Your Mixture-Of-Experts LLM Is Secretly An Embedding Model For Free

Ziyue Li, Tianyi Zhou; UMD

Presenters: Siva, Keshav

Georgia Tech

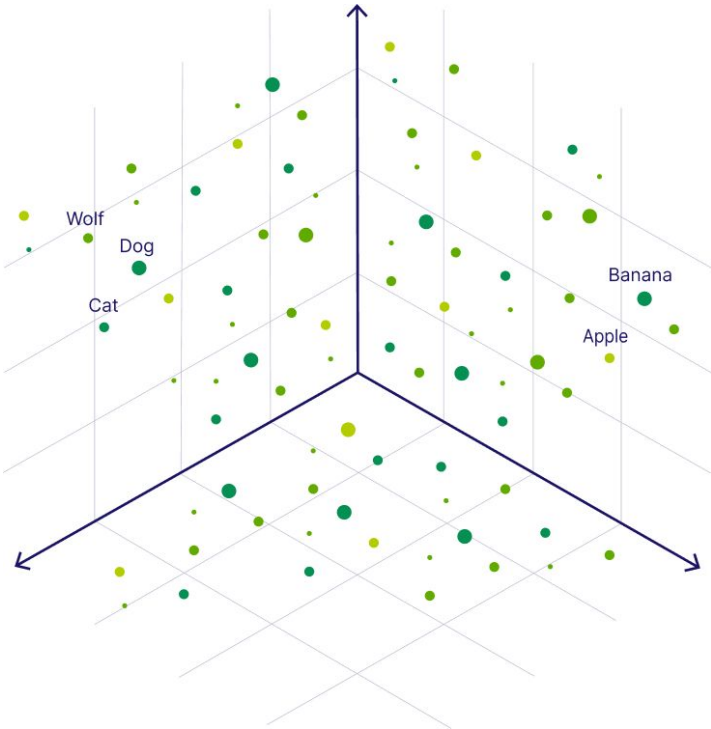
Problem Statement

Can we extract high-quality embeddings directly from LLMs without additional training?

Georgia Tech

Motivation

- Embeddings require extra training and cost
- Useful for various tasks: information retrieval, semantic text similarity, etc.
- Generating better embeddings can have downstream improvements
- If pre-trained LLMs could generate better embeddings without fine-tuning, it would further improve their utility



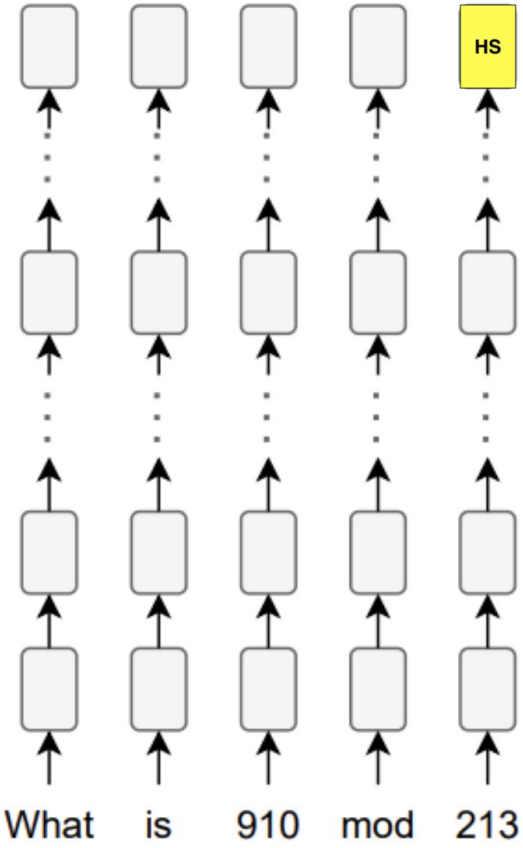
Georgia Tech

Prior Works

- **Word2Vec** - “Efficient Estimation of Word Representations in Vector Space” (Mikolov et al., 2013)
  - Learns word embeddings by predicting surrounding words
- **Sentence-T5** - “Sentence-T5: Scalable Sentence Encoders from Pre-trained Text-to-Text Models” (Ni et al., 2022)
  - Uses a contrastive objective to pull similar sentences together
- **PromptEOL** - “Scaling Sentence Embeddings with Large Language Models” (Muennighoff et al., 2024)
  - Adds a prompt to compress semantics into a hidden state
- **Adaptive Mixture of Experts** - “Adaptive Mixtures of Local Experts” (Jacobs et al., 1991)
  - Introduced Mixture of Experts which breaks down dense FFN

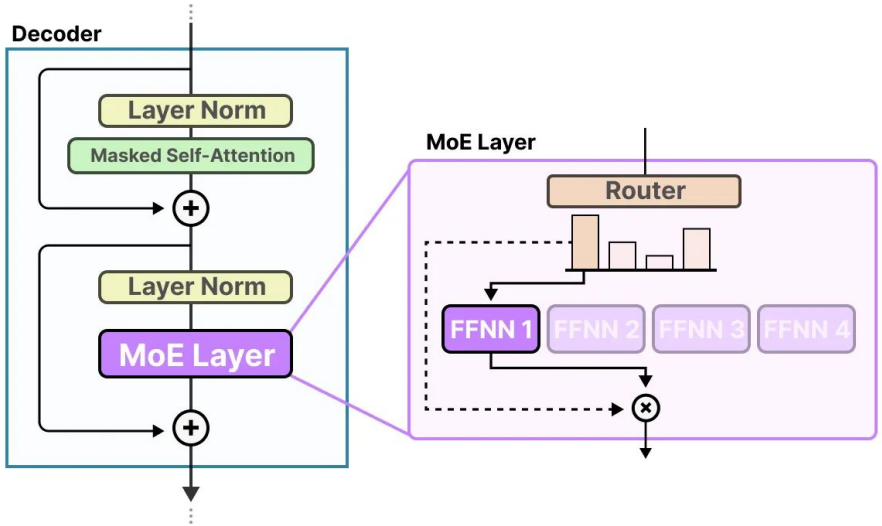
Background - Hidden State (HS)

- Given an input sequence of tokens, the hidden state of a given layer is the vector representation of the last token
- The hidden state of the final layer can represent entire sequence
  - Contains compressed representation of relevant parts via Attention
- Next token generation focus provides more emphasis on overall structure and meaning of sentences



Background - Mixture of Experts

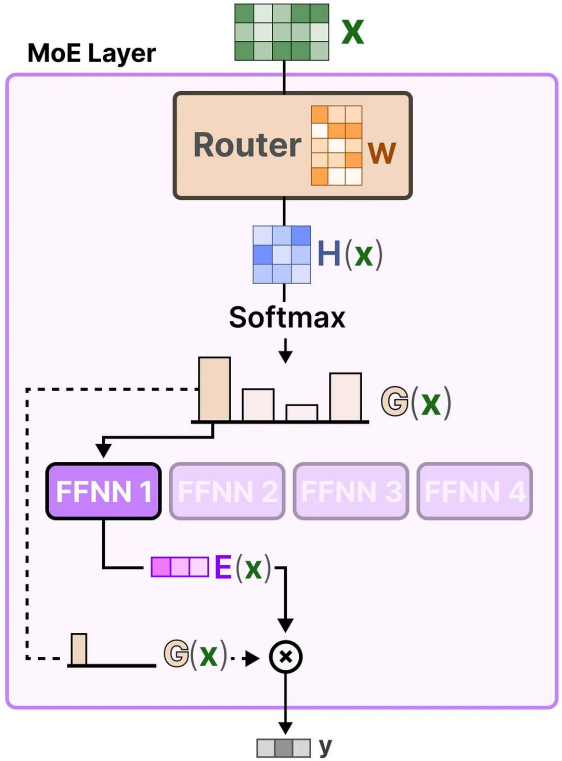
- Each layer has independent “experts” which are smaller feedforward subnetworks
- Each expert specializes in tokens of different domains
- Each layer also has a router with a gating network that chooses which expert(s) will be utilized for each token
- Results from each selected expert and combined together to create the final output of the layer



Georgia Tech

Methods - Routing Weights (RW)

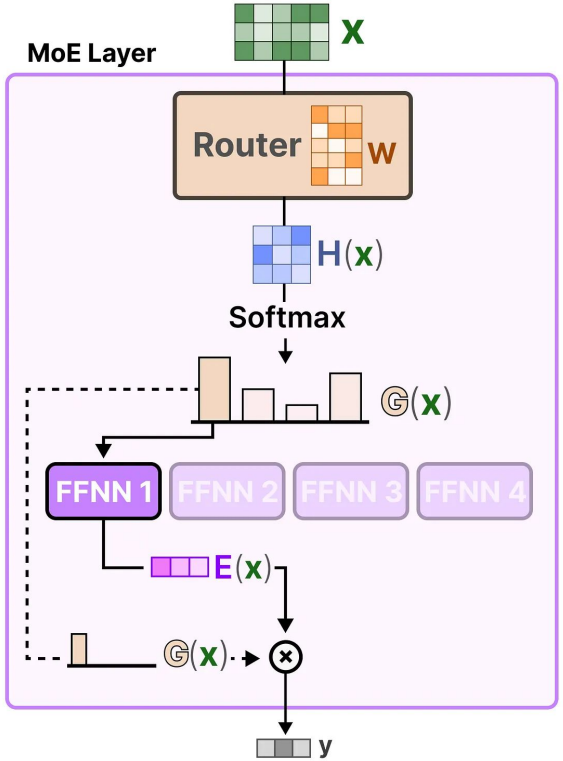
- Training: each router learns parameters for its gating network (linear layer)
- Inference: the hidden state of a layer is matmuled with the gating network ⇒ logits
- Softmax applied on the logits ⇒ probability distribution over the experts (Routing Weights)
  - Note: When using top-k experts only best k experts are chosen with other probabilities set to 0



Georgia Tech

Methods - Routing Weights (RW)

- RWs capture how the input is routed through the model and align token with expert
  - Holistic token understanding
- The RWs from each layer can be concatenated to form an embedding
  - Rich representation accounting for both shallow and deep features with varying levels of abstraction
- Expert selection focus provides more emphasis on capturing deeper contextual changes

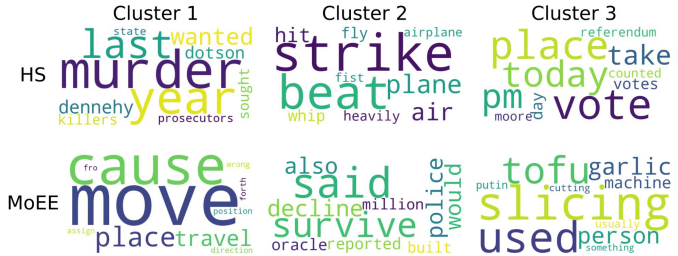


Georgia Tech

Comparative Analysis of RW and HS

- K-means clustering embeddings and perform a correlation analysis
- AMI and NMI ⇒ moderate overlap in clustering
- Low Jaccard Similarity and Exact Matching ⇒ embeddings are different
- Word cloud of top-20 topics further demonstrates this

Metric	Score (*max value*)
Adjusted Mutual Information (AMI)	0.29 (*1.00*)
Normalized Mutual Information (NMI)	0.29 (*1.00*)
Jaccard Similarity	0.06 (*1.00*)
Exact Matching (%)	45.54% (*100.00%*)

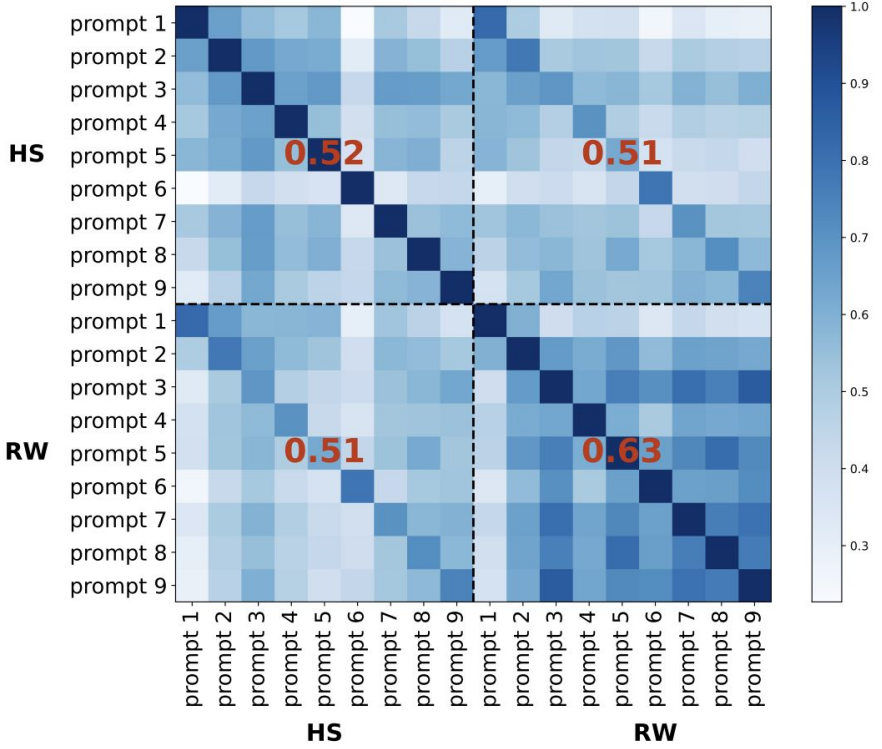


Georgia Tech

Complementary Analysis of RW and HS

- RW is more robust to varying prompts than HS
- RW-HS lowest score indicates the complementary relationship
- RW and HS capture largely unrelated aspects of the data

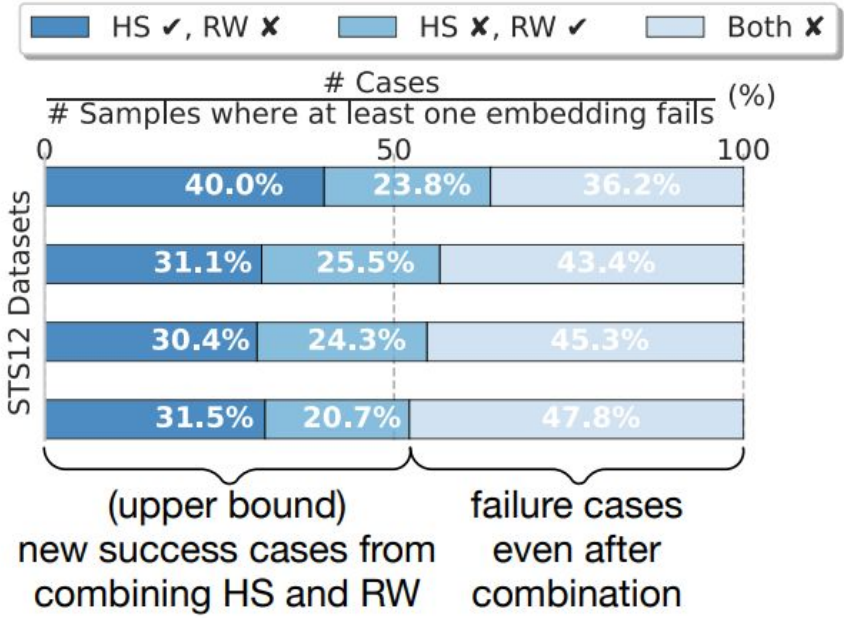
ID	Prompt
1	This sentence: *sent* means in one word:
2	In one word, describe the style of the following sentence - *sent*:
3	In one word, describe the sentiment of the following sentence (positive, neutral, or negative) - *sent*:
4	In one word, describe the tone of the following sentence - *sent* (e.g., formal, informal, humorous, serious):
5	In one word, describe the intent behind the following sentence (e.g., request, suggestion, command) - *sent*:
6	In one word, rate the complexity of the following sentence (simple, moderate, complex) - *sent*:
7	In one word, describe whether the following sentence is subjective or objective - *sent*:
8	In one word, describe the language style of the following sentence (e.g., academic, conversational, literary) - *sent*:
9	In one word, describe the grammatical structure of the following sentence (simple, compound, complex) - *sent*:



Georgia Tech

Complementary Analysis of RW and HS

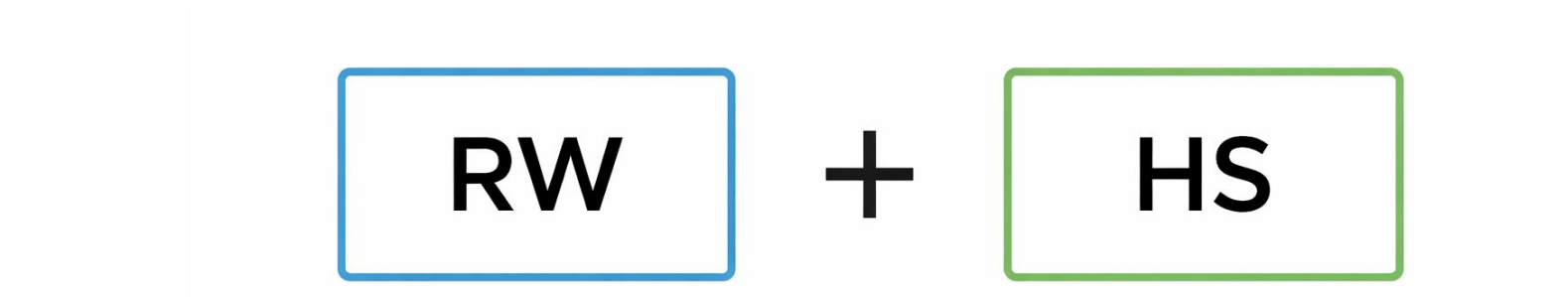
- Error rate of the similarity ranking task
- The proportion of cases where one embedding succeeds and the other fails exceeds 50%, indicating a large room for improvement if combining RW and HS



Georgia Tech

Methods - Mixture of Expert Embeddings (MoEE)

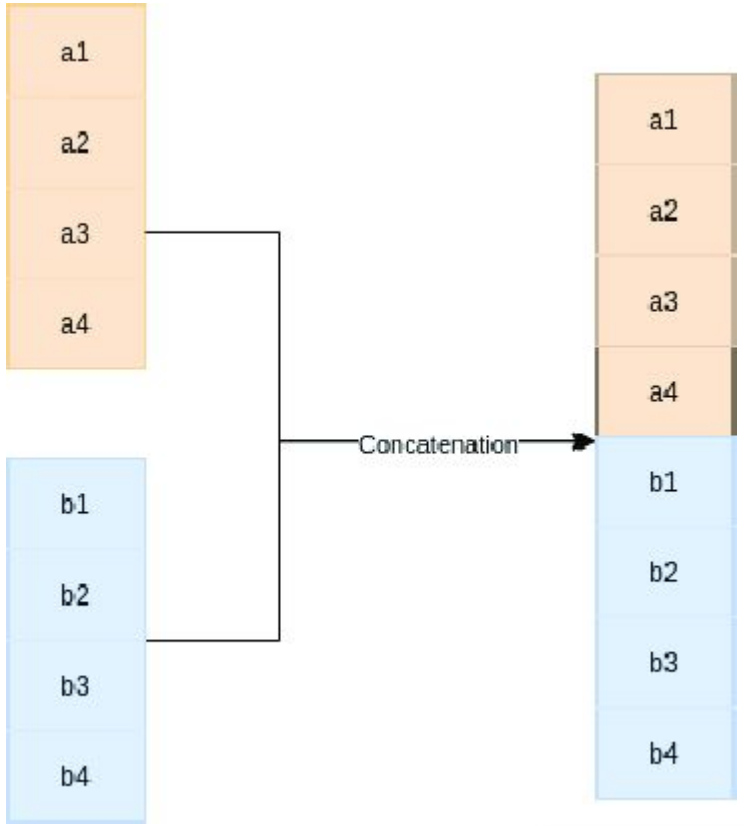
- Given that RW and HS represent different notions with respect to input, they may each contribute distinct information
- Idea: Combine information from RW and HS to provide a more representative embedding



Methods - MoEE(concat)

- Concatenate RW and HS embeddings together
- Preserves the distinct information captured by each component
- Allows downstream tasks to leverage the combined representation

$$\mathbf{e}_{\text{final}} = [\mathbf{e}_{\text{HS}}; \mathbf{e}_{\text{RW}}] \in \mathbb{R}^{d_{\text{HS}} + d_{\text{RW}}}$$



Georgia Tech

Methods - MoEE(sum)

- Take a weighted sum of HS and RW with task-specific adaptation
- Example: Semantic Textual Similarity (STS) on a sentence pair
  - Compute (cosine) similarity under each embedding representation
  - Weighted sum of the two similarity scores to generate the final similarity score

$$\begin{aligned}
\text{sim}_{\text{HS}} &= \text{cosine_similarity}(\mathbf{e}_{\text{HS}}(s_1), \mathbf{e}_{\text{HS}}(s_2)), \\
\text{sim}_{\text{RW}} &= \text{cosine_similarity}(\mathbf{e}_{\text{RW}}(s_1), \mathbf{e}_{\text{RW}}(s_2)) \\
\text{sim}_{\text{final}} &= \text{sim}_{\text{HS}} + \alpha \cdot \text{sim}_{\text{RW}}
\end{aligned}$$

Georgia Tech

Experiments

- Benchmark: Massive Test Embedding Benchmark (MTEB)
  - 20 datasets
- Tasks: Classification, Clustering, Pair Classification, Reranking, Retrieval, Semantic Textual Similarity, Summarization
- Models: DeepSeekMoE-16B, Qwen1.5-MoE-A2.7B, OLMoE-1B-7B
- Compared HS only, RW only, with MOEE version that use concatenate or sum for combining
- Tested embedding extraction without prompt and with PromptEOL

Main Results

- DeepSeekMoE-16B
  - Original HS Averaged Score: 35.36
  - New MOEE Averaged Score: 43.30
- Qwen1.5-MoE-A2.7B
  - Original HS Averaged Score: 35.18
  - New MOEE Averaged Score: 42.25
- OLMoE-1B-7B
  - Original HS Averaged Score: 36.26
  - New MOEE Averaged Score: 42.69

MTEB Tasks	CLF	Clust.	PairCLF	Rerank	STS	Summ.	Avg.
DeepSeekMoE-16b							
Hidden State (HS)	44.79	25.87	44.34	38.13	34.54	24.51	35.36
Routing Weight (RW)	44.06	17.53	50.59	35.94	41.11	26.22	35.91
MoEE (concat)	44.93	24.15	51.88	41.20	46.82	**31.17**	40.03
MoEE (sum)	**48.74**	**32.83**	**52.12**	**47.88**	**48.34**	29.89	**43.30**
Qwen1.5-MoE-A2.7B							
Hidden State (HS)	46.41	24.31	44.43	44.91	28.36	22.65	35.18
Routing Weight (RW)	38.99	10.55	42.26	33.53	23.97	27.44	29.46
MoEE (concat)	44.81	26.75	49.79	49.23	37.93	**27.61**	39.35
MoEE (sum)	**50.70**	**31.35**	**51.87**	**49.82**	**45.75**	24.00	**42.25**
OLMoE-1B-7B							
Hidden State (HS)	44.23	23.79	47.56	45.60	35.44	20.94	36.26
Routing Weight (RW)	43.54	17.66	**53.12**	40.91	44.68	28.68	38.10
MoEE (concat)	44.62	22.83	51.64	46.58	48.84	**31.67**	41.03
MoEE (sum)	**48.54**	**30.67**	50.93	**47.77**	**49.45**	28.77	**42.69**

Main Results

- Compared against self-supervised and supervised methods with the inclusion of PromptEOL
  - This sentence: “ [text] ” means in one word: “
- Competitive with training-based methods even without fine-tuning
  - Only possible when using PromptEOL

Table 4: Performance across MTEB Tasks when *PromptEOL* (Jiang et al., 2023) is applied to MoE. Baselines marked with ⋆ are sourced from the MTEB leaderboard (Muennighoff et al., 2022) and require training.

MTEB Tasks	CLF	Clust.	PairCLF	Rerank	STS	Summ.	Avg.
Self-Supervised Methods							
Glove⋆ (Reimers, 2019)	51.04	23.11	62.90	48.72	60.52	28.87	45.86
Komninos⋆ (Reimers, 2019)	50.21	**24.96**	66.63	50.03	61.73	30.49	47.34
BERT⋆ (Devlin, 2018)	52.36	23.48	66.10	48.47	52.89	29.82	45.52
SimCSE-BERT-unsup⋆ (Gao et al., 2021)	**54.80**	22.59	**70.79**	**52.42**	**75.00**	**31.15**	**51.13**
Supervised Methods							
SimCSE-BERT-sup⋆	**58.98**	29.49	**75.82**	53.61	**79.97**	23.31	53.53
coCondenser-msmarco⋆ (Gao & Callan, 2021)	53.89	**32.85**	74.56	**60.08**	76.41	**29.50**	**54.55**
SPECTER⋆ (Cohan et al., 2020)	42.59	27.94	56.24	55.87	60.68	27.66	45.16
DeepSeekMoE-16b							
Hidden State (HS)	58.24	24.64	48.76	38.13	59.66	24.38	42.30
Routing Weight (RW)	49.52	19.97	68.30	37.48	59.52	**29.26**	44.01
MoEE (concat)	54.21	26.10	**72.44**	53.31	67.59	28.89	50.42
MoEE (sum)	**58.31**	**34.52**	70.95	**55.99**	**70.66**	29.22	**53.28**
Qwen1.5-MoE-A2.7B							
Hidden State (HS)	59.34	29.50	**74.29**	56.51	67.39	23.01	51.67
Routing Weight (RW)	47.84	16.74	64.85	43.55	51.71	27.74	42.07
MoEE (concat)	54.23	27.18	73.93	56.12	68.52	28.57	51.43
MoEE (sum)	**59.57**	**38.33**	72.21	**56.25**	**72.78**	**31.09**	**55.04**
OLMoE-1B-7B							
Hidden State (HS)	**58.18**	32.83	**72.10**	58.31	72.91	27.96	53.72
Routing Weight (RW)	45.02	19.93	61.58	43.91	54.33	29.49	42.38
MoEE (concat)	52.59	33.92	71.85	56.69	71.13	30.21	52.73
MoEE (sum)	57.46	**36.46**	71.26	**60.43**	**74.63**	**30.71**	**55.16**

Georgia Tech

Ablation Study

- One can either use the embeddings corresponding to the last token or use some aggregation across all tokens for the embedding
- Study compares last layer versus average (mean pooling) of all layers
- Study compares last token versus average (mean pooling) of all tokens
- Did not try other aggregation or pooling styles (max, exponential weight, etc)

Table 5: Ablation study on different ways of using routing weights (RW) and hidden state (HS)

STS Datasets	STS12	STS13	STS14	STS15	STS16	Avg.
DeepSeekMoE-16b						
HS - last token, last layer	51.99	**69.56**	**54.68**	**58.04**	**68.47**	**60.40**
HS - last token, all layers	59.82	60.59	45.20	51.08	58.88	55.03
HS - all tokens, last layer	30.95	34.42	26.77	34.90	37.11	32.78
HS - all tokens, all layers	**60.81**	62.46	46.90	52.38	59.99	56.34
RW - last token	**61.97**	**65.86**	**51.38**	**65.86**	**62.49**	**61.18**
RW - all tokens	50.76	46.42	41.47	43.68	48.37	46.03
MoEE (best)	67.39	81.43	68.98	67.76	74.26	71.75

Ablation Study

- When comparing using last token embedding versus aggregation across all tokens embedding, last token often outperforms for both RW and HS
  - Indicates that last token captures most critical information and that aggregation leads to dilution of the critical information and introduces noise
- Utilizing HS from multiple layers does not match performance of RW either
  - Indicates that HS alone cannot replicate the information that RW captures

Table 5: Ablation study on different ways of using routing weights (RW) and hidden state (HS)

STS Datasets	STS12	STS13	STS14	STS15	STS16	Avg.
DeepSeekMoE-16b						
HS - last token, last layer	51.99	**69.56**	**54.68**	**58.04**	**68.47**	**60.40**
HS - last token, all layers	59.82	60.59	45.20	51.08	58.88	55.03
HS - all tokens, last layer	30.95	34.42	26.77	34.90	37.11	32.78
HS - all tokens, all layers	**60.81**	62.46	46.90	52.38	59.99	56.34
RW - last token	**61.97**	**65.86**	**51.38**	**65.86**	**62.49**	**61.18**
RW - all tokens	50.76	46.42	41.47	43.68	48.37	46.03
MoEE (best)	67.39	81.43	68.98	67.76	74.26	71.75

Georgia Tech

Stability Analysis

- Evaluated prompt sensitivity of HS and RW embeddings across 9 different prompts on STS12-16 datasets
- RW often exhibits better performance on each dataset compared to HS
  - Performance is often better (paired analysis would have been better to support this)
- RW often exhibits lower variance on each dataset compared to HS
  - Demonstrates that RW has greater stability and robustness to prompt choice

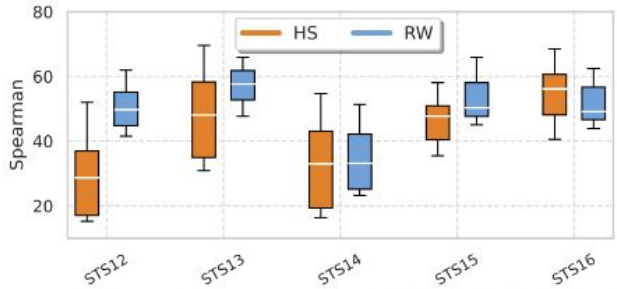


Figure 5: Box plots of the performance of the two embedding methods (RW or HS) using nine different prompts (listed in Table 2) on five STS datasets. The higher variance and wider spread of HS in the box plots indicate its sensitivity to the prompt choice, while RW is more robust (lower variance) with better mean performance.

Georgia Tech

Case Study

- Qualitative analysis of when HS outperforms RW indicates that HS capture formal linguistic consistency
  - Particularly when sentence structure undergoes superficial changes
- Qualitative analysis of when RW outperforms HS indicates that RW capture paraphrasing, synonym use, and nuanced stylistic shifts
  - Captures deeper contextual changes

Table 6: Semantically similar sentence pairs correctly predicted by HS embedding but not by RW embedding. Differences between the sentences are highlighted to show subtle variations that influence prediction outcomes.

	Sentence 1	Sentence 2
1	the vote will take place today at 5.30 p.m	the vote will take place at 17h30
2	the standards are scarcely comparable, let alone transferable	the norms are hardly comparable and still less transferable
3	that provision could open the door wide to arbitrariness	this point of procedure opens the door to the arbitrary
4	A woman puts flour on a piece of meat	A woman is putting flour onto some meat.
5	the fishermen are inactive, tired and disappointed	fishermen are inactive, tired and disappointment

Table 7: Semantically similar sentence pairs correctly predicted by RW embedding but not by HS embedding.

	Sentence 1	Sentence 2
1	He did, but the initiative did not get very far.	What happened is that the initiative does not go very far.
2	then perhaps we could have avoided a catastrophe	we might have been able to prevent a disaster
3	it increases the power of the big countries at the expense of the small countries	it has the effect of augmenting the potency of the big countries to the detriment of babies
4	festive social event, celebration	an occasion on which people can assemble for social interaction and entertainment.
5	group of people defined by a specific profession	organization of performers and associated personnel (especially theatrical).

Takeaways

- HS helps with high level, structural consistency
- RW helps with semantic and more nuanced shifts
- Combining them together via a weighted sum allows to take advantage of both of their strengths
- The routers allow for an embedding signal without training or further fine tuning
- Limitation: Can only be applied to Mixture of Experts models

References

arXiv preprint (PDF)
Li, Z., & Zhou, T. (2024). Your Mixture-of-Experts LLM Is Secretly an Embedding Model for Free. arXiv preprint arXiv:2410.10814. https://doi.org/10.48550/arXiv.2410.10814

Newsletter/Online Guide
Grootendorst, M. (2024, October 7). A visual guide to mixture of experts (MoE). Exploring Language Models Newsletter. https://newsletter.maartengrootendorst.com/p/a-visual-guide-to-mixture-of-experts

Blog Article
Weaviate. (2023, January 16). Vector embeddings explained. Weaviate Blog. https://weaviate.io/blog/vector-embeddings-explained

ResearchGate Figure Reference — Neurosymbolic LLM Paper
Dhanraj, V., & Eliasmith, C. (2025). Improving rule-based reasoning in LLMs via neurosymbolic representations [Preprint]. ResearchGate (Figure illustration). https://www.researchgate.net/figure/A-diagram-of-our-method-showing-how-LLM-hidden-states-are-converted-into-compositional_fig1_388686117

ResearchGate Figure Reference — Feature Concatenation Paper
Amin, H., Darwish, A., Hassanien, A. E., & Soliman, M. (2022). End-to-End deep learning model for corn leaf disease classification [Figure illustration]. ResearchGate. https://www.researchgate.net/figure/Schematic-representation-for-concatenation-of-two-feature-vectors_fig4_359452254

Georgia Tech

YOUR MIXTURE-OF-EXPERTS LLM IS SECRETLY AN EMBEDDING MODEL FOR FREE

Ziyue Li, Tianyi Zhou

Presented by: Siva, Keshav

Motivation

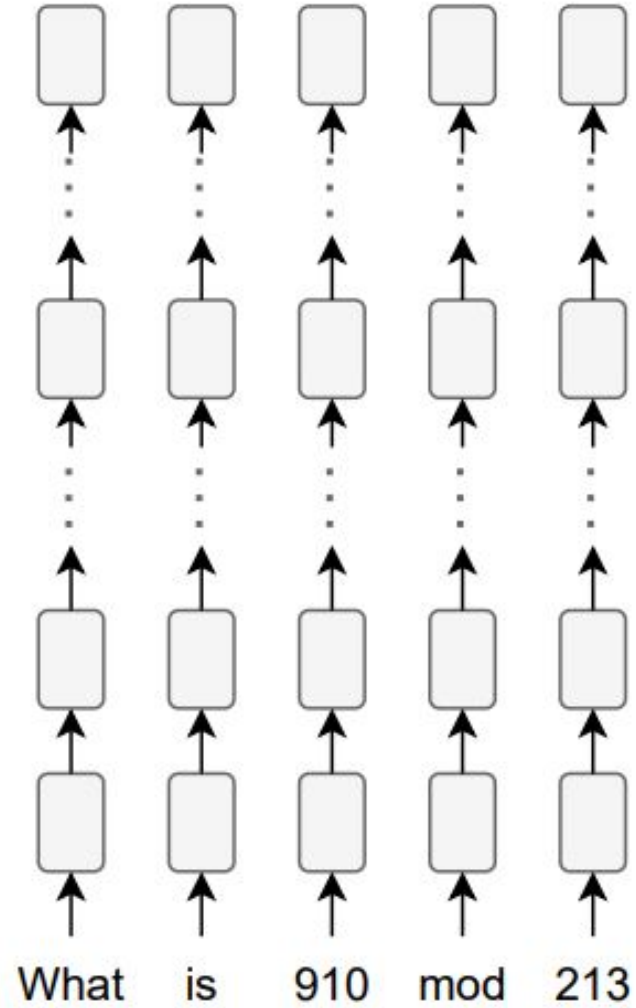
- Embeddings are useful for various NLP tasks: information retrieval, semantic textual similarity, bitext mining, item recommendation
- Generating better embeddings can have downstream improvements on these NLP tasks, and LLMs have been shown to be generate good embeddings [1]
- If pre-trained LLMs could generate better embeddings with no fine-tuning needed, it would further improve the utility of LLMs within NLP in general

Problem Statement

- Can high-quality embeddings be extracted from pre-trained mixture-of-expert LLMs without additional fine-tuning or training?
- How can different embedding representations be utilized in conjunction to effectively generate a representative and high-quality embedding?

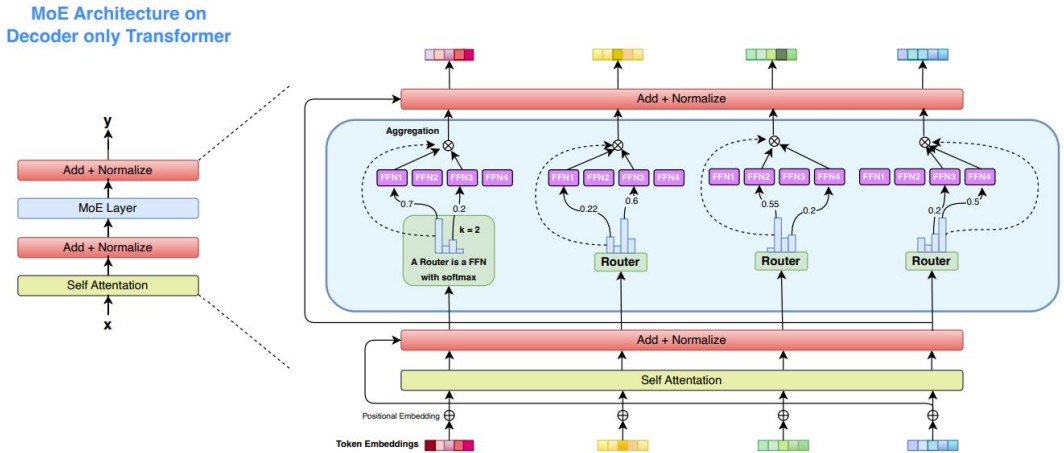
Background

- Hidden State (HS)
  - Given an input sequence $[x_1, \ldots, x_T]$, the hidden state of a given layer is $R^{T \times d}$
  - The last token of the hidden state of the final layer L can represent entire sequence
  - Alternatively, take dimension-invariant aggregation of last layer hidden state



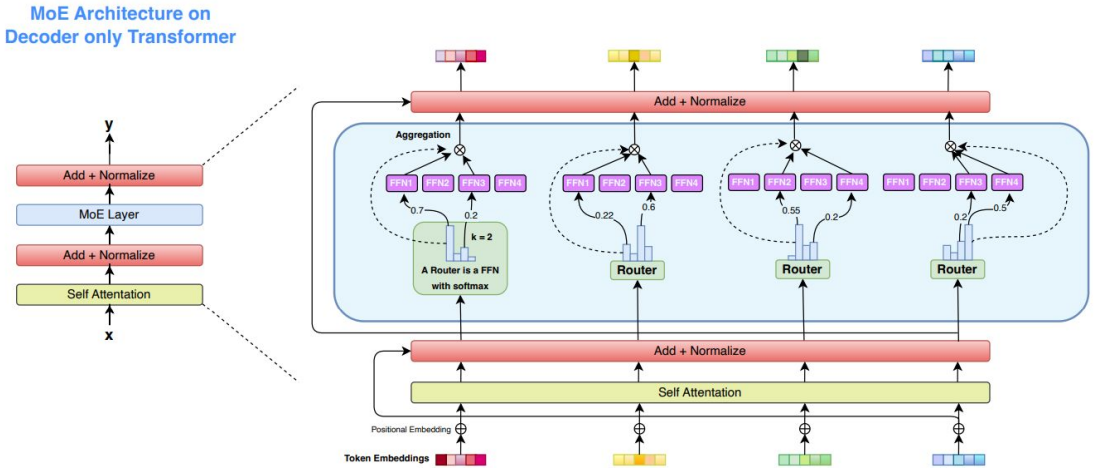
Background

- Mixture of Experts
  - General idea: given E_1, …, E_N experts and a gating functions g_1, …, g_N, the final ensemble output is a weighted sum represented as $y(x) = g_1(x)E_1(x) + … + g_N(x)E_N(x)$
  - MoE layer routes each input to a (sparse) set of expert networks
  - Output is often a sum or average of the output of the selected experts
    - Example of a top-k=2 MoE LLM



Methods

- Extracting Routing Weights (RW)
  - The routing weights can be directly extracted from the softmax output of the logits generated by the feedforward network, and these can be concatenated to form the RW embedding
    - Example below for top-k=2 MoE LLM, note that all probabilities are used for embedding
  - RW embedding captures how the input is routed through the model via the probabilities
    - RW embedding captures information about model holistic interaction with input

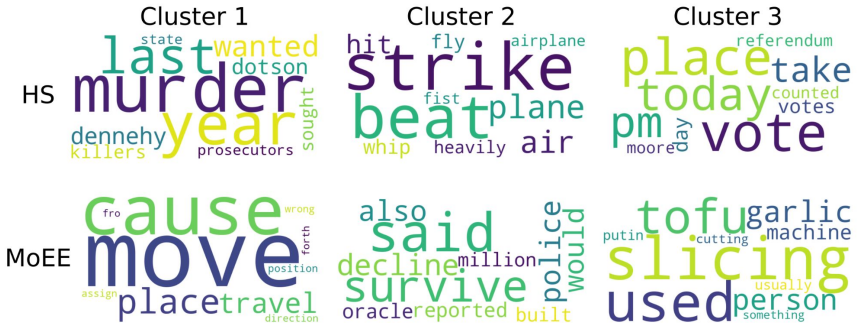


Methods

- Mixture of Expert Embeddings (MoEE) Model: RW + HS
  - Given that RW and HS represent different notions with respect to input, they may each contribute idiosyncratic information that provides a more representative embedding
  - General Idea: combine information from RW and HS
    - Concat: concatenate the embeddings together, which avoids dilution of information
      - This follows the same MoE principle of $y(x) = g_1(x)E_1(x) + \ldots + g_N(x)E_N(x)$
      - g_1 is 1 and g_2 is 1
      - E_1 generates the HS embedding with 0 post-padding
      - E_2 generates the RW embedding with 0 pre-padding

Comparative Analysis of RW and HS

Metric	Score (*max value*)
Adjusted Mutual Information (AMI)	0.29 (*1.00*)
Normalized Mutual Information (NMI)	0.29 (*1.00*)
Jaccard Similarity	0.06 (*1.00*)
Exact Matching (%)	45.54% (*100.00%*)



Complimentary Analysis of RW and HS

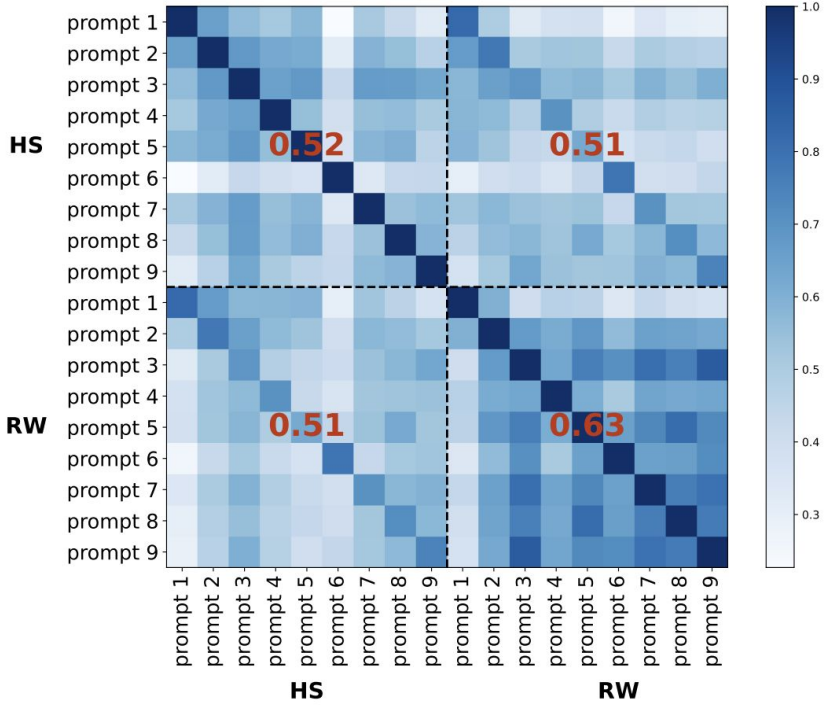


Table 2: Prompts used in Fig 4-5.

ID	Prompt
1	This sentence: *sent* means in one word:
2	In one word, describe the style of the following sentence - *sent*:
3	In one word, describe the sentiment of the following sentence (positive, neutral, or negative) - *sent*:
4	In one word, describe the tone of the following sentence - *sent* (e.g., formal, informal, humorous, serious):
5	In one word, describe the intent behind the following sentence (e.g., request, suggestion, command) - *sent*:
6	In one word, rate the complexity of the following sentence (simple, moderate, complex) - *sent*:
7	In one word, describe whether the following sentence is subjective or objective - *sent*:
8	In one word, describe the language style of the following sentence (e.g., academic, conversational, literary) - *sent*:
9	In one word, describe the grammatical structure of the following sentence (simple, compound, complex) - *sent*:

Methods

- Mixture of Expert Embeddings (MoEE) Model: RW + HS
  - Given that RW and HS represent different notions with respect to input, they may each contribute idiosyncratic information that provides a more representative embedding
  - General Idea: combine information from RW and HS
    - Sum: compute (cosine) similarity under each embedding representation, followed by a weighted sum of the two (cosine) similarity scores to generate the final similarity score
      - Formally, sim = $\text{cosSim}(HS_1, HS_2) + \boldsymbol{\alpha}\text{cosSim}(RW_1, RW_2)$
      - This follows the same MoE principle of $y(x) = g_1(x)E_1(x) + \ldots + g_N(x)E_N(x)$
      - g_1 is 1 and g_2 is $\boldsymbol{\alpha}$ (pre-specified hyperparameter, could be inferred?)
      - E_1 generates the HS embeddings and computes cosine similarity
      - E_2 generates the RW embeddings and computes the cosine similarity

Experiments

- Benchmark: Massive Test Embedding Benchmark (MTEB)
  - 20 datasets
- Tasks: Classification, Clustering, Pair Classification, Reranking, Retrieval, Semantic Textual Similarity, Summarization
- Models: DeepSeekMoE-16B, Qwen1.5-MoE-A2.7B, OLMoE-1B-7B
- Compared HS only, RW only, with MOEE version that use concatenate or sum for combining
- Tested embedding extraction without prompt and with PromptEOL

Main Results

- DeepSeekMoE-16B
  - Original HS Averaged Score: 35.36
  - New MOEE Averaged Score: 43.30
- Qwen1.5-MoE-A2.7B
  - Original HS Averaged Score: 35.18
  - New MOEE Averaged Score: 42.25
- OLMoE-1B-7B
  - Original HS Averaged Score: 36.26
  - New MOEE Averaged Score: 42.69
- Competitive with training based embedding methods even without fine-tuning

MTEB Tasks	CLF	Clust.	PairCLF	Rerank	STS	Summ.	Avg.
DeepSeekMoE-16b							
Hidden State (HS)	44.79	25.87	44.34	38.13	34.54	24.51	35.36
Routing Weight (RW)	44.06	17.53	50.59	35.94	41.11	26.22	35.91
MoEE (concat)	44.93	24.15	51.88	41.20	46.82	**31.17**	40.03
MoEE (sum)	**48.74**	**32.83**	**52.12**	**47.88**	**48.34**	29.89	**43.30**
Qwen1.5-MoE-A2.7B							
Hidden State (HS)	46.41	24.31	44.43	44.91	28.36	22.65	35.18
Routing Weight (RW)	38.99	10.55	42.26	33.53	23.97	27.44	29.46
MoEE (concat)	44.81	26.75	49.79	49.23	37.93	**27.61**	39.35
MoEE (sum)	**50.70**	**31.35**	**51.87**	**49.82**	**45.75**	24.00	**42.25**
OLMoE-1B-7B							
Hidden State (HS)	44.23	23.79	47.56	45.60	35.44	20.94	36.26
Routing Weight (RW)	43.54	17.66	**53.12**	40.91	44.68	28.68	38.10
MoEE (concat)	44.62	22.83	51.64	46.58	48.84	**31.67**	41.03
MoEE (sum)	**48.54**	**30.67**	50.93	**47.77**	**49.45**	28.77	**42.69**

Ablation Study

- As previously mentioned, one can either use the embeddings corresponding to the last token or use some aggregation across all tokens for the embedding
- Study compares last layer versus average (mean pooling) of all layers
- Study compares last token versus average (mean pooling) of all tokens
- Did not try other aggregation or pooling styles

Table 5: Ablation study on different ways of using routing weights (RW) and hidden state (HS)

STS Datasets	STS12	STS13	STS14	STS15	STS16	Avg.
DeepSeekMoE-16b						
HS - last token, last layer	51.99	**69.56**	**54.68**	**58.04**	**68.47**	**60.40**
HS - last token, all layers	59.82	60.59	45.20	51.08	58.88	55.03
HS - all tokens, last layer	30.95	34.42	26.77	34.90	37.11	32.78
HS - all tokens, all layers	**60.81**	62.46	46.90	52.38	59.99	56.34
RW - last token	**61.97**	**65.86**	**51.38**	**65.86**	**62.49**	**61.18**
RW - all tokens	50.76	46.42	41.47	43.68	48.37	46.03
MoEE (best)	67.39	81.43	68.98	67.76	74.26	71.75

Ablation Study

- When comparing using last token embedding versus aggregation across all tokens embedding, last token often outperforms for both RW and HS
  - Indicates that last token captures most critical information, and that aggregation procedure leads to dilution of the critical information and introduces noise
- Utilizing HS from multiple layers does not match performance of RW either
  - Indicates that HS alone cannot replicate the information that RW captures
  - Performance difference seems fairly narrow though (60.40 versus 61.18)

Table 5: Ablation study on different ways of using routing weights (RW) and hidden state (HS)

STS Datasets	STS12	STS13	STS14	STS15	STS16	Avg.
DeepSeekMoE-16b						
HS - last token, last layer	51.99	**69.56**	**54.68**	**58.04**	**68.47**	**60.40**
HS - last token, all layers	59.82	60.59	45.20	51.08	58.88	55.03
HS - all tokens, last layer	30.95	34.42	26.77	34.90	37.11	32.78
HS - all tokens, all layers	**60.81**	62.46	46.90	52.38	59.99	56.34
RW - last token	**61.97**	**65.86**	**51.38**	**65.86**	**62.49**	**61.18**
RW - all tokens	50.76	46.42	41.47	43.68	48.37	46.03
MoEE (best)	67.39	81.43	68.98	67.76	74.26	71.75

Stability Analysis

- Evaluated prompt sensitivity of HS and RW embeddings across 9 different prompts on STS12-16 datasets
- RW often exhibits better performance on each dataset compared to HS
  - Performance is often better (paired analysis would have been better to support this)
- RW often exhibits lower variance on each dataset compared to HS
  - Demonstrates that RW has greater stability and robustness to prompt choice

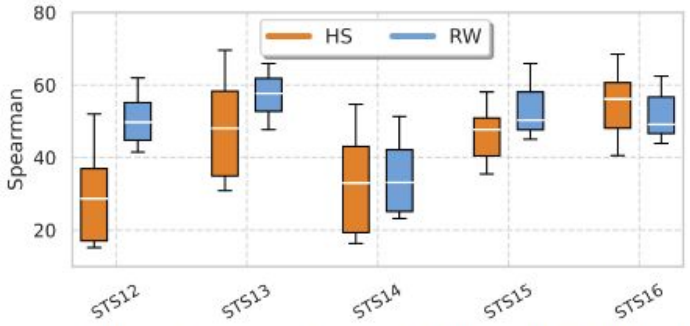


Figure 5: Box plots of the performance of the two embedding methods (RW or HS) using nine different prompts (listed in Table 2) on five STS datasets. The higher variance and wider spread of HS in the box plots indicate its sensitivity to the prompt choice, while RW is more robust (lower variance) with better mean performance.

Case Study

- Qualitative analysis of when HS outperforms RW indicates that HS capture formal linguistic consistency
  - Particularly when sentence structure undergoes superficial changes
- Qualitative analysis of when RW outperforms HS indicates that RW capture paraphrasing, synonym use, and nuanced stylistic shifts
  - Captures deeper contextual changes

Table 6: Semantically similar sentence pairs correctly predicted by HS embedding but not by RW embedding. Differences between the sentences are highlighted to show subtle variations that influence prediction outcomes.

	Sentence 1	Sentence 2
1	the vote will take place today at 5.30 p.m	the vote will take place at 17h30
2	the standards are scarcely comparable, let alone transferable	the norms are hardly comparable and still less transferable
3	that provision could open the door wide to arbitrariness	this point of procedure opens the door to the arbitrary
4	A woman puts flour on a piece of meat	A woman is putting flour onto some meat.
5	the fishermen are inactive, tired and disappointed	fishermen are inactive, tired and disappointment

Table 7: Semantically similar sentence pairs correctly predicted by RW embedding but not by HS embedding.

	Sentence 1	Sentence 2
1	He did, but the initiative did not get very far.	What happened is that the initiative does not go very far.
2	then perhaps we could have avoided a catastrophe	we might have been able to prevent a disaster
3	it increases the power of the big countries at the expense of the small countries	it has the effect of augmenting the potency of the big countries to the detriment of babies
4	festive social event, celebration	an occasion on which people can assemble for social interaction and entertainment.
5	group of people defined by a specific profession	organization of performers and associated personnel (especially theatrical).

Takeaways

- HS helps with high level, structural consistency
- RW helps with semantic and more nuanced shifts
- Combining them together via a weighted sum allows to take advantage of both of their strengths
- The routers allow for an embedding signal without training or further fine tuning
- Limitation: Can only be applied to Mixture of Experts models

References

-