

## 1 A The Effects of Decision Tree Algorithm

2 In order to evaluate the effect of the decision tree algorithm  
 3 to generate the tree for PEACH, we tested several decision  
 4 tree algorithms, including ID3, C4.5, CART, and those with  
 5 Random Forest. Table 3 shows that the overall performance  
 6 between different decision tree algorithms is very similar to  
 7 each other. Generally, the Random forests worked better than  
 8 single trees most of the time, except for SST2 and BBCNews.  
 9 However, the Random Forest is too crowded and too many  
 10 rules in the PEACH decision tree output. Hence, it is better  
 11 to visualise the decision-making path of the specific PLMs in  
 12 a single decision tree.

## 13 B Baseline and Dataset

14 Figure 1 illustrates the pretraining and fine-tuning step. For  
 15 PEACH with the fine-tuned PLMs, we compare ours with  
 16 the original PLMs. As our PEACH involves finetuning pre-  
 17 trained models to extract embedding features, we finetune a  
 18 linear classification layer based on the [CLS] token of the pre-  
 19 trained language models, and report the results on the fine-  
 20 tuned BERT, RoBERTa, ALBERT, XLNet and ELMo mod-  
 21 els to compare with our PEACH. Table 2 shows the detailed  
 22 statistics of the datasets we used for our PEACH experiments.

## 23 C Maximum Depth Analysis

24 To visualise the decision-making pattern clearly, we limit the  
 25 maximum depth of the generated tree. However, it is crucial  
 26 to keep the classification performance and behaviour simi-  
 27 lar to the original fine-tuned contextual embedding. Table 1  
 28 shows that increasing the tree depth from 3 to 15 has little  
 29 effect on binary datasets. However, for datasets with mul-  
 30 tiple classes, decreasing the maximum depth leads to a sig-  
 31 nificant decrease in performance. Insufficient rules to cover  
 32 all classes, especially at a depth of 3, greatly affect these  
 33 datasets. Adequate rule coverage is crucial for effectively  
 34 handling datasets with more classes.

Table 1: The effects of maximum depth on PEACH. The following tests are conducted on the best setting for each dataset. For MSRP, RoBERTa embedding and K-means clustering is applied to train the decision tree with different maximum depth limit; RoBERTa embedding with CNN is applied for MR; BERT embedding with K-means is applied for BBCNews and 20ng.

Depth	MSRP	MR	BBCNews	20ng
3	0.825	0.869	0.806	0.492
5	0.824	0.866	0.975	0.706
10	0.823	0.869	0.975	0.841
15	0.834	0.867	0.975	0.849

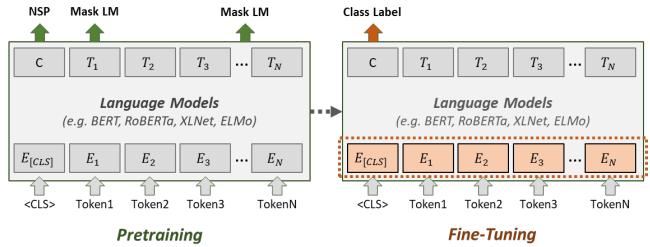


Figure 1: Architecture of pretrained model and fine-tuning process. We extracted the fine-tuned contextual embedding (orange-coloured) as an input of PEACH

BBC News, MR, and SST. Figures 7 and 8 show the global interpretation decision tree generated on BBC News, while Figures 9, 10 and 11, 12, 13, 14 presents the local explanation for each test document. The figure captions explain the detailed information for each figure. For SST, the global interpretations are in Figures 21 and 22, and the local explanations are in Figures 23 and 24. For MR, the global interpretations are in Figures 15 and 16, and the local explanations are in Figures 17, 18, 19, 20. The important points and remarkable patterns are described in the captions for each figure.

## 50 D Local Explanation Comparison on PLMs

In Section 5, we conducted the qualitative analysis of the interpretation and visualisation. In this Appendix, we would like to show some examples presented in 25 and 26 to compare how PEACH explained the best-performing PLM and the worst performing PLM for BBCNews and MR. The figure captions explain the detailed information for each figure.

## 57 E Application User Interface

As shown in Figure 6, we developed an interactive decision tree-based text classification decision-making interpretation system for different PLMs. The application would be helpful for anyone who would like to apply PLMs in their text-based prediction/classification tasks. The following describes the main components of the developed PEACH application. The PEACH application is developed by Python, CSS, and JQuery with the Flask environment. The application has four main components: 1) Dataset Navigator, 2) Parameter Visualisation Panel, 3) Decision Tree Visualisation, and 4) Visualisation Filter.

## 69 F Case Study: High-risk Text Classification

The results of our investigation into the feasibility and interoperability of our proposed visualisation approach, using a high-risk text classification dataset for disease detection, have provided valuable insights. To assess the performance of our approach, we employed fine-tuned versions of both Biomed-BERT and BERT on the Medical Report dataset, sourced from the HuggingFace library.

The Medical Report dataset<sup>1</sup>, comprising 2.83k instances

<sup>1</sup>[https://huggingface.co/datasets/IndianaUniversityDatasetsModels/Medical\\_reports\\_Splits](https://huggingface.co/datasets/IndianaUniversityDatasetsModels/Medical_reports_Splits)

<b>Method</b>	<b>MSRP</b>	<b>SST2</b>	<b>MR</b>	<b>IMDB</b>	<b>SICK</b>	<b>BBCNews</b>	<b>TREC</b>	<b>20ng</b>	<b>Ohsumed</b>
# Class	2	2	2	2	3	5	6	20	23
# Docs	5801	8741	10662	50000	9345	2225	5952	18846	7400
# Train Docs	4076(70.3%)	6920(79.2%)	7108(66.7%)	25000(50%)	4439(47.5%)	1225(55.1%)	5452(91.6%)	11314(60.0%)	3357(45.4%)
# Test Docs	1725(29.7%)	1821(20.8%)	3554(33.3%)	25000(50%)	4906(52.5%)	1000(44.9%)	500(8.4%)	7532(40.0%)	4043(54.6%)
# Words	17873	15481	18334	181061	2318	32772	8900	42106	14127
Avg Length	37.7	17.5	19.4	230.3	19.3	388.3	8.7	189.0	121.6
Task	NLI	SA	SA	SA	NLI	NC	QC	NC	TA

Table 2: Detailed Dataset Statistics

<b>Method</b>	<b>MSRP</b>	<b>SST2</b>	<b>MR</b>	<b>IMDB</b>	<b>SICK</b>	<b>BBCNews</b>	<b>TREC</b>	<b>20ng</b>	<b>Ohsumed</b>
<b>ID3</b>	0.797	0.931	0.860	0.874	0.848	0.969	0.958	0.815	0.561
<b>C4.5</b>	0.796	0.925	0.856	0.884	0.847	0.963	0.956	0.810	0.566
<b>CART</b>	0.784	<b>0.938</b>	0.856	0.871	0.851	<b>0.975</b>	0.960	0.813	0.561
<b>RF(ID3)</b>	0.809	0.935	0.868	0.891	0.872	0.970	0.968	0.845	<b>0.651</b>
<b>RF(C4.5)</b>	0.814	0.934	<b>0.872</b>	0.891	<b>0.877</b>	0.968	0.966	<b>0.849</b>	0.649
<b>RF(CART)</b>	<b>0.819</b>	0.936	0.864	<b>0.893</b>	0.870	0.963	<b>0.978</b>	0.841	0.640
<b>Rule number (best single tree)</b>	341	38	176	702	244	7	92	119	616
<b>Max depth (best one)</b>	22	11	17	30	25	5	9	12	11
<b>Max depth (best single tree)</b>	29	11	18	95	18	5	12	10	80
<b>Max depth (best forest)</b>	22	6	17	30	25	3	9	12	11
<b>Input dimension</b>	70	70	70	220	90	71	31	50	80
<b>Tree number</b>	5	1	5	5	5	1	5	10	10

Table 3: The effects of Decision Tree Algorithm

for training, 250 for validation, and 250 for testing, contains detailed findings and disease diagnoses from chest X-ray collections by Indiana University. We performed binary predictions to determine whether a patient has a disease or is in a normal state based on the finding description. PEACH is constructed for each fine-tuned model.

Our findings indicate that the BiomedBERT, which is specifically pre-trained with medical knowledge from PubMed, outperforms the general pre-trained BERT model, as can be seen in Table 4. Notably, BiomedBERT demonstrates superior performance, particularly when the document case is labelled as a disease class, as highlighted in Figure 2 and Figure 4.

The decision tree visualisations, as shown in Figure 2 and Figure 3 for the same case using BiomedBERT and BERT, reveal interesting insights. BiomedBERT’s decision tree provides an interpretable reasoning chain for classifying the example document. On the contrary, the decision tree visualisation for BERT struggles, with alignment spread across both classes, making it challenging to determine a clear and accurate classification. The same pattern can be found in Figure 4 and Figure 5. This discrepancy in visualisation using our model highlights the significance of utilising domain-specific pre-trained models, such as BiomedBERT, when dealing with medical text data. Our model reveals the importance of such interpretability and visualisation would help medical experts select pre-trained models tailored to the domain of interest to achieve optimal performance and meaningful interpretability in their text classification tasks

Table 4: Settings and results of the PEACH pipeline on the Medical Report dataset.

<b>Model</b>	<b>DT algorithm</b>	<b>Grouping Type</b>	<b>Acc</b>	<b>F1</b>
<b>BiomedBERT</b>	ID3	kmeans	<b>0.976</b>	<b>0.974</b>
<b>BERT</b>	ID3	kmeans	0.956	0.951

## H Human Evaluation Sample Case

Figure 27 shows a sample question for our human evaluation form.

107

108

109

## BiomedBERT

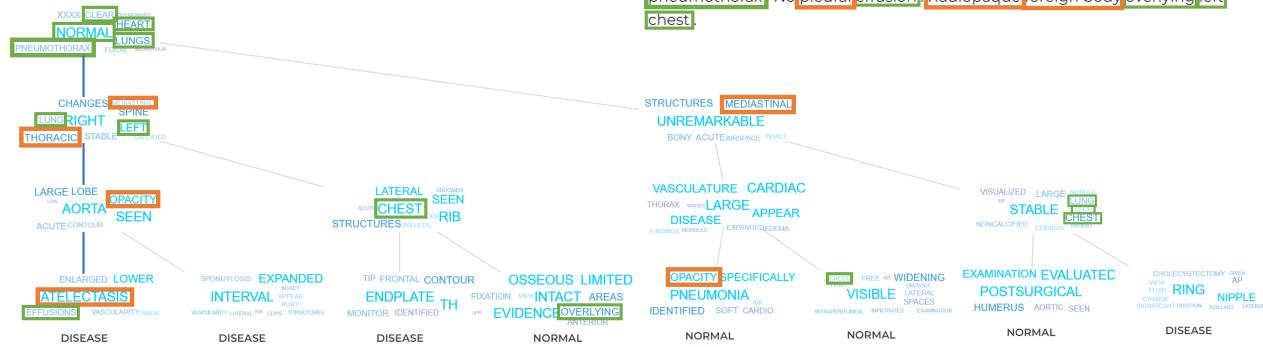


Figure 2: The local explanation decision trees generated for a clinical report **correctly** classified as **Disease** based on fine-tuned BiomedBERT embedding.

## BERT

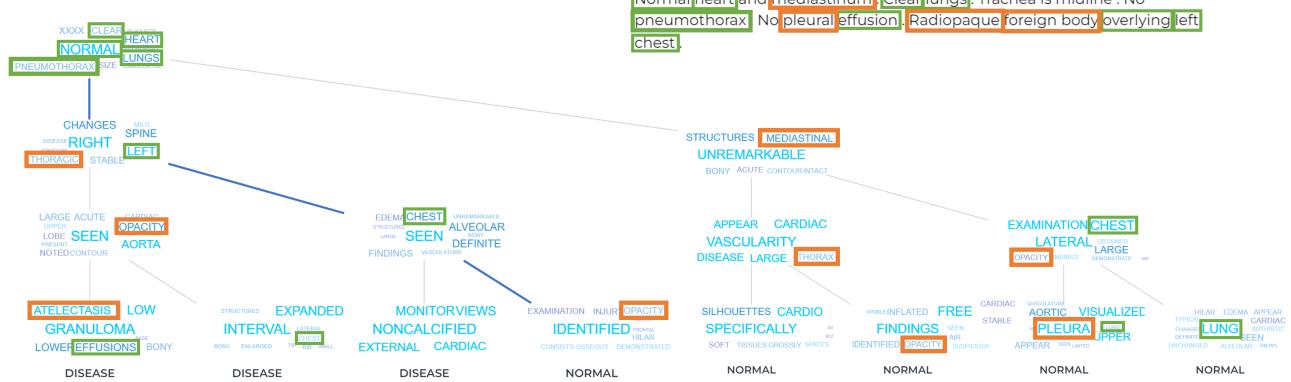


Figure 3: The local explanation decision trees generated for a clinical report **wrongly** classified as **Normal** based on fine-tuned BERT embedding.

## BiomedBERT

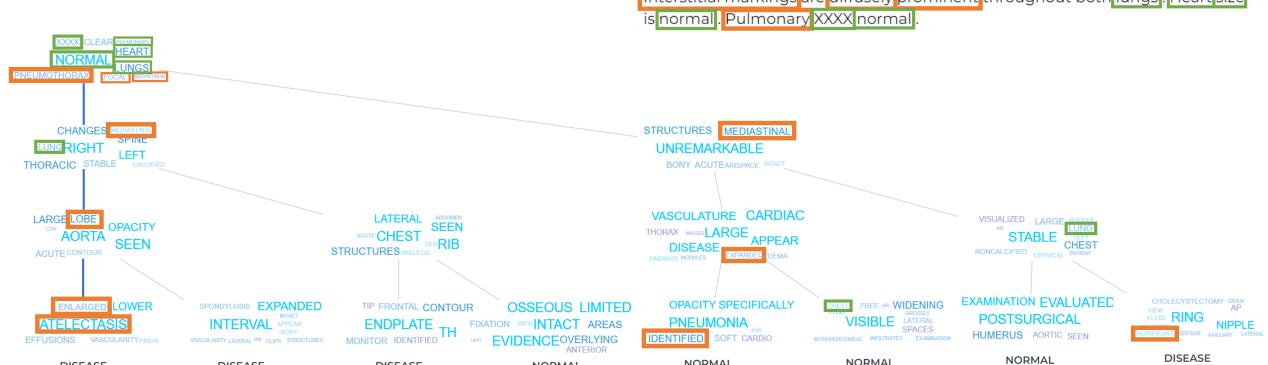


Figure 4: The local explanation decision trees generated for a clinical report **correctly** classified as **Disease** based on fine-tuned BiomedBERT embedding.

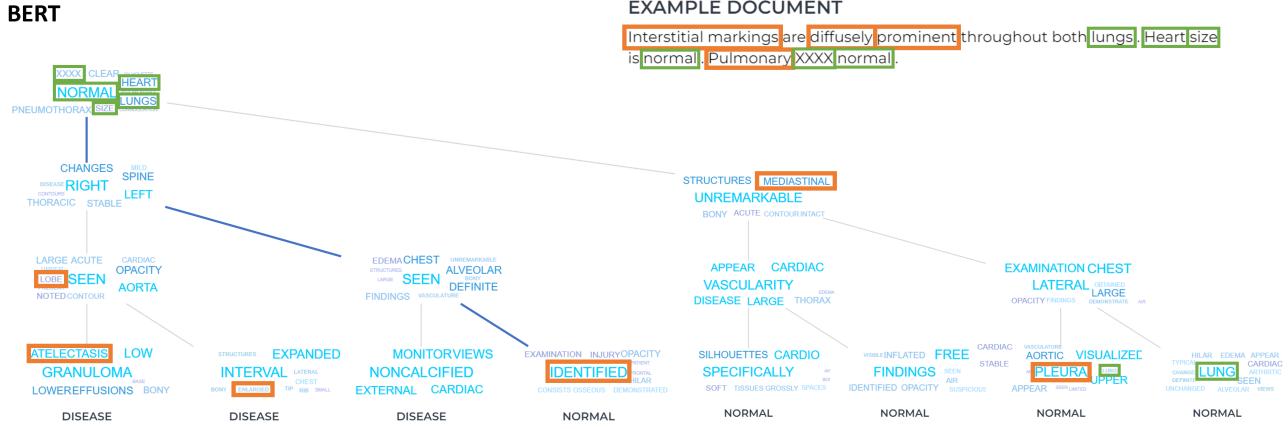


Figure 5: The local explanation decision trees generated for a clinical report **wrongly** classified as **Normal** based on fine-tuned BERT embedding.

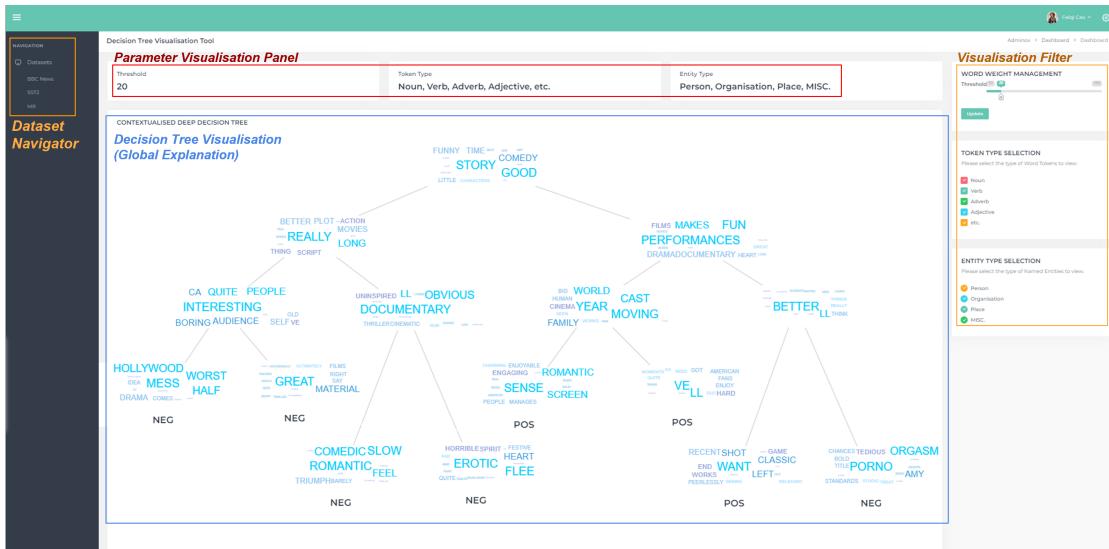


Figure 6: User Interface of the Application PEACH

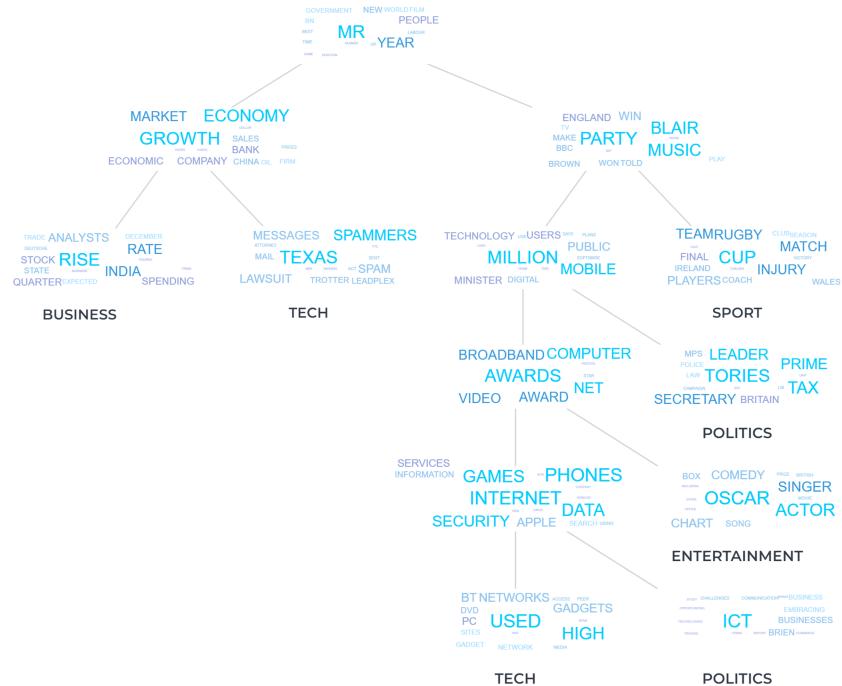


Figure 7: The global explanation decision tree generated on BBCNews dataset based on fine-tuned BERT embedding, where the prototype nodes show the decision path based on the global context for each class. We can observe that Technology news and Business news are grouped together in the first step of the reasoning, before further distinguishing between them. This is probably because there are quite some new articles related to technological corporates, making those two classes share some commonality in the concepts.

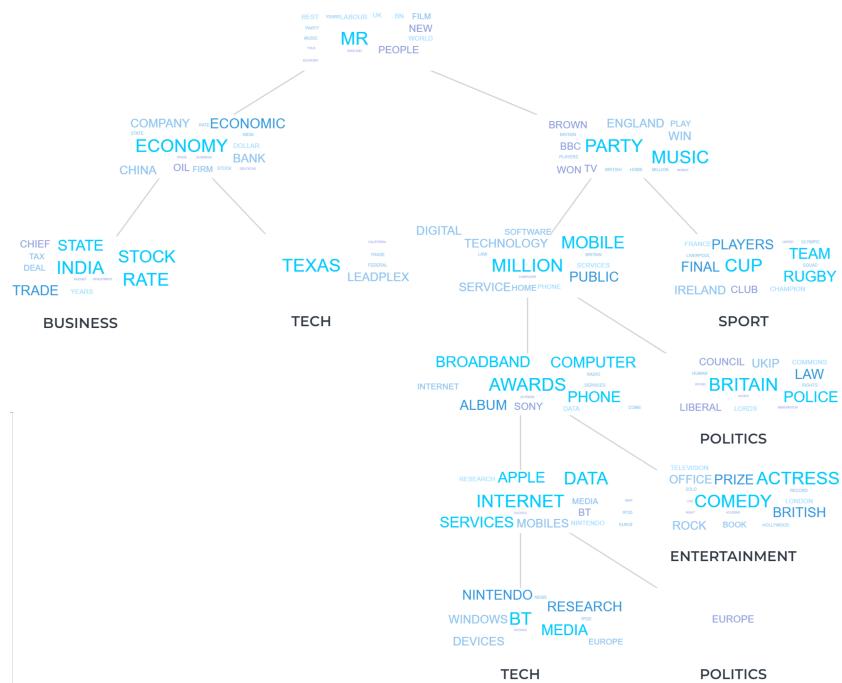


Figure 8: The global explanation decision tree generated on BBCNews dataset based on fine-tuned BERT embedding, with NER filters to display only words with NER tags for organizations and locations, labelled by spaCy library. We can see some country names, sports team names, and company names are coming out more in the prototype nodes to distinguish different types of news. Errors inherited from the NER tagging model will lead to some non-location or non-organisation concepts remaining in the filtered global tree.

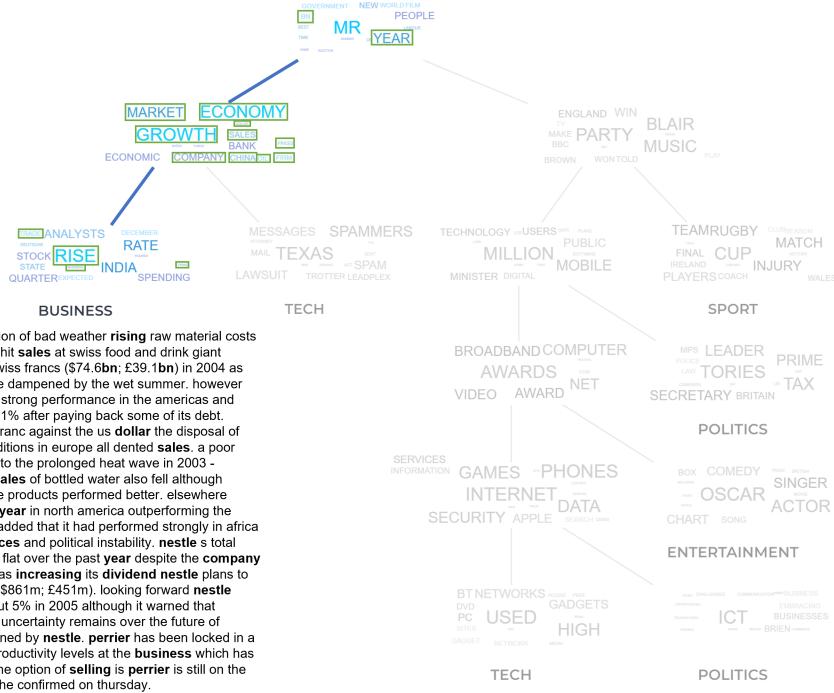


Figure 9: The local explanation decision tree generated for a correctly predicted **Business** news article in BBCNews dataset based on fine-tuned BERT embedding. We can observe that lots of business-related concepts or keywords can be matched from the global trend decision path (highlighted with green squares) to this specific article (highlighted as bolded words).

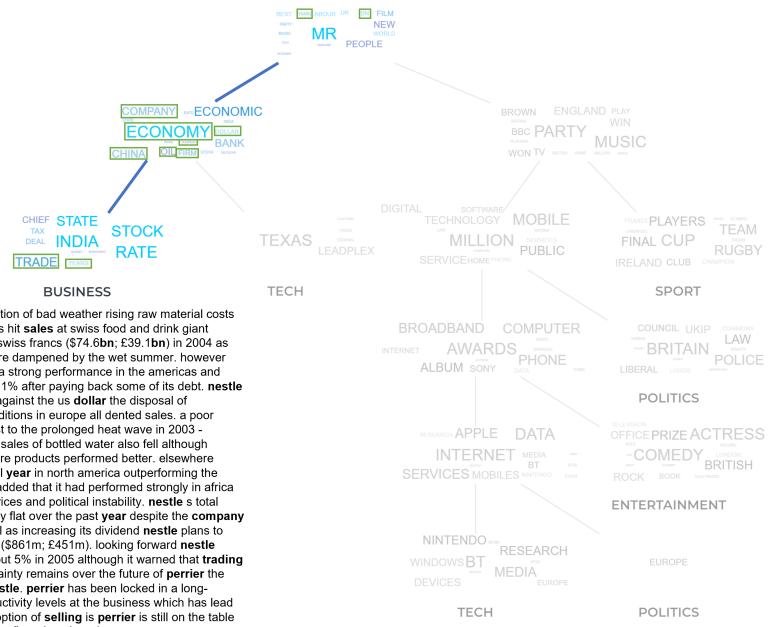


Figure 10: The local explanation decision tree generated for a correctly predicted **Business** news article in BBCNews dataset based on fine-tuned BERT embedding, with NER filters to display only words with NER tags for organizations and locations, labelled by the spaCy library. Business-related concepts or keywords can be matched from the global trend decision path (highlighted with green squares) to this specific article (highlighted as bolded words).

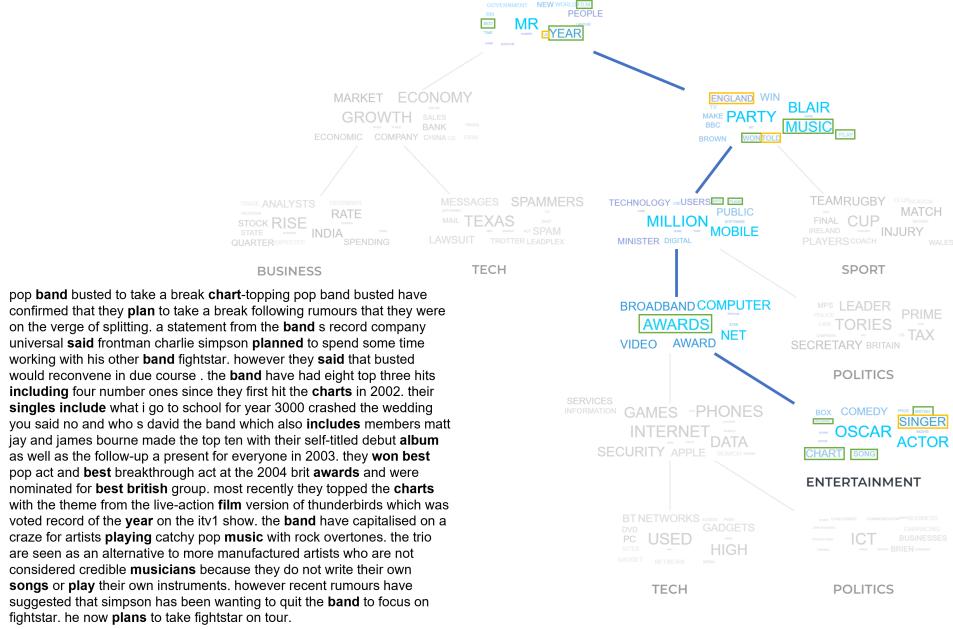


Figure 11: The local explanation decision tree generated for a correctly predicted **Entertainment** news article in BBCNews dataset based on fine-tuned BERT embedding. We can find music-related concepts in this specific news article (highlighted as bolded words), which also come out in the decision path, either as exact matching (highlighted with green squares) or similar concepts like *musician* vs *music* (highlighted with yellow squares).

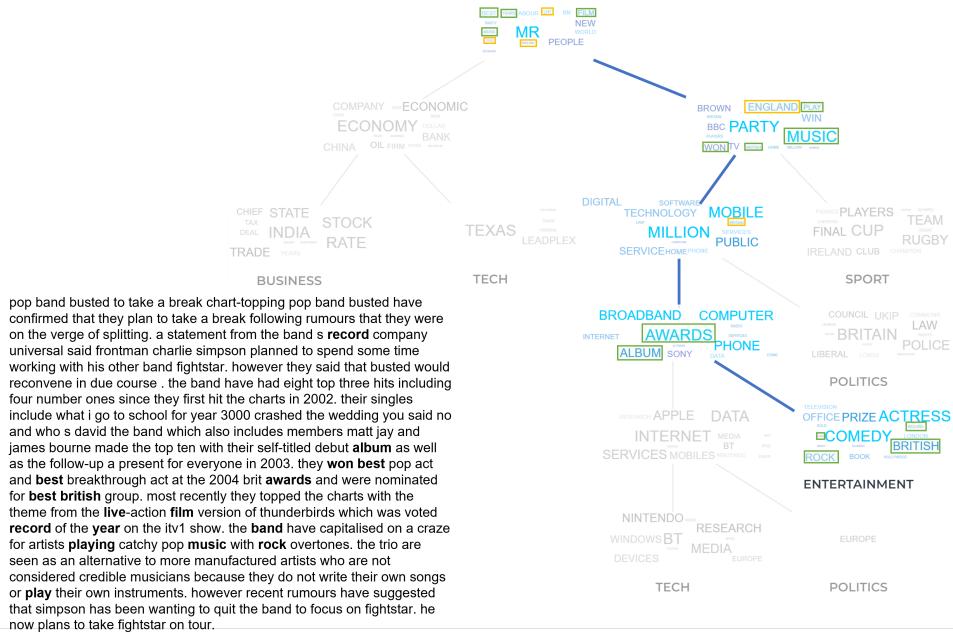


Figure 12: The local explanation decision tree generated for a correctly predicted **Entertainment** news article in BBCNews dataset based on fine-tuned BERT embedding, with NER filters to display only words with NER tags for organizations and locations, labelled by the spaCy library. Not many organization or location names are coming out in this music-related article, as well as the decision path, indicating that the other aspects can be further investigated for more obvious patterns.

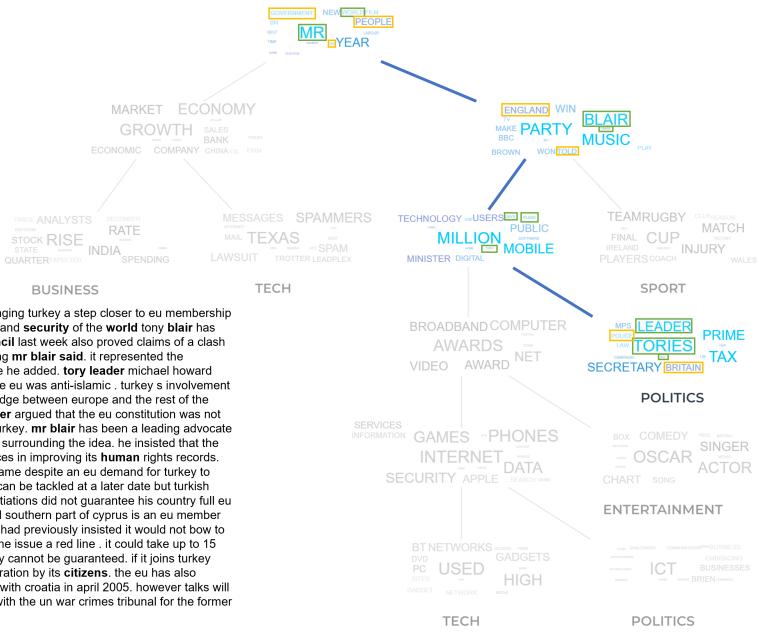


Figure 13: The local explanation decision tree generated for a correctly predicted **Politics** news article in BBCNews dataset based on fine-tuned BERT embedding. We can observe that government people's names or positions from the text (highlighted as bolded text) can be mapped to the decision path (highlighted as green squares for exact mapping, yellow squares for similar concepts).

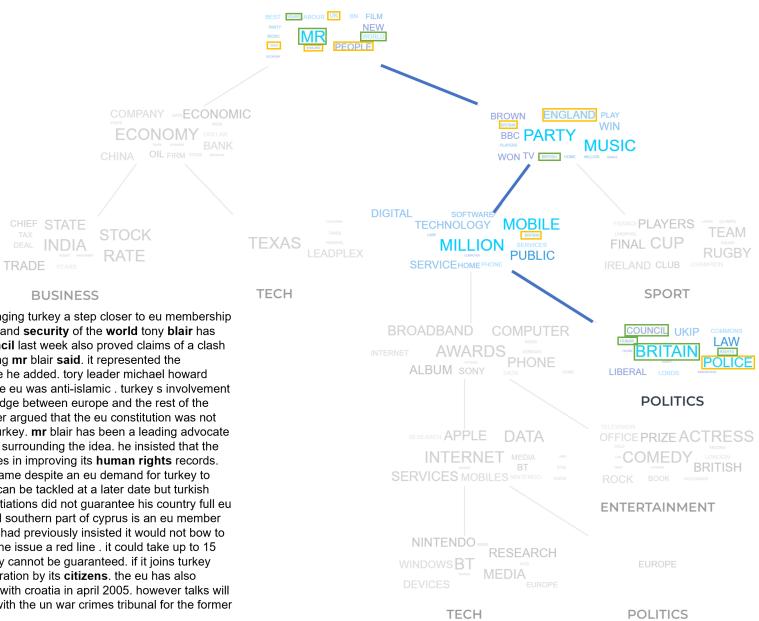


Figure 14: The local explanation decision tree generated for a correctly predicted **Politics** news article in BBCNews dataset based on fine-tuned BERT embedding, with NER filters to display only words with NER tags for organizations and locations, labelled by the spaCy library. In this case, important countries and government departments stand out more in the presence of the NER filter.

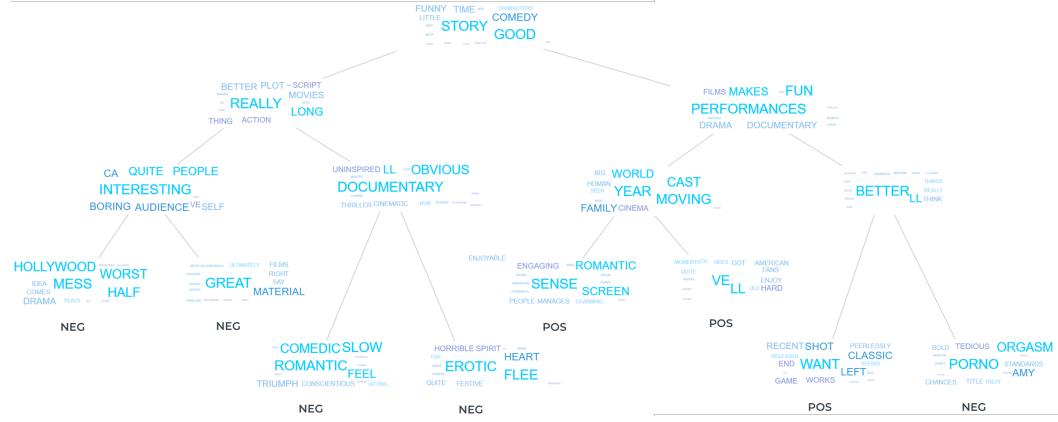


Figure 15: The global explanation decision tree generated on MR dataset based on fine-tuned RoBERTa embedding. Please take note that we explicitly limit the maximum depth of the tree during the training to prevent further branching out beyond 3rd children's level for better performance as well as interpretability, therefore the two children with the same class can be present. We can see words with negative connotations in human understanding tend to appear more in the prototype nodes or decision paths for the negative class, and similarly for the positive class.

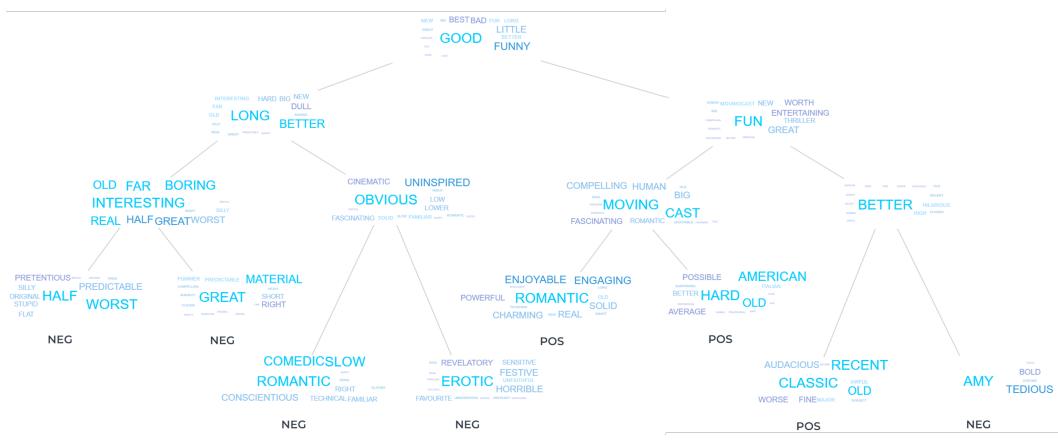


Figure 16: The global explanation decision tree generated on MR dataset based on fine-tuned RoBERTa embedding, with the POS filter to display only words which can be labelled as adjectives by the spaCy library. People tend to use adjectives when giving out movie reviews so investigating the adjectives for this dataset shows a clear pattern for classifying positive or negative reviews. Errors inherited from the POS tagging model will lead to some non-adjectives remaining in the filtered global tree.

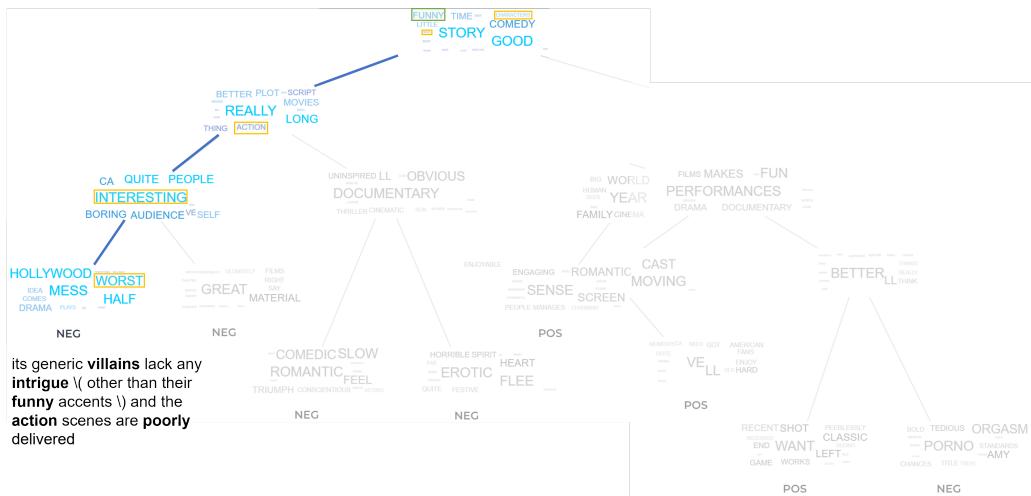


Figure 17: The local explanation decision tree generated for a correctly predicted **Negative** review in MR dataset based on fine-tuned RoBERTa embedding. Concepts related to judging whether something is interesting or not can be found in the decision path and negative word *worst* similar to the concept of *poorly* can be found as well.

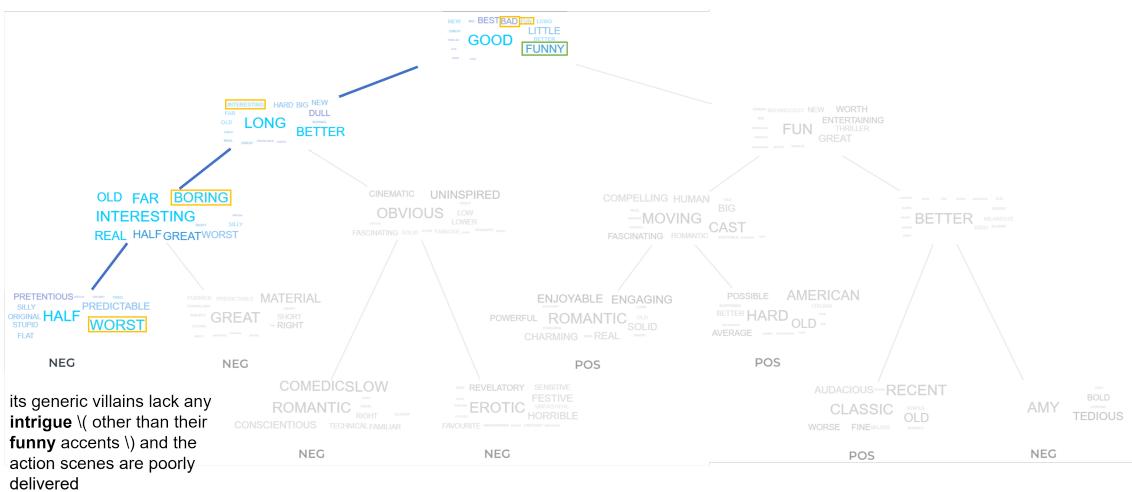


Figure 18: The local explanation decision tree generated for a correctly predicted **Negative** review in MR dataset based on fine-tuned RoBERTa embedding, with the POS filter to display only words which can be labelled as adjectives by the spaCy library. We can see concepts related to the *cast* or *action* is no longer found, leaving only important adjectives for decision-making.

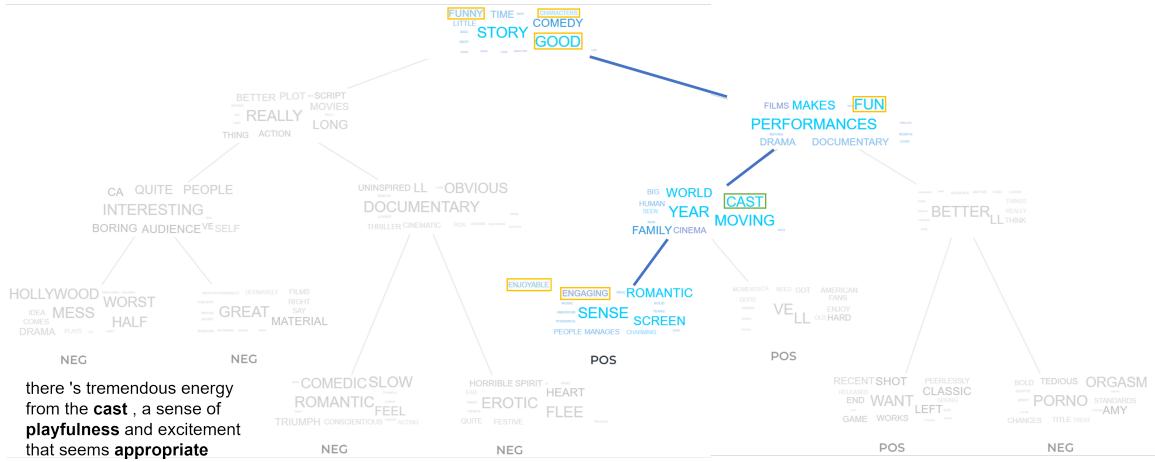


Figure 19: The local explanation decision tree generated for a correctly predicted **Positive** review in MR dataset based on fine-tuned RoBERTa embedding. Multiple positive comments like *good*, *fun*, *engaging*, *enjoyable* can be found in decision-making for the text describing something as *playful* and *appropriate*.

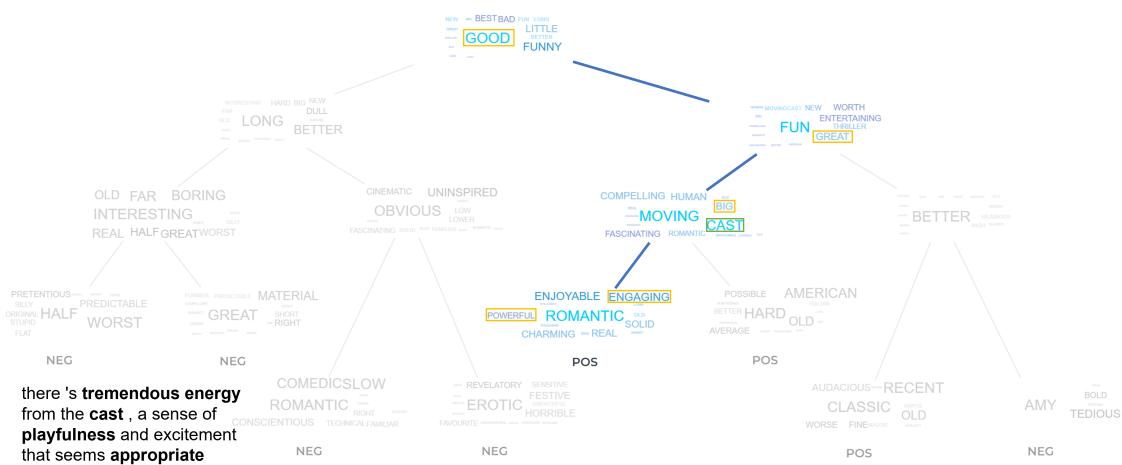


Figure 20: The local explanation decision tree generated for a correctly predicted **Positive** review in MR dataset based on fine-tuned RoBERTa embedding, with the POS filter to display only words which can be labelled as adjectives by the spaCy library.



Figure 21: The global explanation decision tree generated on SST2 dataset based on fine-tuned RoBERTa embedding. Please take note that we explicitly limit the maximum depth of the tree during the training to prevent further branching out beyond 3rd children's level for better performance as well as interpretability, therefore the two children with the same class can be present. Words with negative connotations like *mess*, *worst* in human understanding tend to appear more in the prototype nodes or decision paths for the negative class, and similarly for the positive class with concepts like *worth*, *deliciously*.



Figure 22: The global explanation decision tree generated on SST2 dataset based on fine-tuned RoBERTa embedding, with the POS filter to display only words which can be labelled as adjectives by spaCy library. More variations are of judgment other than common words like *good* or *bad* can be found when emphasising adjectives, such as *tragic*, *pretentious*, *dreadful* for the help of understanding the decision-making process. Errors inherited from the POS tagging model will lead to some non-adjectives remaining in the filtered global tree.

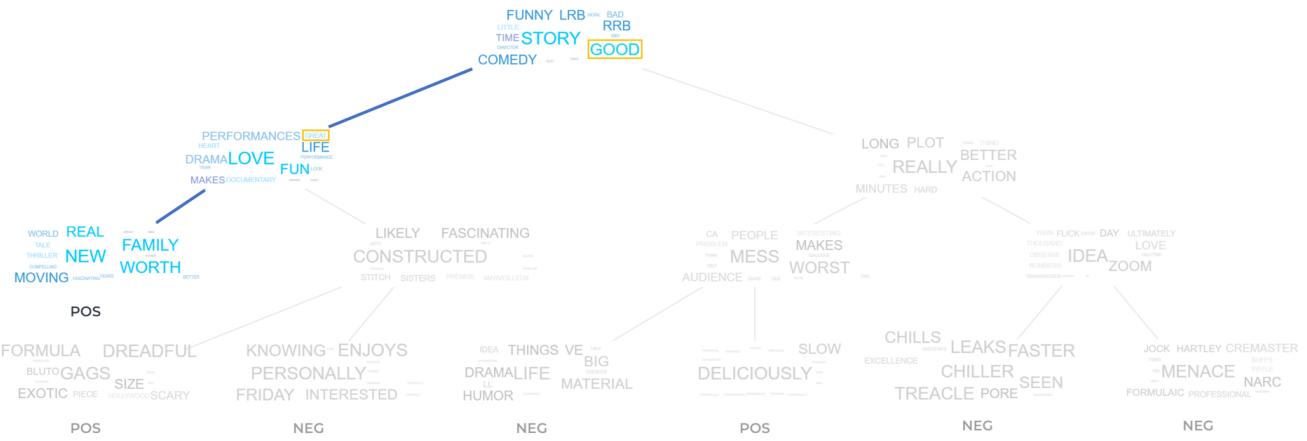


Figure 23: The local explanation decision tree generated for a correctly predicted **Positive** review in SST2 dataset based on fine-tuned RoBERTa embedding. Several positive concepts similar to *admirable* can be found in the decision path.

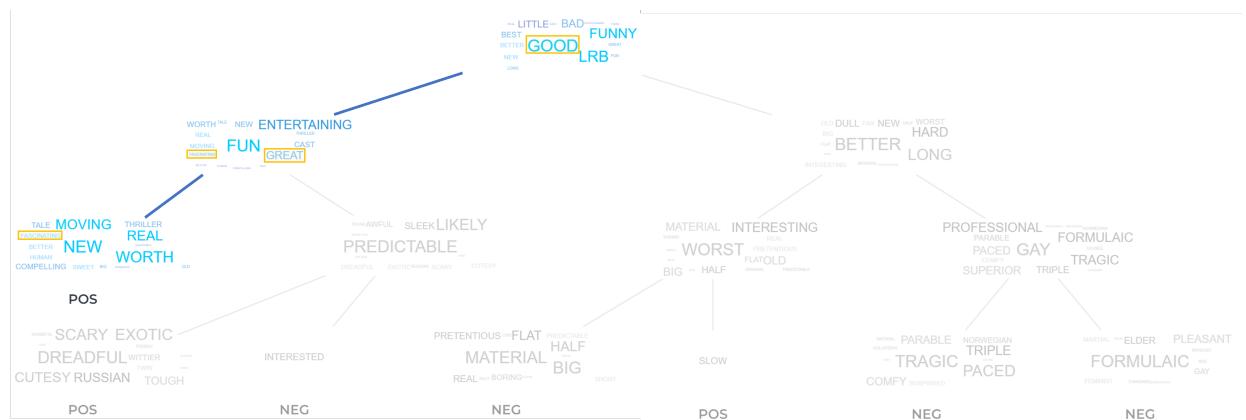


Figure 24: The local explanation decision tree generated for a correctly predicted **Positive** review in SST2 dataset based on fine-tuned RoBERTa embedding, with the POS filter to display only words which can be labelled as adjectives by spaCy library. Several positive concepts similar to *admirable* can be found in the decision path.

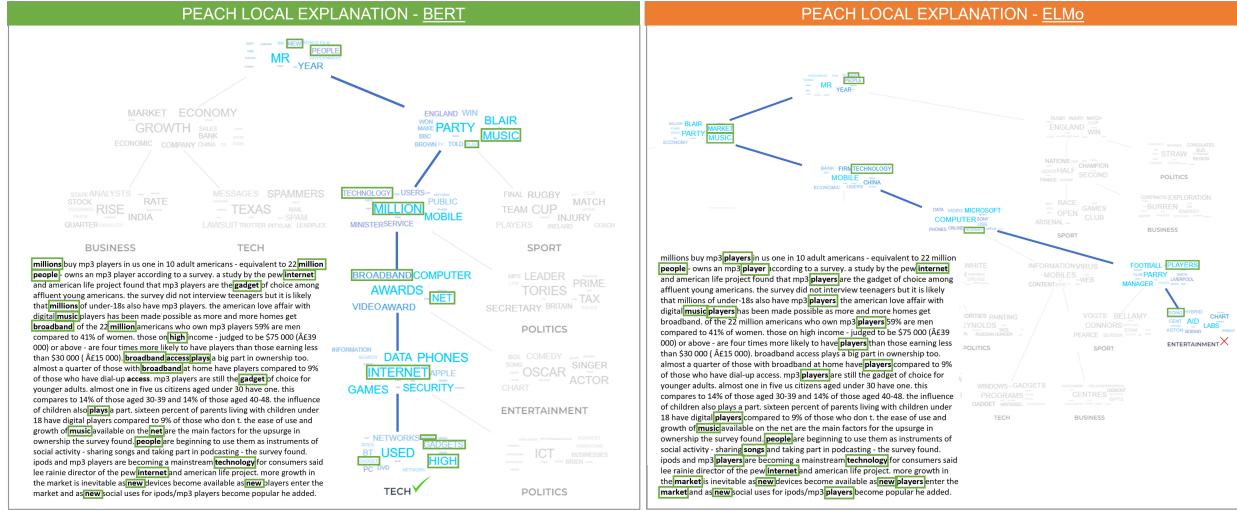


Figure 25: The local explanation decision trees generated for a **Tech** news article in BBCNews dataset based on fine-tuned BERT embedding compared to fine-tuned ELMo embedding, where PEACH(BERT) predicted correctly but PEACH(ELMo) predicted wrongly. While the BERT embedding fine-tuned by BBCNews has a clear decision-making path with exception nodes for technology classification, those with ELMo have the node that is lop-sided by the term ‘players’ and classified into either sports or entertainment.

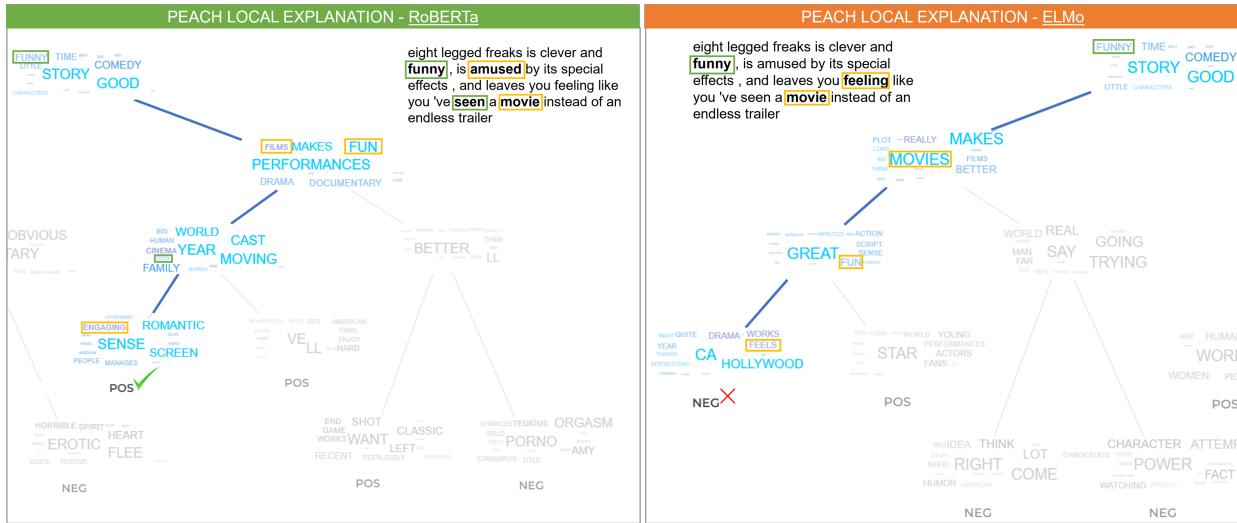


Figure 26: The local explanation decision trees generated for a **Positive** review in MR dataset based on fine-tuned RoBERTa embedding compared to fine-tuned ELMo embedding, where PEACH(RoBERTa) predicted correctly but PEACH(ELMo) predicted wrongly. While the successful embedding (RoBERTa) tends to have several adjectives that can highlight positive aspect (e.g. moving, engaging, romantic) or negative aspect (e.g. horrible, tedious), Elmo embedding does not understand the pattern of both with any remarkable adjectives patterns.

## Text

There's tremendous energy from the cast, a sense of playfulness and excitement that seems appropriate

## Type A

There's tremendous energy from the cast, a sense of playfulness and excitement that seems appropriate

NEG  
POS

## Type B

there's tremendous energy from the cast, a sense of playfulness and excitement that seems appropriate

Type C {"excitement", "tremendous"} → Positive

Figure 27: A sample case in the human evaluation to LIME and Anchor interpretation with our PEACH interpretation.