Understanding the Hidden State of Language Models from the Perspective of Sentiment Analysis

Kaicheng Xue♣, Xulang Zhang♠, Rui Mao♠ and Erik Cambria♠

♣*University of Wisconsin–Madison, United States*
♠*Nanyang Technological University, Singapore*
kxue22@wisc.edu; {xulang.zhang, rui.mao, cambria@ntu.edu.sg}

***Abstract*—Transformer-based Large Language Models (LLMs) are widely used as foundational models for sentiment analysis. However, they have long been criticized for their lack of explainability and transparency. Most existing research focuses on interpreting the salience of input tokens with regard to the model prediction, while the inner workings of LLMs in sentiment analysis remain under-explored. In this work, we attempt to explore the hidden states of Transformer-based LLMs in relation to the sentiment conveyed by input texts. Specifically, we analyze the hidden states outputted by each layer as well as each head in a RoBERTa model finetuned for sentiment analysis, so as to examine the sentiment-related knowledge embedded in them. To achieve this, we apply three different clustering algorithms to probe whether each layer or head encodes sufficient knowledge to distinguish sentiment. Our experiments reveal that text length and frequency affect the tokens of hidden layers, and that not all heads within a layer contribute to the final result, indicating redundancy in parts of the model's internal structure. Additionally, we conduct a part-of-speech analysis, which suggests that hidden states contain information about part-of-speech tags. We further explore the internal mechanism in RoBERTa by performing experiments on word sense disambiguation and entailment.**

***Index Terms*—sentiment analysis, large language models, semantics, pragmatics.**

I. Introduction

Large Language Models (LLMs) have significantly impacted the research community from different aspects [1]. Researchers begin to focus on the state-of-the-art performance of large models in language tasks, the bias and fairness of models in learning and generating words [2], [3], and the efficiency of new architectures of models. Extensive knowledge has been embedded in the parameters of LLMs via pre-training. These parameters together build up LLMs' knowledge base for handling various tasks.

However, the embedded information has not been systematically examined in the context of sentiment analysis. Sentiment analysis typically focuses on improving the performance of models in recognizing sentiments and other sentiment-related tasks [4], with limited exploration of internal mechanisms behind these processes. In this work, we analyzed the hidden states of LLMs in the aspect of sentiments. We used clustering algorithms to isolate relevant information from the hidden states to uncover pattern changes within layers in the model. The targeted information includes typical sentiments of normal texts, texts containing frequently used words, and part-of-speech tag information for each word in texts.

Additionally, we conducted two other experiments, text entailment, and word sense disambiguation, to determine whether the hidden states of the model exhibit patterns similar to those found in the sentiment experiment. In this work, we used clustering methods to process raw hidden states and find pattern changes related to learning sentiments and part-of-speech tags. We found that there exists redundancy within hidden states when the model learns sentiment information from different texts; part of speech tag information could be learned by hidden states. Through experiments on text entailment and word sense disambiguation, we explored potential reasons behind the model's failures and tried to provide a general explanation of the internal mechanisms.

The contribution of this work can be summarized as follows: (1) The study investigates the hidden states of a Transformer-based pre-trained language model, RoBERTa, to understand how sentiment-related knowledge is encoded across different layers and heads. This provides insights into the internal mechanisms of language models in the context of sentiment analysis. (2) The findings reveal that there is redundancy within the hidden states when the model learns sentiment information, indicating that not all heads within a layer contribute equally to the final output. This suggests potential areas for optimization in model architecture.

II. Related Work

A. Sentiment Analysis

Sentiment analysis aims to classify the emotions and attitudes conveyed from people's words at different levels [5], [6]. In most cases, it is concerned with measuring the polarity of language expressions. Sentiment analysis itself can be categorized into four levels: word level, sentence level, document level, and featured-based level [7]. Each level has its own specific requirements, leading to different methodologies and research approaches. Researchers also focus on key features such as syntax, words, frequencies and negation, etc. [8]–[11]. There is research that analyzes sentiment and opinionated text from a cognitive perspective [12]–[14]. All of these reflect that sentiment analysis is a complex field of study.

In recent years, the emergence of Transformer-based language models has inspired extensive research into sentiment analysis. For instance, there are studies that explore the models' ability to handle ambiguous, ironic, or metaphorical

2375-9259/24/$31.00 ©2024 IEEE
DOI 10.1109/ICDMW65004.2024.00072

texts for sentiment analysis [15]–[19]. Other studies propose new cross-lingual approaches to enhance model performance in sentiment analysis [20], [21]. New model architectures for specializing in tasks are also important to the research community [22]. Miah et al. [23] proposed new models for sentiment analysis in new languages that lack enough sentiment datasets. Sun et al. [24] studied the model's inductive capabilities, and explored a new architecture to optimize the decision-making process for sentiment analysis. There are also studies that focused on enhancing the foundation models with sentiment knowledge bases [25]–[27] and employing sentiment analysis on a wide range of downstream tasks [28]–[30] and modalities [31]–[33].

LLMs have also been implemented in various fields and applications. This becomes important in service industries such as finance and healthcare. For instance, Xing [34] applied LLM without fine-tuning to enhance performance in sentiment analysis on finance. Du et al. [35] analyzed the limitation of LLMs in financial sentiment analysis tasks. In the healthcare field, there are many works using LLMs to study health data. A study addressed the problem of low accuracy in sentiment tools on medical data [36]; another study investigates the application of LLMs on health-related social platform [37].

The works mentioned above aim to address gaps in sentiment analysis work in different fields to improve model performance. However, they do not explore the internal mechanism of LLMs when predicting sentiment labels. The internal mechanism of models is called a black box as it is isolated from the outside environment. Studying and interpreting hidden workings is beneficial for understanding the architecture of models and the role of components, which is critical for developing accountable AI systems [38], [39]. In this paper, we analyze the hidden states of LLMs to uncover patterns within the layers and heads of the model.

B. Clustering

In this work, we utilize the following three clustering algorithms:

K-Means. MacQueen first used the term K-means for a method to solve the problem of partitioning N-dimensional population into k sets under some conditions [40]. Stuart Lloyd first proposed a standard algorithm of K-means [41]. The basic algorithm of this method is to assign data points of populations to the nearest cluster centroid and then update the centroids until they stabilize. The pseudocode is shown in Algorithm 1.

Algorithm 1: K-Means Algorithm

Input: Data points X and number of clusters k
Result: Number of clusters with assignments
Initialization: choose cluster centers randomly;
while *centers have not converged* **do**
  Assign each point x_i to the nearest center;
  Update each center c_k to be the mean points of each cluster;
end

Spectral Clustering. The spectral clustering method has developed for a long time, building on concepts of graph theory and the Laplacian Matrix. Shi and Malik [42] popularized spectral clustering methods in the machine learning area. The algorithm steps they proposed can be simplified as in Algorithm 2 for general spectral clustering:

Algorithm 2: Spectral Clustering

Input: Data points X, number of clusters k
Result: Number of clusters with assignments
Construct similarity matrix W
Compute the Laplacian matrix L
Compute the first k eigenvectors of L
Form matrix U from the eigenvectors
Apply k-means clustering to U

Hierarchical Clustering. Danish botanist Sørensen first proposed an algorithm using hierarchical relationships to cluster data [43]. Hierarchical clustering has developed for a long time. Now there are two ways to construct clusters: agglomerative and divisive. The former builds clusters from the bottom up, starting with individual data points and progressively merging them into larger clusters. The latter begins with the entire dataset as a single cluster and iteratively splits it into smaller clusters. In this paper, we used the agglomerative approach for hierarchical clustering. The algorithm is shown in Algorithm 3:

Algorithm 3: Agglomerative Clustering

Input: Data points X, number of clusters k;
Result: Number of clusters with assignments
Initialization: each data point is a single cluster;
while *number of clusters* $> k$ **do**
  Find the two closest clusters based on a distance metric;
  Merge these two closest clusters;
end

III. Experiment

A. Datasets

We conducted our experiments using the following four datasets.

1) Sentiment140 [44] consists of 1.6M tweets with positive, negative, and neutral sentiment labels. Due to the dataset containing only 138 neutral samples, we exclude them to maintain data balance. To conduct our evaluation, we sorted the entire dataset by text length and extracted the 15,000 shortest and 15,000 longest texts.
2) The IMDB-sentiment-reviews dataset [45] contains 50k movie reviews labeled with positive and negative sentiments. We used the entire dataset for our experiment.
3) The Auditor Sentiment dataset [46] consists of 3.88k sentences from financial news, classified in positive,

negative, and neutral sentiment labels. We also remove the neutral labels and use the remaining data for cross-domain experiments in part of speech. For our analysis, we extract between 10000 to 20000 of the shortest and longest texts.

4) The SemCor dataset [47] contains about 20.1k texts with part-of-speech tags and word definitions. We use this dataset for part-of-speech and word sense disambiguation analysis, extracting between 10000 to 20000 texts for the experiment.
5) The SNLI dataset [48] contains about 570k texts consisting of premises, hypothesis and labels. There are three labels, entailment, contradiction, and neutral. We use this dataset for the entailment experiment, extracting 30000 texts from it.

B. Implementation Details

In our work, we use the RoBERTa-base model [49], which has been fine-tuned on the training set of the Sentiment140 dataset. The hyperparameters for training the model are as follows: the $learning\_rate$ is $1e^{-05}$; the $train\_batch\_size$ is 16; the $eval\_batch\_size$ is 8; the $seed$ is 42; the $optimizer$ is Adam with $betas = (0.9, 0.999)$ and $epsilon = 1e^{-08}$; the $lr\_scheduler\_type$ is $linear$; num_epochs is 5. The model achieves an accuracy of 89.33% on Sentiment140 datasets classification.

After feeding the data into the model, we extract hidden states from each hidden layer for further clustering and accuracy testing. As mentioned before, we use three clustering algorithms: K-means, Spectral, and Hierarchical clustering, with the following settings:

- For all clustering methods, the number of clusters $n\_clusters$ is set to 2;
- For Spectral clustering, choices for constructing affinity matrix, $affinity$, is set to `"nearest_neighbors"`; and the number of neighbors $n\_neighbors$ set to 1,000;
- For Hierarchical clustering, we use the linkage package to calculate the matrix. The linkage method is set to `"single"`, and the distance calculating method $metric$ is set to `"euclidean"`.

For visualizing the results, we use the tSNE method to process the hidden states and then apply the predicted labels from the clustering methods for classification visualization. The tSNE hyperparameters are set as follows: the dimension of embedded space, $n\_components$, is set to 2, and the number of neighbors, $perplexity$ is set to 6.

For the part-of-speech analysis (Section IV-D), after feeding the data into the model, we identify the tokens in the hidden states corresponding to the words in the original text, and then perform clustering on them based on part-of-speech tags. We calculate the accuracy of the predicted labels against the ground-true labels, and obtain the F1 scores for each tag as well as the total weighted F1 score for each layer. In the clustering algorithm, the $n\_clusters$ is set to the total number of possible tags provided by the dataset or by function word_tokenize from the NLTK package. We did truncation on texts which have only exceeded 256 tokens in sequence. Given the varying distributions of tags in the texts, we processed distributions of tokens extracted from the hidden states to ensure that different tags are evenly distributed for clustering.

In the entailment experiment (Section IV-E), we pair two sentences as a tuple and feed them into the model to obtain the hidden states of output. We then do K-means clustering on these hidden states to see if the algorithm can correctly predict a label for each tuple. The possible labels are entailment, contradiction, and neutral. We did truncation on texts which have exceeded 128 tokens in sequence.

In the word sense disambiguation experiment (Section IV-E), we input each sentence into the model and collect the tokens in the hidden states that correspond to the target word. Then, we run K-means clustering on these tokens. The number of labels is the total number of definitions of the target word. We did truncation on texts which have exceeded 512 tokens in sequence.

IV. Results

The sentiment analysis experiment (Table V) reveals that the last hidden layer consistently performs best, achieving an accuracy of around 80% under all conditions. Most other layers achieve the accuracy of about 50% (Table I and Table II). However, Layers 10 and 11 (Table III and Table IV) show improved performance on some datasets, reaching the accuracy between 60% and 70%, closing to 80%. Also, we find that the accuracy of some heads in layer 11 on short texts is averaged as 50%, while other heads maintain an accuracy of 78% (Table VI). Additionally, the model's performance improves across more layers when processing more frequent texts. The part-of-speech analysis experiment confirms that the hidden layers of the model have indeed learned underlying patterns from words. Part of the tags in the data exhibit different changing trends.

TABLE I
PERFORMANCE OF LAYERS 1 TO 9 ON SHORT TEXT.

Layer #	K-means	Spectral	Hierarchical
Layer1	54.38%	54.15%	54.2%
Layer2	54.43%	54.11%	54.15%
Layer3	54.44%	54.17%	54.19%
Layer4	54.57%	54.18%	54.17%
Layer5	54.76%	54.19%	54.32%
Layer6	54.75%	54.26%	54.33%
Layer7	50.19%	54.07%	54.65%
Layer8	50.16%	54.13%	54.88%
Layer9	50.16%	54.29%	54.79%

A. Visualization

Fig 3 shows the visualization of the best-performing hidden layer. The first row refers to short texts while the second row refers to long texts. It is obvious that all three algorithms can classify the entire dataset as expected. The last layer of hidden states exhibits the best performance. Fig 1 and Fig 2 show that hidden layers 10 and 11 have increasing accuracy, exceeding 50% in short texts, but reaching around 50% in long texts.

TABLE II
PERFORMANCE OF LAYERS 1 TO 9 ON LONG TEXT.

Layer #	K-means	Spectral	Hierarchical
Layer1	53.43%	54.21%	55.98%
Layer2	53.47%	50.5%	56.2%
Layer3	53.9%	53.86%	56.46%
Layer4	54.04%	53.98%	56.58%
Layer5	54.13%	54.14%	56.17%
Layer6	54.23%	54.19%	52.85%
Layer7	54.21%	54.16%	56.16%
Layer8	52.61%	54.23%	55.43%
Layer9	52.34%	54.11%	54.97%

TABLE III
PERFORMANCE OF LAYER 10.

Clustering Methods	Short Text	Long Text
K-means	76.47%	54.11%
Spectral	61.11%	54.02%
Hierarchical	72%	70.8%

TABLE IV
PERFORMANCE OF LAYER 11.

Clustering Methods	Short Text	Long Text
K-means	78.94%	54.13%
Spectral	79.96%	54.11%
Hierarchical	77.33%	75.85%

TABLE V
PERFORMANCE OF LAYER 12.

Clustering Methods	Short Text	Long Text
K-means	80.52%	81.55%
Spectral	80.13%	81.46%
Hierarchical	79.1%	81.08%

TABLE VI
PERFORMANCE OF THE HEADS IN LAYER 11.

Head #	K-means	Spectral	Hierarchical
Head1	79.84%	79.53%	78.98%
Head2	79.85%	79.63%	79.82%
Head3	78.73%	78.86%	78.29%
Head4	54.05%	54.11%	54.1%
Head5	50.33%	50.31%	50.31%
Head6	53.55%	53.5%	53.5%
Head7	77.27%	56.31%	76.86%
Head8	78.35%	78.08%	78.09%
Head9	78.69%	78.57%	77.39%
Head10	78.83%	78.71%	78.63%
Head11	78.81%	78.71%	77.01%
Head12	78.81%	78.71%	78.59%

We observe that the datasets influence the internal workings of the pre-trained language model. Beyond text length, other factors also affect performance of layers. To uncover these hidden factors, we make frequency-based analysis below.

B. Frequency-based Analysis

We count the occurrences of each word in the texts. For each text, we calculate the total number of occurrences of each word and divide it by the text length to obtain the average frequency. Then, we sort all texts based on this frequency and divide them into low-frequency and high-frequency groups.

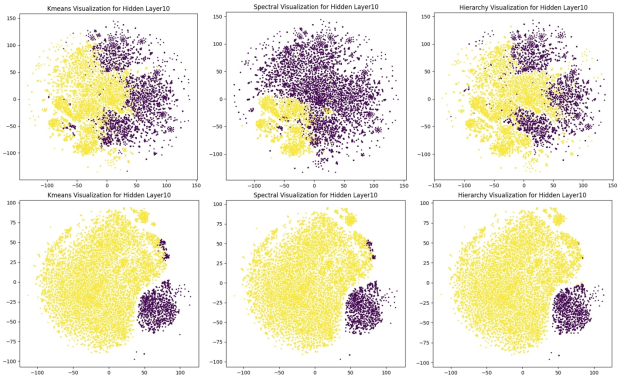


Fig. 1. Layer 10 on Sentiment140 sentiments classification.

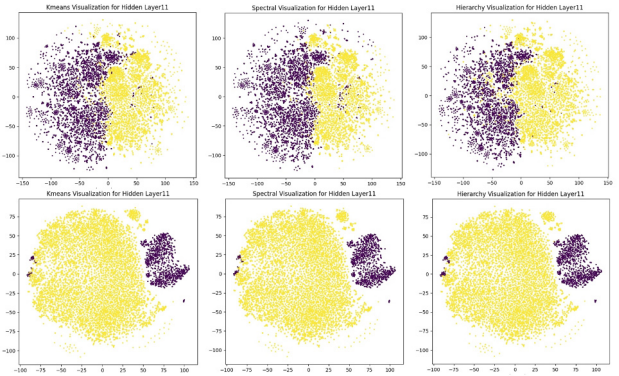


Fig. 2. Layer 11 on Sentiment140 sentiments classification.

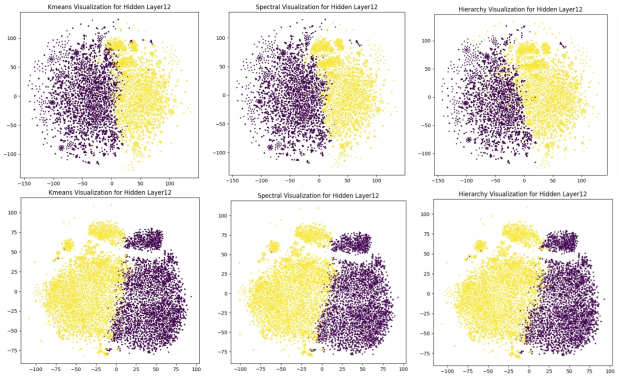
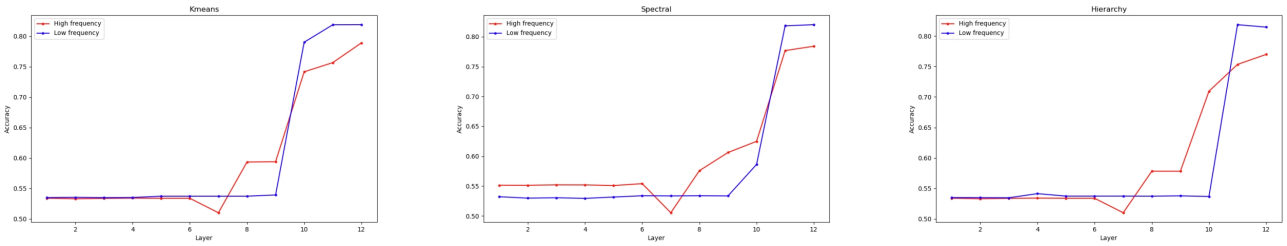


Fig. 3. Layer 12 on Sentiment140 sentiments classification.

We then inputted each group into the trained model, applied clustering algorithms to the model's hidden layers, and visualized the results to find the relationship between frequency and clustering accuracy. From Fig 4, it can be found that there is a spike in the low-frequency group (blue lines) starting at layer 9, where accuracy suddenly increases from 53.34% to 81.82%. In contrast, the high-frequency group (red lines) shows a gradual upward trend beginning at Layer 7.



(a) (b) (c)

Fig. 4. Clustering results of high and low frequency of input text; (a) refers to Kmeans; (b) refers to Spectral; (c) refers to Hierarchical.

Given that high-frequency words are more prevalent in the datasets than low-frequency words, it can be reasonably inferred that the hidden layers are more influenced by these common words. This inference explains the difference in accuracy between the hidden layers for long tweets compared to short tweets. Unlike short tweets dataset, long tweets dataset have fewer common sentences for model to learn and memorize. This is why higher accuracy was observed in Layers 10 and 11 across the three algorithms.

C. Head Analysis

Since all heads of the last hidden layer perform well, we also want to see whether this holds true for heads of other layers which show good results. We selected Layer 11 for short texts, as it achieves an accuracy of 78.94% with K-means, 79.96% with Spectral clustering and 77.33% with Hierarchical clustering. We extract heads of hidden layer 11 and calculate accuracy using three clustering algorithms. Most heads follow the pattern where single head performs as well as the entire layer. However, a few heads, including head 4, head 5 and head 6, only achieve around 50% accuracy. It is reasonable to infer that there exists redundancy in some layers that do not require all heads working on information [50].

D. Part of Speech Analysis

Through the head analysis in Section IV-C, we observe that the information within the tokens of each head is relatively obscure, making it difficult to identify a clear path for exploration. Also, the mechanism of interaction between different heads further complicates the token information, leading to disorder when grouping or extracting tokens. In order to discover more patterns, we shifted our focus from the hidden states themselves to the relationship between the input texts and the hidden states. We designed a part-of-speech experiment to explore the role of part-of-speech patterns in model's internal processing.

In this experiment, we used the dataset SemCor [47], which labels each word in an input sentence with part-of-speech tags. After feeding the text into the model, we obtain the hidden states. We then extract corresponding tokens from hidden states, and perform clustering algorithms on them to determine if the model has indeed learned the patterns of different part-of-speech tags in a sentence.

In this experiment, we used datasets from different domains and used function from NLTK package to perform part-of-speech tagging for each word in a sentence. We first choose a target word which contains different tags in the dataset. After feeding all texts containing target word into the model, we obtained the hidden states. Then we extracted the corresponding tokens for target word and applied clustering algorithms on them to see if there exists potential patterns that is similar to distribution of part-of-speech tags. The evaluation metric used is the F1 score, comparing clustering predictions to the ground-true labels. Since part-of-speech tags are distributed differently in each domain of dataset, we also calculate general weighted F1 scores to compare each of them. Given the varying distribution of part-of-speech tags in different dataset domains, we also calculate general weighted F1 scores, i.e., the sum of all F1 scores of each specific tag in a layer, to compare performance on each dataset.

As shown in Fig 5(a), RoBERTa's performance fluctuates between 3rd and 7th layer in these datasets. Starting from the eighth layer, the performance begins to decline and reaches the lowest point in the last layer. This observation suggests that part-of-speech information is retained in different layers when fine-tuned on different datasets. Nouns and verbs are important components in most sentences. As such, how they are distributed in different layers worth studying. We further analyzed the distribution of different tags in each layer. In Fig 5(b), it can be observed that the lines representing Finance, Review, and SemCor datasets are smoother than the line representing Sentiment140 dataset. The texts in Sentiment140 are tweets, which may lack necessary components as support for structure connections. As a result, the information of nouns may be affected. This infers that nouns are not processed independently by the model; instead, they are influenced by other elements within the sentence. Compared to NOUN F1 scores, VERB F1 scores in Fig 5(c) are much lower in most layers. These scores begin to decline at earlier layers, starting at 7, whereas NOUN scores start to decrease at layer 9. The lines representing Finance, Review and SemCor datasets become steady from layer 6, indicating that verb information becomes more consistent and less variable by this point. The line of Sentiment140 also supports the inference that information of words are not processed independently in RoBERTa.

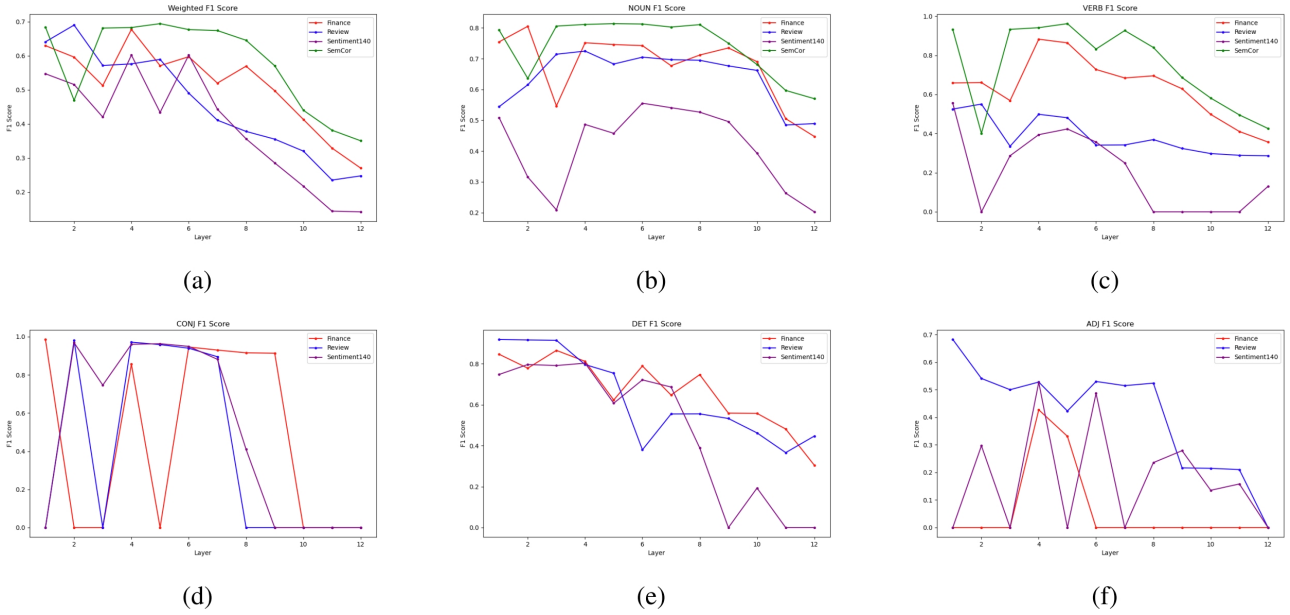


Fig. 5. F1 scores for different tags; (a) refers to total weighted F1 scores of each layer; (b) refers to F1 scores of nouns; (c) refers to F1 scores of verbs; (d) refers to F1 scores of conjunctions; (e) refers to F1 scores of determiners; (f) refers to F1 scores of adjectives.

Other tags, such as CONJ and DET, are also illustrated in Fig 5. Due to the limited range of word definitions for CONJ and DET words, such tokens are less varied and often perform well in certain layers, with F1 scores exceeding 90%. The ADJ tag, showed in third picture of Fig 5(f), indicates the differences among three datasets in their use of adjectives. The Review dataset uses many ADJ words in complete sentence, while Sentiment140 tends to use ADJ words but in phrases or short texts. The Finance datasets, which consists of analytical contents, uses relatively few ADJ words. This also proves that complete information affects words encoding and following passing through layers.

E. Further Experiments on RoBERTa

Learning patterns of part-of-speech tags aligns with human language habits. The part-of-speech experiment in Section IV-D illustrates that the Transformer-based model indeed has similar functions as human brains in learning language structures. Nevertheless, it is well-known that humans learn languages not only through basic structures and feelings but also by studying deeper connections, such as logic within words and sentences. Given this, we would like to explore whether the model also has similar performance as humans in logic aspects, especially in word illustrations and sentence inductions. We then designed two experiments to explore this objective. The results may further provide insight into the model's internal reasoning process. For word illustration, we designed a word sense disambiguation experiment. The word sense disambiguation studies how to determine which sense of a word is used in a particular context when the word has multiple meanings.

Differentiating word definitions requires logic, such as recognizing indicators and signs. Since this experiment usually focuses on the same word for study, it provides a clear way for comparison. These make word sense disambiguation suitable for investigating word logic. In the word sense disambiguation experiment, we first choose a target word that frequently appears in texts, and use the SemCor dataset [47] to gather all texts containing the target word. We define the number of distinct meanings for the target word as the number of ground-true labels. Then, we feed all the selected texts into the model and extract the corresponding tokens of the target word. We performed clustering on these tokens to classify them into the same number of clusters as the ground-true labels and calculate the accuracy of clustering prediction. The results of this experiment suggest that the token distributions for the target word do not align with those of definitions.

TABLE VII
K-MEANS PERFORMANCE COMPARISON FROM LAYERS 1 TO 12 FOR THE TARGET WORD "BE" AND THE TARGET WORDS "BE, IS, ARE, WAS".

Layer #	"be"	"be" extensions
Layer1	25.65%	32.57%
Layer2	44.61%	40.94%
Layer3	47.71%	44.44%
Layer4	51.14%	41.05%
Layer5	48.69%	45.84%
Layer6	47.71%	46.1%
Layer7	48.85%	47.96%
Layer8	43.14%	46.7%
Layer9	45.10%	46.41%
Layer10	36.93%	41.08%
Layer11	30.72%	34.35%
Layer12	31.53%	28.18%

For instance, Table VII shows an example of the target word "be" in the experiment, and none of the layers performs well, as accuracy is below or not over 50% significantly. We then include more extensions for "be" by finding texts containing "is", "are", and "was". However, as shown in Table VII, the accuracy was not improved.

For sentence inductions, we designed an entailment experiment. Entailment describes the logical relationship between two parts: the premise and the hypothesis (or conclusion), where the truth of the premise guarantees the truth of the hypothesis. To produce the correct answer, the model must understand both parts separately and coherently. This experiment is suitable for evaluating the model's ability to find relationships between texts. We use the SNLI dataset [48] with prepared premise and hypothesis texts for this experiment.

Since the premise and hypothesis are paired in the dataset, we group them together as a unit. We then feed these units into the model and obtain the hidden states. Then we apply clustering to these hidden states and calculate accuracy by clustering predictions and ground-true labels from the dataset. The results indicate that the model fails to recognize the logical relationships between premise and hypothesis. As shown in Table VIII, the accuracy is around $\frac{1}{3}$, suggesting that the model is randomly selecting between three possible outputs: entailment, contradiction and neutral.

TABLE VIII
THE PERFORMANCE OF LAYERS 1 TO 12 WAS EVALUATED FOR THE TASK OF ENTAILMENT.

Layer #	K-means
Layer1	37.22%
Layer2	37.33%
Layer3	36.07%
Layer4	35.7%
Layer5	35.98%
Layer6	36.15%
Layer7	36.34%
Layer8	36.04%
Layer9	37.62%
Layer10	34.96%
Layer11	34.5%
Layer12	38.55%

The failures observed for the two tasks above indicate that the internal mechanisms of the model do not mirror human thought processes. The Transformer-based model is highly task-driven and influenced by internal algorithms. It needs directions from a specific task to adapt to a new dataset. The internal algorithm and mechanism guide the model to provide a determined word as the answer. In sentiment fine-tuning, the model is trained to find the pattern of polarity of words. The connections between words in a sentence are a lower-level task in training sentiments, which explains why the model can identify patterns in part-of-speech tags. However, induction and word definition combinations are higher-level features that current internal workings did not learn about. This is why the model fails in these two experiments. Only when a task requires the model to identify higher-level information, it tends to align with those specific patterns.

V. CONCLUSION

In this work, we analyzed the hidden states of the RoBERTa model trained on different datasets. Clustering algorithms reveal that most earlier layers exhibit stead performance without significant fluctuations, while the performance of the last three layers gradually improves. However, the head analysis showed that some heads in the last three layers do not perform well, which is in contrast with the overall results for model layers. This indicates that certain layers and heads in the model may be redundant in processing and transferring information. In frequency-based analysis, we observed that as the number of frequent words in the input sentence increases, more layers would yield better performance. In part-of-speech analysis, we find that the hidden states have learned patterns corresponding to word tags in the first few layers. This also indicates that the last few layers contain more concentrated and abstract features. Further experiments on word sense disambiguation and entailment suggest that the RoBERTa model does not work intuitively as the human brain. The model seems to find and adapt to patterns according to the given algorithms and task prompts, instead of understanding the underlying meanings.

REFERENCES

[1] R. Mao, G. Chen, X. Zhang, F. Guerin, and E. Cambria, "GPTEval: A survey on assessments of ChatGPT and GPT-4," in *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*. Torino, Italia: ELRA and ICCL, 2024, pp. 7844–7866.

[2] R. Mao, Q. Liu, K. He, W. Li, and E. Cambria, "The biases of pre-trained language models: An empirical study on prompt-based sentiment analysis and emotion detection," *IEEE Transactions on Affective Computing*, vol. 14, no. 3, pp. 1743–1753, 2023.

[3] Q. Liu, S. Han, Y. Li, E. Cambria, and K. Kwok, "PrimeNet: A framework for commonsense knowledge representation and reasoning based on conceptual primitives," *Cognitive Computation*, 2024.

[4] R. Mao, M. Ge, S. Han, W. Li, K. He, L. Zhu, and E. Cambria, "A survey on pragmatic processing techniques," *Information Fusion*, vol. 114, p. 102712, 2025.

[5] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Engineering Journal*, vol. 5, no. 4, pp. 1093–1113, 2014.

[6] Y. Susanto, A. Livingstone, B. C. Ng, and E. Cambria, "The Hourglass Model revisited," *IEEE Intelligent Systems*, vol. 35, no. 5, pp. 96–102, 2020.

[7] A. Kumar and M. S. Teeja, "Sentiment analysis: A perspective on its past, present and future," *International Journal of Intelligent Systems and Applications*, vol. 4, no. 10, p. 1, 2012.

[8] E. Cambria, S. Poria, A. Gelbukh, and M. Thelwall, "Sentiment analysis is a big suitcase," *IEEE Intelligent Systems*, vol. 32, no. 6, pp. 74–80, 2017.

[9] H. Patel, X. Zhang, and Q. Liu, "Enhancing negation scope detection using multitask learning," in *2021 International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2021, pp. 179–185.

[10] X. Zhang, R. Mao, and E. Cambria, "A survey on syntactic processing techniques," *Artificial Intelligence Review*, vol. 56, no. 6, pp. 5645–5728, 2023.

[11] R. Mao, K. He, X. Zhang, G. Chen, J. Ni, Z. Yang, and E. Cambria, "A survey on semantic processing techniques," *Information Fusion*, vol. 101, p. 101988, 2024.

[12] R. Mao, K. Du, Y. Ma, L. Zhu, and E. Cambria, "Discovering the cognition behind language: Financial metaphor analysis with MetaPro," in *2023 IEEE International Conference on Data Mining (ICDM)*. Shanghai, China: IEEE, 2023, pp. 1211–1216.

[13] R. Mao, Q. Lin, Q. Liu, G. Mengaldo, and E. Cambria, "Understanding public perception towards weather disasters through the lens of metaphor," in *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, (IJCAI-24)*. Jeju, South Korea: International Joint Conferences on Artificial Intelligence Organization, 2024, pp. 7394–7402.

[14] R. Mao, T. Zhang, Q. Liu, A. Hussain, and E. Cambria, "Unveiling diplomatic narratives: Analyzing United Nations Security Council debates through metaphorical cognition," in *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci)*, vol. 46, Rotterdam, the Netherlands, 2024, pp. 1709–1716.

[15] A. K. Jayaraman, T. E. Trueman, G. Ananthakrishnan, S. Mitra, Q. Liu, and E. Cambria, "Sarcasm detection in news headlines using supervised learning," in *2022 International Conference on Artificial Intelligence and Data Engineering (AIDE)*. IEEE, 2022, pp. 288–294.

[16] X. Zhang, R. Mao, K. He, and E. Cambria, "Neuro-symbolic sentiment analysis with dynamic word sense disambiguation," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023, pp. 8772–8783.

[17] A. Buscemi and D. Proverbio, "Chatgpt vs gemini vs llama on multilingual sentiment analysis," *arXiv preprint arXiv:2402.01715*, 2024.

[18] R. Mao, X. Li, M. Ge, and E. Cambria, "MetaPro: A computational metaphor processing model for text pre-processing," *Information Fusion*, vol. 86-87, pp. 30–43, 2022.

[19] R. Mao, K. He, C. B. Ong, Q. Liu, and E. Cambria, "MetaPro 2.0: Computational metaphor processing on the effectiveness of anomalous language modeling," in *Findings of the Association for Computational Linguistics: ACL*. Bangkok, Thailand: Association for Computational Linguistics, 2024, pp. 9891–9908.

[20] P. Přibáň, J. Šmíd, J. Steinberger, and A. Mištera, "A comparative study of cross-lingual sentiment analysis," *Expert Systems with Applications*, vol. 247, p. 123247, 2024.

[21] X. Zhang, R. Mao, and E. Cambria, "Multilingual emotion recognition: Discovering the variations of lexical semantics between languages," in *2024 International Joint Conference on Neural Networks (IJCNN)*, 2024.

[22] R. Mao and X. Li, "Bridging towers of multi-task learning with a gating mechanism for aspect-based sentiment analysis and sequential metaphor identification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 15, 2021, pp. 13 534–13 542.

[23] M. S. U. Miah, M. M. Kabir, T. B. Sarwar, M. Safran, S. Alfarhood, and M. Mridha, "A multimodal approach to cross-lingual sentiment analysis with ensemble of transformer and llm," *Scientific Reports*, vol. 14, no. 1, p. 9603, 2024.

[24] X. Sun, X. Li, S. Zhang, S. Wang, F. Wu, J. Li, T. Zhang, and G. Wang, "Sentiment analysis through llm negotiations," *arXiv preprint arXiv:2311.01876*, 2023.

[25] K. Du, F. Xing, R. Mao, and E. Cambria, "FinSenticNet: A concept-level lexicon for financial sentiment analysis," in *2023 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2023, pp. 109–114.

[26] X. Zhang, R. Mao, and E. Cambria, "Senticvec: Toward robust and human-centric neurosymbolic sentiment analysis," *Findings of the Association for Computational Linguistics: ACL 2024*, 2024.

[27] E. Cambria, X. Zhang, R. Mao, M. Chen, and K. Kwok, "SenticNet 8: Fusing emotion AI and commonsense AI for interpretable, trustworthy, and explainable affective computing," in *International Conference on Human-Computer Interaction (HCII)*, 2024.

[28] Y. Ma, R. Mao, Q. Lin, P. Wu, and E. Cambria, "Multi-source aggregated classification for stock price movement prediction," *Information Fusion*, vol. 91, pp. 515–528, 2023.

[29] K. Du, F. Xing, R. Mao, and E. Cambria, "Financial sentiment analysis: Techniques and applications," *ACM Computing Surveys*, vol. 56, no. 9, pp. 1–42, 2024.

[30] Y. Ma, R. Mao, Q. Lin, P. Wu, and E. Cambria, "Quantitative stock portfolio optimization by multi-task learning risk and return," *Information Fusion*, vol. 104, p. 102165, 2024.

[31] T. Yue, R. Mao, H. Wang, Z. Hu, and E. Cambria, "KnowleNet: Knowledge fusion network for multimodal sarcasm detection," *Information Fusion*, vol. 100, p. 101921, 2023.

[32] C. Fan, J. Lin, R. Mao, and E. Cambria, "Fusing pairwise modalities for emotion recognition in conversations," *Information Fusion*, vol. 106, p. 102306, 2024.

[33] X. Luwei, R. Mao, X. Zhang, L. He, and E. Cambria, "Vanessa: Visual connotation and aesthetic attributes understanding network for multimodal aspect-based sentiment analysis," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, Miami, Florida, 2024, pp. 4851–4863.

[34] F. Xing, "Designing heterogeneous llm agents for financial sentiment analysis," *arXiv preprint arXiv:2401.05799*, 2024.

[35] K. Du, F. Xing, R. Mao, and E. Cambria, "An evaluation of reasoning capabilities of large language models in financial sentiment analysis," in *IEEE Conference on Artificial Intelligence (IEEE CAI)*, Singapore, 2024.

[36] J. A. Lossio-Ventura, R. Weger, A. Y. Lee, E. P. Guinee, J. Chung, L. Atlas, E. Linos, and F. Pereira, "A comparison of chatgpt and fine-tuned open pre-trained transformers (opt) against widely used sentiment analysis tools: Sentiment analysis of covid-19 survey data," *JMIR Mental Health*, vol. 11, p. e50150, 2024.

[37] L. He, S. Omranian, S. McRoy, and K. Zheng, "Using large language models for sentiment analysis of health-related social media data: empirical evaluation and practical tips," *medRxiv*, pp. 2024–03, 2024.

[38] E. Cambria, R. Mao, M. Chen, Z. Wang, and S.-B. Ho, "Seven pillars for the future of artificial intelligence," *IEEE Intelligent Systems*, vol. 38, no. 6, pp. 62–69, 2023.

[39] E. Cambria, "Understanding natural language understanding," *Springer, ISBN 978-3-031-73973-6*, 2024.

[40] J. Macqueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability/University of California Press*, 1967.

[41] S. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[42] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[43] T. Simpson, "A method of establishing group of equal amplitude in plant society based on similarity of species content," *K. Danske idensk. Selsk*, vol. 5, pp. 1–34, 1948.

[44] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," *CS224N Project Report, Stanford*, vol. 1, no. 12, p. 2009, 2009.

[45] Dame_Rajee, "Imdb sentiment reviews," 2023. [Online]. Available: https://huggingface.co/datasets/damerajee/IMDB-sentiment-reviews

[46] Finance_Inc., "Auditor sentiment," 2022. [Online]. Available: https://huggingface.co/datasets/FinanceInc/auditor_sentiment

[47] Yu-Ting_Chen, "Semcor," 2023. [Online]. Available: https://huggingface.co/datasets/MarkChen1214/SemCor

[48] Stanford_NLP, "snli," 2022. [Online]. Available: https://huggingface.co/datasets/stanfordnlp/snli

[49] Itay_Etelis, "Sentiment140_roberta_5e," 2022. [Online]. Available: https://huggingface.co/pig4431/Sentiment140_roBERTa_5E

[50] J. Ni, R. Mao, Z. Yang, H. Lei, and E. Cambria, "Finding the pillars of strength for multi-head attention," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, vol. 1. Toronto, Canada: Association for Computational Linguistics, 2023, pp. 14 526—14 540.