

We initially validate the retrained Tree Transformer model and the pretrained BiLSTM model on the WSJ-test data, giving the F1 scores in Table 2.

Model	F1-score
Tree Transformer, L=10	49.7
BiLSTM	75.8

Table 2: The F1 scores of Tree Transformer with 10 layers and pre-trained BiLSTM tested on WSJ-test.

The F1 score of the Tree Transformer model is aligned with the results reported by Wang et al. (2019), as we stick close to their original setting for preprocessing the data and computing F1 score. However, for the pretrained BiLSTM model, the F1 score deviates from the score reported by Gómez-Rodríguez and Vilares (2018) (90.6). This is due to few factors: first, they calculate the F1 score directly from the predicted labels, rather than the reconstructed span representation. Second, they use the micro- instead of macro-average F1 score. Third, the preprocessing of the data is different.

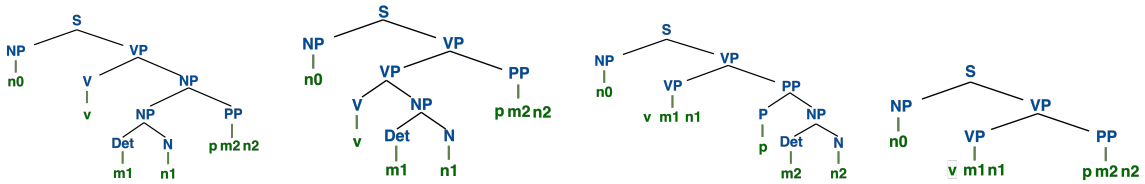
5.1 PP Attachment Results

Given that the PP attachment experiment evaluates parser performance, we first conduct an error analysis on the parse trees produced by the models, before analyzing the classification results.

Parsing error analysis An overview of the types of parse structures produced by the two models is given in Table 3.

For the Tree Transformer, close to 80% of cases are fully correct parses. In cases of incomplete (Partial) parse trees, we could still include 13 out of 14 cases where an attachment preference is clearly present. These cases, as well as the cases where no attachment decision could be made, are further detailed in Figure 3. We classify the remainder as 'Incorrect' in our accuracy results, with the main source of error the incapability of the parser to recognize the prepositional phrase (19.52% of cases).

For the BiLSTM model there are significantly fewer parsing errors, with a total of just 13 out of



(a) Noun attachment (b) Verb attachment (c) Undecidable attachment (d) Undecidable attachment

Figure 3: Partial parse trees produced by the Tree Transformer model.

Parse Structure	Tree Transformer Count	Tree Transformer Perc.	BiLSTM Count	BiLSTM Perc.
Correct parse	1120	78.65%	1406	98.74%
PP not found	278	19.52%	2	0.14%
Partial parse	14	0.98%	5	0.35%
Other incorrect	12	0.84%	11	0.77%

Table 3: Different parse structures from the Tree Transformer model and the BiLSTM model, with their absolute and relative frequency. We include correct parses, partial parses with decidable attachments in our result. The remaining cases are considered incorrect in the classification results.

1424. There are only 2 cases where the prepositional phrase is not found, significantly lower than for the Tree Transformer. Again we find cases where the model provides an incomplete parse structure in which nevertheless the PP attachment can still be determined. These partial parse trees are given in Figure 4 having a total of 5 cases.

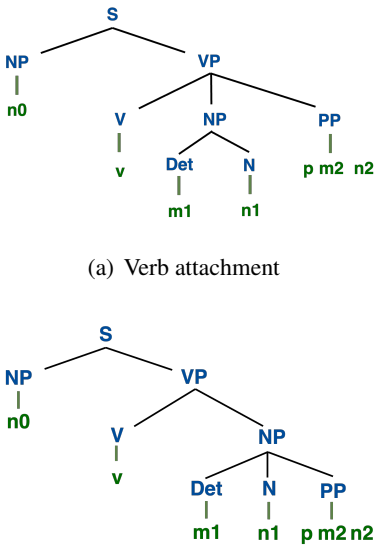
Classification results We display the two models' accuracy in Table 4, offset against the LLM baseline. As already shines through in the parsing

Model	Accuracy
Tree Transformer	47.2%
BiLSTM	79.4%
Zephyr	62.5%

Table 4: Accuracy on pp attachment sentences for Tree Transformer and BiLSTM models.

error analysis, we observe significant performance differences between the Tree Transformer model and the BiLSTM model. The fact that the latter model achieves an accuracy of 79.4% may be attributed to its supervised learning approach, where the high cost of training leads to substantial returns.

A further breakdown of model performance is given in the confusion matrix in Table 5. While the number of prepositions attached to nouns (868) is



(a) Verb attachment

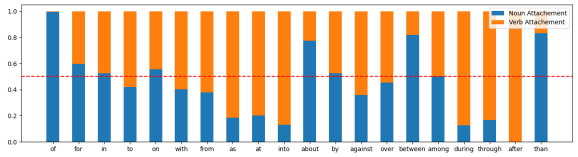
(b) Noun attachment

Figure 4: Partial parse trees produced by the BiLSTM model.

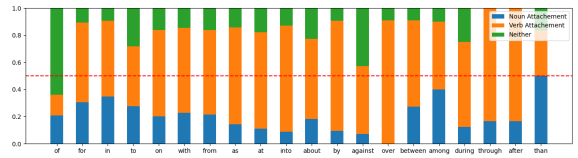
		Tree Transf. Prediction V	N	Incorr.	rec.
Gold	V	299	104	153	54%
	N	357	373	138	43%
	pr.	46%	78%		
BiLSTM					
Gold	V	442	110	4	79%
	N	171	688	9	79%
	pr.	72%	86%		
Zephyr					
Gold	V	53	503	0	10%
	N	31	837	0	96%
	pr.	63%	62%		

Table 5: Confusion matrix for the attachment accuracy on PP attachment sentences for the Tree Transformer and BiLSTM models.

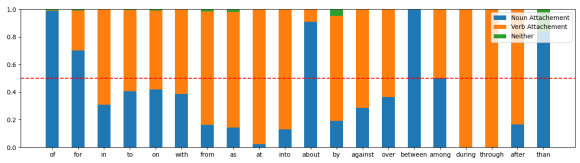
higher than that attached to verbs (556) in the data, the Tree Transformer tends to attach prepositional phrases to verbs (656) rather than nouns (477). Its ability to determine the PP attachment to nouns is particularly weak, and there is a significant gap compared to the gold standard attachments. Ad-



(a) Data

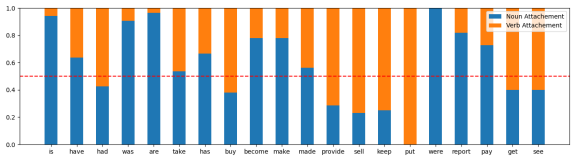


(b) Tree Transformer

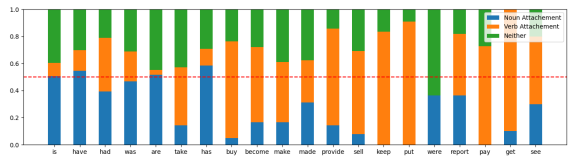


(c) BiLSTM

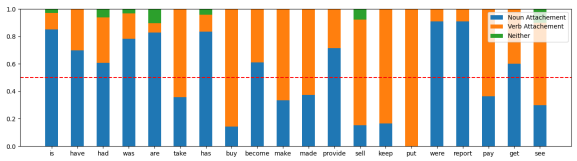
Figure 5: The proportions of noun attachment, verb attachments or neither/incorrect for the 20 most frequent *prepositions* in our dataset (a), by Tree Transformer (b) and BiLSTM (c), ordered by overall frequency.



(a) Data



(b) Tree Transformer



(c) BiLSTM

Figure 6: The proportions of noun attachment, verb attachments or neither/incorrect for the 20 most frequent *verbs* in our dataset (a), by Tree Transformer (b) and BiLSTM (c), ordered by overall frequency.

ditionally, the number of attachments where the attachment decision is neither a noun nor a verb is not negligible, contributing significantly to the low accuracy. This also indicates that the Tree Transformer is not highly sensitive to the attachment of prepositional phrases.

However, while the BiLSTM model also tends to attach prepositional phrases to verbs, this tendency is much less pronounced compared to the Tree Transformer. As shown in Table 5, the number of instances where BiLSTM attaches the pp to a verb is 613, which is higher than the actual attachment count of 556 in the dataset. Conversely, BiLSTM attaches prepositional phrases to nouns 798 times, which is lower than the actual attachment count of 868. Furthermore, the number of attachments that are neither nouns nor verbs is only 13, significantly lower than 291 that the Tree Transformer possesses.

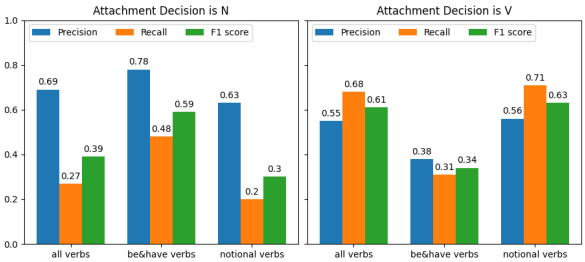
As for a comparison against the LLM prompting, we observe a large imbalance with the LLM deciding on noun attachment in most of the cases, leading to low accuracy.

Linguistic Analysis To investigate this imbalance in attachment decisions, we conducted an analysis of the parsing results combined with the distribution of verbs and prepositions in the sentences. This analysis aims to explore whether specific verbs or prepositions contribute to the attachment decision.

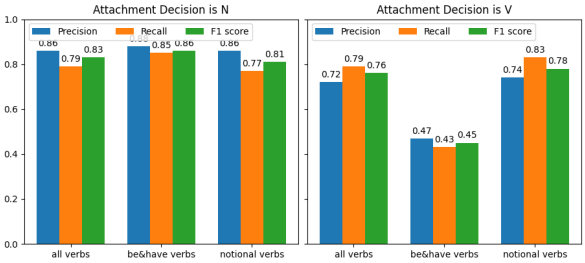
Figure 5 illustrates the distribution of attachment decisions by preposition, in both the data and the model output. Figure 5(a) illustrates the preference in the dataset, with prepositions 'of', 'about', 'between' and 'than' expectedly having a strong preference to attach to the noun. Figure 5(b) shows that Tree Transformer has a proportion trend that is roughly consistent with the data, albeit being skewed towards verb attachment. The prepositions 'than' and 'over' deviate, with the model having no preference for noun over verb attachment for 'than' even though verb attachment barely occurs in the dataset for this preposition; conversely for 'over' we find a relatively equal proportion of attachment choice, whereas the model always chooses verb attachment.

Figure 5(c) displays the same distribution for the BiLSTM model, indicating that the parsing results are closer to the distribution of the original dataset in terms of different prepositions. However, for certain prepositions, the distribution trend becomes more polarized. The BiLSTM model prefers noun attachment more than proportionate in the dataset for the prepositions 'of', 'for', 'about', 'between' and 'than', whereas it disproportionately prefers verb attachment for the prepositions 'during' and 'through'. Overall, BiLSTM exhibits a more pronounced attachment tendency difference in the prepositional dimension compared to Tree Transformer.

We present the same proportions but broken



(a) Tree Transformer



(b) BiLSTM

Figure 7: Evaluation metrics (precision, recall and F1 score) for different types of verbs when attachment decision is noun (N) or verb (V).

down by choice of verb in Figure 6. In the dataset (Figure 6(a)) we find that inflections of the functional verbs 'be' and 'have' generally have a higher preference for noun attachment than notional verbs. This difference tends to become more pronounced in the models' predictions: the Tree Transformer (Figure 6(b)) tends to choose noun attachment for functional verbs, and verb attachment for notional verbs, whereas BiLSTM (Figure 6(c)) displays a similar trend, albeit less extreme.

Given this observed dichotomy, we compute separate precision/recall/F1 scores for the aggregate functional verbs and aggregate notional verbs, displayed in Figure 7. The scores in the figure confirm worse performance in the case of functional verbs, indicating the sensitivity of the models to lexical cues in their decision-making.

5.2 Garden Path Effect Results

We combined Figure 8 and 9 to give a comparative analysis of the performance of each model in the garden path effect. Figure 8(a) depicts the magnitude of the garden path effect generated by Tree Transformer and verb form ambiguity. Figure 9(a) illustrates the garden path effect size by four models and verb form ambiguity directly taken from Futrell et al. (2019). Upon comparison, it is evident that Tree Transformer, like the other models, exhibits a fundamental garden path effect. The garden path effect in Tree Transformer is similar to that of TinyLSTM, but considerably smaller than in the other models. Additionally, if the model uses the morphological form of the verb as a cue for syntactic structure, instances where the verb has not changed to a passive participle form should exhibit a stronger garden path effect compared to situations where the change has already occurred. This is evident in the figures, where the red bars are higher than the green ones. Tree Transformer, along with two large LSTMs and RNNG, displays this pattern. Despite demonstrating crucial human-like garden path effect disambiguation due to the verb form ambiguity, it is noteworthy that significant garden path effect still persist in these models, even when the verb form is unambiguous like passive-participial verb. Regarding TinyLSTM, it does not exhibit sensitivity to ambiguous verb forms and reduced relative clauses.

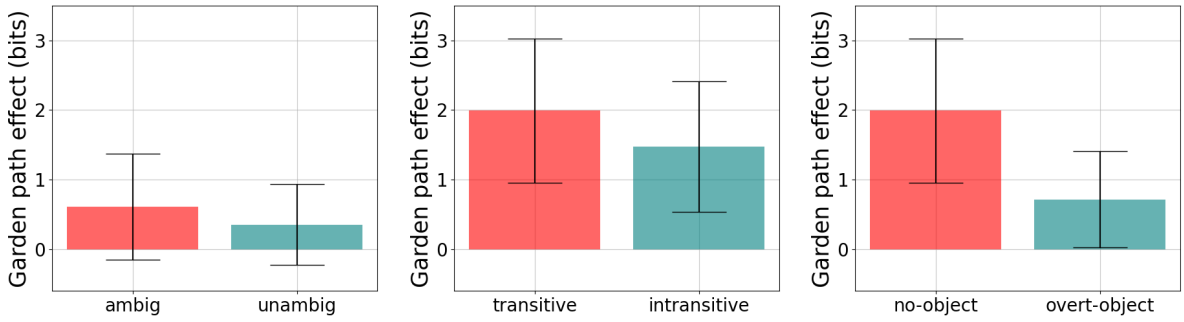
Figures 8(b) and 9(b) respectively illustrate the garden path effect sizes generated by Tree Transformer and four other models along with those of verb transitivity. Similarly, we observe the presence of the garden path effect in all models. Although smaller in Tree Transformer, even markedly smaller than in the two large LSTMs, it is higher than in TinyLSTM. As for the sensitivity, only the large LSTMs seem to be sensitive to the transitivity of embedded verbs, showing smaller garden path effects for intransitive verbs. Tree Transformer demonstrates sensitivity just below them, but noticeably higher than RNNG and TinyLSTM.

Figure 8(c) shows the garden path effect sizes generated by Tree Transformer and presence of an object. In comparison with Figure 8(b), it can be observed that Tree Transformer is much more sensitive to the presence or absence of an object than lexical cues such as verb transitivity.

6 Conclusion

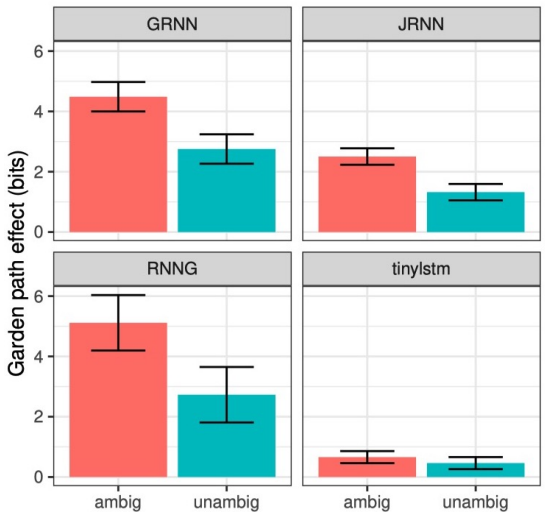
In this work, we prepared a novel prepositional phrase attachment ambiguity dataset, suitable for evaluating modern parsers on their capacity to recognize such ambiguities. We evaluated the Tree Transformer model, which combines an unsupervised parsing objective with masked language modelling. As such, we could evaluate both on prepositional phrase attachment ambiguity by analyzing the induced parse trees, as well as compare with prior work on measuring garden path effects through language model surprisal rate (Futrell et al., 2019).

In order to align with the datasets for our experiment, we trained a Tree Transformer from scratch

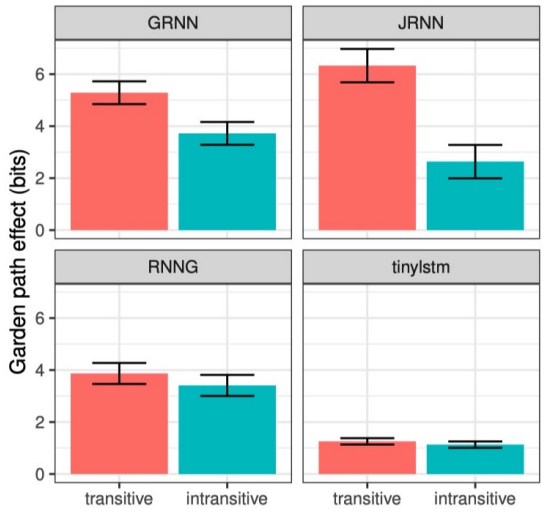


(a) MV/RR (b) NP/Z(verb transitivity) (c) NP/Z(overt object)

Figure 8: Average garden path effect size by Tree Transformer and disambiguation lexical clues on (a) MV/RR dataset; (b) NP/Z (verb transitivity) dataset; (c) NP/Z (overt object) dataset. Error bars depict 95% confidence intervals computed based on the standard error of the surprisals after subtracting out the mean surprisal (Masson and Loftus, 2003).



(a) MV/RR



(b) NP/Z(verb transitivity)

Figure 9: Figures are the original ones taken from the paper of Futrell et al. (2019). Average garden path effect by 4 models and disambiguation lexical clues on (a) MV/RR dataset; (b) NP/Z (verb transitivity) dataset.

and compared its performance with a supervised BiLSTM-based parser trained on constituent parsing as sequence labelling (Gómez-Rodríguez and Vilares, 2018).

Our study reveals that the Tree Transformer model performs suboptimal in disambiguating PP attachments at the sentence level, and less effective than the BiLSTM-based model. Primary factors are its poor ability to identify prepositional phrases and a tendency to attach prepositional phrases to verbs. We argue that this is likely to be caused by Tree Transformer's unsupervised nature. Additionally, the Tree Transformer demonstrates garden path effects across multiple datasets and exhibits varying sensitivity to different subtle lexical cues, generally being more sensitive to the presence of an object than to specific verb form. Overall, the evaluation suggests that an unsupervised model like Tree Transformer may be the lesser choice compared to a supervised model based on a pre-Transformer architecture, when it comes to natural language ambiguities.

7 Limitations

There is still room for improvement, and many related aspects warrant further investigation. For example, enriching the diversity and quantity of pp attachment sentences can enhance the depth of evaluation, and the accuracy of results. Improving Tree Transformer's ability to correctly identify prepositional phrase structures is essential for increased accuracy. Additionally, training Tree Transformer on larger datasets could be attempted to explore differences in performance of garden path effect compared to large LSTMs.

References

Thomas G Bever. 1970. The cognitive basis for linguistic structures. *Cognition and the development of language*.

Michael Collins and James Brooks. 1995. Prepositional phrase attachment through a backed-off model. In *Third Workshop on Very Large Corpora*.

Vera Demberg and Frank Keller. 2008. Data from eye-tracking corpora as evidence for theories of syntactic processing complexity. *Cognition*, 109(2):193–210.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

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.

Christiane Fellbaum. 1998. *WordNet: An electronic lexical database*. MIT press.

Stefan L Frank and Rens Bod. 2011. Insensitivity of the human sentence-processing system to hierarchical structure. *Psychological science*, 22(6):829–834.

L FRAZIER. 1978. On comprehending sentences: Syntactic parsing strategies. *Doctoral dissertation, University of Connecticut*.

Richard Futrell, Ethan Wilcox, Takashi Morita, Peng Qian, Miguel Ballesteros, and Roger Levy. 2019. Neural language models as psycholinguistic subjects: Representations of syntactic state. 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 32–42, Minneapolis, Minnesota. Association for Computational Linguistics.

Carlos Gómez-Rodríguez and David Vilares. 2018. Constituent parsing as sequence labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1314–1324, Brussels, Belgium. Association for Computational Linguistics.

Adam Goodkind and Klinton Bicknell. 2018. Predictive power of word surprisal for reading times is a linear function of language model quality. In *Proceedings of the 8th Workshop on Cognitive Modeling and Computational Linguistics (CMCL 2018)*, pages 10–18, Salt Lake City, Utah. Association for Computational Linguistics.

John Hale. 2001. A probabilistic Earley parser as a psycholinguistic model. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Donald Hindle and Mats Rooth. 1993. Structural ambiguity and lexical relations. *Computational Linguistics*, 19(1):103–120.

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.

John Kimball. 1973. Seven principles of surface structure parsing in natural language. *Cognition*, 2(1):15–47.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Annalen der Physik*, 19(2):313–330.

Michael EJ Masson and Geoffrey R Loftus. 2003. Using confidence intervals for graphically based data interpretation. *Canadian Journal of Experimental Psychology/Revue canadienne de psychologie expérimentale*, 57(3):203.

Adwait Ratnaparkhi. 1998. Statistical models for unsupervised prepositional phrase attachment. In *COLING 1998 Volume 2: The 17th International Conference on Computational Linguistics*.

Adwait Ratnaparkhi, Jeff Reynar, and Salim Roukos. 1994. A maximum entropy model for prepositional phrase attachment. In *Proceedings of the Workshop on Human Language Technology*, HLT '94, page 250–255.

Laurent Sartran, Samuel Barrett, Adhiguna Kuncoro, Miloš Stanojević, Phil Blunsom, and Chris Dyer. 2022. Transformer Grammars: Augmenting Transformer Language Models with Syntactic Inductive Biases at Scale. *Transactions of the Association for Computational Linguistics*, 10:1423–1439.

Yikang Shen, Zhouhan Lin, Chin-Wei Huang, and Aaron C. Courville. 2017. Neural language modeling by jointly learning syntax and lexicon. *ArXiv*, abs/1711.02013.

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

Jiri Stetina and Makoto Nagao. 1997. Corpus based PP attachment ambiguity resolution with a semantic dictionary. In *Fifth Workshop on Very Large Corpora*.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. 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 1556–1566, Beijing, China. Association for Computational Linguistics.

Kristina Toutanova, Christopher D. Manning, and Andrew Y. Ng. 2004. Learning random walk models for inducing word dependency distributions. In *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, page 103, New York, NY, USA. Association for Computing Machinery.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. 2023. Zephyr: Direct distillation of lm alignment.

Marten van Schijndel and Tal Linzen. 2018a. Modeling garden path effects without explicit hierarchical syntax. In *40th Annual Meeting of the Cognitive Science Society: Changing Minds, CogSci 2018*, pages 2603–2608. The Cognitive Science Society.

Marten van Schijndel and Tal Linzen. 2018b. A neural model of adaptation in reading. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4704–4710, Brussels, Belgium. Association for Computational Linguistics.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Neural Information Processing Systems*.

Yau-Shian Wang, Hung yi Lee, and Yun-Nung (Vivian) Chen. 2019. Tree transformer: Integrating tree structures into self-attention. In *Conference on Empirical Methods in Natural Language Processing*.

Jie Yang and Yue Zhang. 2018. NCRF++: An open-source neural sequence labeling toolkit. In *Proceedings of ACL 2018, System Demonstrations*, pages 74–79, Melbourne, Australia. Association for Computational Linguistics.

Dani Yogatama, Phil Blunsom, Chris Dyer, Edward Grefenstette, and Wang Ling. 2016. Learning to compose words into sentences with reinforcement learning. *ArXiv*, abs/1611.09100.

A Appendix

Given a sentence containing a prepositional phrase attachment ambiguity, the task is to indicate whether the prepositional phrase attaches to the main verb in the sentence, or to the main noun.

——

Sentence: He saw the person with the binoculars.
Attachment: Verb
Sentence: They understand the cost of living.
Attachment: Noun
Sentence: He saw the person with the binoculars.
Attachment: Noun
Sentence: I drove home with my bike.
Attachment: Verb
Sentence: We prepare a dinner for the family.
Attachment: [FILL]

Figure 10: The prompt we use to have the LLM predict verb or noun attachment. [FILL] indicates the generation of the model which is constrained to the two possible labels.