SASNet:
Dynamic LLM Embeddings and Siamese Transformer Network for Recommendation Systems

Adam Amer and **Berat Çabuk** and **Federico Chinello** and **Denis Rotov**

Abstract

Recommendation systems are essential in Natural Language Processing due to their vast business potential. Traditional methods, such as content-based and collaborative filtering, struggle to capture the full richness of user preferences. Common issues include overspecialization, the cold-start problem, and the need for in-depth item knowledge. We introduce a hybrid model to address these challenges. Using a quantized Large Language Model (LLM), Phi 3 3.8B-Q4-GPTQ, with custom prompt instructions, we create richer embeddings. These embeddings are input to our Siamese AttentionSetNet (SASNet), an original Transformer-based architecture that creates dynamic embeddings for users and activities by contextually relating multiple reviews. The user rating serves as the label for categorical cross-entropy loss between constructed embeddings. Our custom data generator enhances learning by recombining training reviews into diverse sets each epoch. We evaluate our model on reviews from the Yelp dataset. Although our SASNet does not outperform the baseline using TF-IDF embeddings and a Support Vector Machine (SVM) classifier, our LLM embeddings capture greater semantic similarity than state-of-the-art sentence embedders. Using SVM with our embeddings outperforms the baseline by 12.25%. This study demonstrates that leveraging quantized LLMs and multiple embeddings can significantly enhance recommendation accuracy, even though the SASNet did not perform as expected on this dataset. This discrepancy is likely due to the dataset size, as we have achieved much greater performance on larger datasets with images.

1 Introduction

Recommendation systems are central in Natural Language Processing (NLP). Traditional approaches can be divided into two categories: content-based and collaborative filtering (Roy and Dutta, 2022). However, these methods often struggle with critical challenges. They face issues with **overspecialization**, which hinders recommendations for items with nuanced differences. The **cold-start** problem arises due to a lack of user information. Additionally, content-based methods struggle when there is a **lack of in-depth information** about item characteristics (Mishra et al., 2021).

We address these challenges with SASNet. It uses the textual context of Large Language Models (LLM) to infer nuanced differences between items and captures user similarity through its Siamese architecture. Our implicit augmentation on sets enables effective training on small datasets and alleviates the cold-start problem.

We use a highly efficient model (**Phi 3 3.8B**) (Abdin et al., 2024), quantized at the 4th bit with Accurate Post-Training Quantization (Q4-GPTQ) (Frantar et al., 2023). Inspired by previous research ((Su et al., 2023), (Jiang et al., 2022)), we employ **custom prompt instructions** to create more meaningful embeddings for our recommendation system. Our approach uses 3072-dimensional embeddings created by the LLM, combined with a **Siamese AttentionSetNet (SASNet) network** that features a novel Transformer-based architecture. This architecture **dynamically generates embeddings** for entities (users and businesses) by **contextually integrating information** from multiple documents (reviews). We use the user rating for the business as the label for the classification loss between constructed embeddings.

We evaluate SASNet on reviews from the Yelp Open Dataset (Yelp, Inc.). Our LLM embeddings capture greater semantic similarity than state-of-the-art sentence embedders, and using SVM with our embeddings outperforms the TFIDF+SVM baseline by 12.25%. However, SASNet performs worse by 5% compared to the TFIDF+SVM baseline.

Our study demonstrates that leveraging LLMs and multiple embeddings can significantly enhance recommendation accuracy. While SASNet did not perform as expected on this dataset, it shows promise for applications with larger datasets. By incorporating custom prompt instructions and a dynamic embedding strategy, we provide a more sophisticated and contextually aware recommendation system, improving user satisfaction and engagement.

2 Experiments

2.1 Data overview

We used data from the Yelp Open Dataset (Yelp, Inc.), which includes over 6.2 million reviews of 150,000 businesses by 1.9 million users. Each review links to a business ID and a user ID. Initially, we filtered the reviews to include only users with at least 10 reviews and businesses with at least 20 reviews. This reduced the dataset to 2,787,011 reviews from 103,251 users and 61,272 businesses. To create a manageable and overlapping dataset, we used linear programming to minimize the number of reviews while ensuring at least 7,000 users with 10 reviews and 3,000 businesses with 20 reviews. This process resulted in a sample of 75,379 reviews from 11,117 users and 3,752 businesses. The dataset is highly imbalanced, with over 75% of reviews rated above 4 stars. To address this, we shifted from regression (predicting ratings 1-5) to classification (predicting bad ($\leq$ 3 stars), good (4 stars), or excellent (5 stars)). Finally, we split the dataset by users into training (80% = 60k reviews), validation (10% = 7.5k reviews), and test sets (10% = 7.5k reviews).

2.2 Embedding construction

We compared different methods of embedding reviews, including TFIDF, a state-of-the-art Sentence Transformer (all-MiniLM-L6-v2), and our LLM token embeddings extracted from the last hidden state before the classification head. Since the Sentence Transformer context window is limited to 512 tokens, we truncated longer reviews. We focused on LLM token embeddings and developed a methodology to achieve the best sentence embeddings.

Our chosen LLM is an instruct model. We created non-instructed embeddings by embedding the single review and instructed embeddings using a system prompt: "Analyze the following review, focusing on the aspects discussed and the associated sentiment." For both types of embeddings, we considered four methods to construct sentence embeddings:

1. **Global Average Pooling** across token embeddings.
2. **Global Max Pooling** across token embeddings.
3. **Last Token Embedding** in a Causal LM, as later tokens have more meaningful representations.
4. **Global Weighted Average Pooling** with a weight vector based on the normalized token index.

For instructed embeddings with stored attention scores, we also created **Attentional Average Pooling**. We averaged attention scores from all heads in the last transformer block to form a single attention matrix, representing the relevance of each token to others. Multiplying this matrix by the token embedding matrix produced contextualized token embeddings, which we then averaged.

We used Z-score normalization across all reviews to make embeddings comparable between methods. This normalization was necessary due to the higher dimensionality of LLM embeddings compared to the Sentence-Transformer. It also helped remove the "average review" effect, allowing us to capture differences between users, businesses, sentiment, and categories.

2.3 Embedding Analysis

To determine the best embeddings for training, we evaluated the quality of the embeddings using two key metrics: understanding user sentiment and identifying the type of activity discussed. We computed an average embedding for each sentiment or category class and then calculated the cosine similarity between the average embeddings of different classes.

Sentiment Analysis

For sentiment analysis, we measured the cosine similarity between the "bad" and "excellent" embeddings. We considered the true label to be -1 (maximum dissimilarity) and we computed the absolute error.

Category Understanding

For category understanding, we addressed the complexity of comparing embeddings from different categories, including main categories and subcategories (e.g., Food, Pizza). We used the Sentence Transformer (all-MiniLM-L6-v2) to embed the category names and computed the cosine similarity of these embeddings. This provided an estimate of the "true" cosine similarity across reviews based on their categories. We used Mean Square Error and Pearson's Correlation to estimate how close predicted cosine similarity was to "true" cosine similarity.

Metrics

To make the metrics comparable, we normalized them using min-max normalization. The overall score for each embedding type was calculated as:

$$s = (-0.25 \times \text{NMSE}_c) + (0.25 \times \text{NPR}_c) - (0.5 \times \text{NAE}_s)$$

2.4 Architecture Overview

Our custom Siamese architecture consists of an initial sequential layer to decrease embedding size, a Self-Attention Transformer block, shared across the user and business, which relates multiple input reviews and "contextualizes" them, creating a single embedding per entity. A Cross-Attention Transformer block that let user embeddings attend to business embeddings, and a Classification Layer that classifies the obtained embeddings into three classes (bad, good, excellent).

1. **Self-Attention Transformer**: The input to the Transformer Core is a tensor $n \times s \times d$, where n is the batch size, s is the number of input reviews per entity (user/business), and d is the embedding dimensionality. The Transformer Core includes

two transformer blocks based on 4 heads with dot-product self-attention. Static embeddings, computed from the LLM, do not capture contextual information between different reviews of the same entity. The embeddings are enriched with contextual information from other reviews, becoming ”dynamic”. Global Average Pooling is performed across embeddings to obtain a single tensor of dimension $n \times d$

2. **Cross-Attention Transformer**: In the cross attention transformer, the business embedding is used as key, value for the user embedding, in order to capture business-characteristics and user preferences.

3. **Classification Layer**: Finally, in the classification layer, a simple MLP that classifies the flattened dynamic embeddings into *bad, good, or excellent.*

2.5 Experimental Setting

To better evaluate our implementation's performance, we constructed a baseline model using a Support Vector Machine (SVM) classifier with TF-IDF vectors as input. We used the same train and test splits as the deep learning models but selected a subset of the dataset, maintaining the split ratios while reducing the total samples to 4000 to manage training time. In preprocessing, we initialized the TF-IDF vectorizer on the training set and transformed all review texts into TF-IDF vectors. We averaged these vectors by user and business, then used the averages as input to the SVM classifier. Additionally, we implemented a more sophisticated baseline using the contextual embeddings from Phi3-Instructed as input for the SVM classifier. Our main model, the SASNet architecture, is a large and sophisticated transformer-based architecture with 3.3M parameters.

3 Results

3.1 Embeddings

In Table 1, we observe that the state-of-the-art sentence embedder all-MiniLM outperforms the best model using uninstructed Phi3 (-0.35 vs. -0.41). However, every Phi3 model with instructions significantly outperforms the sentence embedder. The worst Phi3 model with instructions, using GMP, has a raw score of -0.10. The attention construction achieves the best performance, outperforming the all-MiniLM by 263% in normalized score (1.00 vs. 0.38).

3.2 Classification

As reported in Table 2, the baseline model shows 57% accuracy and 57% F1 score on test data when trained on a set of 3200 reviews, and tested on 400. Accuracy and F1 score increase to 69.25% and 69.25%, respectively when using the more sophisticated baseline with LLM embeddings. We conclude that the model performs decently well, also in light of the small amount of data it was trained on.

4 Discussion

4.1 Statistical Significance

We assessed the statistical significance of our embedding analysis results using the F-test for Mean Squared Error (MSE) and the Z-test for Pearson's correlation. The Phi3-Instructed Attentional embedding method showed statistically significant differences compared to all other methods, confirming its robustness and reliability. This demonstrates that the superior performance of these embeddings is not due to random chance.

4.2 Limitations

Our Siamese AttentionSetNet (SASNet) did not perform as well as anticipated. Despite the sophisticated architecture and the promising results of the LLM embeddings, SASNet underperformed compared to a simpler TFIDF+SVM baseline. This discrepancy may be attributed to several factors:

Dataset Size: The small size of the dataset likely limited the performance of SASNet. Larger datasets could better leverage the dynamic embeddings and sophisticated data augmentation techniques inherent in SASNet.

Embedding Computation Cost: The small dataset was influenced from the cost in computation of LLM embeddings. For instance, embedding our relatively small dataset of 75,000 reviews required several hours of computation on an NVIDIA A100 GPU. In contrast, the all-MiniLM model required less than an hour on a laptop CPU. This computational cost, not onaffected our ability to scale the training properly.

Architecture Complexity: While SASNet's architecture is theoretically sound and has shown competitive performance in other tasks (e.g., achieving 87% accuracy in a CV classification task across 22 classes), the complexity might necessitate more data for effective training. The sophisticated architecture may also introduce challenges in capturing the nuances of textual data compared to image data.

Limited Testing for SVM: Our SVM baseline results, while impressive, were based on a subset of only 4,000 reviews from the 60,000 available in the training set due to computational constraints. Although the distribution of this subset was similar to the full dataset, it does not guarantee that performance would scale proportionally with more data.

5 Related Work

Our work introduces a hybrid model that addresses classical challenges in recommender systems, including overspecialization, the cold-start problem, and the need for in-depth item knowledge.

Recommender Sytems Challenges: Classical recommender systems are divided into two main categories: content-based and collaborative filtering (Roy and Dutta, 2022). Content-based systems recommend items with similar properties to those the user has liked, while collaborative filters cluster users (or items) and recommend items liked by users with similar profiles. These approaches face several challenges: capturing nuanced differences (overspecialization) (Mishra et al., 2021), addressing the cold-start problem where users have few reviews, and requiring in-depth item knowledge (Chen et al., 2015).

Our hybrid model addresses these challenges effectively (Murillo et al., 2022). It uses semantic LLM embeddings from text, inherently handling content orientation and mitigating the cold-start problem. Additionally, training on a Siamese network captures similarities between users, combining the strengths of both methods. Previous works have used textual embeddings in recommender systems (Kanwal et al., 2021), such as extracting sentiment from reviews (Betancourt and Ilarri, 2020) or aspect-based sentiment analysis (Bauman et al., 2017), yielding significant performance improvements.

LLM for review embedding: We embed the reviews using a quantized LLM, which offers several advantages over BERT based embeddings. Our LLM, Phi 3 3.8B-Q4-GPTQ, supports a 4k token context window (Abdin et al., 2024), compared to BERT's 512-token limit (Devlin et al., 2019), enabling better capture of meaning and nuances. LLMs also demonstrate superior semantic similarity capture compared to classical transformers like BERT (Freestone and Santu, 2024). The emergence of quantized LLMs, which maintain performance with reduced size (e.g., trimming weights to smaller bit representation) (Li et al., 2024), allows us to leverage these advantages while running the model on a mobile device with a size of 1.8 GB.

Sets and Siamese Network: Our SASNet Transformer architecture processes unordered sets of reviews, a principle seen in DeepSets (Zaheer et al., 2017) and PointNet (Qi et al., 2017). This enables sophisticated embedding of users and businesses by leveraging the transformer's attention mechanism to identify relevant reviews and aspects. We then incorporate user embedding similarities in a Siamese-like architecture, where two inputs from the same class are processed through parallel, identical subnetworks to learn associations via categorical cross-entropy loss (Bromley et al., 1993). This novel approach in recommender systems, though sparsely implemented to date (Serrano and Bellogín, 2023), shows promise in enhancing recommendation performance.

6 Conclusion

SASNet is a novel architecture for recommender systems that leverages the textual context of LLMs to infer nuanced differences between items. Additionally, it captures user similarity through its Siamese architecture. The proposed implicit augmentation on sets enables significant performance improvements even on small datasets, thereby alleviating the cold-start problem. We demonstrated the capability of quantized LLMs, whose embeddings captured semantic similarity **263% better** than all-MiniLM. Our promosed architecture, SASNet achieved a test performance of 52% in classifying a user's preference for an activity (bad, good, excellent). Despite this performance is worse than the one of our baselines, probably due to the low amount of available data and the complexity of SASNet architecture, we believe that our experiments represent a good starting point for further research. Possible improvements include incorporating more detailed information about users and activities into our model, such as activity location and images from the Yelp dataset.

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Caio César Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Parul Chopra, Allie Del Giorno, Gustavo de Rosa, Matthew Dixon, Ronen Eldan, Dan Iter, Amit Garg, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Jamie Huynh, Mojan Javaheripi, Xin Jin, Piero Kauffmann, Nikos Karampatziakis, Dongwoo Kim, Mahoud Khademi, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Chen Liang, Weishung Liu, Eric Lin, Zeqi Lin, Piyush Madan, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Xia Song, Masahiro Tanaka, Xin Wang, Rachel Ward, Guanhua Wang, Philipp Witte, Michael Wyatt, Can Xu, Jiahang Xu, Sonali Yadav, Fan Yang, Ziyi Yang, Donghan Yu, Chengruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. 2024. Phi-3 technical report: A highly capable language model locally on your phone.

Konstantin Bauman, Bing Liu, and Alexander Tuzhilin. 2017. Aspect based recommendations: Recommending items with the most valuable aspects based on user reviews. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 717–726.

Yanelys Betancourt and Sergio Ilarri. 2020. Use of text mining techniques for recommender systems. In *Proceedings of the 22nd International Conference on Enterprise Information Systems (ICEIS 2020) - Volume 1*, pages 780–787.

Jane Bromley, J.W. Bentz, Leon Bottou, I. Guyon, Yann Lecun, C. Moore, Eduard Sackinger, and R. Shah. 1993. Signature verification using a siamese time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(4).

Li Chen, Guanliang Chen, and Feng Wang. 2015. Recommender systems based on user reviews: the state of the art. *User Modeling and User-Adapted Interaction*, 25(2):99–154.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023. Gptq: Accurate post-training quantization for generative pre-trained transformers.

Matthew Freestone and Shubhra Kanti Karmaker Santu. 2024. Word embeddings revisited: Do llms offer something new?

Ting Jiang, Jian Jiao, Shaohan Huang, Zihan Zhang, Deqing Wang, Fuzhen Zhuang, Furu Wei, Haizhen Huang, Denvy Deng, and Qi Zhang. 2022. PromptBERT: Improving BERT sentence embeddings with prompts. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8826–8837, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Safia Kanwal, Sidra Nawaz, Muhammad Kamran Malik, and Zubair Nawaz. 2021. A review of text-based recommendation systems. *IEEE Access*, 9:31638–31652.

Shiyao Li, Xuefei Ning, Luning Wang, Tengxuan Liu, Xiangsheng Shi, Shengen Yan, Guohao Dai, Huazhong Yang, and Yu Wang. 2024. Evaluating quantized large language models.

Nitin Mishra, Saumya Chaturvedi, Aanchal Vij, and Sunita Tripathi. 2021. Research problems in recommender systems. *Journal of Physics: Conference Series*, 1717:012002.

Victor Giovanni Morales Murillo, David Eduardo Pinto Avendaño, Franco Rojas Lopez, and Juan Manuel Gonzales Calleros. 2022. A systematic literature review on the hybrid approaches for recommender systems. *Computación y Sistemas*, 26(1):357–372.

Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85.

Deepjyoti Roy and Mala Dutta. 2022. A systematic review and research perspective on recommender systems. *Journal of Big Data*, 9(1):59.

Nicolás Serrano and Alejandro Bellogín. 2023. Siamese neural networks in recommendation. *Neural Computing and Applications*, 35:13941–13953.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2023. One embedder, any task: Instruction-finetuned text embeddings. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1102–1121, Toronto, Canada. Association for Computational Linguistics.

Yelp, Inc. Yelp open dataset. `https://www.yelp.com/dataset`. Accessed: 2024-05-26.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbhakhsh, Barnabás Póczos, Ruslan Salakhutdinov, and Alexander J. Smola. 2017. Deepsets. *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS)*.

7 Appendix

Model	NMSE/NPER/ NAE	Score/ Normalized
Phi3-Instructed		
Attentional	0.13/1.00/0.00	0.22/1.00
LastToken	0.48/0.76/0.01	0.07/0.84
Positional	0.75/0.99/0.01	0.05/0.82
GAP	0.89/0.97/0.02	0.01/0.78
GMP	0.78/0.45/0.04	-0.10/0.66
all-MiniLM	0.00/0.10/0.75	-0.35/0.38
Phi3-NotInstructed		
GMP	0.42/0.08/0.65	-0.41/0.32
LastToken	0.45/0.00/0.62	-0.42/0.30
GAP	0.91/0.18/0.99	-0.68/0.03
Positional	1.00/0.18/1.00	-0.71/0.00

Table 1: Embedding methods and their performance metrics.

Model	F1-score	Accuracy
Baseline		
TFIDF+SVM	0.57	0.57
LLM+SVM	0.6925	0.6925
SASNet	0.52	0.52

Table 2: Comparison of models based on F1-score and Accuracy.

Model	p_{mse}	$p_{pearson}$
Phi3 Instructed+Attentional		
Phi3 Instructed+LastToken	$\leq 1.73^{-20}$	$\leq 1^{-20}$
Phi3 Instructed+Positional	$\leq 1^{-20}$	$\leq 1^{-20}$
Phi3 NotInstructed+GMP	$\leq 1^{-20}$	≤ 0.05
allMiniLM	$\leq 1.62^{-19}$	$\leq 1^{-20}$

Table 3: P-values for chosen embedding model with others. Test for MSE difference computed using F-test and for Pearson's correlation (r) differences using Z-test.