Resolving Prepositional Phrase Attachment Ambiguities with Contextualized Word Embeddings

Adwait Ratnaparkhi
University of California Santa Cruz
U.S.A.
adratnap@ucsc.edu

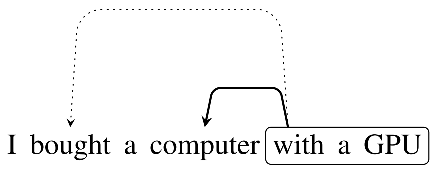
Atul Kumar
Roku
U.S.A.
kumatul06@gmail.com

Abstract

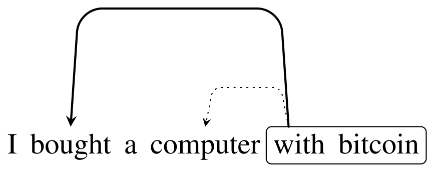
This paper applies contextualized word embedding models to a long-standing problem in the natural language parsing community, namely prepositional phrase attachment. Following past formulations of this problem, we use data sets in which the attachment decision is both a binary-valued choice as well as a multi-valued choice. We present a deep learning architecture that fine-tunes the output of a contextualized word embedding model for the purpose of predicting attachment decisions. We present experiments on two commonly used datasets that outperform the previous best results, using only the original training data and the unannotated full sentence context.

1 Introduction

Prepositional phrase (PP) attachment is a subproblem of natural language parsing in which the objective is to determine the likely attachment site of the preposition. The attachment site should correspond to the preferred semantic interpretation of the sentence.



In the above example, the preposition *with* could attach to *bought* or *computer*. The likely interpretation is that *GPU* is a sub-component of *computer*, which implies an attachment to *computer*.



In this example, the likely interpretation is that *bitcoin* is a payment method, which implies an attachment to *bought*.

PP attachment ambiguities are difficult to resolve because the candidate attachment sites look equally plausible from the perspective of natural language syntax. Deciding the best attachment site for a preposition often requires a semantic interpretation of the words in the sentence.

2 Previous work

Early work on this task uses relationships between *head words* of the phrases involved in the attachment decision. Hindle and Rooth (1993) predict attachments using co-occurrence statistics between the preposition and the candidate heads, drawn from an automatically built corpus of partial parses. Ratnaparkhi et al. (1994); Brill and Resnik (1994); Collins and Brooks (1995) use a wider variety of machine learning techniques to learn the attachment decision from annotated tuples of head words.

Later work uses a variety of external data sources to help with the attachment decision. E.g., Stetina and Nagao (1997) use features from WordNet (Miller, 1995) while Olteanu and Moldovan (2005) use features from FrameNet (Baker et al., 1998) and co-occurrence statistics drawn from the World Wide Web.

More recent work uses word embeddings and neural models. Belinkov et al. (2014) explore a number of neural composition architectures to combine the embeddings from the head words, while Dasigi et al. (2017) use the WordNet ontology to create context-sensitive word embeddings. Yu et al. (2016) use a scoring function on low-rank tensors, created from a variety of PP attachment features, while Madhyastha et al. (2017) use the tensor products of the word vectors in a multi-linear model. Recently, Do and Rehbein (2020) present German language PP attachment experiments using a neural scoring model with a biaffine transformation.

Field	Example
Verb h_1	made
Noun h_2	paper
Preposition p	for
Child noun $n2$	filters
Label	N

Table 1: Example in RRR data set. h_1 and h_2 are the verb and noun candidate attachment sites for p, respectively.

Field	Example
Candidate heads $h_1 \dots h_n$	made trip compete
Preposition p	for
Child noun $n2$	slots
Full sentence	eventually , about 250 made the trip to florida to compete for the available slots .
Annotated label	3

Table 2: Example in BLBG data set with joined sentence. h_3, or "compete", is the annotated attachment site for the preposition.

3 Data

In this work, we report results on the English-language data sets from Ratnaparkhi et al. (1994) and Belinkov et al. (2014), henceforth referred to as the RRR and BLBG data sets, respectively. Both datasets were extracted from the Penn Treebank (Marcus et al., 1993).

Each example in the RRR data set[1] consists of a tuple $(h_1, h_2, p, n2, L)$, where h_1 is the candidate verb head, h_2 is the candidate noun head, p is the preposition, $n2$ is the noun head child of the preposition, and $L \in \{N, V\}$ is the label that indicates a noun or verb attachment. We map $\{N, V\}$ to $\{2, 1\}$ for consistency with the BLBG format, described below.

Each example in the BLBG data set[2] consists of a tuple $(h_1 \dots h_n, p, n2, L)$, in which $h_1 \dots h_n$ are a list of candidate noun and verb heads, p is the preposition, $n2$ is the noun head child of the preposition, and $L \in \{1 \dots n\}$ denotes the index of the correct attachment in the list of heads. Compared to the RRR dataset, the BLBG dataset is a better approximation of the attachment decision faced in a full parsing task since it allows more than two possible attachment sites.

[1]https://github.com/adwaitratnaparkhi/ppa_transformer

[2]http: //groups.csail.mit.edu/rbg/code/pp

	Training	Development	Test
RRR	20801	4039	3097
BLBG	35359	n/a	1951

Table 3: Data set sizes of the RRR and BLBG data sets

In order to enable experiments in which the full sentence context is used as input, the original head word tuples were joined with the full sentences from which they were extracted. Starting from data generation scripts provided to us by the author[3] of the BLBG data set, we joined each example in the original BLBG data set with its full sentence context from the Penn Treebank Version 3. The sentences were lower cased to match the convention of the BLBG data. As we could not obtain the matching version of the Penn Treebank for the RRR dataset[4], the full sentence experiments were only conducted on the BLBG data set.

Table 3 shows the size of the data sets. Tables 1 and 2 contain example training instances.

4 Our method

In recent years, contextualized word embeddings (CWE), particularly those built using the Transformer (Vaswani et al., 2017) architecture, have accelerated progress in many NLP tasks. Past work on NLP tasks has typically followed a transfer learning strategy, in which a large pre-trained model is fine-tuned on a small annotated training set with task-specific labels. We apply this strategy using pre-trained BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019) models on data specific to the prepositional phrase attachment task, with the objective of improving task accuracy.

Figure 1 introduces a deep learning architecture that fine-tunes the output of a CWE module in order to predict attachments. We henceforth refer to it as the Fine Tuning for Headword Attachment (FTHA) model[5]. The full sentence or phrase is first passed through the CWE module, which yields a vector of token embeddings, shown in layer (1). In cases where the original word has been split into multiple tokens by the CWE tokenizer, we follow the convention in (Devlin et al., 2018) and use the

[3]Many thanks to Yonatan Belinkov for the data generation code.

[4]The RRR dataset cannot be joined with Penn Treebank Version 3 due to missing data.

[5]The FTHA implementation can be found at https://github.com/adwaitratnaparkhi/ppa_transformer

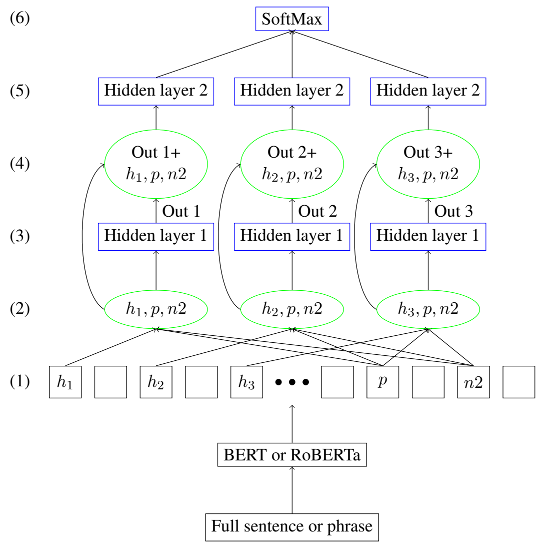


Figure 1: The Fine Tuning for Headword Attachment (FTHA) model. The diagram shows only 3 candidate head word attachment sites for the preposition, but the network can operate on an arbitrary number of candidate head words.

embeddings of the first sub-token. Next, a mask operation extracts a vector of n embedding triples $(h_1, p, n2) \dots (h_n, p, n2)$, shown in layer (2). Each triple represents a possible attachment. The embedding triples are passed to a hidden layer (3), and then concatenated with the hidden layer output using a "skip" connection, shown in layer (4). The result (4) is then passed to a second hidden layer (5). Finally, a softmax layer (6) returns a score for each attachment. The hidden layers in (3) and (5) share their parameters across their respective layer.

Each input example contains the information shown in Tables 1 and 2, together with a token sequence, and token indices of the head words. Depending on the experiment configuration, the token sequence is either the full sentence from which the example was drawn, or a phrase synthesized by concatenating the words in $(h_1 \dots h_n, p, n2)$.

5 Experiments

The FTHA model is trained and evaluated on both the RRR and BLBG data sets. Furthermore, we compare with past results on both data sets, as well as against an existing implementation[6] of a multiple choice fine-tuning architecture in the HuggingFace (HF) code library (Wolf et al., 2019).

The HF multiple choice implementation was designed for the SWAG (Zellers et al., 2018) task, and is described in Devlin et al. (2018). We configure the implementation to predict up to 8 choices, where each choice is a phrase formed from the head word triple $(h_i, p, n2)$, or a dummy label. For context input, we also give it either a full sentence or a phrase synthesized from all of the head words, depending on the experiment configuration. For our experiments, we configure it to use RoBERTa.

The HF implementation encodes the text of the choice (i.e., the head word triples) and any additional input, "pools" the embeddings, and then passes them to linear and softmax layers. In contrast, the FTHA model retains the head word embeddings that were computed in the context of the full sentence or phrase. This representation allows the FTHA model to work with a higher granularity of information from the input, compared to the HF implementation.

All experiments are measured using accuracy of the label classification. The allowable labels for each data set and their distributions are shown in Table 4.

Data set	Label	Frequency	Label	Frequency
RRR	N	10865	V	9936
BLBG	1	1585	5	4957
	2	7961	6	1478
	3	10113	7	225
	4	9022	8	18

Table 4: Allowable labels and their distribution in the RRR and BLBG training sets.

Table 5 shows results on the RRR dataset using the FTHA model with both BERT and RoBERTa. Only the head word tuples are used as input. The best result, FTHA with RoBERTa, is 0.7% higher than the one reported in Stetina and Nagao (1997), which is the previously best known result for the RRR dataset.

Table 6 shows results on the BLBG dataset with multiple experiment configurations. The best result in the `Head words only` configuration, FTHA with RoBERTa, outperforms the previously best result from Yu et al. (2016) by 1.7%. The best result in the `Full sentence` configuration, again FTHA with RoBERTa, outperforms that result by 4.1%. Table 7 shows an example where having the full sentence context helps.

The higher granularity representation of the FTHA model (with RoBERTa) gives slight accu-

[6]https://github.com/huggingface/transformers/tree/master/examples/multiple-choice

Paper	Dev	Test
(Stetina and Nagao, 1997)	n/a	88.1%
HF multiple choice	89.1%	88.4%
FTHA (RoBERTa)	89.3%	**88.8%**
FTHA (BERT)	88.8%	87.9%

Table 5: Results on the RRR data set.

Paper/Configuration and Model	Accuracy
(Belinkov et al., 2014)	
HPCD-full	88.7%
HPCD-full with parser	90.1%
(Dasigi et al., 2017)	
OntoLSTM-PP	89.7%
OntoLSTM-PP with parser	90.11%
(Yu et al., 2016)	
LRFR1-TUCKER & LRFR2-CP	90.3%
`Head words only` configuration	
FTHA (RoBERTa)	**92.0%**
FTHA (BERT)	91.1%
HF multiple choice	91.1%
`Full sentence` configuration	
FTHA (RoBERTa)	**94.4%**
FTHA (BERT)	93.2%
HF multiple choice	94.2%

Table 6: Results on the BLBG data set

racy gains compared to the HF baselines, for both the RRR and BLBG experiments. However, the differences are not statistically significant according to McNemar's test at $\alpha = 0.05$ significance level.

All RoBERTa and BERT experiments use the `roberta-base` and `bert-base-uncased` models, respectively. All models have at most 126M parameters, and training times were at most 1 hour on a UNIX server with an NVIDIA Titan RTX GPU with 24Gb RAM.

5.1 Hyperparameters

In our experiments, the hidden layers 1 and 2 in Figure 1 have a size of $3N$ and $4N$, respectively, where $N = 768$. During training, we use 3 epochs, a batch size of 16, the `AdamW` optimizer, a warmup step of 500, a weight decay of 0.01, and a learning rate of 10^{-3}.

6 Discussion

In past work, data sparsity has been a major challenge for corpus-based approaches to prepositional phrase attachment. To remedy data sparsity, re-

Field	Example
Candidate heads $h_1 \ldots h_n$	plan impose freeze
Preposition p	on
Child noun $n2$	fees
Full sentence	the plan would impose a brief freeze on physician fees next year .
Annotated label	2

Table 7: Test example where the `Head words only` configuration predicts incorrectly and the `Full sentence` configuration predicts correctly, for the FTHA (RoBERTa) model. h_2, or "impose", is the annotated attachment site for the preposition.

searchers have incorporated features from external resources like WordNet and FrameNet, with the idea that semantic features on words will have far less than sparsity than words themselves.

In our work, the CWE, built from a huge amount of unsupervised data, seem to compensate for the sparsity in the relatively small training sets. Our results on both the RRR and BLBG sets exceed those reported in past work, but without using WordNet, FrameNet, or any other external resources. All other results compared in Tables 6 and 5 use resources external to the training set.

We do not compare our experimental results with those in Do and Rehbein (2020) and Olteanu and Moldovan (2005) due to differences in test data.

The approach of Do and Rehbein (2020), conducted on German-language PP attachment data, resembles our work in that it uses CWE in a neural model. It also presents a scoring function for $(h_i, p, n2)$ triples, using a biaffine transformation of word representations derived from a bidirectional LSTM. It differs from our approach in that it considers all words in the sentence as potential attachment sites. And unlike our work, it incorporates additional information like part-of-speech tags, topological field tags, and auxiliary distributions computed from a large newspaper corpus.

The approach of Olteanu and Moldovan (2005) uses support vector machines and requires rich features that cannot be derived from the RRR dataset. Therefore it creates a new and larger data set with complex features from syntax trees, FrameNet, and co-occurrence statistics derived from an internet search engine. In contrast, our work focuses on only the CWE and excludes external resources.

Both the FTHA and HF experiments show benefit from using the full sentence context. For FTHA, the embeddings computed in the context of the whole sentence are likely more accurate than those computed from the head word phrase. For the HF implementation, the pooled embedding of the sentence carries enough information to boost the accuracy over just using the head words. Notably, in both cases, the sentence information is not explicitly annotated with any syntactic information, yet gives a sizable boost (+ 2% to 3%) in PP attachment accuracy.

Our work studies the PP attachment problem in isolation, and does not compare against the attachment decisions of a full parser. Other work (Belinkov et al., 2014; Dasigi et al., 2017) shows that PP attachment is still a problematic area for full parsers, and that an independently trained PP attachment model can improve the decisions of a full parser.

7 Conclusion

We present deep learning experiments for the prepositional phrase attachment task that exceed the accuracy of all previously published results on two widely used data sets. The results in our paper are 0.7% and 4.1% higher in absolute percentage points over the best previously published results on the RRR and BLBG data sets, respectively. We present a novel fine-tuning architecture that uses a higher granularity of information from the input, compared to a baseline implementation from HuggingFace. All our results were obtained without using external semantic data sources like WordNet or FrameNet. Lastly, we observe a big accuracy gain when the model is given the full sentence context vs. only the head words, despite having no syntactic annotation in the full sentence context.

References

C. Baker, C. Fillmore, and J. Lowe. 1998. The Berkeley FrameNet Project. In *COLING-ACL*.

Yonatan Belinkov, Tao Lei, Regina Barzilay, and Amir Globerson. 2014. Exploring compositional architectures and word vector representations for prepositional phrase attachment. *Transactions of the Association for Computational Linguistics*, 2:561–572.

Eric Brill and Philip Resnik. 1994. A rule-based approach to prepositional phrase attachment disambiguation. In *COLING 1994*.

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

Pradeep Dasigi, Waleed Ammar, Chris Dyer, and Eduard Hovy. 2017. Ontology-aware token embeddings for prepositional phrase attachment.

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.

Bich-Ngoc Do and Ines Rehbein. 2020. Parsers know best: German PP attachment revisited. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2049–2061.

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

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Pranava Swaroop Madhyastha, Xavier Carreras, and Ariadna Quattoni. 2017. Prepositional phrase attachment over word embedding products. In *Proceedings of the 15th International Conference on Parsing Technologies*, pages 32–43.

M. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Comput. Linguistics*, 19:313–330.

G. Miller. 1995. Wordnet: a lexical database for english. *Commun. ACM*, 38:39–41.

Marian Olteanu and Dan Moldovan. 2005. PP-attachment disambiguation using large context. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 273–280.

Adwait Ratnaparkhi, Jeff Reynar, and Salim Roukos. 1994. A maximum entropy model for prepositional phrase attachment. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Mo Yu, Mark Dredze, Raman Arora, and Matthew R Gormley. 2016. Embedding lexical features via low-rank tensors. In *Proceedings of NAACL-HLT*, pages 1019–1029.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. *CoRR*, abs/1808.05326.