Linguistic representations in multi-task neural networks for ellipsis resolution

Ola Rønning
Cobiro
ronning@pm.me

Daniel Hardt
Copenhagen Business School
dh.msc@cbs.dk

Anders Søgaard
University of Copenhagen
soegaard@di.ku.dk

Abstract

Sluicing resolution is the task of identifying the antecedent to a question ellipsis. Antecedents are often sentential constituents, and previous work has therefore relied on syntactic parsing, together with complex linguistic features. A recent model instead used partial parsing as an auxiliary task in sequential neural network architectures to inject syntactic information. We explore the linguistic information being brought to bear by such networks, both by defining subsets of the data exhibiting relevant linguistic characteristics, and by examining the internal representations of the network. Both perspectives provide evidence for substantial linguistic knowledge being deployed by the neural networks.

1 Introduction

Sluices are questions where material beyond the *wh*-word is missing and must be retrieved from context. Consider the following example from Rønning et al. (2018):

(1) If [this is not practical], explain *why*.

Here, the antecedent is the complete sentential constituent, *this is not practical*.

Anand and Hardt (2016) present a sluice resolution system, in which candidate antecedents are required to be sentential constituents. Furthermore, each candidate is represented by features manually defined over syntactic dependency structures. Anand and Hardt report an accuracy of antecedent selection of 0.72, and a token-level F1 score 0.72, applied to a dataset based on news content (Anand and McCloskey, 2015). Rønning et al. (2018) show that neural network architectures with multi-task learning are able to achieve comparable results to Anand and Hardt, without relying on structured syntactic annotation or handcrafted features. On a slightly different version of the news dataset, Rønning et al. report a token-level F1 score of 0.70, compared to 0.67 for Anand and Hardt's system. Furthermore, it is far superior to Anand and Hardt's system at adapting from the newswire to a dialogue dataset.

This is quite surprising as sluicing is traditionally understood to be constituent-based. Two explanations present themselves; first, the traditional view might simply be wrong – that is, linguistic structure is not actually needed for ellipsis resolution. The second, and perhaps more reasonable, explanation is that Rønning et al.'s multi-task neural network architectures have learned to extract and incorporate the relevant linguistic representations.

In this paper, we investigate the linguistic knowledge learned implicitly in the experiments in Rønning et al. (2018). We take two approaches to this:

1. We select linguistically-defined subsets of the data, and examine the output of different systems on these subsets; and

2. we examine activations of the networks, focusing in particular on the activations associated with the *wh*-word that identifies a sluice, to assess how well the network notices, remembers, and classifies them.

2 Systems

The two sluicing resolution systems we compare, are: The linguistic system (AH) by Anand and Hardt (2016) and the neural network architectures introduced by Rønning et al. (2018).

2.1 Linguistic System (AH)

The system presented in Anand and Hardt (2016) defines a set of linguistically motivated features over pre-parsed input to determine the most likely

Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, pages 66–73
Brussels, Belgium, November 1, 2018.

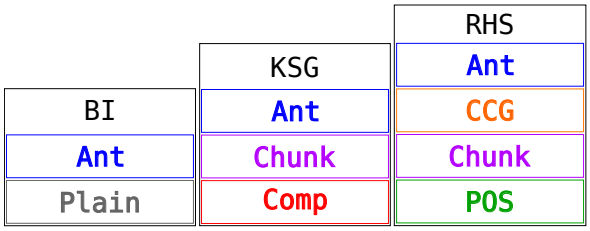

Figure 1: Task hierarchy of the three networks considered; the tasks are (Ant)ecedent Selection, (Chunk)ing, Sentence (Comp)ression, (CCG) super-tagging and (POS) tagging. Plain denotes a RNN-layer with no associated task.

antecedent among a set of candidates. The candidate set consist of all sentential constituents within a predefined context window of the sluice. AH parameterizes a log-linear score akin to Denis and Baldridge (2008) using hill-climbing. Anand and Hardt (2016) evaluates AH by the accuracy of chosen constituents, but also report token-level F1 scores. We focus on token-F1 score in this paper to compare with the neural networks as these do not operate with predefined constituents.

2.2 Neural Network Architectures

We examine three neural network architectures, all defined in Rønning et al. (2018), and depicted in Figure 1. In all three systems, the input is a sequence of tokens without syntactic annotations. All our neural networks use 50 dimensional fixed GloVe embeddings, obtained by applying the model in Pennington et al. (2014) on Wikipedia and Gigaword 5. The sluice expression is not specifically marked in the text. Instead, a copy of the sluice expression is prefixed to the sequence. The networks assign either a begin, inside or outside (BIO) label to each token and the task is to align these with the span of the antecedent. The three networks are:

BI: This is a single-task, two-layered long-short-term memory (LSTM) network, with a projection layer and a softmax layer.

KSG: This is a cascading, three-layered LSTM, as described by Klerke et al. (2016). The KSG system is trained with the following auxiliary tasks:

> **Chunking**: a partial parsing task, in which we need to identify the boundaries of the phrases in a sentence; and
>
> **Sentence compression:** the task of finding sentence parts that can be dropped without losing coherence or important information

During an epoch, k batches of size b are sampled from each of the three tasks such that $k\dot{b}$ is the number of examples in the antecedent selection task. We choose batches from each of the auxiliary tasks in the fixed order: sentence compression, chunking, antecedent selection.

RHS: This system also cascades the auxiliary tasks. However, it uses a different set of auxiliary tasks than KGS, computes label embeddings that are also passed on to subsequent layers, and has skip connections from the embedding layer to all layers in the network. RHS uses the auxiliary tasks described below:

> **CCG super-tagging:** another form of partial parsing, using a more fine-grained tagset.
>
> **Chunking**: same task as described for KSG.
>
> **POS tagging**: determining the syntactic category (part of speech) of a word in context.

RHS cycles through all data for each of the auxiliary tasks during a epoch, only layers up to and including the layer associated with the tasks currently being optimized is active during training of that task. Over the course of an epoch, the network trains on POS, Chunk, CCG-super tagging and, then, antecedent selection.

Table 1 gives the token-level F1 score for each system on the dataset used in Rønning et al. (2018). It also includes the baseline performance of choosing a random constituent within the two sentence window of the sluice site. This is the same window size AH uses to determine its candidate set.

System	Score
AH	0.67
Random	0.45
RHS	0.70
KSG	0.64
BI	0.54

Table 1: Token-F1 Score on complete test set.

3 Data Subsets

Below we introduce various linguistic dimensions of the ellipsis resolution data, which we can use to

study the models' behavior on subsets of the data, indirectly probing what linguistic distinctions they make.

3.1 Adjacency

It is very common for the correct antecedents to be sentential constituent immediately preceding the ellipsis site. Call this constituent the *adjacent* candidate to the sluice. The antecedent in Example 1 is adjacent. However, adjacency is not ubiquitous. We refer to all other candidates as non-adjacent. Example 2 from Anand and Hardt (2016) is non-adjacent:

(2) [$_{S-3}$ Deliveries would increase as a result of the acquisition] , [$_{S-2}$ he predicted] , but [$_{S-1}$ he would not say by how much]

The sluice expression *how much* has S_2 as the adjacent constituent, while the correct antecedent is the non-adjacent S_3. Table 2 gives token-level F1 scores for all systems with the dataset partitioned by adjacency.

System	Adjacent	Non-Adjacent	Difference
AH	84.8	56.5	28.3
RHS	84.7	65.5	19.2
KSG	74.5	60.6	13.9
BI	62.3	51.0	11.3

Table 2: F1 score for Adjacent vs. Non-adjacent Sluice Antecedents

All systems score higher with adjacent candidates than with non-adjacent ones, i.e., when the true antecedent is adjacent. AH scores marginally higher on adjacent candidates than RHS, but has a significantly higher difference between adjacent and non-adjacent compared to the three neural systems. Since AH has explicit constituency information, it makes sense that it would have a high token-level F1 score, when the antecedent is adjacent. For the MTL systems, RHS and KSG, adjacency also makes a big difference in F1 scores, albeit less than for AH. The smallest performance drop is seen with BI, the single-task system.

This is somewhat comparable to the case of subject-verb agreement studied in Linzen et al. (2016), where it was found that an LSTM could learn structural information necessary to identify the subject, but that performance decreased when the subject was not the nearest noun phrase to the verb. In their case, the neural model had learned a representation that was too dependent on adjacency information; in our case, it seems like the neural architectures have successfully learned a representation that makes them less dependent on adjacency than the baseline BI system.

3.2 Punctuation/Boundary Tokens

A major difference between AH and the neural models is that the neural models do not have explicit marking of the boundaries of candidate antecedents, as AH does. The neural models may, however, rely on specific tokens that signal these boundaries. We hypothesize that punctuation can play this role. Based on this, we would expect that the neural networks do better when antecedents are marked by punctuation, while this should matter less, if at all, to the AH system.

We define the following subsets of the data: first, we restrict ourselves to cases where the antecedent is adjacent. Then we define three subsets:

- (L) the correct antecedent has a punctuation token on its left edge
- (R) the correct antecedent has a punctuation token on its right edge
- (LR) the correct antecedent has a punctuation token on both its left edge and its right edge

System	LR	R	L
AH	85.0	84.9	85.8
RHS	85.6	85.5	80.7
KSG	75.2	74.1	71.1
BI	62.3	62.3	58.7

Table 3: F1 score for punctuation as boundary token for antecedent

The results, which are given in Table 3, partially support our hypothesis. Focusing on LR (where punctuation marks both edges of the adjacent antecedent), we observe that RHS and KSG see improvements of .9 and .7 respectively, while AH is improved only by .2. This confirms our expectation that punctuation marking would help the neural networks more than AH. However, BI is surprising here, since it is not helped at all. Furthermore, the pattern is much less clear when we look at the cases of R and L. For the R case, we see that RHS is helped quite a bit, while AH is not; this again supports our hypothesis. However, we

are surprised to see that KSG is worse with R than adjacency overall, and the pattern with L does not at all support the hypothesis. Thus while our results are indeed suggestive that punctuation might play an important role here, the picture seems to be complicated by other factors.

3.3 Discontinuity

Since the sluicing antecedent is a constituent, it often consists of a continuous sequence of tokens, as in Example 1. This is not *always* true, however; it is well known that constituents are sometimes *discontinuous*, and this can in fact be observed in our sluicing data, as in the following example (Anand and Hardt, 2016):

(3) [$_{S-2}$ A major part of the increase in coverage , [$_{S-1}$ though Mitchell 's aides could not say just **how much** ,] would come from a provision providing insurance for children and pregnant women .]

Here, the antecedent is *A major part of the increase in coverage would come from a provision providing insurance for children and pregnant women*; the tokens constituting S_1 are not included. However, such cases of discontinuous antecedents are rare in our data – the vast majority of sluice antecedents consist of a continuous sequence of tokens.

The AH system nearly always selects continuous antecedents, since the underlying syntactic parser is unable to predict discontinuous constituents. The other systems are under no such constraint; however, implicit linguistic knowledge might be expected to result in a higher degree of continuity, which we measure as follows:

$$1 - \frac{|\text{holes}|}{|\text{inner-span}|}$$

where the span is the subsequence starting with the first begin or inside tag and ending with the last begin or inside tag; the *inner-span* is the span without the boundary tags, and *holes* is the set of outside tags within the span.

System	Contiguity
RHS	84.5
KSG	82.0
BI	78.4

Table 4: Degree of token continuity

BI has a lower degree of token continuity than the multi-task systems, RHS and SG. This suggests that the multi-task architectures learn implicit knowledge about linguistic constituency.

3.4 Matching Content

Anand and Hardt (2016) point out that, in sluicing, "the wh-phrase must semantically cohere with the main predicate of the antecedent", which they illustrate with example 4. Here S-3 is a more likely antecedent than S-2 because *increase* is more likely to take an implicit extent than *predict*. In other words *increase* is a better match for *how much* than *predict* is.

(4) . [$_{S-3}$ Deliveries would increase as a result of the acquisition] , [$_{S-2}$ he predicted] , but [$_{S-1}$ he would not say by how much]

To capture this information, Anand and Hardt collect data on cooccurrences between wh-phrases and main predicates, and based on this they calculate a feature WHGOVNPMI, the Pointwise Mutual Information (PMI) of the wh-phrase and the main verb of each candidate. In general, the adjacent candidate S_2 would normally be preferred by the AH system. But this could be overruled by the fact that the non-adjacent candidate, S_3, has a higher value for the WHGOVNPMI feature.

We wish to explore whether the neural networks can also take advantage of this information. While the networks do not have access to the PMI information Anand and Hardt computed, they do have access to embeddings for each input word, and one might imagine that pairs with high PMI would also tend to be closer in embedding space. To see why we would expect that, consider that Anand and Hardt collected statistics on *overt* WH-constructions, where the WH-phrase and associated main predicate would co-occur in close proximity, as in cases like **how much** it **increased**, or **why** did they **attack**. Thus highly related such pairs would tend to cooccur frequently within a fairly small context window, while less related pairs would not. The distances of word embeddings (such as Glove, used here), tend to reflect such differences in cooccurence frequency.

To explore this, we examine cases where the correct antecedent receives a comparatively high WHGOVNPMI score, but where it is not adjacent to the sluice (since adjacent antecedents tend to be chosen as the default option). We define various

thresholds for the WHGOVNPMI, and define the subset of only those examples where the correct antecedent receives a PMI score above that threshold. These should be examples where the PMI score provides a strong signal about the correct antecedent, although the antecedent is not adjacent to the sluice. We examine the F1 scores of the systems on these subsets. As we raise the threshold, we would expect F1 scores to increase, if the system is indeed making use of this information.

Figure 2 shows F1 scores as a function of PMI score thresholds. For RHS and BI, we do observe an increase of F1 scores as thresholds increase, but this is not consistently the case for either AH or KSG. It is difficult to draw any firm conclusions here. While the plot does not show a consistent pattern for the AH system, (Anand and Hardt, 2016) give feature ablation results that show that the WHGOVNPMI does make a positive, although modest, contribution. The plots here suggest that the word embedding comparisons of the neural networks might in fact contain more useful information than the PMI statistics.

4 Probing the Networks

We probe the networks by examining particular network states, posing three specific questions that are fundamental to the sluice resolution task:

- How well does the network distinguish *adjacent* from *non-adjacent* antecedents?
- How well does the network *notice* the sluice word?
- How well does the network *remember* the sluice word, when the antecedent is encountered?

4.1 Classification of Adjacent vs. Non-adjacent

We collect the final state activation for the network for each instance, and divide these states into two classes: in one class, the correct antecedent is adjacent to the sluice, and in the other class it is not. We perform logistic regression, using class balancing, with five-fold cross validation.

The MTL systems, RHS and KSG, score higher than chance, while the single-task baseline is a bit below chance. KSG performance significantly better than RHS; however, all networks are close to chance. From Table 2, we saw that all networks perform substantially worse on non-adjacent sluices, which suggests the networks are treating non-adjacent antecedents as if they were adjacent.

System	Adjacent
RHS	54.3
KSG	55.9
BI	48.0

Table 5: Accuracy of classifying adjacent antecedents from non-adjacent antecedents.

4.2 Noticing the Sluice Word

The AH system uses input in which the sluice occurrence is explicitly marked. This is not the case for the neural networks. Instead, a copy of the sluice phrase is prefixed to the example. Ideally, the neural network would remember this phrase and locate its copy within the text. Then the system would search for the antecedent in proximity to the sluice phrase. The following example illustrates the representation for Example 1:

(5) why If this is not practical , explain why .

We term the first occurrence of *why* a *prefixed* wh-word, and the second occurrence is an *in-situ* wh-word. Wh-word occurrences that are neither prefixed or in-situ, are termed *other*.

We collect all the activations associated with in-situ wh-words. This is the positive class in our classification task. Next we collect all activations associated with wh-words occurring in the data, when they are *not* sluice words. This constitutes the negative class. We perform logistic regression, again with class balancing. We want to see if the network activations clearly distinguish the sluice wh-words from other wh-words.

System	Sluiced Wh-word
RHS	78.0
KSG	80.2
BI	76.6

Table 6: Accuracy of classifying *wh*-words in sluiced and non-sluiced positions.

Results in Table 6 show that all three networks are doing substantially better than chance. This is interesting, since noticing the sluice wh-word would seem to be a crucial first step in the sluice

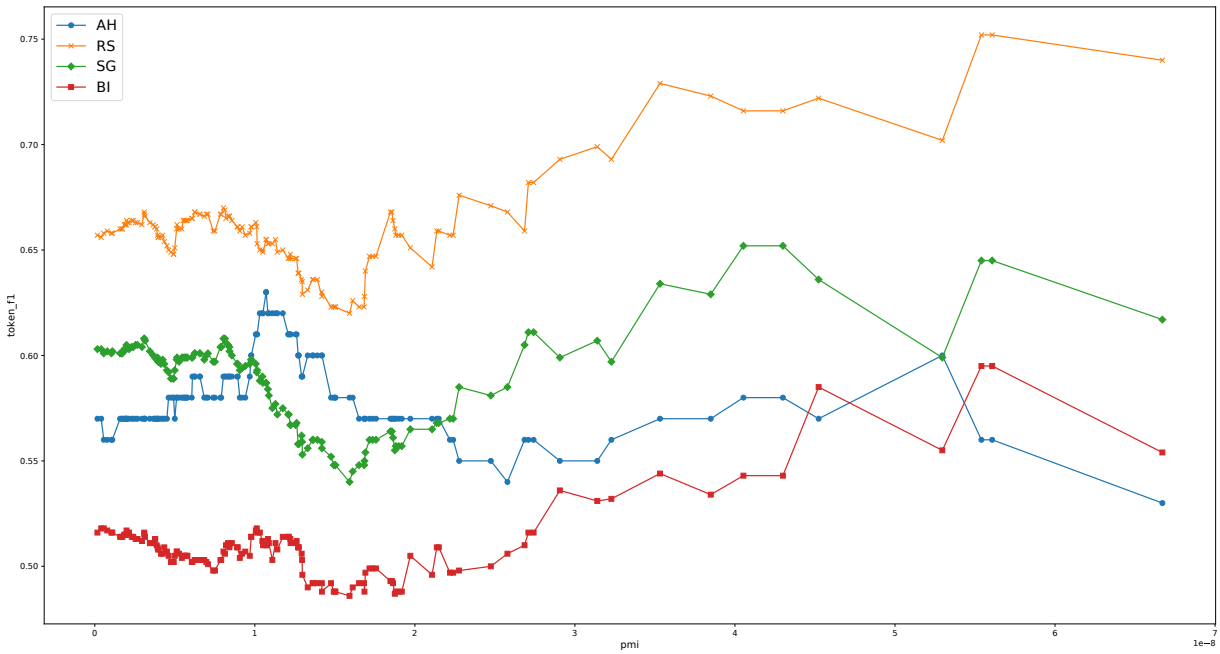

Figure 2: Minimum pointwise mutual information vs token-F1 score on those examples, only thresholds with at least one hundred examples included in plot.

resolution task. We note further that the activations from both multi-task systems (KSG and RHS) provide a better basis for classification than the baseline BI. We suggest that the linguistic auxiliary tasks might explain this difference, since distinguishing the sluice word is facilitated by knowledge of linguistic structure. It is, however, somewhat surprising that KSG performs better than RHS on this task, since RHS does better on the sluice resolution task, overall.

These classification results provide an indication that, in some sense, the networks are indeed noticing whether the wh-words appear in a sluice or not. We suggest further that this is reflected quite directly in network activations. In general, we suggest that the distance between a word embedding and its associated activation provides a measure of how much the network notices that word. Table 7 supports this idea; here, we compare embedding and activation distances for prefixed, in-situ and other wh-words. The results support the idea that in-situ wh-words are noticed the most, as they are crucial to task of determining the antecedent. The prefix wh-word has somewhat larger distances, while other wh-words, which play no role is sluicing, have the highest distances. Furthermore, we note that the distances suggest that the KSG system is "best" at noticing the in-situ wh-word, while BI is worst. In Table 6 the accuracy of in-situ sluice word classification follows the same ordering.

System	other-wh	in-situ	prefix	all
RHS	5.82	5.16	5.33	5.61
KSG	5.61	5.03	5.29	5.43
BI	6.32	5.31	5.61	6.11

Table 7: Average distance between embedding and activation for same token.

4.3 Remembering Sluice Word at Antecedent

We have shown that the networks are able to distinguish adjacent from non-adjacent antecedents, and they are also able to notice the sluice *wh*-word. The next question is: Can the network draw a connection between the sluice *wh*-word and the antecedent? This is fundamental to the task of sluice resolution – connecting the sluice occurrence with the antecedent.

To address this question, we propose to measure how well the sluice wh-word is "remembered" by the network when the edge of the antecedent is encountered. We compute the vector distance between the word embedding for the sluice wh-word and the state associated with the token appearing at the edge of the antecedent. We suggest that this distance provides a metric of how much the network remembers the wh-word, when the an-

System	Forward			Backward		
	WH-Ant Dist.	Avg. Dist.	Normalized	WH-Ant Dist.	Avg. Dist.	Normalized
RHS	6.34	10.72	0.59	6.22	11.52	0.54
KSG	5.07	9.47	0.54	5.33	11.08	0.48
BI	5.88	9.99	0.59	5.71	12.37	0.46

Table 8: Euclidean distance between the antecedent left boundary activation and avg. sluice word vector representation. Distances compared to average Euclidean norm distance between word representations and activations separated by the same number of tokens as the antecedent sluice pair.

tecedent is encountered. Table 8 shows the relevant measurements, both for the forward and backward directions of the each of the neural networks. The column *WH-Ant Dist* gives the average vector distance between the wh-word embedding and the state vector when the antecedent is encountered. The *Avg. Dist.* column gives a corresponding average over all token occurrences. Our intention is to provide a relevant comparison to see if the *wh*-word is remembered more than words are in general. So the Avg. Dist. column has a weighted average of the vector distances for all words, using the distribution of token distances seen for the WH-Ant Dist values. Finally, the Normalized column is the ratio, WH-Ant Dist/Avg.Dist.

Overall, we see a strong effect of the networks remembering the sluice word at the antecedent site, to a much higher degree than an average word is remembered. Furthermore, there is a difference with directionality. Avg Dist is higher in the Backwards direction – in general, words are remembered less when moving backwards. But WH-Ant Dist. is lower in Backward, which makes sense, since the system needs to keep track of the wh-word to help identify the antecedent. There are also modest differences among the three systems.

We would like to see how this develops over time. Our hypothesis is that the neural networks will remember the sluice word more at the point where it is within the antecedent, and less when it is outside the antecedent. We define four areas of interest:

1. **Between:** tokens between the sluice and the antecedent (Except the **Right** token)
2. **Right:** the token just to the right of the antecedent
3. **Ant:** tokens within the antecedent
4. **Left:** the token just to the left of the antecedent

In Figure 3, we can observe a modest effect of the sort we hypothesized: distances are indeed lower within the antecedent than in the other regions, suggesting that the neural networks do in fact have a stronger memory of the wh-word at that point in the computation.

5 Conclusion

Ellipsis resolution is widely believed to require sophisticated knowledge of linguistic structure. Thus, it is interesting that the neural architectures presented by Rønning et al. (2018), are able to match and even surpass systems like that of Anand and Hardt (2016), which rely on pre-parsed input and linguistically engineered features. In this paper, we investigated the linguistic knowledge implicit in the neural models. We have done so in two ways: (1) We have defined subsets of the data based on adjacency, boundary tokens, discontinuity, and matching content. In general, we have observed that these linguistic factors clearly play a role in the network performance, and there is further evidence that the systems with MTL have a higher degree of linguistic sophistication in their performance, compared to a single task baseline network. (2) We then examined the internal states of the networks. Focusing on the *wh*-word of the sluice, we have shown that the networks are sensitive to the occurrence of the sluice *wh*-word. Furthermore, we find some evidence that the networks remember the *wh*-word more at the point where the antecedent is encountered, compared to other points in the computation.

References

Pranav Anand and Daniel Hardt. 2016. Antecedent selection for sluicing: Structure and content. In *EMNLP*. pages 1234–1243.

Pranav Anand and Jim McCloskey. 2015. Annotating the implicit content of sluices. In *Proceedings*

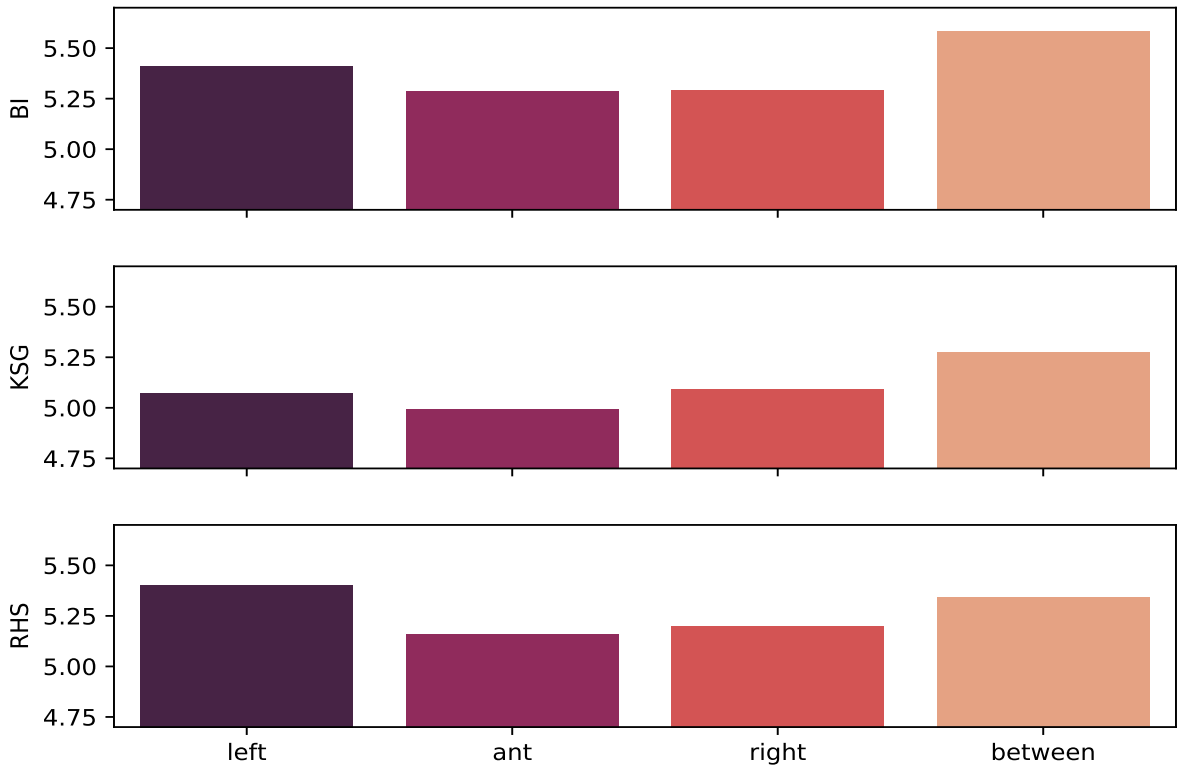

Figure 3: Euclidean distance over antecedent tokens.

of The 9th Linguistic Annotation Workshop. pages 178–187.

Pascal Denis and Jason Baldridge. 2008. Specialized models and ranking for coreference resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 660–669.

Sigrid Klerke, Yoav Goldberg, and Anders Søgaard. 2016. Improving sentence compression by learning to predict gaze. *arXiv preprint arXiv:1604.03357* .

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of lstms to learn syntax-sensitive dependencies. *arXiv preprint arXiv:1611.01368* .

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. pages 1532–1543.

Ola Rønning, Daniel Hardt, and Anders Søgaard. 2018. Sluice resolution without hand-crafted features over brittle syntax trees. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. Association for Computational Linguistics, pages 236–241. http://aclweb.org/anthology/N18-2038.