Dynamic Feature Selection with Attention in Incremental Parsing

Ryosuke Kohita†, Hiroshi Noji§ and Yuji Matsumoto‡
†IBM Research
§Artificial Intelligence Research Center,
National Institute of Advanced Industrial Science and Technology (AIST)
†‡Graduate School of Information Science,
Nara Institute of Science and Technology (NAIST)
ryosuke.kohita1@ibm.com, hiroshi.noji@aist.go.jp, matsu@is.naist.jp

Abstract

One main challenge for incremental transition-based parsers, when future inputs are invisible, is to extract good features from a limited local context. In this work, we present a simple technique to maximally utilize the local features with an attention mechanism, which works as context-dependent dynamic feature selection. Our model learns, for example, which tokens should a parser focus on, to decide the next action. Our multilingual experiment shows its effectiveness across many languages. We also present an experiment with augmented test dataset and demonstrate it helps to understand the model's behavior on locally ambiguous points.

1 Introduction

This paper explores better feature representations for incremental dependency parsing. We focus on a system that builds a parse tree incrementally receiving each word of a sentence, which is crucial for interactive systems to achieve fast response or human-like behavior such as understanding from partial input (Baumann, 2013). The most natural way to achieve incremental parsing is using a transition system (Nivre, 2008), and for such parsers, the main challenge is to choose an appropriate action with only the local context information. While some recent transition-based parsers alleviate this difficulty by exploiting the entire input sentence with recurrent neural networks (Kiperwasser and Goldberg, 2016; Shi et al., 2017), one possible disadvantage is to require that all inputs are visible from the beginning, which should be problem when we try more strict incremental conditions such as simultaneous translation. Therefore there are still demands to explore the effective way to extract better feature representation from incomplete inputs.

In this paper, we incorporate a simple attention mechanism (Bahdanau et al., 2015) with an incremental parser and investigate its effectiveness during the feature extraction. Attention mechanism itself has firstly succeeded in machine translation, capturing relative importances of tokens on a certain step for a proper output (Bahdanau et al., 2015; Luong et al., 2015). The characteristic to weight on some features automatically and effectively can be applied to various tasks such as seq-to-seq parsing model (Vinyals et al., 2015), text summarization (Rush et al., 2015), dialogue generation (Shang et al., 2015), image captioning (Xu et al., 2015) in which the systems can enjoy performance gain by attending to specific clues depending on a given situation. We can also expect this behavior is helpful to fix the error which transition-based parsers often commits due to local ambiguities.

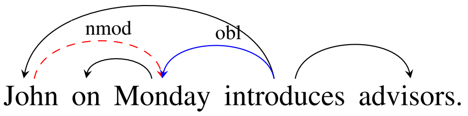

Figure 1: A locally ambiguous sentence. "*Monday*" should be analyzed as oblique of "*introduce*" while tends to be analyzed as a noun modifier of "*John*".

Proceedings of the 27th International Conference on Computational Linguistics, pages 785–794
Santa Fe, New Mexico, USA, August 20-26, 2018.

Figure 1 shows our motivating example, on which the standard transition-based parser fails and attaches "*Monday*" to "*John*", since on the POS level and the usual behavior of "*on*", this sequence is misleading as a typical noun phrase. By introducing attention on feature extraction, we expect the model to attend to important tokens, in this case "*Monday*", which is not likely to attach to a person and suggests the parser to anticipate the following predicate. Our technique can be applied to any models with feed-forward networks on concatenated feature embeddings, and in this work, we apply it on the standard transition-based parser of Chen and Manning (2014).

On the multilingual experiment on Universal Dependencies (UD) 2.0 (Zeman et al., 2017), we find our attention brings performance gain for most languages. To inspect the model's behavior, we also introduce a controlled experiment with manually created data. For this experiment, we prepare a set of sentences for which the parser must attend to the key points for correct disambiguation, as in Figure 1, and see whether the model behaves as expected. There we give detailed error analysis to suggest what makes it difficult to solve the local ambiguities and how attention achieves it. This type of analysis is common in psycholinguistics (Levy, 2008), and a similar idea has recently begun to be explored in NLP neural models (Shekhar et al., 2017).

2 Model

2.1 Base model

Our base model is a transition-based neural parser of Chen and Manning (2014).[1] For each step, this parser first creates feature vectors of words ($\mathbf{x}^w$), POS tags ($\mathbf{x}^p$), and labels ($\mathbf{x}^l$), each of which is a concatenation of embeddings around a stack and a buffer. These vectors are transformed with corresponding weights, i.e., $\mathbf{h} = \mathbf{W}^w\mathbf{x}^w + \mathbf{W}^p\mathbf{x}^p + \mathbf{W}^l\mathbf{x}^l + \mathbf{b}$, followed by nonlinearity. A next softmax layer then provides action probabilities.

Although this method is actually old, the approach which creates the feature vector from independent embeddings becomes useful in our second experiment inspecting our attention behaviors (See section 3.3 in detail). In addition, UDPipe (Straka et al., 2016) which is the baseline parser in the latest shared task (Zeman et al., 2017) also adopts this approach and holds good performance compared to others using recent techniques.

2.2 Attention on local features

We introduce attention in feature computation from the input embeddings to $\mathbf{h}$. Note that three components $\mathbf{W}^w\mathbf{x}^w$, $\mathbf{W}^p\mathbf{w}^p$, and $\mathbf{W}^l\mathbf{x}^l$ are independent; in the following we focus on just one part, abstracted by $\mathbf{W}\mathbf{x}$, and describe how attention is applied for this computation.

Our attention calculates the importance of input elements. First, note that $\mathbf{x}$ is a concatenation of embeddings of input elements, and when the number of elements is n, $\mathbf{W}$ can also be divided into n blocks as in Figure 2. When these parts are denoted by $\mathbf{W}_i$ and $\mathbf{x}_i$, $\mathbf{W}\mathbf{x} = \sum_i \mathbf{W}_i\mathbf{x}_i$ holds. We define $\mathbf{c}_i = \mathbf{W}_i\mathbf{x}_i$, which corresponds to the hidden representation for the i-th input element.

Our core idea is to apply attention on decomposed hidden vectors $\{\mathbf{c}_i\}$. Using attention vector $\mathbf{a} = (a_1, a_2, \cdots, a_n)$, the new hidden representation becomes $\mathbf{h}_g = \sum_i a_i\mathbf{c}_i$. We obtain attention a_i using $\mathbf{c}_i$ and parameters $\mathbf{q}$ as follows:

$$a_i = \frac{\exp(\sigma(\mathbf{q} \cdot \mathbf{c}_i))}{\sum_{i=1}^{n} \exp(\sigma(\mathbf{q} \cdot \mathbf{c}_i))},$$

where σ is a sigmoid function. We use different attention parameters $\mathbf{q}^w$, $\mathbf{q}^p$, and $\mathbf{q}^l$ for word, POS, and label inputs, respectively.

3 Experiments

Our first experiment is on the multilingual UD treebanks used in CoNLL 2017 shared task (Zeman et al., 2017). In addition to this, we present another experiment using augmented test data. This is a set of

[1] As described in Section 3.1 we slightly extend their parser to use additional features. In this section, we first present our model with the original features for simplicity.

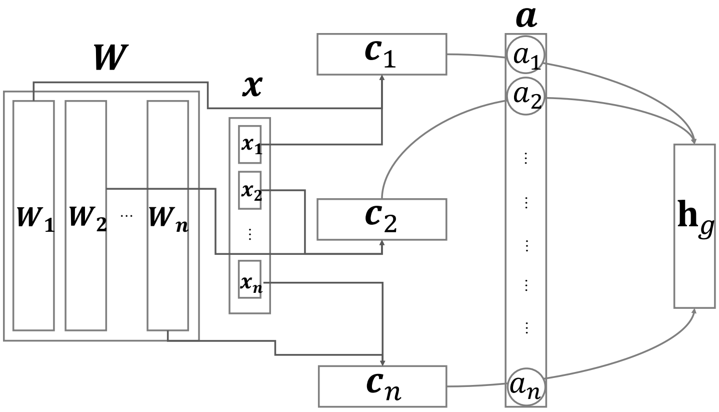

Figure 2: Our attention mechanism on the decomposed hidden vectors $\mathbf{c}_i$, obtained by $\mathbf{W}_i\mathbf{x}_i$.

sentences for which there is a key token for correct disambiguation. We will see whether our model is capable of disambiguating them by attending to the critical points.

For both experiments, our baselines are our parser without attention, and UDPipe v1.1 (Straka et al., 2016), which was the state-of-the-art among transition-based parsers with local features on the shared task.[2]

3.1 Parser

We extract features from the same positions as Chen and Manning (2014); top three tokens from the stack and the buffer, the first and second leftmost or rightmost children of the top two tokens on the stack, and the leftmost or rightmost children of leftmost or rightmost children of the top two tokens on the stack. However, from each position we extract more information such as LEMMA (see also footnote 1). The embedding sizes are: 50 dimensions for WORD, 20 dimensions for LEMMA, UPOS, XPOS, FEATS, and DEPREL. We also extract 32 dimensional character encoding of a token by bi-LSTMs (Ling et al., 2015), though we do not apply attention on this. The size of the hidden dimension is 200, on which we apply 50% dropout. We use pre-trained embeddings used in the baseline UDPipe.[3] To handle non-projectivity, we employ the arc-standard swap algorithm (Nivre et al., 2009). We also use beam search with width 5. To learn the representation for unknown words, we stochastically replace singletons with the dummy token (Dyer et al., 2015). These hyperparameters are the same across languages except Kazakh. This is apart from UDPipe, which tunes the setting for each language. For Kazakh, which is extremely small, we find increasing the model size as 100, 50, and 50 dimensions for WORD, UPOS, and XPOS embeddings works well so we choose this setting.

3.2 Multilingual evaluation

We use 63 treebanks in 45 languages on Universal Dependencies v2.0 (Nivre et al., 2017), with the same data split as the setting of official UDPipe.[3] We evaluate F1 LAS of each treebank and their macro average. For the development sets, we use the gold preprocessed data while for the test sets, we parse the raw text preprocessed by UDPipe.

With respect to the macro averaged score, in the Table 1 below, we can see that our model without attention (w/o Att.) is comparable to UDPipe; with attention, it outperforms both. When inspecting in detail, we see that our attention improves the scores on 54 treebanks on the development set and 57 treebanks on the test set. We also see that the treebanks for which our attention degrades the performance are relatively small, e.g., en_partut (1,035 sentences) and hu (864 sentences), which indicates our attention may be more data-hungry.

[2] There are three systems (Straka and Straková, 2017; Kanerva et al., 2017; Yu et al., 2017) that outperform UDPipe v1.1 but the improvements come not from parsing models but from preprocessing, such as improvements to the POS tagger.

[3] https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-1990

	Development			Test		
Treebank	UDPipe	w/o Att.	w/ Att.	UDPipe	w/o Att.	w/ Att.
ar	78.11	78.73	**79.84**	65.30	64.72	**65.43**
bg	**87.56**	86.90	87.35	**83.64**	83.23	83.44
ca	**88.35**	87.52	88.28	85.39	84.66	**85.43**
cs	**88.19**	86.29	87.06	**82.87**	81.15	82.27
cs_cac	86.57	86.13	**87.01**	**82.46**	81.48	82.15
cs_cltt	78.95	**79.96**	79.22	71.64	72.08	**72.56**
cu	79.44	79.46	**81.69**	62.76	63.19	**65.40**
da	81.13	80.57	**82.01**	73.38	73.16	**74.22**
de	84.06	83.58	**84.25**	**69.11**	67.54	68.66
el	83.71	**84.59**	83.88	79.26	79.56	**80.03**
en	**85.82**	84.96	85.29	**75.84**	74.65	75.06
en_lines	**80.51**	80.40	80.46	72.94	73.75	**74.11**
en_partut	81.29	**81.49**	79.95	**73.64**	73.37	73.15
es	**86.69**	86.17	86.66	81.47	80.55	**81.58**
es_ancora	87.55	86.98	**87.89**	83.78	83.59	**84.39**
et	**76.37**	74.00	75.63	**58.79**	57.62	58.74
eu	76.88	76.31	**77.93**	69.15	68.24	**70.29**
fa	**85.16**	82.69	83.60	**79.24**	77.20	78.28
fi	82.12	81.83	**83.10**	73.75	73.73	**74.73**
fi_ftb	85.14	84.70	**86.20**	74.03	73.45	**74.54**
fr	**89.02**	87.94	88.82	**80.75**	79.87	80.70
fr_partut	80.61	78.81	**82.42**	77.38	77.62	**78.08**
fr_sequoia	**86.66**	**86.66**	86.60	79.98	80.00	**80.29**
ga	71.09	70.49	**72.75**	61.52	62.37	**62.62**
gl	80.55	81.16	**82.22**	77.31	77.82	**78.71**
gl_treegal	74.48	**75.46**	75.13	**65.82**	65.06	65.30
got	76.51	77.32	**77.86**	59.81	60.16	**60.80**
grc	61.65	62.80	**65.51**	**56.04**	54.83	55.66
grc_proiel	75.72	74.58	**76.78**	65.22	64.80	**66.79**
he	**83.18**	81.94	82.87	**57.23**	55.13	55.07
hi	91.07	91.72	**92.15**	**86.77**	86.02	86.46
hr	80.76	79.46	**81.17**	77.18	76.35	**77.59**
hu	73.98	**75.42**	75.36	**64.30**	64.23	64.01
id	78.43	78.24	**79.15**	74.61	74.41	**75.31**
it	88.44	87.27	**88.89**	**85.28**	84.47	85.20
it_partut[a]	**85.16**	84.20	83.85	-	-	-
ja	**95.48**	95.28	95.23	72.21	72.68	**72.69**
kk	34.83	**37.08**	22.47	24.51	**25.14**	22.77
ko	62.06	79.28	**80.10**	59.09	73.52	**74.38**
la	60.04	61.44	**63.11**	43.77	43.78	**46.51**
la_ittb	77.91	77.01	**79.98**	**76.98**	75.78	76.67
la_proiel	74.36	72.48	**75.06**	57.54	57.11	**58.28**
lv	72.71	72.58	**73.37**	59.95	58.65	**60.13**
nl	82.43	81.85	**83.51**	68.90	68.02	**68.93**
nl_lassysmall	80.34	79.22	**80.61**	78.15	76.15	**78.86**
no_bokmaal	**88.78**	87.54	88.70	**83.27**	81.61	82.71
no_nynorsk	87.99	87.56	**88.04**	**81.56**	80.51	80.94
pl	**87.35**	87.79	86.66	78.78	**78.99**	78.65
pt	89.45	92.10	**92.59**	**82.11**	78.79	78.91
pt_br	89.57	88.97	**89.58**	**85.36**	84.91	85.35
ro	82.25	81.80	**82.59**	79.88	78.93	**80.07**
ru	80.84	81.79	**82.53**	74.03	74.79	**75.36**
ru_syntagrus	**89.63**	88.10	89.38	**86.76**	85.55	86.54
sk	83.83	82.95	**84.13**	72.75	72.14	**73.66**
sl	89.15	89.18	**90.05**	81.15	80.06	**81.18**
sl_sst	67.31	66.81	**68.09**	46.45	46.05	**46.50**
sv	80.40	78.94	**80.91**	**76.73**	76.11	76.32
sv_lines	81.38	81.07	**81.73**	**74.29**	73.62	74.07
tr	60.27	59.45	**61.48**	53.19	54.50	**55.50**
ug	**53.85**	58.65	49.04	34.18	**36.26**	35.62
uk	69.30	70.61	**70.68**	60.76	60.91	**61.13**
ur	81.62	85.47	**85.68**	76.69	76.36	**76.98**
vi	66.22	68.13	**69.27**	37.47	37.85	**38.10**
zh	**79.37**	77.45	78.21	**57.40**	56.18	56.22
avg.	79.52	79.65	**80.18**	70.34 [b]	70.06	**70.79**

[a] The test set of it_partut treebank was excluded in the shared task as well.

[b] The UDPipe's official score is 68.35 because it includes the scores for extra treebanks in the shared task, called *surprise language*.

Table 1: Labeled attachment scores of 63 treebanks in UD v2.0

3.3 Augmented data evaluation

Why does attention help for disambiguation? To inspect this, now we perform a controlled experiment by parsing a set of sentences that for correct disambiguation may require attending to some specific points. We present two different sets on English, which differ in the points where the model should attend.

Oblique vs. noun modifier The first set is related to the difficulty of the left of Figure 3, where as we discussed the parser may be confused and attach "*Monday*" to "*John*" as a noun modifier since the POS sequence of "*John on Monday*" and the usual behavior of "*on*" are typical for a noun phrase. The right of Figure 3 shows the step where the parser must decide the head of "*Monday*"; here the correct action is shift and right-arc leads to the wrong analysis. At this step, though the important token for a typical NP is "*on*", we expect the parser to focus more on "*Monday*", which is likely to attach to a subsequent predicate as oblique.

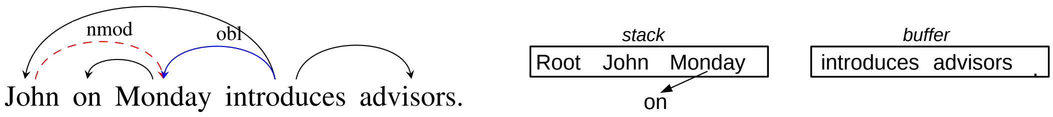

Figure 3: Left - An local ambiguous sentence, reprint of Figure 1. Right - The configuration on which the parser must decide whether "*Monday*" works as an oblique or a modifier.

To inspect the model's ability for correctly handling these ambiguities, we prepare 14 pairs of sentences.[4] Each pair differs minimally, as in "*John on Monday introduces advisors*" and "*John on a balcony introduces advisors*", in which the former should be analyzed as oblique (obl) while the latter as a modifier (nmod). Table 2 contrasts the inputs to parsers when gold preprocessing is given, where the differences always appear at third and forth tokens ("*Monday*" in obl vs. "*a*" and "*balcony*" in nmod). All words in these items occur at least one in the training corpus, therefore no unknown words are used.

	1	2	3	4	5	6	7
obl	John	on	Monday		introduces	advisors	.
	PROPN	ADP	PROPN		VERB	NOUN	PUNCT
	NNP	IN	NNP		VBZ	NNS	.
	Sing	_	Sing		Ind	Plur	_
nmod	John	on	a	balcony	introduces	advisors	.
	PROPN	ADP	DET	NOUN	VERB	NOUN	PUNCT
	NNP	IN	DT	NN	VBZ	NNS	.
	Sing	_	Ind	Sing	Ind	Plur	_

Table 2: Minimal pair in oblique ("*John on Monday introduces advisors*") vs. noun modifier ("*John on a balcony introduces advisors*") experiment

The result is summarized in Table 3, where we count the number of sentences on which the parser outputs are perfect. We can see that nmod sentences are analyzed near perfectly, which is intuitive as this structure is typical. obl sentences are more difficult, but the system with attention is capable of handling them. The other systems fail, even assuming gold tags. For pred tags, all systems receive the same inputs tagged by UDPipe. The accuracy for obl decreases, and we find the errors are due to incorrect POS tags for the predicate at 5th word, which are sometimes tagged as a noun. This suggests our attention parser can handle these local ambiguities unless a crucial tag error occurs, while the other systems cannot at all.

Finally, we show in Figure 4 the attention weights on features at a branching step (The right of Figure 3) for the sample sentences in Table 2. We can see that for the obl sentence the parser attends more

[4] All items are shown in Appendix A.

	Gold tags		Pred tags	
	obl	nmod	obl	nmod
UDPipe	0 / 14	14 / 14	0 / 14	13 / 14
w/o Att.	0 / 14	14 / 14	0 / 14	13 / 14
w/ Att.	**12** / 14	14 / 14	**6** / 14	13 / 14

Table 3: # of correct analysis for obl vs. nmod pairs.

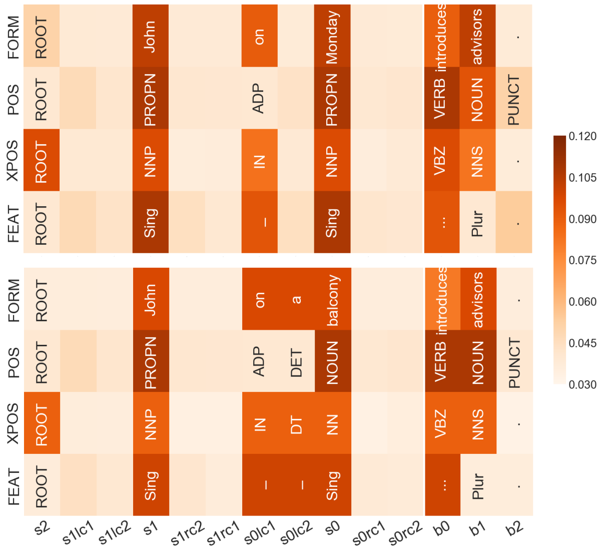

Figure 4: Attention weights on obl sentence (above) and nmod sentence (below). $\mathsf{s_i}$ and $\mathsf{b_i}$ are the i-th top-most position on the stack and buffer, respectively. $\mathsf{lc_i}$ and $\mathsf{rc_i}$ are their (inward) i-th left and right child.

on the key tokens of "*Monday*" on the stack and "*introduces*" on the buffer. This suggests the attention mechanism works as we expected and its behavior matches our intuition.

Object complement vs. that clause The second set is about different ambiguities from the previous experiment; an example pair is shown in the left of Figure 5, where to correctly parse the lower one, the parser must recognize the implicit that clause (that), rather than an object-complement (oc). The right of the figure shows the configuration on which the parser must choose the structure, by shift or right-arc. The key token for correct analysis is at the last, which can be accessed as the second token on the buffer.

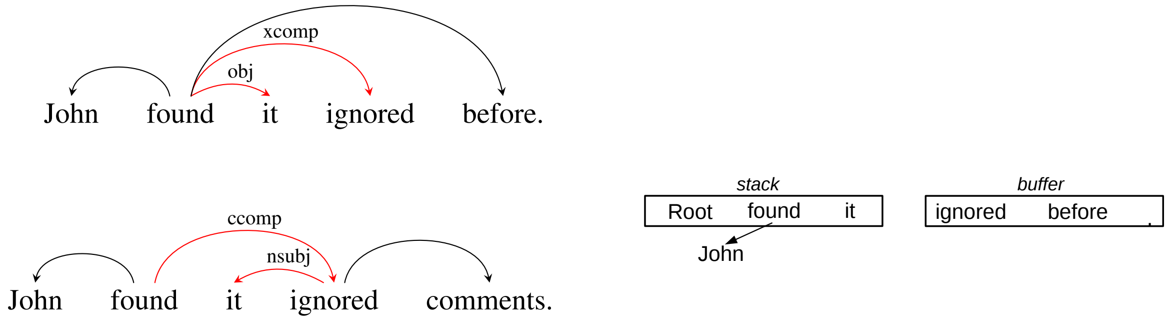

Figure 5: The representative pair for the second set: object-complement (left above) vs. that clause (left below) and the branching configuration (right).

We prepare 24 pairs of sentences. Table 4 shows an example of differences of a pair. In these sentences, tokens from third to fifth differ. Note that contrary to Table 2 these two condition are distinctive with

POS tags (e.g., VBN or VBD), so the main challenge is whether the model can attend to the key tokens when the predicted noisy tags are used.

	1	2	3	4	5	6
oc	John	found	it	ignored	before	.
	PROPN	VERB	PRON	VERB	ADV	PUNCT
	NNP	VBD	PRP	VBN	RB	.
	Sing	Ind	Acc	Past	_	_
that	John	found	it	ignored	comments	.
	PROPN	VERB	PRON	VERB	NOUN	PUNCT
	NNP	VBD	PRP	VBD	NNS	.
	Sing	Ind	Nom	Ind	Plur	_

Table 4: Minimal pair in object-complement (“*John found it ignored before*”) vs. that-clause experiment (“*John found it ignored comments*”)

Table 5 summarizes the results. As we expected, all systems succeed with gold tags, but perform badly in particular on that sentences, with predicted tags. Inspecting errors, we find that this is due to error propagation from an incorrect tag for *it* (3rd token), on which UDPipe assigns Acc(suative) feature due to that-omission. By this error, another error is induced on the POS tag of the next token (e.g., *ignored*), which becomes participle or adjective. These erroneous tags make it hard for parsers to recognize an implicit *that*.

Though all models fail, we notice that for 30% of sentences (7/24), our attention parser recognizes the existence of that-clause, by wrongly analyzing the last noun (e.g., comments) as the head of the clause (*it* becomes nsubj of the noun). Inspecting the attention weights for succeeded and failed cases (Figure 6), we find the last noun is slightly attended more in the succeeded case (above), which may lead the parser to predict a ccomp arc (but to a wrong word).

	Gold tags		Pred tags	
	oc	that	oc	that
UDPipe	23 / 24	24 / 24	19 / 24	0 (0) / 24
w/o Att.	24 / 24	24 / 24	16 / 24	1 (2) / 24
w/ Att.	23 / 24	24 / 24	18 / 24	1 (7) / 24

Table 5: # of correct sentences for oc vs. that. Numbers in brackets mean the cases where that-omission is correctly predicted but other errors exist (see body).

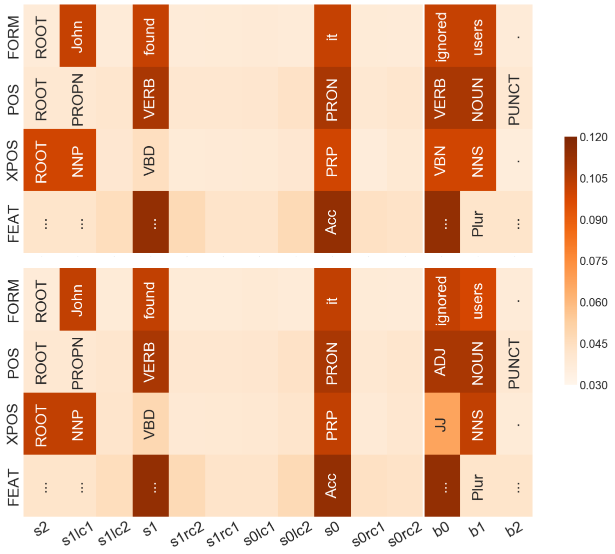

Figure 6: Attention weights on that sentences when that-omission is predicted (above) or failed (below).

4 Conclusion

We have presented an simple attention mechanism for dynamic feature selection, which can be applied to any feed-forward networks on concatenated feature embeddings. When applying to an incremental parser, the parser performance increased across many languages. Also our augmented-data experiment showed that the parser successfully learns where to focus on each context, and becomes more robust to erroneously tagged sentences.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate.

Timo Baumann. 2013. *Incremental Spoken Dialogue Processing: Architecture and Lower-level Components*. Ph.D. thesis.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, Doha, Qatar, October. Association for Computational Linguistics.

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, July. Association for Computational Linguistics.

Jenna Kanerva, Juhani Luotolahti, and Filip Ginter. 2017. Turkunlp: Delexicalized pre-training of word embeddings for dependency parsing. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 119–125, Vancouver, Canada, August. Association for Computational Linguistics.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional lstm feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):11261177.

Wang Ling, Chris Dyer, Alan W Black, Isabel Trancoso, Ramon Fermandez, Silvio Amir, Luis Marujo, and Tiago Luis. 2015. Finding function in form: Compositional character models for open vocabulary word representation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1520–1530, Lisbon, Portugal, September. Association for Computational Linguistics.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1412–1421, Lisbon, Portugal, September. Association for Computational Linguistics.

Joakim Nivre, Marco Kuhlmann, and Johan Hall. 2009. An improved oracle for dependency parsing with online reordering. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT)*, pages 73–76.

Joakim Nivre, Željko Agić, Lars Ahrenberg, Maria Jesus Aranzabe, Masayuki Asahara, Aitziber Atutxa, Miguel Ballesteros, John Bauer, Kepa Bengoetxea, Riyaz Ahmad Bhat, Eckhard Bick, Cristina Bosco, Gosse Bouma, Sam Bowman, Marie Candito, Gülşen Cebiroğlu Eryiğit, Giuseppe G. A. Celano, Fabricio Chalub, Jinho Choi, Çağrı Çöltekin, Miriam Connor, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Arantza Diaz de Ilarraza, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Marhaba Eli, Tomaž Erjavec, Richárd Farkas, Jennifer Foster, Cláudia Freitas, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta Gonzáles Saavedra, Matias Grioni, Normunds Grūzītis, Bruno Guillaume, Nizar Habash, Jan Hajič, Linh Hà Mỹ, Dag Haug, Barbora Hladká, Petter Hohle, Radu Ion, Elena Irimia, Anders Johannsen, Fredrik Jørgensen, Hüner Kaşıkara, Hiroshi Kanayama, Jenna Kanerva, Natalia Kotsyba, Simon Krek, Veronika Laippala, Phng Lê H`ông, Alessandro Lenci, Nikola Ljubešić, Olga Lyashevskaya, Teresa Lynn, Aibek Makazhanov, Christopher Manning, Cătălina Mărănduc, David Mareček, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Anna Missilä, Verginica Mititelu, Yusuke Miyao, Simonetta Montemagni, Amir More, Shunsuke Mori, Bohdan Moskalevskyi, Kadri Muischnek, Nina Mustafina, Kaili Müürisep, Lng Nguy˜ên Thị, Huy`ên Nguy˜ên Thị Minh, Vitaly Nikolaev, Hanna Nurmi, Stina Ojala, Petya Osenova, Lilja

Øvrelid, Elena Pascual, Marco Passarotti, Cenel-Augusto Perez, Guy Perrier, Slav Petrov, Jussi Piitulainen, Barbara Plank, Martin Popel, Lauma Pretkalniņa, Prokopis Prokopidis, Tiina Puolakainen, Sampo Pyysalo, Alexandre Rademaker, Loganathan Ramasamy, Livy Real, Laura Rituma, Rudolf Rosa, Shadi Saleh, Manuela Sanguinetti, Baiba Saulīte, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Lena Shakurova, Mo Shen, Dmitry Sichinava, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Alane Suhr, Umut Sulubacak, Zsolt Szántó, Dima Taji, Takaaki Tanaka, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Larraitz Uria, Gertjan van Noord, Viktor Varga, Veronika Vincze, Jonathan North Washington, Zdeněk Žabokrtský, Amir Zeldes, Daniel Zeman, and Hanzhi Zhu. 2017. Universal dependencies 2.0. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Joakim Nivre. 2008. Algorithms for deterministic incremental dependency parsing. *Computational Linguistics*, 34(4):513–554.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal, September. Association for Computational Linguistics.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. 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 1577–1586, Beijing, China, July. Association for Computational Linguistics.

Ravi Shekhar, Sandro Pezzelle, Yauhen Klimovich, Aurélie Herbelot, Moin Nabi, Enver Sangineto, and Raffaella Bernardi. 2017. Foil it! find one mismatch between image and language caption. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 255–265, Vancouver, Canada, July. Association for Computational Linguistics.

Tianze Shi, Liang Huang, and Lillian Lee. 2017. Fast(er) exact decoding and global training for transition-based dependency parsing via a minimal feature set. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 12–23, Copenhagen, Denmark, September. Association for Computational Linguistics.

Milan Straka and Jana Straková. 2017. Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada, August. Association for Computational Linguistics.

Milan Straka, Jan Hajic, and Jana Strakov. 2016. Udpipe: Trainable pipeline for processing conll-u files performing tokenization, morphological analysis, pos tagging and parsing. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may. European Language Resources Association (ELRA).

Oriol Vinyals, Ł ukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2773–2781. Curran Associates, Inc.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention.

Kuan Yu, Pavel Sofroniev, Erik Schill, and Erhard Hinrichs. 2017. The parse is darc and full of errors: Universal dependency parsing with transition-based and graph-based algorithms. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 126–133, Vancouver, Canada, August. Association for Computational Linguistics.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajic, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinkova, Jan Hajic jr., Jaroslava Hlavacova, Václava Kettnerová, Zdenka Uresova, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria dePaiva, Kira Droganova, Héctor Martínez Alonso, Çağrı Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung

Kwak, Gustavo Mendonca, Tatiana Lando, Rattima Nitisaroj, and Josie Li. 2017. Conll 2017 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19. Association for Computational Linguistics.

A Experiment Items

No.	Type	Sentence
1	obl	John on Monday agrees friends.
	nmod	John on a table agrees friends.
2	obl	John on Monday needs colleagues.
	nmod	John on a chair needs colleagues.
3	obl	John on Tuesday introduces advisors.
	nmod	John on a balcony introduces advisors.
4	obl	John on Tuesday calls leaders.
	nmod	John on a bike calls leaders.
5	obl	John on Wednesday employs people.
	nmod	John on a ship employs people.
6	obl	John on Wednesday meets relatives.
	nmod	John on a bus meets relatives.
7	obl	John on Thursday sees workers.
	nmod	John on a stage sees workers.
8	obl	John on Thursday praises owners.
	nmod	John on a roof praises owners.
9	obl	John on Friday believes teachers.
	nmod	John on a seat believes teachers.
10	obl	John on Friday hits professors.
	nmod	John on a car hits professors.
11	obl	John on Saturday protects soldiers.
	nmod	John on a tank protects soldiers.
12	obl	John on Saturday supports doctors.
	nmod	John on a hill supports doctors.
13	obl	John on Sunday worries visitors.
	nmod	John on a mountain worries visitors.
14	obl	John on Sunday contacts managers.
	nmod	John on a plane contacts managers.

Table 6: Items in the oblique vs. noun modifier experiment.

No.	Type	Sentence
1	oc	John found it ignored before.
	that	John found it ignored comments.
2	oc	John found it ignored again.
	that	John found it ignored opinions.
3	oc	John found it contained before.
	that	John found it contained layers.
4	oc	John found it contained again.
	that	John found it contained plants.
5	oc	John considered it classified before.
	that	John considered it classified species.
6	oc	John considered it classified again.
	that	John considered it classified words.
7	oc	John considered it involved before.
	that	John considered it involved issues.
8	oc	John considered it involved again.
	that	John considered it involved changes.
9	oc	John felt it abandoned before.
	that	John felt it abandoned soldiers.
10	oc	John felt it abandoned again.
	that	John felt it abandoned people.
11	oc	John felt it protected before.
	that	John felt it protected ideas.
12	oc	John felt it protected again.
	that	John felt it protected students.
13	oc	John guessed it reccommended before.
	that	John guessed it reccommended graphics.
14	oc	John guessed it reccommended again.
	that	John guessed it reccommended services.
15	oc	John guessed it employed before.
	that	John guessed it employed officials.
16	oc	John guessed it employed again.
	that	John guessed it employed relatives.
17	oc	John understood it infected before.
	that	John understood it infected computers.
18	oc	John understood it infected again.
	that	John understood it infected animals.
19	oc	John understood it pasted before.
	that	John understood it pasted pictures.
20	oc	John understood it pasted again.
	that	John understood it pasted posters.
21	oc	John believed it transmitted before.
	that	John believed it transmitted signals.
22	oc	John believed it transmitted again.
	that	John believed it transmitted images.
23	oc	John believed it replaced before.
	that	John believed it replaced lights.
24	oc	John believed it replaced again.
	that	John believed it replaced positions.

Table 7: Items in the object complement vs. that clause experiment.