Eyes on the Parse: Using Gaze Features in Syntactic Parsing

Abhishek Agrawal **Rudolf Rosa**
Charles University, Prague
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
abhi1994.aa@gmail.com, rosa@ufal.mff.cuni.cz

Abstract

In this paper, we explore the potential benefits of leveraging eye-tracking information for dependency parsing on the English part of the Dundee corpus. To achieve this, we cast dependency parsing as a sequence labelling task and then augment the neural model for sequence labelling with eye-tracking features. We also augment a graph-based parser with eye-tracking features and parse the Dundee Corpus to corroborate our findings from the sequence labelling parser. We then experiment with a variety of parser setups ranging from parsing with all features to a delexicalized parser. Our experiments show that for a parser with all features, although the improvements are positive for the LAS score they are not significant whereas our delexicalized parser significantly outperforms the baseline we established. We also analyze the contribution of various eye-tracking features towards the different parser setups and find that eye-tracking features contain information which is complementary in nature, thus implying that augmenting the parser with various gaze features grouped together provides better performance than any individual gaze feature.

1 Introduction

When it comes to natural language processing (NLP), most of the research work that is done is based on textual data that is usually annotated in some way while human language processing data like eye movement recordings, etc. are often overlooked. The solutions for most tasks in NLP these days are based on neural networks which require annotated data on a large scale to learn good representations and provide meaningful results. Annotating such large quantities of data is not an easy task usually requiring a lot of manual labour which is time consuming and can be expensive not to mention resolving the inter-annotator agreement. However, every single day, millions of people read their daily newspapers, books, magazines, articles on the internet, etc. in a multitude of languages. In the past couple of years some studies have shown that using behavioral data for NLP tasks involving syntax and semantics can be useful (Mishra et al., 2016a; Mishra et al., 2016b; Mishra et al., 2016c; Barrett and Søgaard, 2015a; Barrett and Søgaard, 2015b). This leads us to believe that if there was a way to tap into cognitive data generated by unconscious human parsing of text, it could be leveraged in some manner to support NLP tools thereby reducing our dependency on annotated textual data.

Another reason for choosing to work with eye-tracking features is that it is highly likely that eye-tracking technology will be available on a much larger scale in the near future and hence can be leveraged easily for NLP tasks. This is evidenced by the availability of eye-tracking through regular webcams (San Agustin et al., 2009) and smartphones (Krafka et al., 2016). San Agustin et al. (2010) developed the ITU Gaze Tracker which was a low-cost eye-tracking system based on a webcam that is mounted close to the user's eye. The developers of this system soon established their company *"The Eye Tribe"*[1] with a focus on providing low cost eye-tracking and providing eye control technology for mass

[1] https://theeyetribe.com

market consumer devices. Although eye-tracking recording might seem an even more expensive solution to manually annotating data, several studies and even efforts made by the industry indicate that costs associated with eye-tracking hardware and software are just going to go down and that next generation mass consumer goods like smartphones, tablets, laptops and gaming consoles will come equipped with eye-tracking capabilities.

Dependency parsing is a fairly well studied phenomenon because of its relevance in several downstream NLP applications and because of the overwhelming success of the *Universal Dependencies* (UD) project (Nivre et al., 2016) which is an endeavor to provide guidelines for consistent dependency annotations across multiple languages. This study primarily presents an investigation in leveraging eye-tracking features for dependency parsing and the knowledge they bring in identifying syntactic categories and relations. There have been only two previous studies, to the best of our knowledge, which try to leverage gaze features[2] for dependency parsing (Strzyz et al., 2019b; Barrett and Søgaard, 2015b). Strzyz et al. (2019b) cast dependency parsing as sequence labelling in a multi-task learning (MTL) setup where the gaze features are predicted as an auxiliary task while considering that the eye-tracking data will be available only during training time. For our study, we try an alternative approach to Strzyz et al. (2019b) wherein we focus on determining the effects of incorporating eye-tracking data directly in the model for dependency parsing and assume that eye-tracking data would also be available to us at inference time as opposed to the approach followed by Strzyz et al. (2019b). To provide further evidence, we also employ a graph-based parser and augment it with gaze features to see if they improve parsing. We try to address the following research questions through our study:

1. Does eye-tracking data from a reader reading a sentence contain syntactically relevant information?
2. Can the accuracy of syntactic parsing be improved by adding eye-tracking features?

2 Background and Related Work

Eye-tracking information has been used over several years to try to understand cognitive processes. Eye movement recordings of a person taken when they are reading a text have been known to reflect a person's comprehension of the text (Gaskell et al., 2012). Intuitively, one can understand how information about the piece of text that is being focused on and how long it takes to read a piece of text would help in understanding how text is comprehended while being read.

2.1 Factors affecting eye-movements while reading

Gaskell et al. (2012) review several works and provide a good overview of the various factors which influence the eye-movements while reading. The two main factors influencing eye-movements are the word length and word frequency with shorter and more frequent words being skipped more often and longer fixation durations for less frequent and longer words (Brysbaert and Vitu, 1998; Rayner and McConkie, 1976; Liversedge et al., 2004; Rayner et al., 2006; Rayner et al., 2003). Other factors affecting eye-movements are familiarity with the word and the age of acquisition of the word. The degree of familiarity of a word to its reader inversely corresponds to the duration of the first fixation by the reader (Chaffin et al., 2001; Juhasz and Rayner, 2003; Williams and Morris, 2004) and words that are acquired early in life by a reader are processed much faster than words they acquire in the later stages of their life (Juhasz and Rayner, 2003; Juhasz, 2005; Juhasz and Rayner, 2006). The ambiguous meanings of a word also affect the eye-movement patterns with the fixation duration depending upon the frequency of the ambiguous meanings as well as the dominant/sub-ordinate meaning of the word (Rayner and Duffy, 1986; Duffy et al., 1988; Rayner and Frazier, 1989; Sereno et al., 2006).

Studies also found that whenever the predictability of a word based on the past context increases, its reading time decreases and some words that are highly predictable from the past context are altogether skipped (Ehrlich and Rayner, 1981; Rayner and Well, 1996). Morris (1994) found that the fixation duration on a word decreases if it is semantically associated with any prior word. Frazier and Rayner (1982) and Meseguer et al. (2002) found that a reader is innately aware of the point in text where their

[2]Gaze features are the same as eye-tracking features and the two terms are used interchangeably throughout this study.

incorrect syntactic analysis of a sentence differs from the correct analysis as the reader's eyes move back to this point after a regression.

2.2 Overview of eye-tracking in NLP

In this sub-section we describe some of the studies in the field of NLP which make use of eye-tracking data. Barrett et al. (2016a) use eye-tracking data to improve a POS tagger and found significant improvements in the results. A very interesting study by Barrett et al. (2016b) found that the correlation between gaze features and POS tags can be transferred across languages and they use English gaze features to improve a French POS tagger. Barrett and Søgaard (2015a) found that eye-tracking information can be used to predict the syntactic categories of words and Barrett and Søgaard (2015b) used gaze features to predict the grammatical function of a word. The study done by Barrett et al. (2018) leveraged eye-tracking information to find out human attention and used it to regularize the attention function used in an RNN and found improvements for a range of NLP tasks like sentiment analysis, abusive language detection, etc.

Bingel et al. (2018) use eye-tracking information to predict reading errors of children that have some reading disability. Metrics for evaluating the quality of machine translation output derived from eye-tracking data were found to be better than the automatic metrics in use (Klerke et al., 2015). The idea that eye-tracking data could be used to evaluate machine translation output seems quite reasonable as bad translations would result in longer fixation durations and more regressions by the reader which can be picked up from the data. Klerke et al. (2016) in their work use gaze data in a multi-task learning setup to improve sentence compression. Søgaard (2016) and Hollenstein et al. (2019) couple fMRI data along with eye-tracking data to evaluate the quality of word embeddings. Hollenstein and Zhang (2019) leveraged eye-tracking data for improving named entity recognition. An important feature of their study was to leverage type-aggregated gaze features to eliminate the need for recording eye-tracking data at test time and also make the features useful for cross-domain settings.

A couple of studies have also experimented with some sort of parsing of text using eye-tracking data. Strzyz et al. (2019b) leverage gaze data by learning eye-movement features as an auxiliary task in a multi-task learning setup where both dependency parsing and gaze prediction are addressed as sequence labelling. Their experiments resulted in small positive improvements to dependency parsing, however they did not measure the statistical significance of their results. Lopopolo et al. (2019) go about dependency parsing the other way around by showing that there is a relation between regressions and the syntactic structure of sentences. They tested if the path of regressions from a word to an earlier word coincide – at least partially – with the edges of dependency relations between these words by using dependency parsing features to predict eye-regressions during training. One of their important findings indicates that eye regressions are involved predominantly in dependency parsing at the local level (vast majority being shorter than three words with a predominance of one position backwards), rather than at long distance. Cheri et al. (2016) utilize the eye-movements of several annotators resolving coreference to improve automatic coreference resolution.

Mathias et al. (2018) rate the quality of a piece of text by using eye-tracking data and Mishra et al. (2017) try to quantify the effort needed in reading a piece of text by measuring the complexity of the scanpath of various readers. A few studies also found that eye-tracking data can be used to improve sentiment analysis as well as sarcasm detection (Mishra et al., 2016a; Mishra et al., 2016b; Mishra et al., 2016c).

3 Parser Descriptions

In this section, we describe the two parsers we use in our work. Our primary parser is a modification of the system created by Strzyz et al. (2019b) who in turn adapted the *NCRF++* system (Yang and Zhang, 2018) which is an open source neural sequence labelling toolkit. Our secondary parser is a more traditional graph-based parser known as the *BIST* parser from Kiperwasser and Goldberg (2016). One of the main differences between our parsers and the parser by Strzyz et al. (2019b) is that we assume the availability of gaze features both at training and test time whereas Strzyz et al. (2019b) only use the gaze

features during training. Another difference is that we incorporate gaze features as the input to the parser whereas Strzyz et al. (2019b) predict the gaze features as an auxiliary task in a multi-task learning setup.

The parsers are evaluated with respect to the *Labelled Attachment Score* (LAS) and the *Unlabelled Attachment Score* (UAS). The LAS is concerned with the number of words that are assigned a correct head as well as the correct dependency relation whereas the UAS is just concerned with the words that are assigned a correct head.

3.1 Dependency parsing as sequence labelling

The first step while casting dependency parsing as a sequence labelling problem is to convert the dependency tree representation into a set of labels. Our data is in the form of the CoNLL-X format with additional columns containing the gaze features. To encode a dependency tree as labels for sequence labelling, Strzyz et al. (2019c) found that it is sufficient to just encode a word's head and dependency relation associated with it for all the words in a sentence. So following the strategy by Strzyz et al. (2019c) for relative positional encoding, we provide each word with a label that contains its dependency relation label as well as the relative position of its head. The relative position of the head is an encoding in the form of a tuple (p_i, o_i) of a POS tag p_i and a positive or negative number o_i. If the number is positive then it means the head of the word is the o_ith closest word to its right having the POS tag p_i and if the number is negative then the head of the word is the o_ith closest word to its left having the POS tag p_i. For example, (N, 1) would mean "the first noun on the right" of the word is its head. Figure 1 shows an example of an encoded dependency tree.

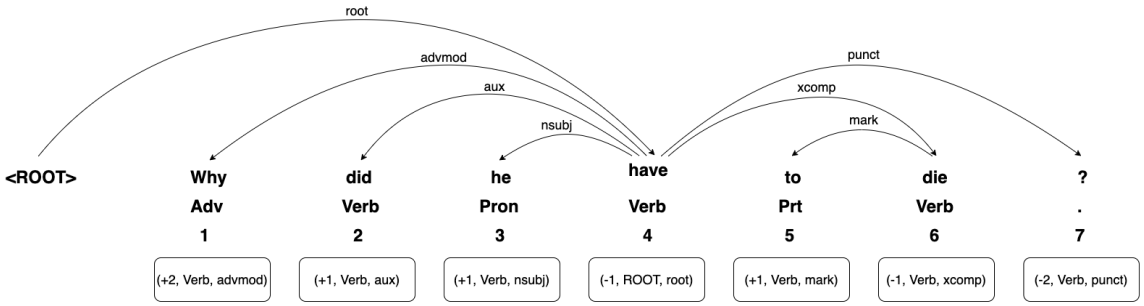

Figure 1: Example of an encoded dependency tree

The generated labels are also conditioned on the chosen multi-task learning setup. In our case, we make use of hard parameter sharing for the multi-task learning setup. As described by Strzyz et al. (2019a), for the combined multi-task learning setup the label will be treated as a single task where its components are separated by the symbol "@". Each component in the label separated by "{}" will be treated as a separate task. The information about the relative position and word's head (e.g. +1@Verb) is combined into one task and the dependency relation is considered as the second task. As per Strzyz et al. (2019a) this gives us the best performance. An example sentence with this setup is as below.

Why	Adv	+2@Verb{}advmod
did	Verb	+1@Verb{}aux
he	Pron	+1@Verb{}nsubj
have	Verb	-1@ROOT{}root
to	Prt	+1@Verb{}mark
die	Verb	-1@Verb{}xcomp
?	.	-2@Verb{}punct

The architecture of the parser is the same as the one employed by Strzyz et al. (2019b) and makes use of bi-directional long short term memory (bi-LSTM) networks (Hochreiter and Schmidhuber, 1997; Schuster and Paliwal, 1997) for the sequence labelling model. For the inputs, we use word, character and POS tag embeddings. The gaze features are also a part of the input, however since they are already real-valued, they are not 'embedded'. One of the changes we make to the parser by Strzyz et al. (2019b)

is to concatenate the word,character and POS embeddings with the corresponding gaze features of the word and pass it through a bi-LSTM, which generates vectors that take context into account. For the combined multi-task learning setup, the total loss is computed as the sum of the individual cross-entropy losses:

$$\mathcal{L} = \mathcal{L}_{(o,p)} + \mathcal{L}_d$$

For all our experiments, we evaluate the sequence labelling parser with respect to the UAS and LAS scores excluding punctuations. We exclude the punctuations following the previous work done in this area (Strzyz et al., 2019b). The source code for our sequence labelling parser can be found online[3].

3.2 Graph-based BIST parser

To provide a better perspective about incorporating gaze features with dependency parsing, we also try out a more standard or traditional parser which is commonly known as the *BIST* parser from Kiperwasser and Goldberg (2016). Kiperwasser and Goldberg (2016) employ their approach for both a transition-based parser and a graph-based parser, however we only make use of the graph-based parser in our work. The model that they use for the graph-based parser is the simple first-order arc-factored model (Mcdonald, 2006).

So according to Kiperwasser and Goldberg (2016), if for a sentence s comprising of n words $w_1, \ldots, w_n$ where we also have the corresponding POS tag p_i for each word w_i, we get the embeddings for the words and POS tags denoted by $e(w_i)$ and $e(p_i)$ and then create a sequence of input vectors $x_{1:n}$ where each vector x_i is a concatenation of the word embedding and POS tag embedding and $\circ$ is the concatenation operator:

$$x_i = e(w_i) \circ e(p_i)$$

For our work, we only make one addition to the parser. We use seventeen different gaze features denoted by $g_1, \ldots, g_{17}$ and we concatenate each of these gaze features directly to the word and POS tag embeddings. So our input vector would look like:

$$x_i = e(w_i) \circ e(p_i) \circ g_1 \circ \ldots \circ g_{17}$$

For all our experiments, we evaluate the BIST parser with respect to the UAS and LAS scores. We also include punctuations in the evaluation this time since for our previous parser we excluded them and we hope to demonstrate the overall effect of gaze features on dependency parsing by covering all possible cases. The source code for the graph-based BIST parser can be found online[4].

4 Using gaze features for dependency parsing

In this section, we describe the dataset that we use along with a description of the gaze features that we intend to make use of. We then describe a set of experiments with these gaze features and report the unlabelled and labelled attachment scores (UAS and LAS) on the development and test set and analyze the result of all the scenarios.

4.1 Data

For all our experiments, we make use of the Dundee Corpus (Kennedy et al., 2003). The reason we chose this corpus was because it is the only available eye-tracking corpus having a UD style syntactic annotation layer on top of the English side of the corpus as described by Barrett et al. (2015) in their work on the Dundee Treebank. We only use the English part of the Dundee Corpus for our experiments. We choose 17 different gaze features spread out across four distinct groups (*basic*, *early*, *late* and *contextual* features) similar to Barrett et al. (2016a) and Hollenstein and Zhang (2019).

For all the experiments in this section, we use a 80-10-10 train-dev-test split with a test set of 241 sentences, dev set of 230 sentences and training set of the remaining 1,897 sentences following Barrett et al. (2016a).

[3] https://github.com/balthamel/dep2label-up

[4] https://github.com/balthamel/bist-parser

Basic	
n fixations	total number of fixations on a word w
fixation probability	the probability that a word w will be fixated
mean fixation duration	mean of all fixation durations for a word w
total fixation duration	sum of all fixation durations for a word w
Early	
first fixation duration	duration of the first fixation on a word w
first pass duration	sum of all fixation durations when it is first visited
Late	
n re-fixations	number of times a word w is fixated (after the first fixation)
re-read probability	the probability of revisiting word w after making a first pass
Context	
total regression-from duration	combined duration of the regressions that began at word w
w-2 fixation probability	fixation probability of the word before the previous word
w-1 fixation probability	fixation probability of the previous word
w+1 fixation probability	fixation probability of the next word
w+2 fixation probability	fixation probability of the word after the next word
w-2 fixation duration	fixation duration of the word before the previous word
w-1 fixation duration	fixation duration of the previous word
w+1 fixation duration	fixation duration of the next word
w+2 fixation duration	fixation duration of the word after the next word

Table 1: Gaze features spread over four groups as adapted from Hollenstein and Zhang (2019).

Table 1 provides a description of all the gaze features used in our experiments. In our experiments, for the gaze features, we normalize all the values across the data. For normalizing the values, we use `scikit-learn`'s normalization from its pre-processing library with all the three norms i.e. '`l1`', '`l2`' and '`max`' norm. The choice of the norm seems to have a considerable effect on the results as is shown in the next couple of sections.

Using just the raw gaze feature values doesn't work well as we found out with one of our initial experiments. It degraded the performance of the parser severely compared to the baseline. Since the ranges for the different gaze features are considerably different, normalizing them makes a lot of sense.

We have two variations in terms of the gaze features that we use. In the first case, we use all seventeen normalized gaze features and these features were picked keeping in mind the previous works in this area (Hollenstein and Zhang, 2019; Strzyz et al., 2019b).

In the second case, we consider a mixture of normalized and raw gaze feature values. We consider raw values only for those features which deal with the number of fixations or probabilities namely: *n fixations*, *fixation probability*, *n re-fixations*, *re-read probability*, *w-2 fixation probability*, *w-1 fixation probability*, *w+2 fixation probability* and *w-2 fixation probability*. For the other features listed in Table 1, we consider their normalized values.

4.2 Experiments

We run our experiments with data that is averaged over all ten readers of the English part of the Dundee corpus. We evaluate several models on the development set varying two hyper-parameters – subset of gaze features to use, and the normalization applied to their values – and identify the best setup for each experiment based on UAS and LAS performance on development data and then run those setups on the test set to verify whether the improvements are consistent. In particular, we carry out three experiments:

- Experiment 1: The first thing we do is to run both parsers with all the available features at hand

	Gaze Features	Sequence Labelling Parser				Graph-Based Parser			
		lexicalized dev set		delexicalized dev set		lexicalized dev set		delexicalized dev set	
		UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
	baseline	83.18	76.45	72.83	64.36	83.41	77.57	73.83	66.49
L1 norm	17 feats. normalized	**84.42**	**77.38**	73.10	64.88	83.52	77.68	73.41	66.90
	17 feats. normalized + raw	83.28	76.78	73.77	65.55	**84.04**	**78.68**	**75.10**	**69.28**
L2 norm	17 feats. normalized	84.30	**77.38**	73.15	64.90	82.62	77.24	73.06	66.81
	17 feats. normalized + raw	83.45	76.82	73.91	**65.73**	83.54	78.16	74.22	67.99
max norm	17 feats. normalized	83.39	76.66	73.94	65.19	83.38	77.66	74.68	68.34
	17 feats. normalized + raw	83.76	77.15	**74.02**	65.71	83.23	77.85	74.22	68.50

Table 2: UAS and LAS scores for lexicalized and delexicalized parsers.

Parser setup	Sequence Labelling Parser				Graph-Based Parser			
	lexicalized test set		delexicalized test set		lexicalized test set		delexicalized test set	
	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
baseline	82.19	75.82	72.86	64.89	**82.29**	76.16	74.14	66.80
best setup	**82.46**	**75.93**	****74.11**	***65.84**	81.90	**76.31**	**74.89**	****68.69**

Table 3: Evaluation of best parser setups on test set with UAS and LAS scores (best results in bold; *, ** indicates significant improvement over the baseline; * $p < 0.05$, ** $p < 0.01$ McNemar's test).

i.e. word features, character features (in case of sequence labelling parser), POS tags and the gaze features.[5] The baseline system is a model where no gaze features were used.

- Experiment 2: The next setup that we try out is delexicalized parsing of the data. Delexicalized parsing has been found quite useful in low resource and cross-lingual settings (Zeman and Resnik, 2008; Aufrant et al., 2016). A part of the motivation behind choosing this setup was also that in case of successful delexicalized parsing, it would be interesting to explore if gaze and syntax co-relations can be transferred across a pair of languages, helping to create a cross-lingual dependency parser. That being said, our interest in this setup is merely to explore if eye-tracking features are useful in any form of parsing. In this case we omit the word and character level features and only use the POS tag embeddings and the gaze features. The baseline system is a model where no gaze features were used.

- Experiment 3: Based on the best setups from the first two experiments, to estimate how much individual gaze features and grouped gaze features as per their category (i.e. basic, early, late and context) contribute towards lexicalized and delexicalized parsing, we compare scores from the baseline model where no gaze features are used to models with individual gaze features or grouped gaze features only for the sequence labelling parser.

5 Results

Table 2 shows the results for Experiment 1 and 2. For the lexicalized sequence labelling parser, the best results are obtained by using all 17 features with the L1 norm. There is an improvement of +1.24 for UAS and +0.93 for LAS score as compared to the baseline scores. For the delexicalized sequence labelling

[5]From here on out we call this the 'lexical' parser.

	Gaze features	lexicalized dev set		delexicalized dev set		(Strzyz et al., 2019b) dev set	
		UAS	LAS	UAS	LAS	UAS	LAS
	baseline	83.18	76.45	72.83	64.36	85.36	79.40
Basic	*n* fixations	+0.60	+0.31	+0.34	+0.23	-0.04	-0.11
	fixation probability	+0.87	+0.46	+0.09	-0.02	-0.04	+0.17
	mean fixation duration	+0.62	+0.19	-0.04	-0.04	-0.15	-0.02
	total fixation duration	+0.68	+0.08	-0.06	-0.14	-0.02	-0.05
	basic features	+0.91	+0.50	**+0.71**	**+0.42**	0.00	+0.17
Early	first fixation duration	+0.97	**+0.73**	-0.12	-0.37	-0.06	+0.06
	first pass duration	+0.83	+0.56	-0.02	+0.04	+0.14	+0.09
	early features	+0.77	+0.62	+0.25	0.00	**+0.25**	+0.17
Late	*n* re-fixations	+0.60	+0.19	+0.32	+0.15	+0.16	-0.15
	re-read probability	+0.25	+0.35	+0.07	+0.08	-0.02	+0.17
	late features	+0.85	+0.70	+0.11	-0.08	+0.18	+0.24
Context	total regression from duration	+0.27	+0.23	+0.19	-0.14	—	—
	w-2 fixation probability	+0.50	+0.33	-0.12	-0.39	—	—
	w-1 fixation probability	+0.60	+0.08	+0.13	-0.52	-0.19	+0.07
	w+1 fixation probability	+0.48	-0.06	-0.16	-0.10	0.00	-0.33
	w+2 fixation probability	+0.50	+0.33	-0.12	-0.39	—	—
	w-2 fixation duration	+0.87	+0.14	+0.30	-0.10	—	—
	w-1 fixation duration	+0.39	+0.23	-0.18	-0.29	+0.07	+0.28
	w+1 fixation duration	+0.77	+0.50	+0.03	-0.29	+0.03	+0.13
	w+2 fixation duration	+0.18	+0.14	+0.34	+0.25	—	—
	context features	**+0.99**	**+0.73**	+0.48	+0.04	**+0.25**	**+0.32**

Table 4: Impact of various gaze features on lexicalized (17 feats. L1 norm) and delexicalized sequence labelling parser (17 feats. max norm + raw). The values in the last two columns are taken directly from Strzyz et al. (2019b) and is an instance of lexicalized parsing. The values reflect improvement or deterioration over the corresponding baseline scores.

parser, the best improvement for the LAS score is +1.37 with the mixture of raw and normalized features with the L2 norm and the best improvement for the UAS score is +1.19 with the mixture of raw and normalized features with max norm. Curiously enough, for the graph-based lexicalized parser, although the best results are also obtained via the L1 normalization of the gaze features, it is the mixture of raw and normalized gaze features that results in the highest scores. We see an improvement of +0.63 for UAS and +1.11 for LAS score as compared to the baseline scores. For the graph-based delexicalized parser, the best improvement we get is of +1.27 for the UAS score and +2.79 for the LAS score by using the L1 norm and a mixture of raw and normalized gaze features. Although the highest scores across both delexicalized parsers vary amongst all three norms used, the one commonality that can be observed from the results is that using a mixture of raw and normalized gaze features provides the best results when it comes to delexicalized parsing. We also observe that the improvements for both parser setups vary considerably depending on the normalization and the features used.

Table 4 shows the results for Experiment 3. For the lexical sequence labelling parser, as seen in Table 2, the L1 norm with all 17 features performs the best in terms of both UAS and LAS scores and for delexicalized parsing we consider the setup with mixture of normalized and raw 17 features with max norm as the best since even though it has lesser LAS score compared to the best one (65.73) the

difference is quite marginal. Although the results of our parser cannot be directly compared with the the results of Strzyz et al. (2019b) since we use different training, dev and test splits we still include their results in Table 4. We see that Strzyz et al. (2019b) achieve higher baseline results and our only estimate for this is that they managed to get the CRF layer working properly for them boosting their result. We do, however, note that our improvements with gaze features are higher than that of Strzyz et al. (2019b) and this could be because we directly incorporate the gaze features instead of predicting them as an auxiliary task.

The results from the dev set for the lexical sequence labelling parser show that the grouped *context* gaze features provide the best improvements as compared to the baseline with an improvement of +0.99 for UAS score and +0.73 for the LAS score. Individually, *first fixation duration* provides almost the same improvements as the grouped *context* features with an improvement of +0.97 for UAS score and +0.73 for LAS score. These results are in line with the results of Strzyz et al. (2019b), who found improvements with the *early* and *context* grouped gaze features.

Interestingly, the results from the dev set show that in case of delexicalized parsing, the set of *basic* gaze features, grouped together provide the best improvements as compared to the baseline. They provide an improvement of +0.71 for the UAS score and +0.42 for the LAS score. Individually, the *w+2 fixation duration* and *n fixations* features seem to provide the best improvements over the baseline.

Table 3 shows the results of the baseline parsers and the lexical parsers with the best setups from Table 2 on the test data. For the lexical sequence labelling parser, the best setup uses all 17 gaze features with the L1 norm and for the lexical graph-based parser, using a mixture of all 17 normalized and raw gaze features with L1 norm is the best setup. Table 3 shows that for the lexical sequence labelling parser, although there is an improvement of +0.27 for the UAS score and +0.11 for the LAS score, these improvements are quite small and not statistically significant. We also see that for the lexical graph-based parser, the performance actually deteriorates by -0.39 for the UAS score as compared to the baseline and for the LAS score we see an improvement of +0.15 as compared to the baseline although this improvement is also statistically insignificant. On comparison with their corresponding scores on the development set, these improvements aren't as large and the performance even deteriorates for the graph-based parser in terms of the UAS score leading us to believe that although the gaze features do help marginally in parsing at least for improving the LAS score, the improvement that we saw with the development data was more due to the data and not because of the gaze features.

Table 3 also shows the results of the baseline delexicalized parsers and the delexicalized parsers with the best setups from Table 2 on the test data. For the delexicalized sequence labelling parser, the best setup uses a mixture of all 17 normalized and raw gaze features with max norm and for the delexicalized graph-based parser, using a mixture of all 17 normalized and raw gaze features with L1 norm is the best setup. The results in Table 3 seem to indicate that eye-tracking features most definitely help in delexicalized parsing. There is an improvement of +1.25 over the baseline for the UAS score and an improvement of +0.95 over the baseline for the LAS score, both of which are statistically significant for the delexicalized sequence labelling parser. As for the delexicalized graph-based parser, there is an improvement of +0.75 for the UAS score and +1.89 for the LAS score over the baseline with the improvement for LAS score being statistically significant. The delexicalized sequence labelling parser seems to give better scores for UAS whereas the delexicalized graph-based parser seems to improve the LAS scores more.

A further analysis of the output[6] of the delexicalized sequence labelling parser shows that the dependency relations towards whose identification the gaze features help the most are *root*, *cop* and *aux*. The relative improvements in the prediction of these relations are 9%, 13% and 15% respectively. The POS tags for which gaze features help in correctly identifying the head and dependency relation the most are *NOUN*s, *CONJ* and *VERB*s. The high relative improvements for *root*, *aux*, *cop* and *VERB* makes us conclude that gaze features contain some important syntactic information related to verbs allowing the parser to parse the verbs much better. As verbs are heads of syntactic clauses, it may also be that gaze features help the parser distinguish the main clause (headed by a verb with the 'root' label) from

[6]Detailed results are available in the appendix.

subordinate clauses. The improvement for *CONJ* is also high meaning gaze features somehow help the parser in understanding co-ordination structures.

An alternate hypothesis to explain the improvements in delexicalized parsing could also be that since our data makes use of Google universal POS tags (Petrov et al., 2012), which is more coarse than the UD tagset, the gaze features may simply be distinguishing the POS categories into more fine grained ones like distinguishing between full verbs and auxiliary verbs or co-ordinating and sub-ordinating conjunctions which then in turn helps the parser. It might be the case that were the UD POS tagset had been used in the data, the improvements in the delexicalized parser might have been smaller.

6 Conclusion

In this paper, we explored the benefits of using eye-tracking features for dependency parsing. We performed a set of experiments wherein we tried different parser settings and different eye-tracking feature selection along with various feature normalization techniques to try to answer the question of relevance of gaze features for syntactic parsing.

Our experiments show that although eye-tracking features seem to help in dependency parsing where all features (word level, character level and POS tags) are used for the LAS score, the improvements over the baseline are not statistically significant. However, when we use eye-tracking features for delexicalized parsing, our experimental results show that there is statistically significant improvement over the baseline for both the UAS and LAS scores. For this particular setup, our results seem to indicate that using a mixture of raw and normalized eye-tracking features seems to provide the best improvements.

The results also seem to indicate that grouped gaze features perform better than individual gaze features and that only on combining different gaze feature groups can we obtain significant improvements over the baseline if any. This seems to suggest that the various eye-tracking features contain information which is complementary in nature.

Our experiments also seem to show that gaze features contain syntactic information allowing the parser to parse verbs better and also to better understand co-ordination structures.

Acknowledgements

We would like to especially thank Maria Barrett for providing us with the extracted gaze features without which this work would not have been possible. We also thank the reviewers for their informative comments.

References

Lauriane Aufrant, Guillaume Wisniewski, and François Yvon. 2016. Zero-resource dependency parsing: Boosting delexicalized cross-lingual transfer with linguistic knowledge. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 119–130, Osaka, Japan, December. The COLING 2016 Organizing Committee.

Maria Barrett and Anders Søgaard. 2015a. Reading behavior predicts syntactic categories. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 345–349, Beijing, China, July. Association for Computational Linguistics.

Maria Barrett and Anders Søgaard. 2015b. Using reading behavior to predict grammatical functions. In *Proceedings of the Sixth Workshop on Cognitive Aspects of Computational Language Learning*, pages 1–5, Lisbon, Portugal, September. Association for Computational Linguistics.

Maria Jung Barrett, Zeljko Agic, and Anders Søgaard. 2015. The dundee treebank. In *Proceedings of the Fourteenth International Workshop on Treebanks and Linguistic Theories*, pages 242–248. Association for Computational Linguistics.

Maria Barrett, Joachim Bingel, Frank Keller, and Anders Søgaard. 2016a. Weakly supervised part-of-speech tagging using eye-tracking data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 579–584, Berlin, Germany, August. Association for Computational Linguistics.

Maria Barrett, Frank Keller, and Anders Søgaard. 2016b. Cross-lingual transfer of correlations between parts of speech and gaze features. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1330–1339, Osaka, Japan, December. The COLING 2016 Organizing Committee.

Maria Barrett, Joachim Bingel, Nora Hollenstein, Marek Rei, and Anders Søgaard. 2018. Sequence classification with human attention. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 302–312, Brussels, Belgium, October. Association for Computational Linguistics.

Joachim Bingel, Maria Barrett, and Sigrid Klerke. 2018. Predicting misreadings from gaze in children with reading difficulties. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 24–34, New Orleans, Louisiana, June. Association for Computational Linguistics.

Marc Brysbaert and Françoise Vitu, 1998. *Word skipping: Implications for theories of eye movement control in reading.*, pages 125–147. Eye guidance in reading and scene perception. Elsevier Science Ltd, Oxford, England.

Roger Chaffin, Robin Morris, and Rachel Seely. 2001. Learning new word meanings from context: A study of eye movements. *Journal of experimental psychology. Learning, memory, and cognition*, 27:225–35, 02.

Joe Cheri, Abhijit Mishra, and Pushpak Bhattacharyya. 2016. Leveraging annotators' gaze behaviour for coreference resolution. In *Proceedings of the 7th Workshop on Cognitive Aspects of Computational Language Learning*, pages 22–26, Berlin, August. Association for Computational Linguistics.

Susan A Duffy, Robin K Morris, and Keith Rayner. 1988. Lexical ambiguity and fixation times in reading. *Journal of Memory and Language*, 27(4):429 – 446.

Susan F. Ehrlich and Keith Rayner. 1981. Contextual effects on word perception and eye movements during reading. *Journal of Verbal Learning and Verbal Behavior*, 20(6):641 – 655.

Lyn Frazier and Keith Rayner. 1982. Making and correcting errors during sentence comprehension: Eye movements in the analysis of structurally ambiguous sentences. *Cognitive Psychology*, 14(2):178 – 210.

M. Gareth Gaskell, Adrian Staub, and Keith Rayner. 2012. Eye movements and on-line comprehension processes, 09.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November.

Nora Hollenstein and Ce Zhang. 2019. Entity recognition at first sight: Improving NER with eye movement information. 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 1–10, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Nora Hollenstein, Antonio de la Torre, Nicolas Langer, and Ce Zhang. 2019. CogniVal: A framework for cognitive word embedding evaluation. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 538–549, Hong Kong, China, November. Association for Computational Linguistics.

Barbara J. Juhasz and Keith Rayner. 2003. Investigating the effects of a set of intercorrelated variables on eye fixation durations in reading. *Journal of experimental psychology. Learning, memory, and cognition*, 29 6:1312–8.

Barbara Juhasz and Keith Rayner. 2006. The role of age of acquisition and word frequency in reading: Evidence from eye fixation durations. *Visual Cognition - VIS COGN*, 13:846–863, 05.

Barbara J. Juhasz. 2005. Age-of-acquisition effects in word and picture identification. *Psychological Bulletin*, 131(5):684–712.

Alan Kennedy, Robin Hill, and Jöel Pynte. 2003. The dundee corpus. In *Proceedings of the European Conference on Eye Movements (ECEM)*.

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.

Sigrid Klerke, Sheila Castilho, Maria Barrett, and Anders Søgaard. 2015. Reading metrics for estimating task efficiency with MT output. In *Proceedings of the Sixth Workshop on Cognitive Aspects of Computational Language Learning*, pages 6–13, Lisbon, Portugal, September. Association for Computational Linguistics.

Sigrid Klerke, Yoav Goldberg, and Anders Søgaard. 2016. Improving sentence compression by learning to predict gaze. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1528–1533, San Diego, California, June. Association for Computational Linguistics.

K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba. 2016. Eye tracking for everyone. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2176–2184.

Simon P. Liversedge, Keith Rayner, Sarah J. White, Dorine Vergilino-Perez, John M. Findlay, and Robert W. Kentridge. 2004. Eye movements when reading disappearing text: is there a gap effect in reading? *Vision Research*, 44(10):1013 – 1024.

Alessandro Lopopolo, Stefan L. Frank, Antal van den Bosch, and Roel Willems. 2019. Dependency parsing with your eyes: Dependency structure predicts eye regressions during reading. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 77–85, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Sandeep Mathias, Diptesh Kanojia, Kevin Patel, Samarth Agrawal, Abhijit Mishra, and Pushpak Bhattacharyya. 2018. Eyes are the windows to the soul: Predicting the rating of text quality using gaze behaviour. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2352–2362, Melbourne, Australia, July. Association for Computational Linguistics.

Ryan Mcdonald. 2006. *Discriminative Learning and Spanning Tree Algorithms for Dependency Parsing*. Ph.D. thesis, USA. AAI3225503.

Enrique Meseguer, Manuel Carreiras, and Charles Clifton. 2002. Overt reanalysis strategies and eye movements during the reading of mild garden path sentences. *Memory & cognition*, 30:551–61, 07.

Abhijit Mishra, Diptesh Kanojia, and Pushpak Bhattacharyya. 2016a. Predicting readers' sarcasm understandability by modeling gaze behavior. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, page 3747–3753. AAAI Press.

Abhijit Mishra, Diptesh Kanojia, Seema Nagar, Kuntal Dey, and Pushpak Bhattacharyya. 2016b. Harnessing cognitive features for sarcasm detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1095–1104, Berlin, Germany, August. Association for Computational Linguistics.

Abhijit Mishra, Diptesh Kanojia, Seema Nagar, Kuntal Dey, and Pushpak Bhattacharyya. 2016c. Leveraging cognitive features for sentiment analysis. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 156–166, Berlin, Germany, August. Association for Computational Linguistics.

Abhijit Mishra, Diptesh Kanojia, Seema Nagar, Kuntal Dey, and Pushpak Bhattacharyya. 2017. Scanpath complexity: Modeling reading effort using gaze information. In *AAAI Conference on Artificial Intelligence*.

Robin K. Morris. 1994. Lexical and message-level sentence context effects on fixation times in reading. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 20(1):92–103.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož, Slovenia, May. European Language Resources Association (ELRA).

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2089–2096, Istanbul, Turkey, May. European Language Resources Association (ELRA).

Keith Rayner and Susan A. Duffy. 1986. Lexical complexity and fixation times in reading: Effects of word frequency, verb complexity, and lexical ambiguity. *Memory & Cognition*, 14(3):191–201.

Keith Rayner and Louise Frazier. 1989. Selection mechanisms in reading lexically ambiguous words. *Journal of experimental psychology. Learning, memory, and cognition*, 15 5:779–90.

Keith Rayner and George W. McConkie. 1976. What guides a reader's eye movements? *Vision Research*, 16(8):829–837.

Keith Rayner and Arnold D. Well. 1996. Effects of contextual constraint on eye movements in reading: A further examination. *Psychonomic Bulletin & Review*, 3(4):504–509, Dec.

Keith Rayner, Simon P. Liversedge, Sarah Jane White, and Dorine Vergilino-Perez. 2003. Reading disappearing text. *Psychological Science*, 14:385 – 388.

Keith Rayner, Simon P. Liversedge, and Sarah J. White. 2006. Eye movements when reading disappearing text: The importance of the word to the right of fixation. *Vision Research*, 46(3):310 – 323.

Javier San Agustin, Henrik Skovsgaard, John Paulin Hansen, and Dan Witzner Hansen. 2009. Low-cost gaze interaction: Ready to deliver the promises. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '09, page 4453–4458, New York, NY, USA. Association for Computing Machinery.

Javier San Agustin, Henrik Skovsgaard, Emilie Mollenbach, Maria Barret, Martin Tall, Dan Witzner Hansen, and John Paulin Hansen. 2010. Evaluation of a low-cost open-source gaze tracker. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, ETRA '10, page 77–80, New York, NY, USA. Association for Computing Machinery.

M. Schuster and K.K. Paliwal. 1997. Bidirectional recurrent neural networks. *Trans. Sig. Proc.*, 45(11):2673–2681, November.

Sara Sereno, Patrick O'Donnell, and Keith Rayner. 2006. Eye movements and lexical ambiguity resolution: Investigating the subordinate-bias effect. *Journal of experimental psychology. Human perception and performance*, 32:335–50, 05.

Anders Søgaard. 2016. Evaluating word embeddings with fMRI and eye-tracking. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 116–121, Berlin, Germany, August. Association for Computational Linguistics.

Michalina Strzyz, David Vilares, and Carlos Gómez-Rodríguez. 2019a. Sequence labeling parsing by learning across representations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5350–5357, Florence, Italy, July. Association for Computational Linguistics.

Michalina Strzyz, David Vilares, and Carlos Gómez-Rodríguez. 2019b. Towards making a dependency parser see. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1500–1506, Hong Kong, China, November. Association for Computational Linguistics.

Michalina Strzyz, David Vilares, and Carlos Gómez-Rodríguez. 2019c. Viable dependency parsing as sequence labeling. 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 717–723, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Rihana S. Williams and Robin K. Morris. 2004. Eye movements, word familiarity, and vocabulary acquisition. *European Journal of Cognitive Psychology*, 16(1-2):312–339.

Jie Yang and Yue Zhang. 2018. Ncrf++: An open-source neural sequence labeling toolkit. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Daniel Zeman and Philip Resnik. 2008. Cross-language parser adaptation between related languages. In *Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages*.

A Analysis of results for delexicalized sequence labelling parser

Table 5 shows the most common correctly predicted dependency relations by our delexicalized sequence labelling parser compared to baseline on the test set. The counts are simply the number of times the delexicalized parser correctly identified that particular dependency relation compared to the baseline parser and the relative frequency is the count divided by the total number of occurrences of that dependency relation in the gold test data. Similarly, Table 6 shows the most common POS tags with correctly predicted heads and dependency relations by our delexicalized parser compared to baseline on the test set.

Deprel	Counts	Relative frequency
root	23	9%
nmod	23	3%
aux	21	15%
nsubj	15	4%
conj	13	6%
case	13	2%
cop	12	13%
det	11	2%
dobj	10	4%
mark	9	3%

Table 5: Most common correctly predicted deprels by our delexicalized sequence labelling parser compared to baseline

POS tag	Counts	Relative frequency
NOUN	112	8%
VERB	82	9%
CONJ	27	15%
ADV	21	6%
ADP	19	3%
DET	17	3%
PRON	16	5%
ADJ	15	3%
PRT	14	6%
NUM	3	4%

Table 6: Most common POS tags with correctly predicted heads and deprels by our delexicalized sequence labelling parser compared to baseline

Parameter	Value
Word embedding dim.	100
POS embedding dim.	25
Optimizer	SGD
Epochs	150
Batch size	8
LSTM hidden state dim.	800
LSTM layers	2
Learning rate	0.02
Learning rate decay	0.05
Momentum	0.9
Dropout	0.5
Char. embedding dim.	30
Char. LSTM dim.	50
Char dropout	0.5

Table 7: Hyperparameters for the sequence labelling parser

B Hyperparameters

Table 7 describes the hyperparameters for the sequence labelling parser. We keep most of the original parameters from the original architecture as the goal of this study is more of an exploratory nature as opposed to achieving state-of-the-art results in dependency parsing. We trained each model upto 150 epochs and kept the model with the highest score on the development set. During multitask learning, for the combined setup we weigh both the tasks as 1.0 since both are equally important and are our main tasks.

Table 8 describes the hyperparameters for the graph-based BIST parser. Again we keep most of the original parameters from the original architecture and only changed the dimension of the LSTM hidden state to 800 because the original parameter was giving poorer results. We trained each model upto 50 epochs and saved the model after every epoch.

Parameter	Value
Word embedding dim.	100
POS embedding dim.	25
Optimizer	Adam
Epochs	50
LSTM hidden state dim.	800
LSTM layers	2
Learning rate	0.1
MLP hidden state dim.	100

Table 8: Hyperparameters for the BIST graph-based parser

C Architecture

Figure 2 depicts the broad architecture of our sequence labelling parser in combined MTL setup.

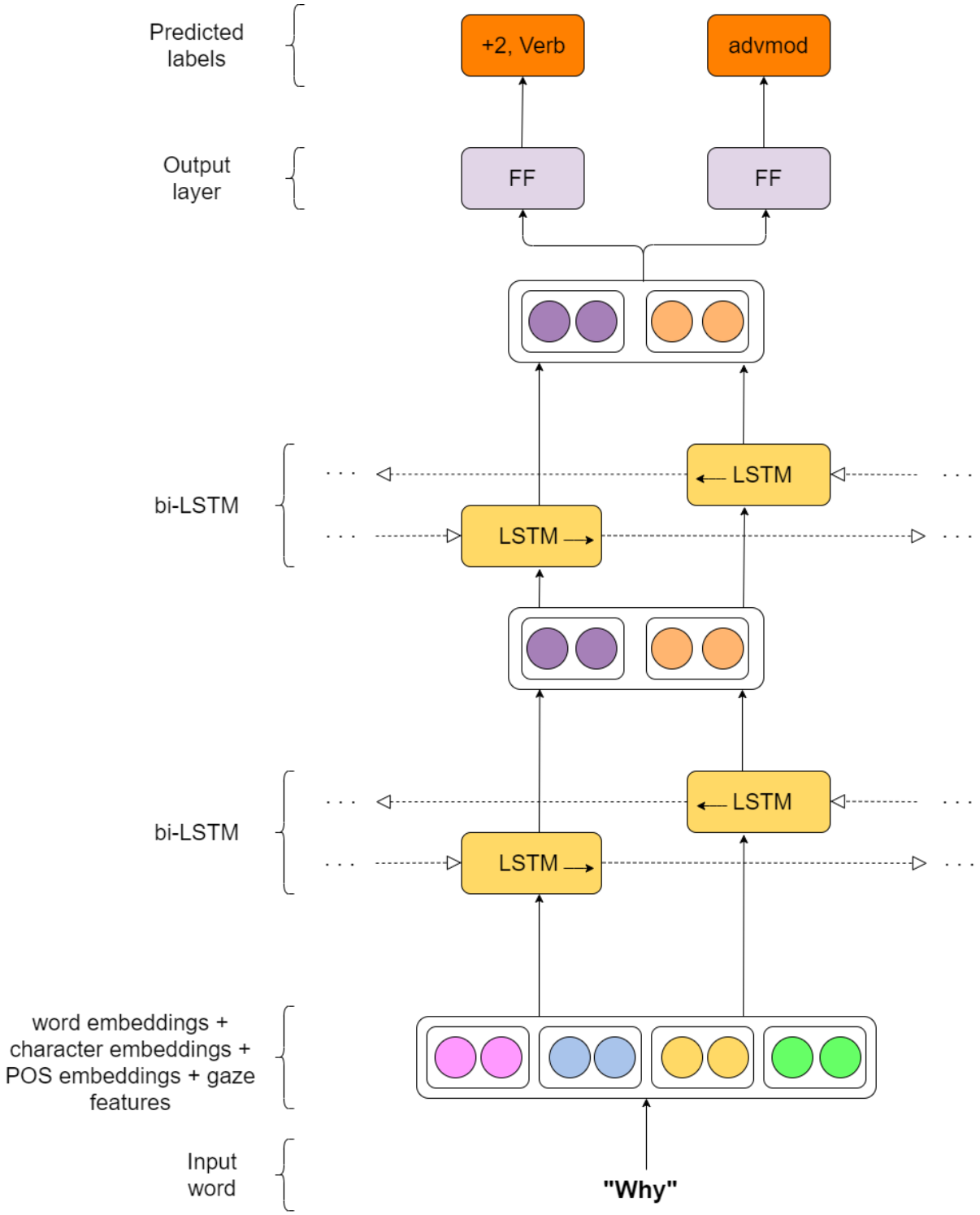

Figure 2: Architecture of sequence labelling parser in combined MTL setup