

# Examining the rhetorical capacities of neural language models

Zining Zhu<sup>1,2</sup>, Chuer Pan<sup>1</sup>, Mohamed Abdalla<sup>1,2</sup>, Frank Rudzicz<sup>1,2,3,4</sup>

1: University of Toronto; 2: Vector Institute

3: Li Ka Shing Knowledge Institute, St Michael’s Hospital

4: Surgical Safety Technologies

zining@cs.toronto.edu, chuer.pan@mail.utoronto.ca  
{msa, frank}@cs.toronto.edu

## Abstract

Recently, neural language models (LMs) have demonstrated impressive abilities in generating high-quality discourse. While many recent papers have analyzed the syntactic aspects encoded in LMs, to date, there has been no analysis of the inter-sentential, rhetorical knowledge. In this paper, we propose a method that quantitatively evaluates the rhetorical capacities of neural LMs. We examine the capacities of neural LMs understanding the rhetoric of discourse by evaluating their abilities to encode a set of linguistic features derived from Rhetorical Structure Theory (RST). Our experiments show that BERT-based LMs outperform other Transformer LMs, revealing the richer discourse knowledge in their intermediate layer representations. In addition, GPT-2 and XLNet apparently encode less rhetorical knowledge, and we suggest an explanation drawing from linguistic philosophy. Our method presents an avenue towards quantifying the rhetorical capacities of neural LMs.

## 1 Introduction

In recent years, neural LMs (especially contextualized LMs) have shown profound abilities to generate texts that could be almost indistinguishable from human writings (Radford et al., 2019). Neural LMs could be used to generate concise summaries (Song et al., 2019), coherent stories (See et al., 2019), and complete documents given prompts (Keskar et al., 2019). It is natural to question their source and extent of rhetorical knowledge: What makes neural LMs articulate, and how? While some recent works query the linguistic knowledge (Hewitt and Manning, 2019; Liu et al., 2019a; Chen et al., 2019; Belinkov et al., 2017), this open question remain unanswered. We hypothesize that contextualized neural LMs encode rhetorical knowledge in their intermediate repre-

sentations, and would like to quantify the extent they encode rhetorical knowledge.

To verify our hypothesis, we hand-craft a set of 24 rhetorical features including those used to examine rhetorical capacities of students (Mohsen and Alshahrani, 2019; Liu and Kunnan, 2016; Zhang, 2013; Powers et al., 2001), and evaluate how well neural LMs encode these rhetorical features in the representations while encoding texts.

Recent work has started to evaluate encoded features from hidden representations. Among them, probing (Alain and Bengio, 2017; Adi et al., 2017) has been a popular choice. Previous work probed morphological (Belinkov et al., 2017; Bisazza and Tump, 2018), agreement (Giulianelli et al., 2018), and syntactic features (Hewitt and Manning, 2019; Hewitt and Liang, 2019). Probing involves optimizing a simple projection model from representations to features. The loss of this optimization measures the difficulty to decode features from the representations.

In this work, we use a probe containing self attention mechanism. We first project the variable-length embeddings to a fixed-length latent representation per document. Then, we apply a simple diagnostic classifier to detect rhetorical features from this latent representation. This design of probe reduces the total number of parameters, and enable us to better understand each model’s ability to encode rhetorical knowledge. We find that:

- The BERT-based LMs encode more rhetorical features, and in a more stable manner, than other models.
- The semantics of non-contextualized embeddings also pertain to some rhetorical features, but less than most layers of contextualized language models.

These observations allow us to investigate the mechanisms of neural LMs to better understand the

degree to which they encode linguistic knowledge. We demonstrate how discourse-level features can be queried and analyzed from neural LMs. All of our code and parsed tree data will be available at github.

## 2 Structural analysis of discourse

Various frameworks exist for “good discourse” (Lawrence and Reed, 2019; Irish and Weiss, 2009; Toulmin, 1958), but most of them are inaccessible to quantitative analysis. In this work, we use Rhetorical Structure Theory (Mann and Thompson, 1988; Mann et al., 1989) since it represents the structures of discourse using trees, allowing straightforward quantitative analysis. There are two components in an RST parse-tree:

- Each leaf node represents an elementary discourse unit (EDU). The role of an EDU in an article is similar to that of a word in a sentence.
- Each non-leaf node denotes a relation involving its two children. Often, one of the children is more dependent on the other, and less essential to the writer’s purpose. This child is referred to as “satellite”, while the more central child is the “nucleus”.

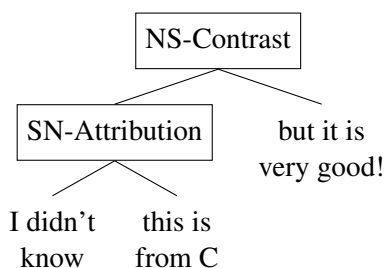


Figure 1: A portion of an RST tree, selected from IMDB (Maas et al., 2011) `train/pos/1.7.txt`, and parsed with Feng and Hirst (2014). Nodes with rectangle borders are discourse relations, and those without borders are individual EDUs. The “N” and “S” prefix for discourse relations stand for “nucleus” and “satellite” respectively.

Tree representations are clear, easy to understand, and allow us to compute features to numerically depict the rhetorical aspects of documents.

### 2.1 Rhetorical features

Previous work used RST features to analyze the quality of discourse, to assess writing abilities

(Wang et al., 2019; Zhang, 2013), examine linguistic coherence (Feng et al., 2014; Abdalla et al., 2017), and to analyze arguments (Chakrabarty et al., 2019). In this project, we extract similar RST features in the following three categories:

**Discourse relation occurrences (Sig)** We include the number of relations detected in each document. There are 18 relations in this category<sup>1</sup>. Unfortunately, the relations adopted by open-source RST parsers are not unified. To allow for comparison against other parsers, we do not differentiate subtle differences between relations, therefore grouping very similar relations, following the approach in (Feng and Hirst, 2012). (E.g., we consider both Topic-Shift and Topic-Drift to be a Topic-Change). Specifically, this approach does not differentiate between the sequence of nucleus and satellite (e.g., NS-Evaluation and SN-Evaluation are both considered as an Evaluation).

**Tree property features (Tree)** We compute the depth and the Yngve depth (the number of right-branching in the tree) (Yngve, 1960) of each tree node, and include their mean and variance as characteristic features, following previous work extracting tree linguistic features (Li et al., 2019; Zhu et al., 2019).

**EDU related features (EDU)** We include the mean and variance of EDU lengths of each document. We hypothesize the longer EDUs indicate higher levels of redundancy in discourse, hence extracting rhetorical features require memory across longer spans.

Overall, there are 24 features from three categories. We normalize them to zero mean and unit variance, and take these RST features for probing. The features are not independent of each other. Specifically, the features of each group tend to describe the same property from different aspects.<sup>2</sup>

<sup>1</sup>The 18 relations are: Attribution, Background, Cause, Comparison, Condition, Contrast, Elaboration, Enablement, Evaluation, Explanation, Joint, Manner-Means, Topic-Comment, Summary, Temporal, Topic-Change, Textual-organization, and Same-unit.

<sup>2</sup>For example, Sig features describe the composition of the document in a histogram. For the same document, if a relation is changed, e.g., from Contrast to Attribution, then the occurrence of both Contrast and Attribution are affected.

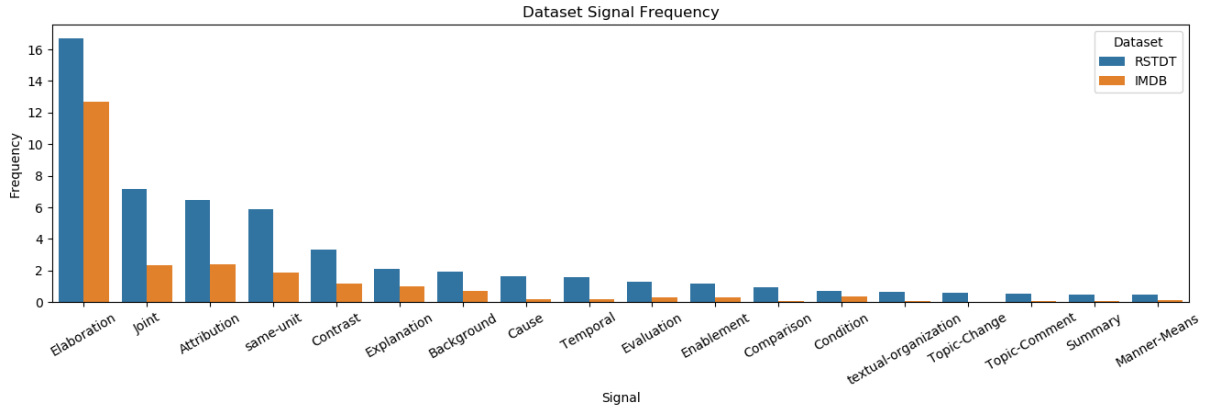


Figure 2: RST relation occurrences per document. RST-DT contain longer documents than IMDB on average. However, the distributions of frequencies between these two datasets are relatively consistent, with Elaboration, Joint, and Attribution the most frequent signals.

## 2.2 Probe

Our probing method contains two weight parameters,  $W_d$  and  $W_p$ . First, we embed a document with  $L$  tokens using a neural LM with  $D$  dimensions to get a raw representation matrix  $X \in \mathbb{R}^{L \times D}$ . We use a projection matrix  $W_d \in \mathbb{R}^{D \times d}$  to reduce the embedding dimension from  $D$  (e.g.,  $D = 768$  for BERT and 2048 for XLM) to a much smaller one,  $d$ . Then, we use self attention similar to Lin et al. (2017) to collect the information spread across the document to a condensed form:

$$A = (XW_d)^T(XW_d) \in \mathbb{R}^{d \times d}$$

We flatten  $A$  into a vector with fixed size:  $\tilde{A} = (d^2, 1)$ . We use a probing matrix  $W_p \in \mathbb{R}^{d^2 \times m}$  to extract RST features  $v \in \mathbb{R}^m$  from attention, normalize them to zero mean and unit variance, and optimize based on the expected  $L_2$  error:

$$\min_{W_d, W_p} \mathbb{E} \|W_p^T \tilde{A} - v\|^2$$

Note that the reduction from  $D$  to  $d$  using  $W_p$  is necessary, because it significantly lowers the number of parameters of the probing model. If there were no  $W_d$  (i.e.,  $d = 768$ ), then  $W_p$  alone would require  $768^2 m$  parameters to probe  $m$  features. Now, we let  $d = 10$ , then  $W_d$  and  $W_p$  combined have  $D \times d + d^2 m \approx 7680 + 100m$  parameters. Considering  $m \in \mathcal{O}(10^1)$ , the total parameter size is reduced from  $\mathcal{O}(10^6)$  to  $\mathcal{O}(10^3)$ .

There is one more step before we can use this loss to measure the difficulty of probing rhetorical features.  $L_2$  error scales linearly with the dimension of features  $m$ , so it is necessary to normalize the  $L_2$  error by  $m$ , to ensure that the losses can be

compared across linguistic feature sets. The *difficulty* of probing a group of  $m$  features  $v \in \mathbb{R}^m$  therefore is:

$$\text{Difficulty} = \frac{1}{m} \mathbb{E} \left\| W_p^T \tilde{A} - v \right\|^2$$

## 3 Experiments

### 3.1 Data

Most state-of-the-art rhetorical parsers are trained on either Penn Discourse Treebank (Ji and Eisenstein, 2014; Feng and Hirst, 2012) or RST-DT (Feng and Hirst, 2014; Joty et al., 2015; Surdeanu et al., 2015; Heilman and Sagae, 2015; Li et al., 2016; Wang et al., 2017; Yu et al., 2018). Although the documents contain accurate discourse annotations, RST-DT (Carlson et al., 2001) only has 385 documents. The Penn Discourse Treebank (Prasad et al., 2008) has 2,159 documents but their annotations do not follow the RST framework. So in addition to RST-DT, we extend the analysis to a 100 times larger dataset, IMDB (Maas et al., 2011).

IMDB contains 50,000 movie reviews without discourse annotations. In these reviews, the authors explain and elaborate upon their opinions towards certain movies and give ratings. We removed html tags, and attempt to parse all of them (i.e., both train and test data) using a two-pass parser from Feng and Hirst (2014). We discarded 1,977 documents that the RST parser generate ill-formatted trees<sup>3</sup>. Of the remaining documents, we additionally filtered out those with sequence lengths greater than 512 tokens<sup>4</sup>, resulting in 40,833 documents.

<sup>3</sup>As determined by `nltk.tree`.

<sup>4</sup>As determined by any one of the tokenizers, since these

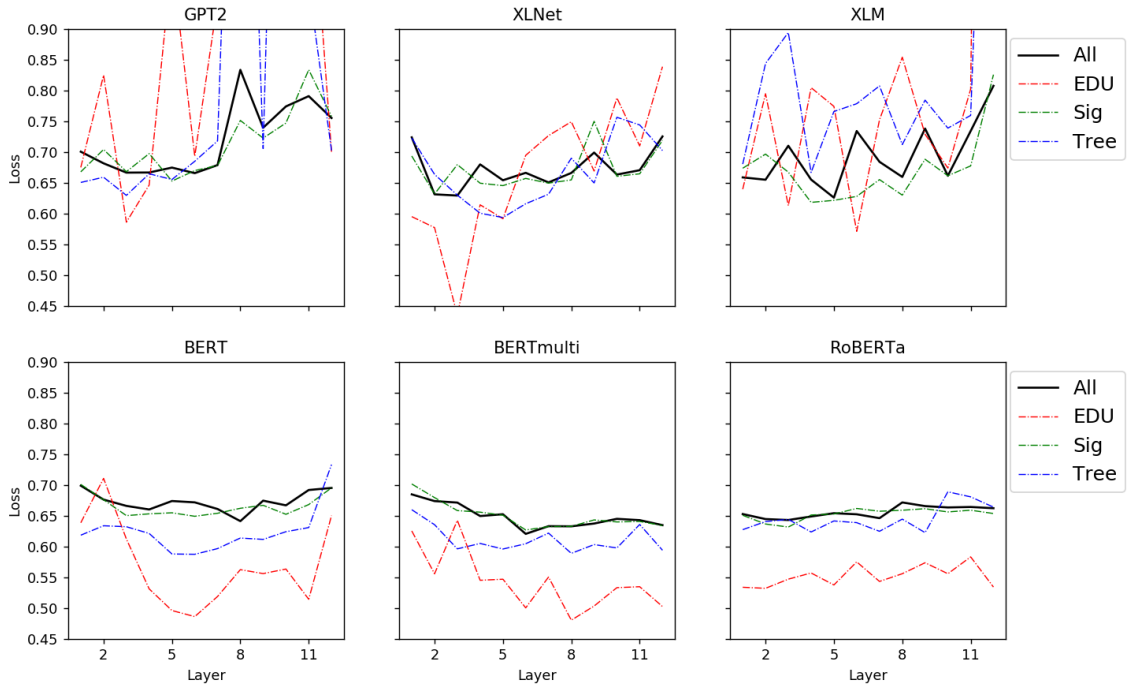


Figure 3: Loss vs layer plot of six neural LMs on four RST feature sets on IMDB. The solid lines represent all RST features combined, while each dash-dotted line denotes one component (EDU, Sig, or Tree feature group for red, green, and blue respectively). In general, BERT-based LMs (BERT, BERT-multi, RoBERTa) encode rhetorical features in a more stable and easy-to-probe manner than the rest.

After parsing each document into an “RST-tree”, we extracted the features mentioned in Section 2.1 from these parsed trees. Figure 2 shows the occurrence of the 18 RST relations per document, and Table 1 shows the statistics of remaining 6 features. In addition, we include several examples of parsed RST trees in Appendix.

| Feature name    | Mean $\pm$ stdev  |
|-----------------|-------------------|
| tree_depth_mean | 3.9 $\pm$ 1.4     |
| tree_depth_var  | 4.6 $\pm$ 4.2     |
| tree_Yngve_mean | 9.2 $\pm$ 8.8     |
| tree_Yngve_var  | 100.6 $\pm$ 164.6 |
| edu_len_mean    | 8.6 $\pm$ 1.4     |
| edu_len_var     | 21.8 $\pm$ 16.0   |

Table 1: Statistics of the 6 non-occurrence-based RST features. The prefix “tree\_” here refers to the parsed “RST-tree”.

language models come with their own tokenizers. Note that RoBERTa adds two special tokens, so this threshold becomes 510 for RoBERTa.

### 3.2 Language models

We considered the following popular neural LMs:

- BERT<sub>BASE</sub> (Devlin et al., 2019) This LM with 110M parameters is built with 12-layer Transformer encoder (Vaswani et al., 2017) with 768 hidden dimensions. It is trained with masked LM (i.e., cloze) and next sentence prediction objectives using 16GB text.
- BERT-multi (Wolf et al., 2019) Same as BERT, BERT-multi is also a 12-layer Transformer encoder with 768 hidden dimensions and 110M parameters. Its difference from BERT is that, BERT-multi is trained on top 104 languages with the largest Wikipedia.
- RoBERTa (Liu et al., 2019b) is an enhanced version of BERT with the same architecture, similar masked LM objectives, and 10 times larger training corpus (over 160GB).
- GPT-2 (Radford et al., 2019) is a 12-layer Transformer decoder with 768 hidden dimensions. There are 117M parameters in total. GPT-2 is pretrained on 40GB of text. Unlike BERT, GPT-2 is a uni-directional LM.

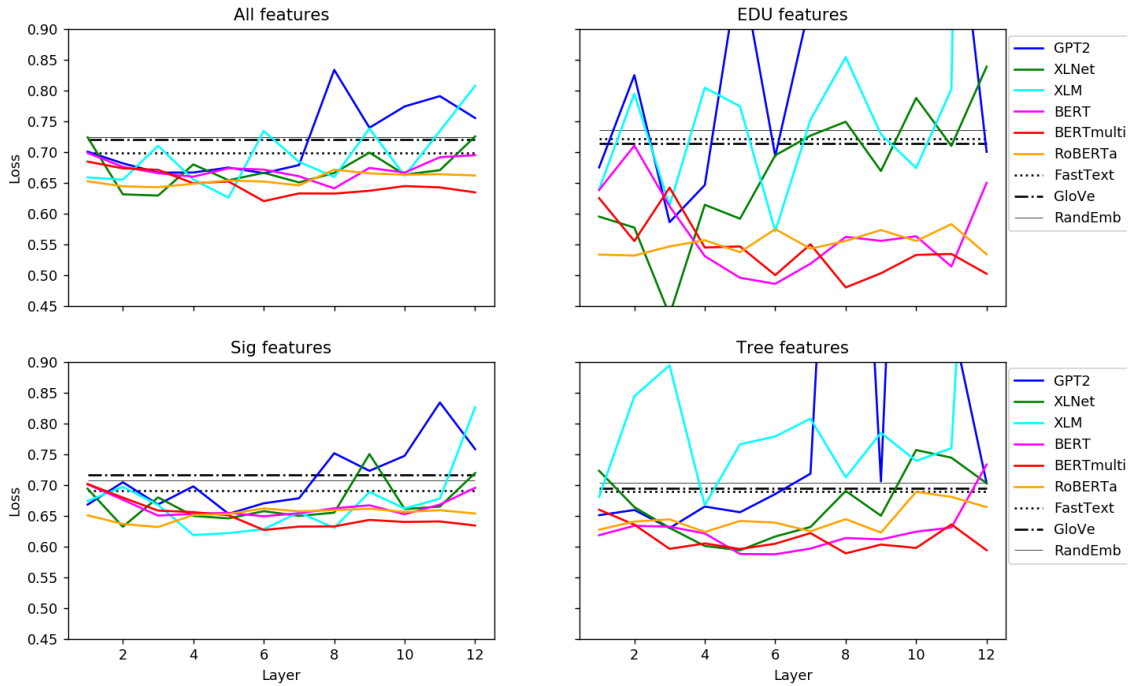


Figure 4: Probing loss, compared to those from non-contextualized baselines, for four feature groups, on IMDB. BERT-based neural LMs stably outperform the word embedding baselines in almost all layers.

- XLM (Lample and Conneau, 2019) is 12-layer Transformer with 2048-hidden dimensions. We use the English model trained with masked language model (MLM) objective. Different from BERT (taking sentence pairs as input), XLM takes continuous streams of tokens as input.
- XLNet (Yang et al., 2019) is a 12-layer Transformer-XL (Dai et al., 2019) with two streams of self attention and 768 hidden dimensions and 110M parameters. The XLNet we use is trained on 33GB texts using the “permutation language modeling” objective, with its LM factorization according to shuffled orders, but its positional encoding correspond to the original sequence order. The permutation LM objective introduces diversity and randomness to the context.

To make comparisons between models fair, we limit to 12-layer neural LMs. The models are pre-trained by Huggingface (Wolf et al., 2019).

### 3.3 Implementation

We formulated probing as an optimization problem, and implemented our solution with PyTorch

(Paszke et al., 2019) and the Adam optimizer (Kingma and Ba, 2014) for 40 epochs. If the training loss stalls (i.e., does not change by  $\geq 10^{-3}$ ), or if the training loss rises by more than 10% from the previous epoch, we stop the optimization. All optimizations follow the same learning rate tuning schemas.

In our experiments, the representation dimension  $d$  is taken to be 10, while the LM dimensions  $D$  is 2048 for XLM and 768 for the rest.

## 4 Results and Discussion

### 4.1 Where do LMs encode RST features?

From Figure 3, neural LMs encode RST features in different manners, depending on their structures. In general, for BERT-based models, features seem to distribute evenly across layers. On GPT-2 and XLNet, lower layers seem to encode slightly more EDU and Sig features than higher levels, whereas Tree features seem to be more concentrated in layers 2-6. The results on XLM are relatively noisy, possibly because the uni-language version does not benefit from the performance boost of cross-language modeling.

Contrasting with previous work that suggested



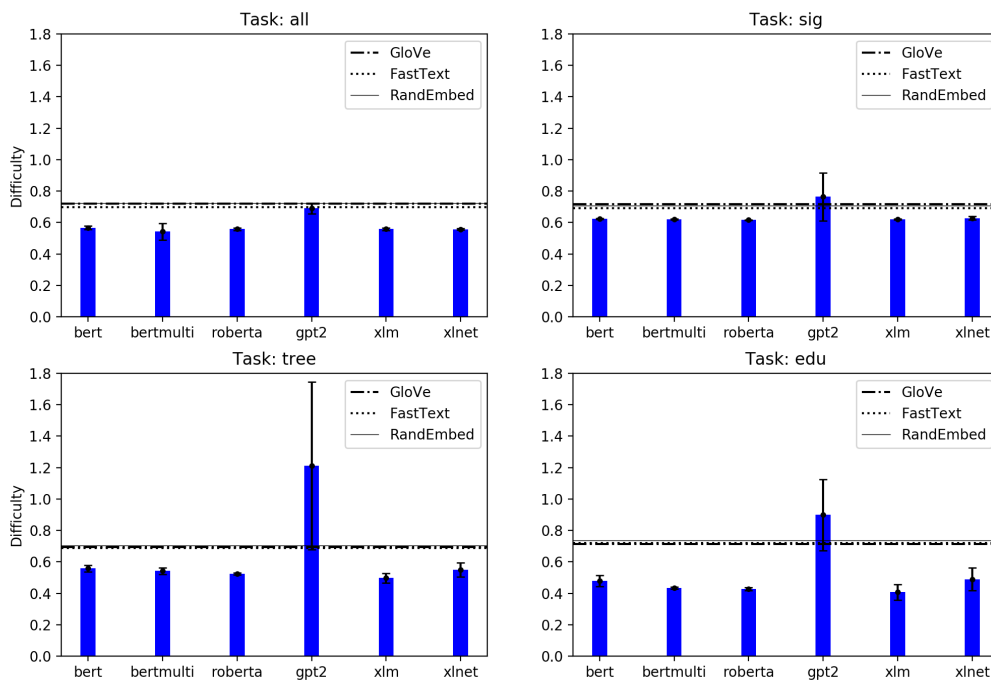


Figure 5: Probing performances of averaging 12 layers for 6 neural LMs on 4 tasks in IMDB, compared to the three non-contextual baselines. All LMs except GPT-2 outperform non-contextual LM baselines. Plots for RST-DT (Figure 6 in Appendix) reveal similar patterns.

that middle layers most contain syntactic features (Hewitt and Manning, 2019; Jawahar et al., 2019), our results indicate a less definitive localization for discourse features, except for the first and final layers. We suggest that the reason they encode less discourse information is that the first layer focuses on connections between “locations”, while the final layer focuses on extracting representations most relevant to the final task.

**Are RST features equally hard to probe?** Figure 3 also shows the difficulty in probing features across feature sets. In BERT-based models, `EDU` and `Tree` features are comparably easier to probe, whereas the `Sig` feature groups is more challenging. However, GPT-2, XLNet, and XLM do not regard `EDU` or `Tree` features easier to probe than other groups. Nevertheless, the results on all features correlate more to the `Sig` features.

**How about averaging layers?** For comparison, we also used the mean of all 12 layers for each neural LM. Figure 5 shows the probing results. Except GPT-2, other LMs show similar performances when the representations of layers are averaged. In

addition, the performances show that `Sig` features are harder to probe than `Tree` and `EDU` features, whereas the aggregation task (using all features) appears harder than each of its three component feature groups.

## 4.2 Deconstructing the probe

We perform ablation studies to illustrate the effectiveness of probing, deconstructing the language model probe step-by-step. First, we get rid of the contextualization component in language modeling by using non-contextualized word embeddings, GloVe and FastText. Then, we discard the semantic component of word embedding by mapping tokens to randomly generated vectors (RandEmbed). Finally, we remove all information pertaining to the text, leading to a random predictor for RST features, RandGuess.

**Non-contextualized word embeddings** We consider two popular word embeddings here:

- GloVe (Pennington et al., 2014) contains 2.2M vocabulary items and produces 300-dimensional word vectors. The GloVe embedding we use is pretrained on Common Crawl.

- FastText (Bojanowski et al., 2017) is trained on Wikipedia 2017 + UMBC (16B tokens) including subword information, and produces 300-dimensional word vectors.

Word embeddings map each token into a  $D$ -dimensional semantic space. Therefore, for a document of length  $L$ , the embedded matrix also has shape  $L \times D$ . The difference from the contextualized neural LMs is that, the  $D$ -dimensional vectors of every word do not depend on their contexts.

**Random embeddings** In this step, we assign a non-trainable random embedding vector per token in the vocabulary. This removes the semantic information encoded by GloVe and FastText word embeddings.

As shown in Figure 4, 5: RandEmbed is worse than GloVe and FastText (except for GloVe in `Sig` features task). This verifies some semantic information is preserved in word embeddings.

**Contextualized LMs against baseline** First, the lack of context restrict the probing performance of non-contextualized baselines. They are worse than most layers in contextualized LMs (in Figure 4), and are worse than all except GPT-2 if we average the layers (in Figure 5).

Second, it is impossible for any LM to have a “negative” rhetorical capacity. If the probing loss is worse than RandEmbed baseline, that means the RST probe can not detect rhetorical features of the given category encoded in the representations. This is what happens in some layers of GPT-2, XLM, and XLNet, and the mean of all layers of GPT-2.

**Random guesser** To measure the capacity of baseline embeddings, we set up a random guesser as a “baseline-of-baseline”. The random guesser outputs the arithmetic mean of RST features plus a small Gaussian noise (with s.d.  $\sigma \in \{0, 0.01, 0.1, 1.0\}$ ) The output of RandGuess is completely independent of the discourse. As shown in Table 2, the best of the four random guessers is much worse than any of the three word embedding baselines, which is expected.

### 4.3 Why are some LMs better?

From probing experiments (Figure 3, 4, and 5) we can see that BERT-based LMs have slightly better rhetorical capacities than XLNet, and much better capacities than GPT-2. We present two hypotheses as following.

| Config    | RST Feature Set |       |       |        |
|-----------|-----------------|-------|-------|--------|
|           | All             | EDU   | Sig   | Tree   |
| FastText  | .6987           | .7215 | .6911 | .6889  |
| GloVe     | .7204           | .7142 | .7166 | .6942  |
| RandEmbed | .7238           | .7365 | .7077 | .7034  |
| RandGuess | 1101.5          | 128.9 | 3.1   | 6799.0 |

Table 2: Comparison between RST probing losses of non-contextual word embeddings (FastText, GloVe), random embedding (RandEmbed), and a trivial guessor (RandGuess).

### Rhetorics favor contexts from both directions

BERT-based LMs use Transformer encoders, whereas GPT-2 use Transformer decoders. Their main difference is that a Transformer encoder considers contexts from both “past” and “future”, while a Transformer decoder only conditions on the context from the “past” (Vaswani et al., 2017). GPT-2 attends to uni-directional contexts. Apparently both the “past” and “future” context would contribute to the rhetorical features of words. Without “future” contexts, GPT-2 would encode less rhetorical information.

### Random permutation makes encoded rhetorics harder to decode

The difference between XLNet and other LMs is the permutation in context. While permutation increases the diversity in discourse, they could also bring in new meaning to the texts. For example, the sentence in Figure 1 (“I didn’t know this is from C, but it is very good!”) has several syntactically plausible factorization sequences:

- I didn’t know C ...
- ... this is C ...
- I know it is very good ...
- I didn’t know this is good ...
- ... didn’t this C good ...

Apparently such diversity in contexts makes the upper layers of XLNet contain harder-to-decode rhetorical features. If we average the representations of all layers, XLNet has larger variance than BERT-based LMs. We hypothesize that larger layer-wise difference is a factor of such instability for averaged representations.

### 4.4 Limitations

RST probing is not perfect. While we designed our comparisons to be rigorous, there are still several limitations to the RST probe, described below.

- RST signals are noisy. The RST relation classification task is less defined than established tasks like POS tagging. Humans tend to disagree with the annotators, resulting in a merely 65.8% accuracy in relation classification (i.e., the task introduced by [Marcu \(2000\)](#)). Regardless, state-of-the-art discourse parsers currently have performances slightly higher than 60% ([Feng and Hirst, 2014](#); [Ji and Eisenstein, 2014](#); [Wang et al., 2017](#)).
- Train / test corpus discrepancy of RST parsers. Most available RST parsers are trained on RST-DT consisting of Wall Street Journal articles. The results of parsers are affected by the corpus. As shown in some examples in Appendix, the IMDB movie review dataset contains less formal languages, introducing noise in segmentations and relation signals. To counteract noise of this type, we recommend evaluating LMs using a corpus similar to the scenario of applying the LM.
- Only 12-layer LMs are involved, to compare across various layers fairly. But our approach would be applicable to 3-layer ELMO and deeper LMs as well. Appropriate statistical controls would naturally need to be applied.
- Not all documents can be analyzed. First, documents longer than 512 tokens cannot be encoded into one vector in our probing model. Second, while RST provides elegant frameworks for analyzing rhetorical structures of discourse, in practice, the RST pipeline does not guarantee a successful analysis for an arbitrary document scraped online.

## 5 Related work

Recent work has considered the interpretability of contextualized representations. For example, [Jain and Wallace \(2019\)](#) found attention to be uncorrelated to gradient-based feature importance, while [Wiegrefe and Pinter \(2019\)](#) suggested such approaches allowed too much flexibility to give convincing results. Similarly, [Serrano et al. \(2019\)](#) considered attention representations to be noisy indicators of feature importance.

Many tasks in argument mining, similar to our task of examining neural LMs, require understanding the rhetorical aspects of discourse ([Lawrence and Reed, 2019](#)). This allows RST to be applied in relevant work. For example, RST enables understanding and analyzing argument struc-

tures of monologues ([Peldszus and Stede, 2016](#)) and, when used with other discourse features, RST can improve role-labelling in online arguments ([Chakrabarty et al., 2019](#)).

Probing neural LMs is an emergent diagnostic task on those models. Previous work probed morphological ([Bisazza and Tump, 2018](#)), agreement ([Giulianelli et al., 2018](#)), and syntactic features ([Hewitt and Manning, 2019](#)). [Hewitt and Liang \(2019\)](#) compared different probes, and recommended linear probes with as few parameters as possible, for the purpose of reducing overfitting. Recently, [Pimentel et al. \(2020\)](#) argued against this choice from an information-theoretic point of view. [Voita and Titov \(2020\)](#) presents an optimization goal for probes based on minimum description length.

[Liu et al. \(2019a\)](#) proposed 16 diverse probing tasks on top of contextualized LMs including token labeling (e.g., PoS), segmentation (e.g., NER, grammatical error detection) and pairwise relations. While LMs augmented with a probing layer could reach state-of-the-art performance on many tasks, they found that LMs still lacked fine-grained linguistic knowledge. DiscoEval ([Chen et al., 2019](#)) showed that BERT outperformed traditional pre-trained sentence encoders in encoding discourse coherence features, which our results echo.

## 6 Conclusion

In this paper, we propose a method to quantitatively analyze the amount of rhetorical information encoded in neural language models. We compute features based on Rhetorical Structure Theory (RST) and probe the RST features from contextualized representations of neural LMs. Among six popular neural LMs, we find that contextualization helps to generally improve the rhetorical capacities of LMs, while individual models may vary in quality. In general, LMs attending to contexts from both directions (BERT-based) encode rhetorical knowledge in a more stable manner than those using unidirectional contexts (GPT-2) or permuted contexts (XLNet).

Our method presents an avenue towards quantitatively describing rhetorical capacities of neural language models based on unlabeled, target-domain corpus. This method may be used for selecting suitable LMs in tasks including rhetorical acts classifications, discourse modeling, and response generation.



## Acknowledgement

We thank the anonymous reviewers for feedback. Rudzicz is supported by a CIFAR Chair in artificial intelligence. Abdalla is supported by a Vanier scholarship.

## References

- Mohamed Abdalla, Frank Rudzicz, and Graeme Hirst. 2017. Rhetorical structure and Alzheimer’s disease. *Aphasiology*, 32(1):41–60.
- Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *ICLR*, Toulon, France.
- Guillaume Alain and Yoshua Bengio. 2017. Understanding intermediate layers using linear classifier probes. In *ICLR*, Toulon, France.
- Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do Neural Machine Translation Models Learn about Morphology? In *ACL*, pages 861–872, Vancouver, Canada. Association for Computational Linguistics.
- Arianna Bisazza and Clara Tump. 2018. The lazy encoder: A fine-grained analysis of the role of morphology in neural machine translation. In *EMNLP*, pages 2871–2876, Brussels, Belgium. Association for Computational Linguistics.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *TACL*, 5:135–146.
- Lynn Carlson, Daniel Marcu, and Mary Ellen Okurovski. 2001. Building a Discourse-Tagged Corpus in the Framework of Rhetorical Structure Theory. In *SIGDIAL Workshop*.
- Tuhin Chakrabarty, Christopher Hidey, Smaranda Muresan, Kathy McKeown, and Alyssa Hwang. 2019. AMPERSAND: Argument Mining for PER-SuAsive oNline Discussions. In *EMNLP*, pages 2933–2943, Hong Kong, China. Association for Computational Linguistics.
- Mingda Chen, Zewei Chu, and Kevin Gimpel. 2019. Evaluation Benchmarks and Learning Criteria for Discourse-Aware Sentence Representations. In *EMNLP*, pages 649–662, Hong Kong, China. Association for Computational Linguistics.
- Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. In *ACL*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*. Association for Computational Linguistics.
- Vanessa Wei Feng and Graeme Hirst. 2012. Text-level Discourse Parsing with Rich Linguistic Features. In *ACL*, pages 60–68, Jeju Island, Korea. Association for Computational Linguistics.
- Vanessa Wei Feng and Graeme Hirst. 2014. A Linear-Time Bottom-Up Discourse Parser with Constraints and Post-Editing. In *ACL*, pages 511–521, Baltimore, Maryland. Association for Computational Linguistics.
- Vanessa Wei Feng, Ziheng Lin, and Graeme Hirst. 2014. The Impact of Deep Hierarchical Discourse Structures in the Evaluation of Text Coherence. In *COLING*, pages 940–949, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.
- Mario Giulianelli, Jack Harding, Florian Mohnert, Dieuwke Hupkes, and Willem Zuidema. 2018. Under the hood: Using diagnostic classifiers to investigate and improve how language models track agreement information. In *EMNLP BlackBoxNLP*, pages 240–248, Brussels, Belgium. Association for Computational Linguistics.
- Michael Heilman and Kenji Sagae. 2015. Fast rhetorical structure theory discourse parsing. *arXiv preprint 1505.02425*.
- John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. In *EMNLP*, pages 2733–2743, Hong Kong, China. Association for Computational Linguistics.
- John Hewitt and Christopher D Manning. 2019. A Structural Probe for Finding Syntax in Word Representations. In *NAACL*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.
- Robert Irish and Peter Eliot Weiss. 2009. *Engineering communication: from principles to practice*. Oxford University Press Canada.
- Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. In *NAACL*, pages 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.
- Ganesh Jawahar, Benoît Sagot, and Djamel Seddah. 2019. What Does BERT Learn about the Structure of Language? In *ACL*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.
- Yangfeng Ji and Jacob Eisenstein. 2014. Representation learning for text-level discourse parsing. In *ACL*, pages 13–24, Baltimore, Maryland. Association for Computational Linguistics.

- Shafiq Joty, Giuseppe Carenini, and Raymond T Ng. 2015. **CODRA: A Novel Discriminative Framework for Rhetorical Analysis**. *Computational Linguistics*, 41(3):385–435.
- Nitish Shirish Keskar, Bryan McCann, Lav Varshney, Caiming Xiong, and Richard Socher. 2019. **CTRL - A Conditional Transformer Language Model for Controllable Generation**. *arXiv preprint 1909.05858*.
- Diederik Kingma and Jimmy Ba. 2014. **Adam: A method for stochastic optimization**. In *ICLR*, Banff, Canada.
- Guillaume Lample and Alexis Conneau. 2019. **Cross-lingual Language Model Pretraining**. *arXiv preprint arXiv:1901.07291*.
- John Lawrence and Chris Reed. 2019. **Argument Mining: A Survey**. *Computational Linguistics*, pages 1–54.
- Bai Li, Yi-Te Hsu, and Frank Rudzicz. 2019. **Detecting dementia in Mandarin Chinese using transfer learning from a parallel corpus**. In *NAACL*, pages 1991–1997, Minneapolis, Minnesota. Association for Computational Linguistics.
- Qi Li, Tianshi Li, and Baobao Chang. 2016. **Discourse Parsing with Attention-based Hierarchical Neural Networks**. In *EMNLP*, pages 362–371, Austin, Texas. Association for Computational Linguistics.
- Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. **A structured self-attentive sentence embedding**. In *ICLR*, Toulon, France.
- Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. **Linguistic Knowledge and Transferability of Contextual Representations**. pages 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.
- Sha Liu and Antony Kunnan. 2016. **Investigating the Application of Automated Writing Evaluation to Chinese Undergraduate English Majors: A Case Study of WriteToLearn**. *CALICO*, 33(1):71–91.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. **RoBERTa: A Robustly Optimized BERT Pretraining Approach**. *arXiv preprint 1907.11692*.
- Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. **Learning Word Vectors for Sentiment Analysis**. In *ACL*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.
- William C Mann, Christian M I M Matthiessen, and Sandra A Thompson. 1989. **Rhetorical structure theory and text analysis**. *ISI Research Report*.
- William C. Mann and Sandra A. Thompson. 1988. **Rhetorical Structure Theory: Toward a functional theory of text organization**. *Interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.
- Daniel Marcu. 2000. *The Theory and Practice of Discourse Parsing and Summarization*. MIT Press, Cambridge, MA, USA.
- Mohammed Ali Mohsen and Abdulaziz Alshahrani. 2019. **The Effectiveness of Using a Hybrid Mode of Automated Writing Evaluation System on EFL Students’ Writing**. *Teaching English with Technology*, 19(1):118–131.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. **PyTorch: An Imperative Style, High-Performance Deep Learning Library**. In *NeurIPS*.
- Andreas Peldszus and Manfred Stede. 2016. **Rhetorical structure and argumentation structure in monologue text**. In *ArgMining Workshop*, pages 103–112, Berlin, Germany. Association for Computational Linguistics.
- Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. **GloVe: Global Vectors for Word Representation**. In *EMNLP*, pages 1532–1543, Doha, Qatar.
- Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. 2020. **Information-Theoretic Probing for Linguistic Structure**. Association of Computational Linguistics.
- Donald E Powers, Jill C Burstein, Martin Chodorow, Mary E Fowles, and Karen Kukich. 2001. **Stumping E-Rater: Challenging the Validity of Automated Essay Scoring**. Technical report, Educational Testing Service, Princeton, New Jersey.
- Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltasakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. **The Penn Discourse TreeBank 2.0**. In *LREC*, Marrakech, Morocco. European Language Resources Association (ELRA).
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. **Language models are unsupervised multitask learners**. *OpenAI Blog*, 1(8).
- Abigail See, Aneesh Pappu, Rohun Saxena, Akhila Yerukola, and Christopher D Manning. 2019. **Do Massively Pretrained Language Models Make Better Storytellers?** In *CoNLL*, pages 843–861, Hong Kong, China. Association for Computational Linguistics.

- Sofia Serrano, Noah A Smith, and Paul G Allen. 2019. [Is Attention Interpretable?](#) In *ACL*, pages 2931–2951, Minneapolis, Minnesota. Association for Computational Linguistics.
- Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. [MASS: Masked Sequence to Sequence Pre-training for Language Generation](#). In *ICML*, Long Beach, California.
- Mihai Surdeanu, Thomas Hicks, and Marco A Valenzuela-Escárcega. 2015. [Two Practical Rhetorical Structure Theory Parsers](#). In *NAACL*, Denver, Colorado.
- Stephen Toulmin. 1958. *The Uses of Argument*. Cambridge University Press.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. [Attention Is All You Need](#). In *NeurIPS*, Long Beach, California.
- Elena Voita and Ivan Titov. 2020. [Information-Theoretic Probing with Minimum Description Length](#). *arXiv preprint arXiv:2003.12298*.
- Xinhao Wang, Binod Gyawali, James V Bruno, Hillary R Molloy, Keelan Evanini, and Klaus Zechner. 2019. [Using Rhetorical Structure Theory to Assess Discourse Coherence for Non-native Spontaneous Speech](#). In *DisRPT*, pages 153–162, Minneapolis, Minnesota. Association for Computational Linguistics.
- Yizhong Wang, Sujian Li, and Houfeng Wang. 2017. [A Two-Stage Parsing Method for Text-Level Discourse Analysis](#). In *ACL*, pages 184–188, Vancouver, Canada. Association for Computational Linguistics.
- Sarah Wiegreffe and Yuval Pinter. 2019. [Attention is not not Explanation](#). In *EMNLP*, pages 11–20, Hong Kong, China. Association for Computational Linguistics.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R’emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. [HuggingFace’s Transformers: State-of-the-art Natural Language Processing](#). *arXiv preprint 1910.03771*.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. [XLNet: Generalized Autoregressive Pretraining for Language Understanding](#).
- Victor H Yngve. 1960. [A model and an hypothesis for language structure](#). *Proceedings of the American philosophical society*, 104(5):444–466.
- Nan Yu, Meishan Zhang, and Guohong Fu. 2018. [Transition-based neural RST parsing with implicit syntax features](#). In *COLING*, pages 559–570, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Mo Zhang. 2013. [Contrasting Automated and Human Scoring of Essays](#). Technical report, Educational Testing Service.
- Zining Zhu, Jekaterina Novikova, and Frank Rudzicz. 2019. [Detecting cognitive impairments by agreeing on interpretations on linguistic features](#). In *NAACL*, pages 1431–1441, Minneapolis, Minnesota. Association for Computational Linguistics.

## A Experiments on RST-DT

As a sanity check, we include experiments on RST-DT (Carlson et al., 2001) corpus with the same pre-processing and feature extraction procedures (i.e., perform feature extraction and embedding on the article level, and ignoring the overlength articles). As shown in Figure 6, BERT-family and XLM outperform GPT-2 and XLNet. Also, the noncontextualized embedding baselines show worse performances than contextualized embeddings in general, with some exceptions (e.g., GPT-2 on EDU features). These are similar to the IMDB results.

What are different is that the probing losses of RST-DT are lower than the IMDB experiments in general. We consider two possible explanations. First, the IMDB signals contain more noise, so that probing rhetorical features from IMDB would be naturally more difficult than probing from the RST-DT dataset. Second, it is possible that the probes overfit the much smaller RST-DT dataset.

## B Examples of parse trees

We include several examples of IMDB parse trees in Appendix here, including some examples where the RST parser makes mistakes on a new domain, movie review. For clarity of illustration, these examples are among the shorter movie reviews. More parse trees can be generated by our visualization code, which is contained in our submitted scripts.

### Averaging all layers on RST-DT

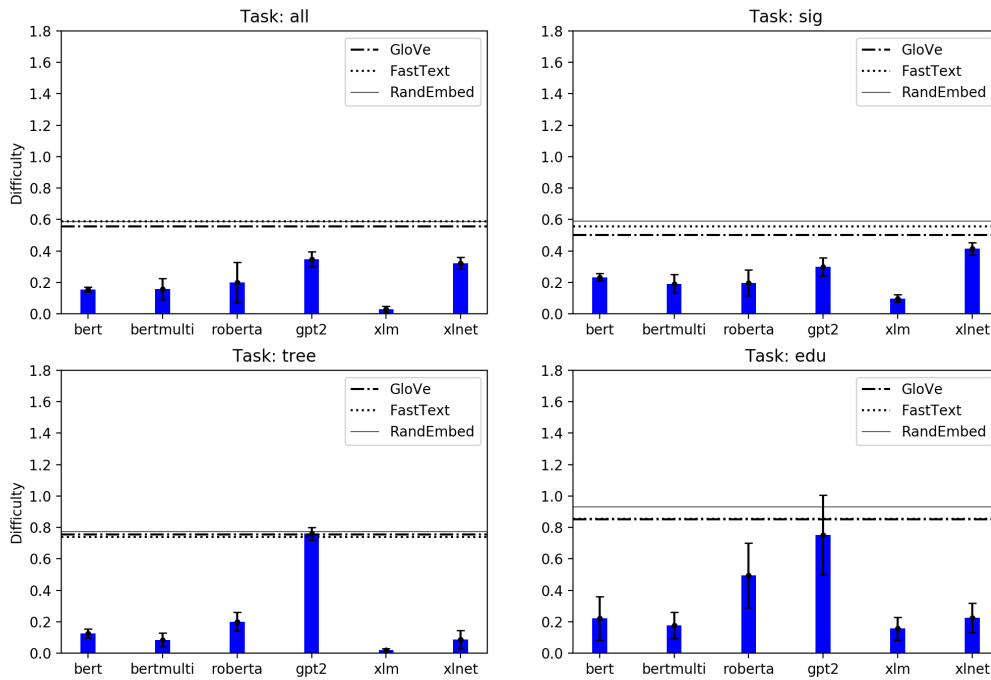


Figure 6: Probing performances of averaging 12 layers for 6 neural LMs on 4 tasks in RST-DT, compared to the three non-contextual baselines.

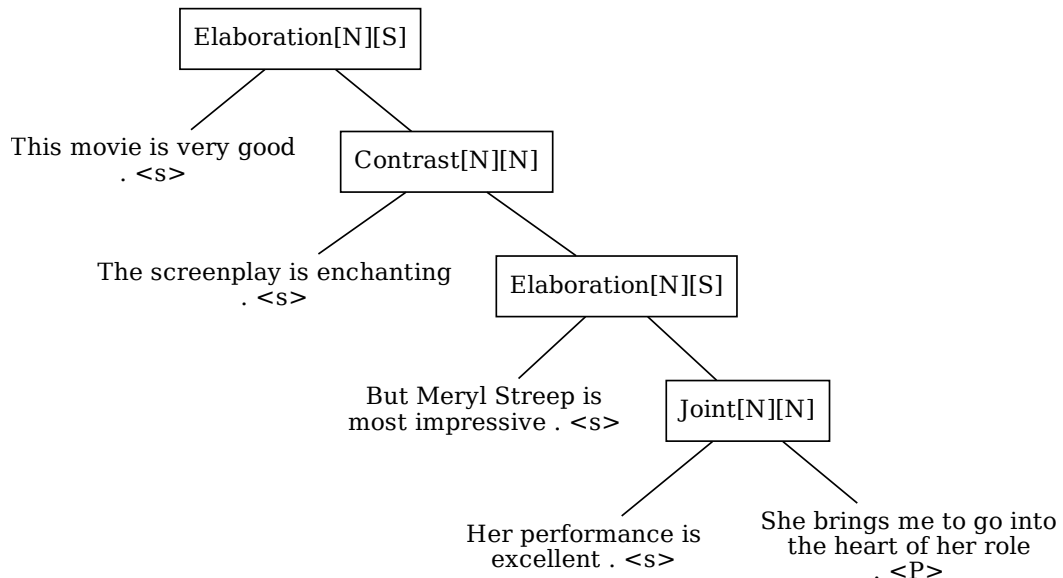


Figure 7: IMDB train/pos/10348\_8.txt. The <s> and <P> are appended automatically by the parser, marking the end of sentences and paragraphs respectively.



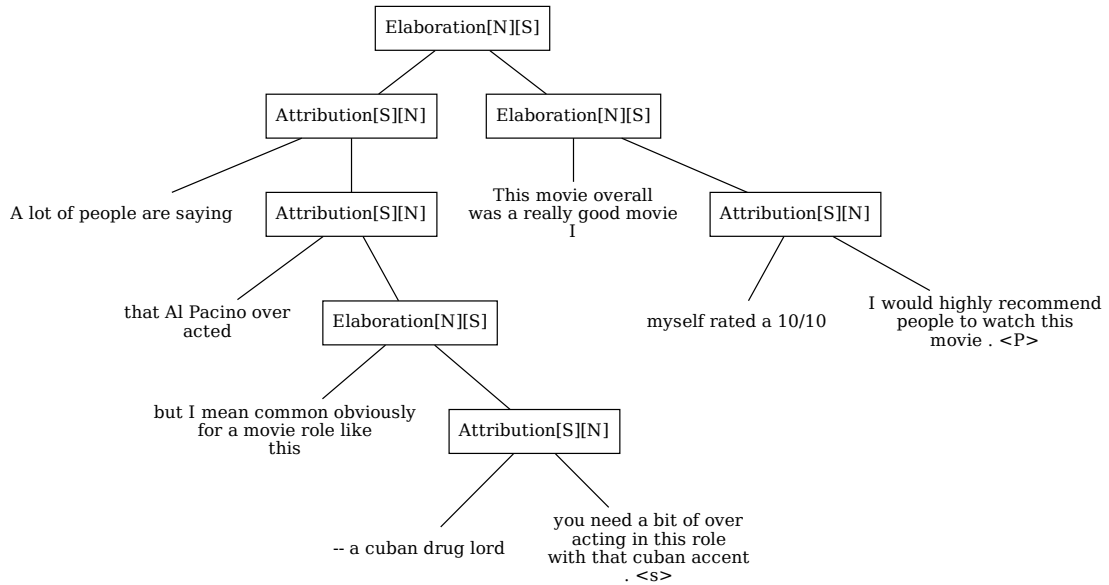


Figure 8: IMDB train/pos/11857\_10.txt. There is an EDU segmentation error: the “I” is incorrectly assigned to the previous sentence “This movie overall was a really good movie”. Apparently some lexical cues the EDU segmentator relies on (e.g., sentence finishes with a period sign) is not always followed in IMDB.

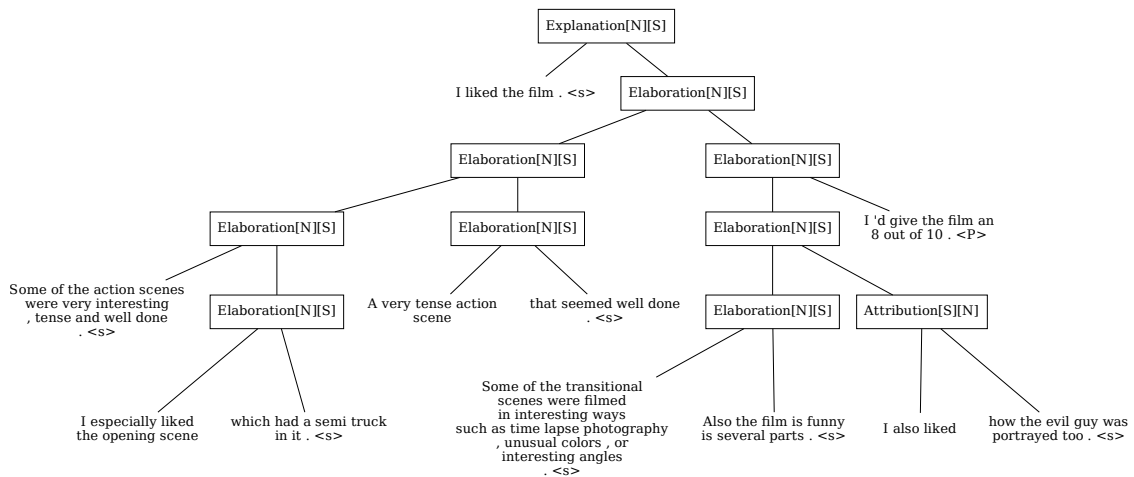


Figure 9: IMDB train/pos/1000\_8.txt. The parser captures the key sentence of this review. All sentences following the first one act as reasons to explain how the reviewer liked the film.

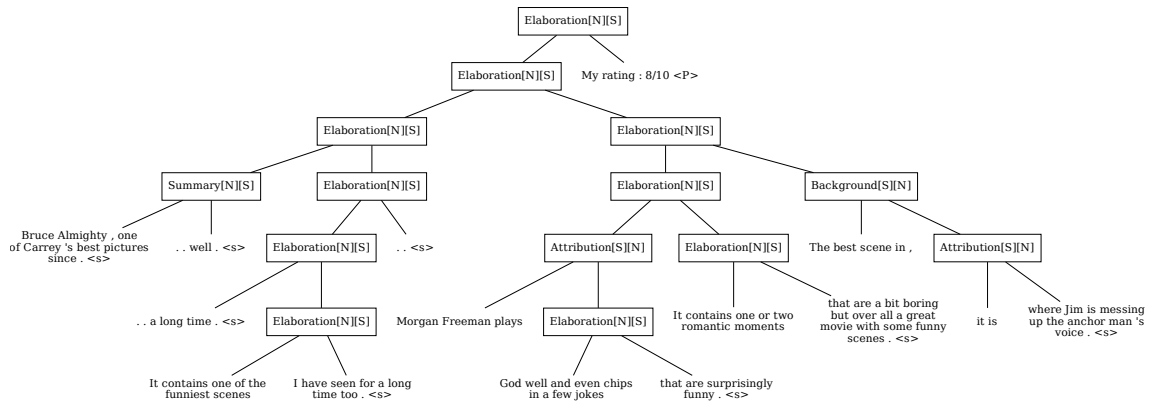


Figure 10: IMDB train/pos/10301-8.txt. The interjection, “well”, is incorrectly identified as the satellite of the summary signal. This is likely caused by the discrepancy between the train (RST-DT) and test (IMDB) corpus discrepancy for the RST parser. The RST-DT dataset contains news articles, which are more formal than the online review in IMDB. The term “well” is therefore more likely to be identified as other senses.

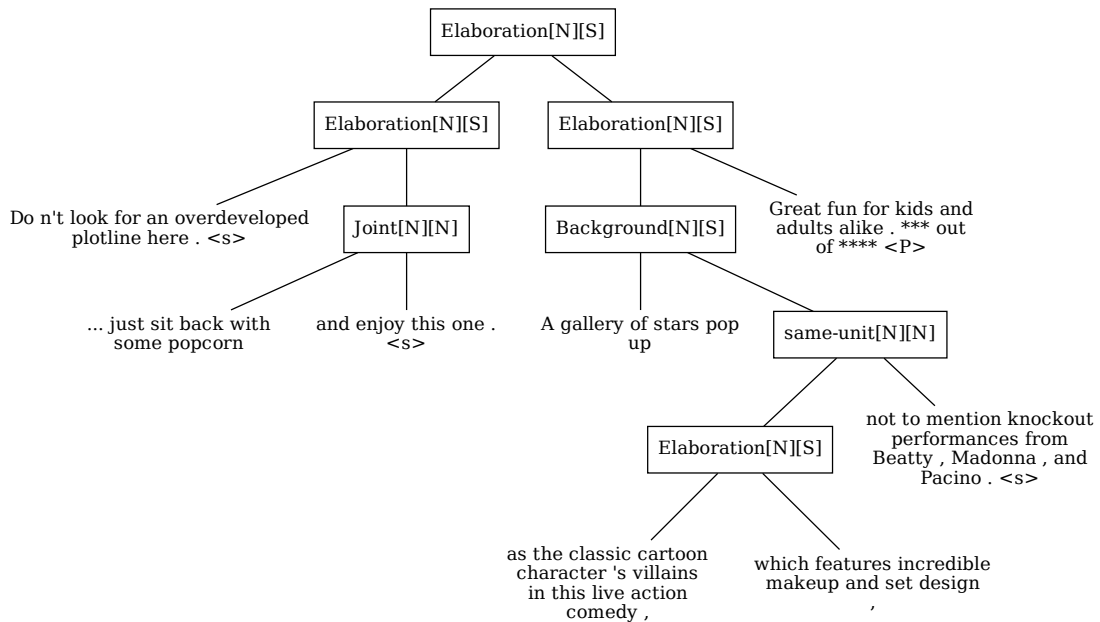


Figure 11: IMDB train/pos/11825-8.txt. One might suggest that the last EDU could be moved one level higher (so that it summarizes the whole review), but this parsing is also reasonable, since the mention of kids elaborates the descriptions of the makeup and the views.

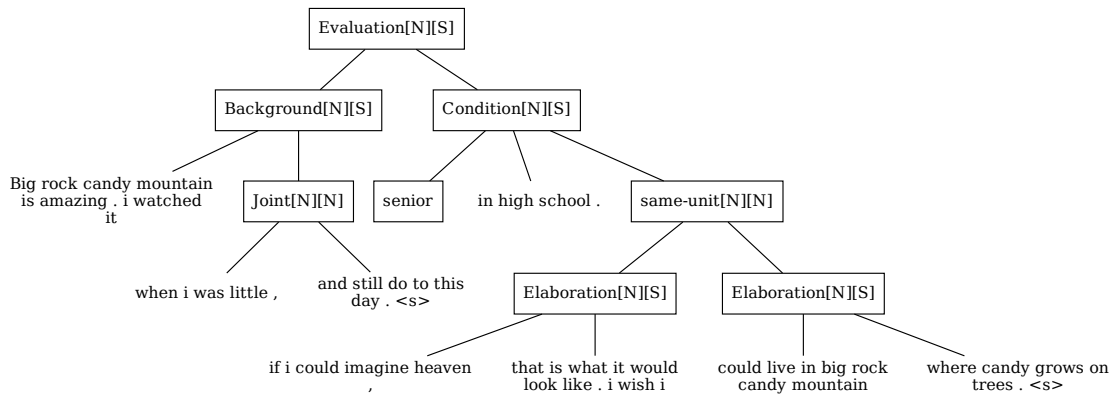


Figure 12: IMDB train/pos/10788-10.txt. This is an example of the EDU segmentation contains mistake. The “i wish i” should be merged with the subsequent EDU, “could live in big rock candy mountain”. Note that the sentence starts with two lowercase “i” (which should be uppercase). The non-standard usages like these are unique for less formal texts like IMDB.

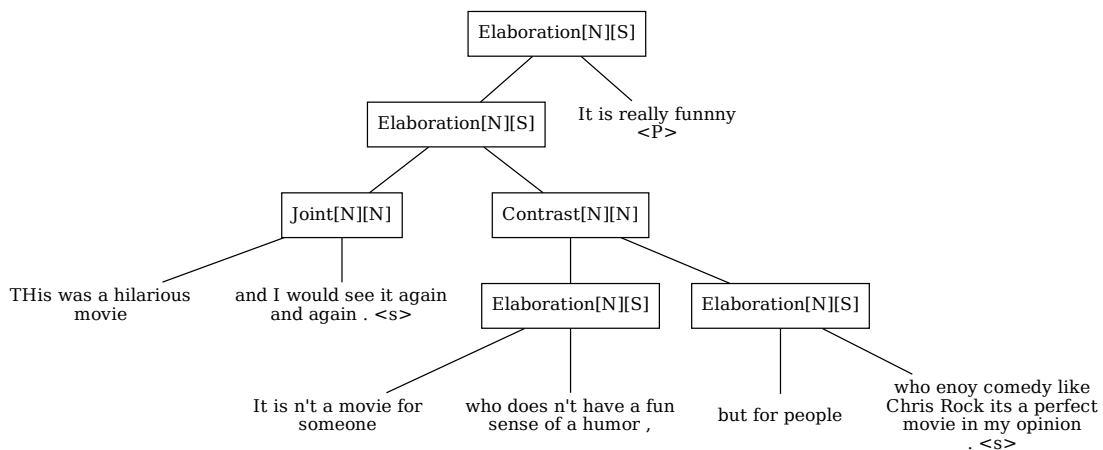


Figure 13: IMDB train/pos/11686-10.txt.

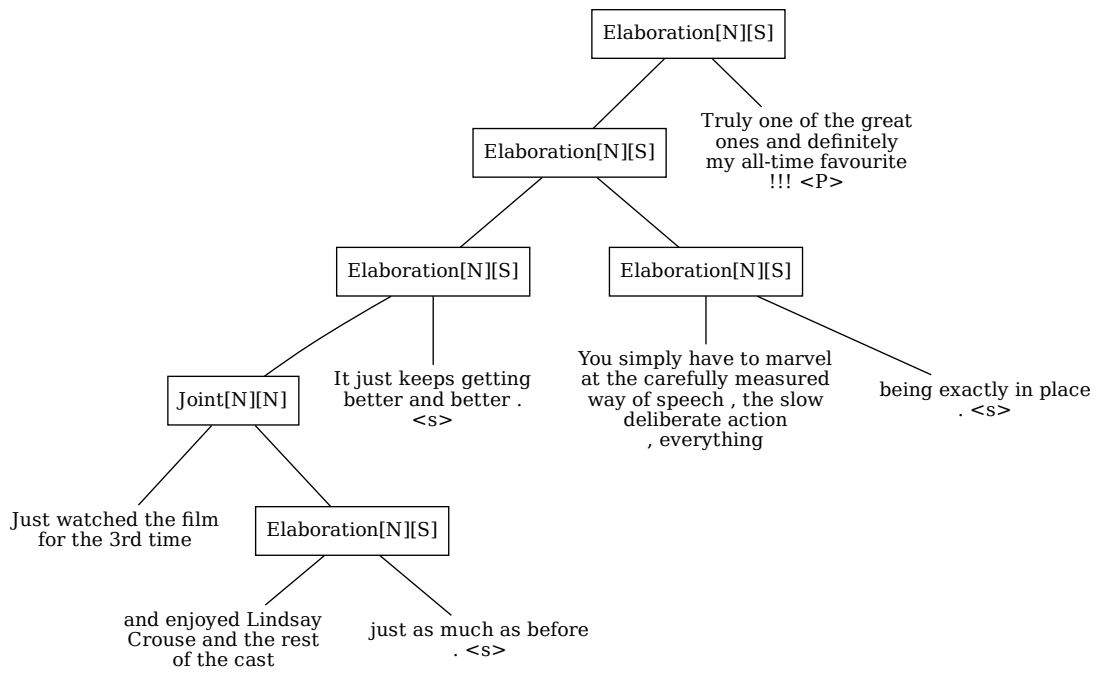


Figure 14: IMDB train/pos/10264-10.txt