Natural Language Processing Project

Group 1

Abstract

Natural Language Inference (NLI) is a central problem in natural language understanding as it encapsulates the fundamental challenge of linguistic variability. In this study, we aim to investigate how neural architectures affect NLI task on a textual entailment dataset. The key contributions are that we implemented a baseline Bidirectional LSTM (Bi-LSTM), ESIM and Transformer encoder model, and qualiativive & ablation studies on ESIM. The main findings are ESIM outpoms with **80.3%** test accuracy, which highlights that explicit alignment and interaction modeling allow architectures like ESIM to outperform more generalized models.

1 Introduction

The NLI task is a foundational benchmark for Natural Language Understanding. It aims to analyze a pair of sentences, a premise p, and a hypothesis h, and to classify their logical relationship as entailment, contradiction, or neutral (Bowman et al., 2015).

Numerous studies have applied various neural architectures to this task. Benchmarks like the General Language Understanding Evaluation (GLUE) have tracked the progression from Bi-LSTMs to attention-based models (Wang et al., 2018). Attention-based models, such as ESIM, proved highly effective by explicitly modeling local inference (Chen et al., 2016).

Based on these researches, our study explores three neural architectures of increasing complexity to analyze how structural and representational factors affect inference performance. Specifically, we evaluate **Bi-LSTM Model, ESIM Model and Transformer-based encoder Model.**

The ESIM model performs best with **80.3%** test accuracy. To further interpret this model, we employ attention visualizations to analyze word alignments and conduct ablation studies on key components.

This study provides an empirical comparison and interpretive analysis of neural architectures for NLI, demonstrating that explicit relational modeling is the crucial factor in reasoning and offering guidance for future model design.

2 Methods

2.1 Data Preprocessing and Representation

To ensure a fair comparison, a consistent preprocessing pipeline was applied to all models. Each sentence pair is tokenized, lowercased, lemmatized and converted into integer indices using a shared vocabulary.

We initialized word representations using a FastText model trained with a 300-dimensional skip-gram configuration. To accommodate variable sentence lengths, padding and corresponding masks are applied during both training and evaluation.

2.2 Bi-LSTM

The Bi-LSTM model serves as our baseline (Khot et al., 2018). As illustrated in Figure 1, this architecture adopts an early-fusion strategy to model the relationship between the premise and hypothesis.

The token embeddings of the premise (X_p) and hypothesis (X_h) are first concatenated into a single input sequence X. This combined sequence is then fed into a Bi-LSTM layer. By processing sequential data in both forward and backward directions, the Bi-LSTM captures contextual information across the entire concatenated sequence. The hidden state h_t at each time step t is formed by concatenating the forward hidden state $\overrightarrow{h_t}$ and the backward hidden state $\overleftarrow{h_t}$.

This approach allows information to flow between the premise and hypothesis implicitly, as the backward pass enables the context from the hypothesis to flow back and influence the represen-

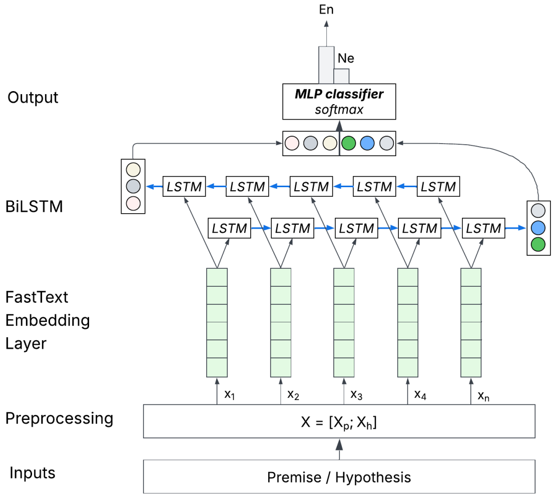

Figure 1: Bi-LSTM architecture diagram.

tations of the premise tokens. Finally, a fixed-size representation of the entire sequence is obtained by applying a pooling strategy over all the hidden states. This pooled vector is then passed through a MLP with a softmax classification.

2.3 ESIM

As the Bi-LSTM model adopts an early-fusion strategy and lacks explicit token-level alignment and interaction modeling between sentence pairs, we employ the ESIM model (Chen et al., 2016) to address these limitations.

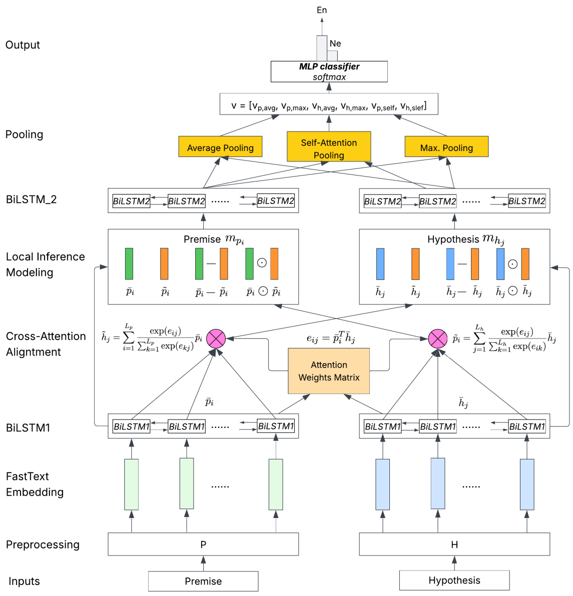
En
Ne
Output
MLP classifier
softmax
v = [v$_{p,avg}$, v$_{p,max}$, v$_{h,avg}$, v$_{h,max}$, v$_{p,self}$, v$_{h,slef}$]
Pooling
Average Pooling
Self-Attention Pooling
Max. Pooling
BiLSTM_2
BiLSTM2
......
Local Inference Modeling
Premise m_{p_i}
$\bar{p}_i$
$\tilde{p}_i$
$\bar{p}_i - \tilde{p}_i$
$\bar{p}_i \odot \tilde{p}_i$
Hypothesis m_{h_j}
$\bar{h}_j$
$\tilde{h}_j$
$\bar{h}_j - \tilde{h}_j$
$\bar{h}_j \odot \tilde{h}_j$
Cross-Attention Alignment
$\tilde{h}_j = \sum_{i=1}^{L_p} \frac{\exp(e_{ij})}{\sum_{k=1}^{L_p} \exp(e_{kj})} \bar{p}_i$
$e_{ij} = \bar{p}_i^T \bar{h}_j$
$\tilde{p}_i = \sum_{j=1}^{L_h} \frac{\exp(e_{ij})}{\sum_{k=1}^{L_h} \exp(e_{ik})} \bar{h}_j$
Attention Weights Matrix
$\bar{p}_i$
$\bar{h}_j$
BiLSTM1
FastText Embedding
Preprocessing
P
H
Inputs
Premise
Hypothesis

Figure 2: ESIM architecture diagram.

Input Encoding: Each sentence is independently encoded to capture contextual semantics. As shown in the figure 2, the word embeddings for the p and h are passed through a Bi-LSTM.

Local Inference Modeling (Attention Alignment): An attention alignment matrix $e \in R^{L_p \times L_h}$ is computed to quantify pairwise semantic similarity using the dot product:

$$e_{ij} = \bar{p}_i^T \bar{h}_j \tag{1}$$

Each word in one sentence is represented as a weighted sum of the words in the other sentence based on the attention scores. This produces two new attended vectors, $\tilde{p}$ and $\tilde{h}$:

$$\tilde{p}_i = \sum_{j=1}^{L_h} \frac{\exp(e_{ij})}{\sum_{k=1}^{L_h} \exp(e_{ik})} \bar{h}_j, \quad \tilde{h}_j = \sum_{i=1}^{L_p} \frac{\exp(e_{ij})}{\sum_{k=1}^{L_p} \exp(e_{kj})} \bar{p}_i \tag{2}$$

$\tilde{p}_i$ represents the parts of the hypothesis most relevant to the i-th word of the premise, and vice-versa for $\tilde{h}_j$. The model computes the element-wise difference and product between the original and the attended vectors. These features are concatenated to form a richer representation m:

$$m_{p_i} = [\bar{p}_i; \tilde{p}_i; \bar{p}_i - \tilde{p}_i; \bar{p}_i \odot \tilde{p}_i] \tag{3}$$

$$m_{h_j} = [\bar{h}_j; \tilde{h}_j; \bar{h}_j - \tilde{h}_j; \bar{h}_j \odot \tilde{h}_j] \tag{4}$$

It enables the model to more explicitly capture evidence of agreement and contradiction between aligned token pairs.

Inference Composition: The locally inferred features (m_p and m_h) are then passed through a second Bi-LSTM layer to compose them into higher-level inference features. They are aggregated using average, self-attention, and max pooling:

$$v = [v_{p,\text{avg}}, v_{p,\text{max}}, v_{h,\text{avg}}, v_{h,\text{max}}, v_{p,\text{self}}, v_{h,\text{self}}] \tag{5}$$

The final composed vector v is passed through a MLP followed by a softmax classifier.

2.4 Transformer-based Encoder

We also evaluate a Transformer-based model, which was inspired by (Lin et al., 2017) self-attention pooling captures contextual dependencies, and (Reimers and Gurevych, 2019) using Transformer-based pairwise operations for sentence similarity. Combining these ideas, our design integrates Transformer self-attention with interaction features.

Input Representation: Each input pair is concatenated with special tokens [CLS], [SEP], and [EOS]. Each token is represented by the sum of three embeddings: token embeddings, positional embeddings, and segment embeddings.

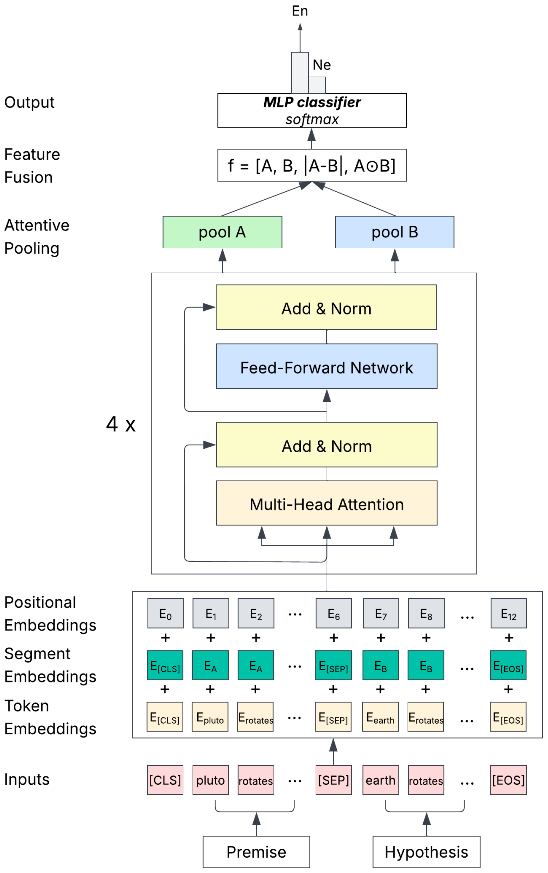

Figure 3: Transformer architecture diagram.

Multi-Head Self-Attention: The concatenated embeddings are passed through multiple stacked Transformer encoder layers. In each layer, token representations are projected into Query ($\mathbf{Q}$), Key ($\mathbf{K}$), and Value ($\mathbf{V}$) spaces and updated using scaled dot-product attention:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V} \tag{6}$$

where d_k denotes the dimension of the Key vectors.

Encoder Stack and Normalization: Each encoder layer contains residual connections, layer normalization, and a feed-forward network to stabilize training and enhance representational depth. Four such layers are stacked sequentially to form the full encoder.

Output and Classification: The contextualized representations are pooled using mean or self-attention pooling, producing fixed-length vectors for the premise and hypothesis. These are combined through feature fusion and passed through by a softmax classifier.

3 Experiment Set Up

3.1 Dataset Description

Datasets	Instances	Neutral	%	Entails	%
Train	23,088	14,618	63.3%	8,470	36.7%
Val	1,304	647	49.6%	657	50.4%
Test	2,126	1,284	60.4%	842	39.6%
Total	26,518	16,549	62.4%	9,969	37.6%

Table 1: Dataset distribution.

The training set is moderately imbalanced with neutral roughly double the amount of entailments.

Datasets	Premise Length			Hypothesis Length		
	min	mean	max	min	mean	max
Train	1	21	14,556	4	13	38
Val	3	20	59	6	14	34
Test	2	19	53	5	14	31

Table 2: Premise and hypothesis length distribution.

The training set contains several extreme outliers. we constrained the lengths by setting the minimum length to 3, the maximum premise length to 48, and the maximum hypothesis length to 27. After filtering, 22,613 samples remain in this dataset.

3.2 Implementation Details

All models were implemented in PyTorch and trained on a single NVIDIA T4 GPU under a consistent setup using FastText embeddings.

Each model was trained for 20 epochs. In the ESIM model, embeddings were frozen for the first three epochs and subsequently unfrozen for joint fine-tuning with early stopping. For the Transformer model, a similar progressive unfreezing strategy with early stopping was adopted to stabilize convergence.

To achieve optimal performance, we fine-tuned the hyperparameters of each model using a grid search strategy, and the main tuned parameters are summarized in Table 3.

Model	Lr	Dropout	Weight decay	Running time
Bi-LSTM	1e-3	0.3	0	16 mins
ESIM	5e-4	0.1	1e-4	25 mins
Transformer	3e-4	0.2	1e-3	60 mins

Table 3: Hyperparameters and running time.

4 Results

4.1 Main Performance Comparison

Table 4 below shows the performance of all models.

Model	Train Accuracy	Val Accuracy	Test Accuracy
BiLSTM	82.01%	71.01%	**60.11%**
ESIM	**97.00%**	**82.90%**	**80.29%**
Transformer	68.37%	71.55%	**66.60%**

Table 4: Model performance summary.

Fine-tuned ESIM model outperforms with the highest testing accuracy at 80.29%. Bi-LSTM model has the lowest testing accuracy at 60.11% while Transformer encoder obtained a slightly higher testing accuracy 66.6%.

According to Table 5, ESIM shows the best generalisation a stark contrast with Bi-LSTM model showing bias towards the majority neutral class while our Transformer encoder model shows overgeneralisation towards the majority class.

Model	True Label	Pred: Entails	Pred: Neutral
ESIM	Entails	0.75	0.25
ESIM	Neutral	0.16	0.84
BiLSTM	Entails	0.01	0.99
BiLSTM	Neutral	0.01	0.99
Transformer	Entails	0.31	0.69
Transformer	Neutral	0.10	0.90

Table 5: Confusion matrices on the test set.

The results exceed our expectations. Compared with the BiLSTM baseline, ESIM performs the best and shows strong generalisation and interpretability. This confirms that explicit alignment and compositional inference were effective, while the Transformer's weaker results highlight that complexity alone does not guarantee better performance.

4.2 Ablation Study on Attention Mechanism

We explore the effect of attention mechanism design on ESIM model by testing it against four types, namely self-attention-only, cross-attention only, combined self + cross-attention and no attention while all other hyperparameters were kept constant.

Attention Type	Validation Accuracy	Test Accuracy
Cross	80.83	77.56
Both	80.75	78.46
None	71.17	72.77
Self	70.17	73.19

Table 6: Effect of attention type on accuracy.

As shown in Table 6, the cross-attention and combined self + cross attention mechanisms perform better compared to the self and no-attention. The highest testing accuracy was the combined attention achieving 78.46% accuracy while cross attention is slightly lower at 77.56%. The lowest was no attention at 72.77%. Self attention performs slightly higher than no attention at 73.19%.

Figure 4 supports our findings with both and cross-attention having consistently higher performance.

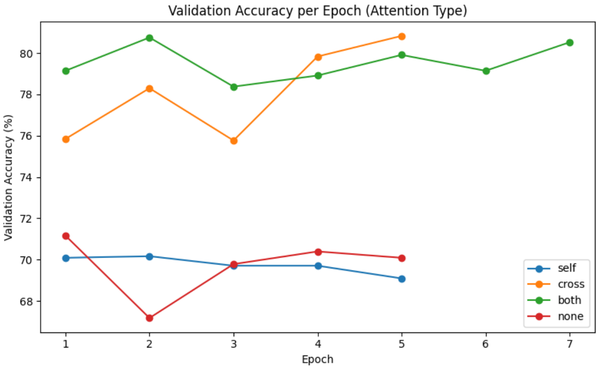

Figure 4: Validation accuracy across attention types.

4.3 Additional Ablation Study I: Hidden Size

An ablation study was conducted to examine the effect of hidden layer size on ESIM performance. Table 7 shows the validation and testing accuracy across 4 hidden sizes. From hidden size 64 to 256, the performance marginally increases then slightly drops on h 384.

Hidden Size	Validation Accuracy	Test Accuracy
256	81.60	80.62
384	80.67	80.24
128	80.21	80.01
64	79.68	79.45

Table 7: Effect of hidden size on accuracy.

Smaller hidden sizes lacked representational depth, while larger ones led to diminishing returns. The final configuration at h = 256 offered the best balance between performance and efficiency.

4.4 Additional Ablation Study II: Interaction features

ESIM uses element-wise difference (Diff) to represent semantic comparison and multiplicative interaction (Mult) to represent similarity between aligned token pairs. To investigate the effect of interaction features against ESIM model's performance, four interaction types were used.

As shown in Table 8, mult interaction type attains the highest test accuracy 79.3% while combined performs slightly under at 78.32% accuracy.

Interaction Type	Validation Accuracy	Test Accuracy
diff_mul	81.29	78.32
diff_only	78.76	76.53
mul_only	78.68	79.30
none	75.00	74.84

Table 8: Effect of interaction type on accuracy.

No interaction has the lowest testing accuracy at 74.84 while Diff only achieved a middle ground of 76.53% testing accuracy.

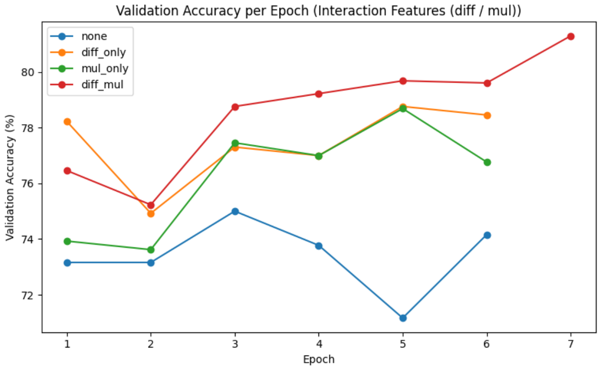

Figure 5: Validation accuracy by interaction features.

Figure 5 supports our finding that combined interaction type substantially enhances performance by improving semantic alignment and reducing overfitting.

5 Qualitative Results

5.1 Misclassified Prediction

A qualitative analysis was conducted on ESIM model. ESIM does good in many samples (Appendix 7, 8), here we chose to highlight an example that shows a misclassification case where the true label is entails and our model predicted neutral.
p: "A polyploid is simply an organism that contains more than the usual two sets of chromosomes."
h: "A(n) polyploid is an individual with more than the correct number of chromosome sets."

Although *h* and *p* describe the same biological concepts, the model predicts neutral, which is false.

Looking further, the attention map shown in Figure 6 below shows that the model attends strongly to "chromosome-chromosome" and "set-set" but fails to propagate the alignment across the entire phrase, which results in incomplete semantic integration. This suggests that our model lacks robustness in reasoning over longer compositional structures. As such, the error shows that potential limitations in inference capability lie where the attention alignment is not fully used for entailment reasoning.

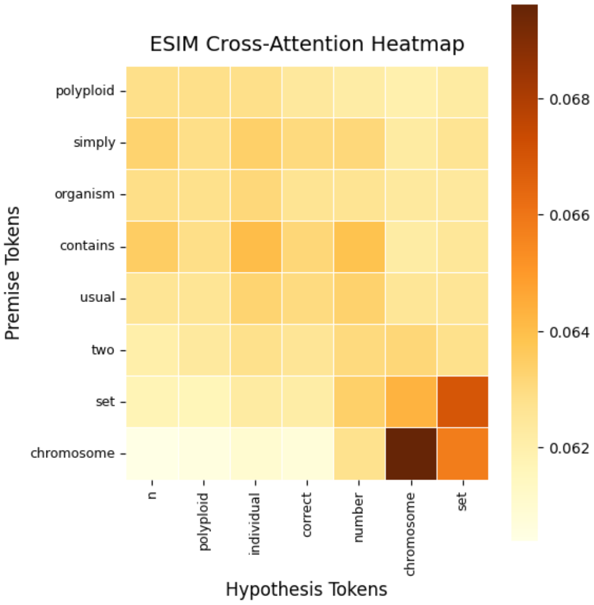

Figure 6: ESIM cross-attention map.

Overall, the attention mechanism contributes directly to the models predictive behavior and in the misclassification example, attention focuses too narrowly on local token matches, thus failing to capture full sentence semantics.

6 Conclusion

We designed and compared the performance of Bi-LSTM, ESIM, and Transformer-based Encoder Models. ESIM model achieved the best overall testing accuracy by using cross-attention and interaction features, which enable detailed semantic matching between *p* and *h* sentences. In ablation study, we investigated how different attention and interaction mechanisms affect the reasoning of the ESIM model, which confirms the functional contributions of each architectural component. Qualitative visualisation also showed interpretable reasoning patterns, where misclassifications stemmed from incomplete or overly localised attention.

Overall, with limited domain data, this project highlights that models combining explicit alignment with compositional inference mechanisms are well suited for the NLI task with the provided dataset. Future work can be done on exploring pretrained contextual embeddings and their effect on model performance.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 632–642. Association for Computational Linguistics.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2016. Enhanced lstm for natural language inference. *arXiv preprint arXiv:1609.06038*.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

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 *International Conference on Learning Representations (ICLR)*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Appendix

Name	Contribution
Bassiman Bin Anuar (24389611)	Transformer Final report
Boya Zhang (24324257)	ESIM, ablation, architecture diagram Final report
Justin Lu (24326939)	Data Processing, Bi-LSTM Final report

Table 9: Group contribution summary.

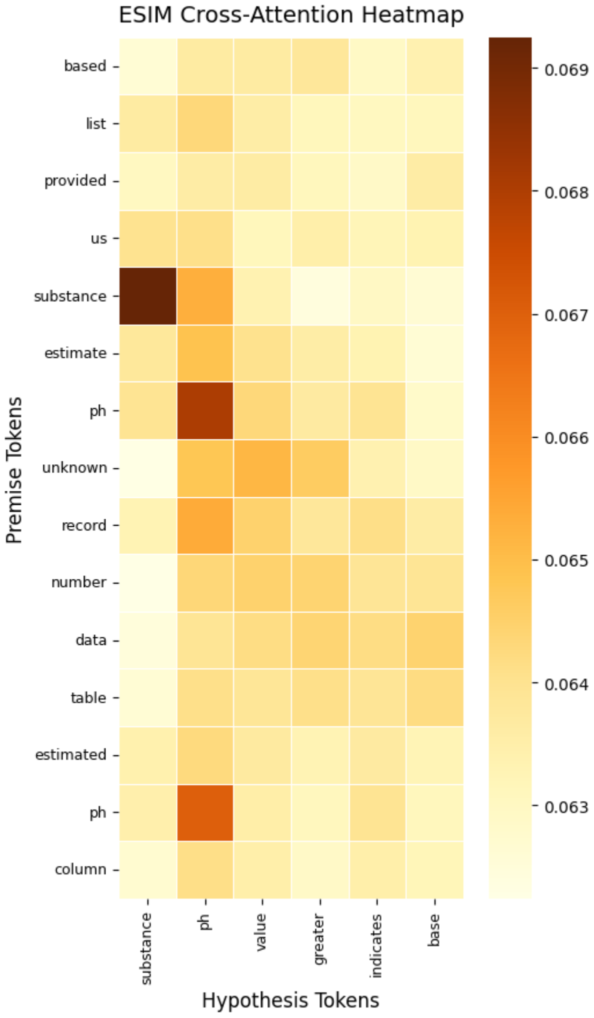

Figure 7: ESIM cross-attention map for correct prediction example 1.

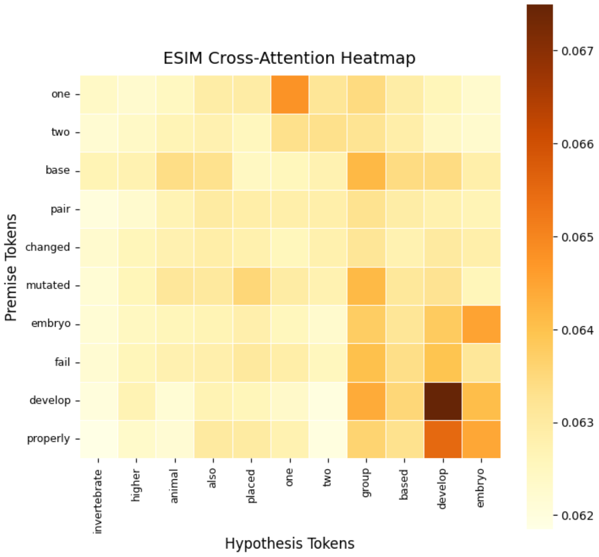

Figure 8: ESIM cross-attention map for correct prediction example 2.