

# Knowledge-Infused Self Attention Transformers

Kaushik Roy  
kaushikr@email.sc.edu  
Artificial Intelligence Institute  
University of South Carolina  
USA

Yuxin Zi  
yzi@email.sc.edu  
Artificial Intelligence Institute  
University of South Carolina  
USA

Vignesh Narayanan  
vignar@sc.edu  
Artificial Intelligence Institute  
University of South Carolina  
USA

Manas Gaur  
manas@umbc.edu  
KAI<sup>2</sup>, University of Maryland  
Baltimore County  
USA

Amit Sheth  
amit@sc.edu  
Artificial Intelligence Institute  
University of South Carolina  
USA

## ABSTRACT

Transformer-based language models have achieved impressive success in various natural language processing tasks due to their ability to capture complex dependencies and contextual information using self-attention mechanisms. However, they are not without limitations. These limitations include hallucinations, where they produce incorrect outputs with high confidence, and alignment issues, where they generate unhelpful and unsafe outputs for human users. These limitations stem from the absence of implicit and missing context in the data alone. To address this, researchers have explored augmenting these models with external knowledge from knowledge graphs to provide the necessary additional context. However, the ad-hoc nature of existing methods makes it difficult to properly analyze the effects of knowledge infusion on the many moving parts or components of a transformer. This paper introduces a systematic method for infusing knowledge into different components of a transformer-based model. A modular framework is proposed to identify specific components within the transformer architecture, such as the self-attention mechanism, encoder layers, or the input embedding layer, where knowledge infusion can be applied. Additionally, extensive experiments are conducted on the General Language Understanding Evaluation (GLUE) benchmark tasks, and the findings are reported. This systematic approach aims to facilitate more principled approaches to incorporating knowledge into language model architectures.

## KEYWORDS

knowledge graphs, language models, knowledge-infusion

## 1 INTRODUCTION

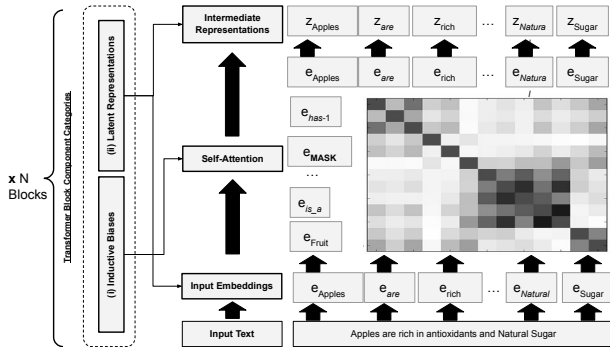
Language modeling has witnessed significant advancements with the introduction of self-attention-based transformer architectures (e.g., GPT-3, ChatGPT, PaLM, etc.) [1, 2]. These models have achieved remarkable success in a wide range of natural language processing tasks, demonstrating their ability to generate coherent and contextually relevant text. By utilizing self-attention mechanisms, transformers excel at capturing long-range dependencies and establishing meaningful relationships between words, enabling them to generate high-quality, context-aware text. However, despite their successes, self-attention-based transformer models have limitations when it comes to capturing all the necessary context solely from the

available data. Language models often struggle with comprehending missing or implicit information, particularly in scenarios where the training data is incomplete or lack the desired context [3]. This limitation can lead to generated text that is plausible but semantically incorrect or inconsistent, diminishing the model’s ability to fully understand and generate language with nuanced meaning [4]. To address these limitations, incorporating external knowledge into language models can provide the missing and implicit context required for accurate language generation. External knowledge, such as factual information, world knowledge, or domain-specific expertise, can supplement the training data by offering additional context that may not be explicitly present in the data alone. By integrating external knowledge, language models can enhance their understanding of complex concepts, disambiguate ambiguous statements, and generate more coherent and contextually accurate text.

However, the existing methods used to incorporate external knowledge into language models often lack a systematic and well-defined approach. These methods seem rather ad hoc, as they introduce knowledge at various components of the transformer architecture based mainly on empirical justifications related to improved performance in downstream tasks. Transformers comprise several interconnected components, including input embedding matrices, encoder layers, and self-attention operations. One concern is that augmenting knowledge in an ad hoc manner may lead to the exploitation of statistical artifacts by the numerous moving parts of the transformer [5]. For instance, it could involve overfitting by utilizing additional parameters provided by the knowledge or fitting to task-specific hidden or spurious patterns to achieve high downstream performance scores. Consequently, it remains unclear, based solely on performance metrics, to what extent such augmentation truly enhances the language comprehension and understanding of the model [6].

In light of these limitations, we propose a systematic approach to infusing knowledge in language models that adopts different strategies depending on the specific transformer component. Initially, we categorize the architectural elements of a transformer into two groups: (i) inductive biases, such as the self-attention matrices, and (ii) latent representations, including the input embeddings and the intermediate representations between encoder layers (Figure 1

illustrates this categorization). Subsequently, we introduce three distinct categories of knowledge infusion: (i) shallow knowledge infusion, which involves incorporating knowledge at the latent representations of the first transformer block; (ii) semi-deep knowledge infusion, where knowledge is also integrated at the self-attention matrix (inductive bias) of the first transformer block, and (iii) deep knowledge infusion, which interleaves knowledge incorporation at the latent representations and self-attention matrices (inductive biases) of the various transformer blocks. In essence, these three methods aim to infuse knowledge at either the level of inductive biases, latent representations, or both. To evaluate the effectiveness of the three categories of knowledge infusion, we conduct extensive experimentation and ablation studies on various tasks from the General Language Understanding Evaluation (GLUE) benchmark, reporting our findings[7]. As mentioned earlier, due to the potential exploitation of statistical artifacts by knowledge infusion methods, traditional downstream task performance metrics can be an ineffective measure of knowledge infusion. Therefore, we also introduce new evaluation metrics for a more robust measurement of the effectiveness of knowledge infusion (see Section 3.2). Our results indicate that deep knowledge infusion outperforms the other two categories of knowledge infusion using both traditional metrics of accuracy and F1-score, as well as the newly introduced metrics.



**Figure 1:** The figure shows the different components of a transformer block, categorized into inductive biases and latent representations. We categorize the input embeddings and the representations obtained after each encoder as latent representations and the self-attention matrices as inductive biases. This structure is repeated  $N$  times (e.g., 12 times in BERT).

## 2 KNOWLEDGE-INFUSED SELF-ATTENTION TRANSFORMERS

### (a) Knowledge Graph Compression for Knowledge Infusion

Since a transformer architecture consists of two distinct types of components, namely (i) inductive biases represented as matrices and (ii) latent representations represented as vectors (as depicted in Figure 1), the challenge lies in compressing external knowledge from knowledge graphs into one of these mathematical representations. This compression process enables the infusion of external

knowledge into the transformer-based models. For this, we first obtain knowledge graph node embeddings (vectors) using a graph encoder network (e.g., numberbatch embeddings obtained by compression of the knowledge graph ConceptNet) and then compute pair-wise inner products between the node embeddings to obtain a graph node embedding-based correlation matrix[8]. Figure 2 (a) illustrates this knowledge compression process.

### (b) Defining Knowledge Infusion Operations

Having obtained the knowledge graph representation in the form of matrices and vectors, we proceed to define two distinct infusion operations. The first operation involves infusing knowledge at the latent representations of a transformer block, which is achieved by adding the graph node embedding vectors to the existing latent representations within the transformer block. The second operation pertains to infusing knowledge at the inductive biases of a transformer block. In this case, the infusion entails adding the graph node embedding-based correlation matrix to the inductive bias, which corresponds to the self-attention matrix of the transformer block. Figure 2 (b) provides a visual depiction of these two operations.

### (c) Shallow Knowledge Infusion

Shallow Infusion involves the operation of infusing the knowledge at the latent representations of just the first transformer block of a transformer-based architecture. It is important to note that a single block usually consists of multiple heads, each associated with its own set of latent representations. In such cases, the knowledge infusion operation is performed on all the latent representations corresponding to the multiple heads. Figure 2 (c) visually represents the shallow infusion approach.

### (d) Semi-deep Knowledge Infusion

Semi-deep Infusion involves both the operations of infusing the knowledge at the latent representations and the inductive biases of just the first transformer block. Similar to the shallow infusion method, if there are multiple heads present, the infusion operations are performed across all heads. Figure 2 (d) provides a visual illustration of the semi-deep infusion approach.

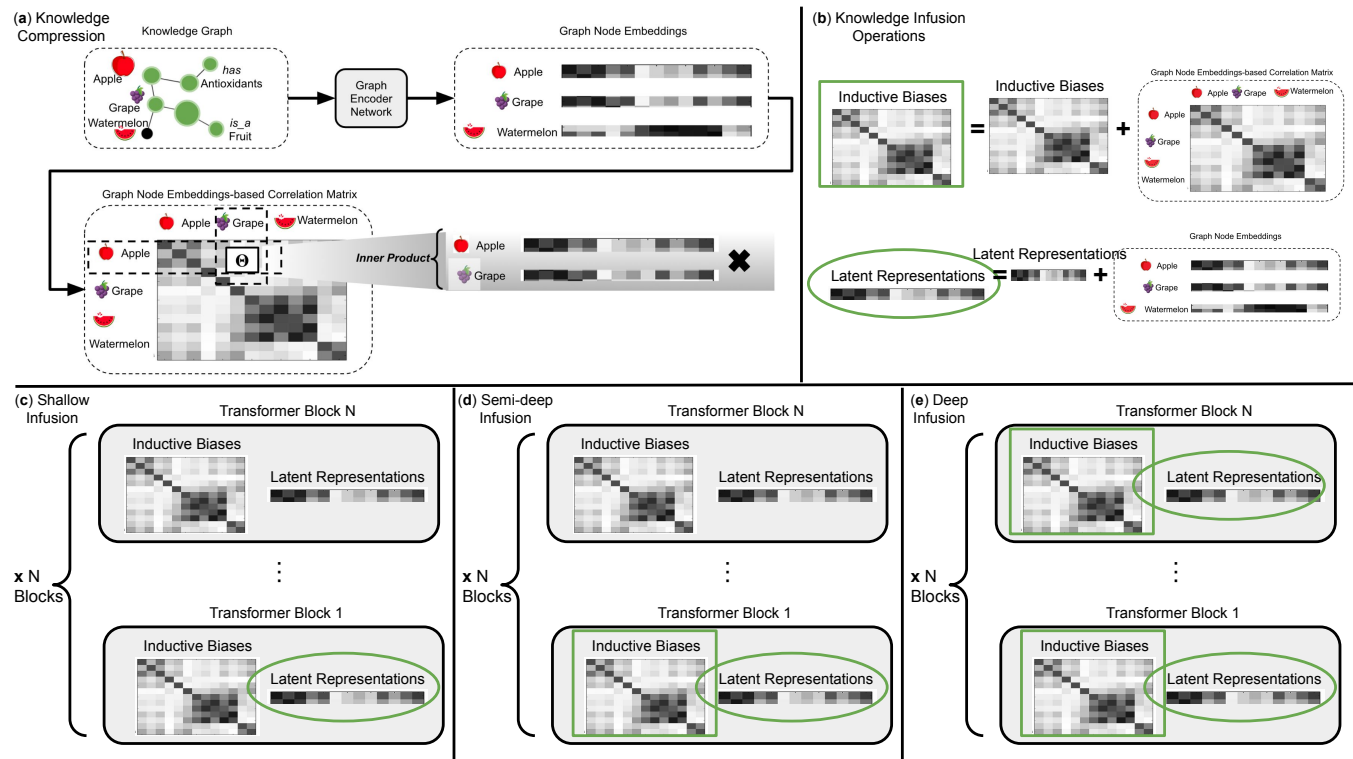
### (e) Deep Knowledge Infusion

Deep Infusion involves both operations of infusing the knowledge at the latent representations and the inductive biases of all the  $N$  transformer blocks. Once again, if there are multiple heads present, these operations are performed across all heads. Figure 2 (e) illustrates the deep-infusion approach, showcasing the infusion of knowledge throughout the transformer blocks.

## 3 EVALUATION TASKS AND METRICS USED IN EXPERIMENTATION

### 3.1 Evaluation Tasks

To assess the performance, we utilize the General Language Understanding Evaluation (GLUE) benchmark tasks. We categorize the tasks we experiment with into three distinct types, which serve as significant indicators of a model’s language comprehension abilities.:



**Figure 2: (a) Knowledge compression - Compressing the information in the knowledge graph into graph node embeddings (vectors) and graph node embedding-based correlations (matrices) for infusion in transformer architectures, namely at the latent representations of the transformer (vectors), or the inductive biases (self-attention matrices). (b) Knowledge infusion operations - Once compressed, we define the operation of knowledge infusion using vectors and matrices as summing the inductive biases (matrices) with the graph node embedding-based correlations and summing the latent representations (vectors) with the graph node embeddings. (c) Shallow infusion - Performing the knowledge infusion operation of adding the graph node embeddings to the latent representations of the first transformer block. (d) Semi-deep infusion - Performing the knowledge infusion operation of adding the graph node embedding-based correlation matrix to the inductive bias matrix of the first transformer block. (e) Deep infusion - Performing both adding the graph node embeddings and graph node embedding-based correlations to the latent representations and inductive biases across all  $N$  transformer blocks.**

*Natural Language Inference (NLI) tasks.* The GLUE tasks MNLI (Multi-genre Natural Language Inference), QNLI (Question Answering Natural Language Inference), and WNLI (Winograd Natural Language Inference) test NLI capabilities from varying angles. MNLI tests whether the model can appropriately judge if a sentence logically follows from another, i.e., logical entailment. QNLI tests similar logical entailment between question and statement pairs - does it make logical sense to ask a follow-up question? WNLI tests logical entailment in the presence of pronouns and the nouns they reference.

*Textual Entailment (TE) tasks.* The GLUE task RTE (Recognizing Textual Entailment) tests for logical entailment similar to the NLI task MNLI. However, RTE emphasizes on the meaning - given two text fragments, whether the meaning of one can be entailed (or can be inferred) from the other.

*Textual Similarity (TS) tasks.* The GLUE task QQP (Quora Question Pairs) tests for the ability to assess the semantic equivalence,

measured as the similarity between a pair of questions that appear on the social media forum Quora.

### 3.2 Evaluation Metrics

In Section 1, we discussed how ad hoc knowledge infusion techniques can lead to models exploiting statistical artifacts towards achieving high downstream task performance. Therefore, to evaluate the different knowledge infusion methods, in addition to the traditional performance metrics of accuracy, F1-scores across GLUE tasks, we also devise the following metrics:

**3.2.1 Combined Graph Encoder and KSAT model Accuracy (CGKA).** This metric evaluates the combined accuracies of two components. Firstly, it measures the accuracy of link prediction by the graph encoder on a separate test set that includes triples from the knowledge graph it encodes. Link prediction is carried out by summing the subject and predicate vectors and finding the closest object vector in a (subject, predicate, object) triple (we check if the similarity is greater than a threshold in our experiments, see

| Model              | QQP         | QNLI        | WNLI         | MNLI         | RTE         |
|--------------------|-------------|-------------|--------------|--------------|-------------|
| Baseline (XLNET)   | 74.79       | 84.17       | 79.9         | 72.3         | 83.6        |
| Shallow Infusion   | 76.11       | 89.9        | 80.5         | 75.3         | 85.1        |
| Semi-Deep Infusion | 80.2        | 90.5        | 90.5         | 82.1         | 90.3        |
| Deep-Infusion      | <b>80.9</b> | <b>92.3</b> | <b>90.91</b> | <b>88.53</b> | <b>90.4</b> |

**Table 1: Shows the comparison of accuracy across the different GLUE tasks from Section 3 for the baseline XLNET model and its variants using the different kinds of knowledge infusion - Shallow, Semi-deep and Deep Infusion. We see that there is steady improvement as external knowledge is included at different components, i.e., latent representations and Inductive biases of a transformer-based model.**

Section 5.2). These triples were not used during runtime. Secondly, it calculates the average accuracy of the KSAT model across all the GLUE tasks. The underlying idea is that if we observe high scores on conventional accuracy and F1-score metrics but low scores on the CGKA metric, it suggests that the KSAT model has possibly learned misleading patterns. A low CGKA score indicates a lack of capturing knowledge graph information by the graph encoder.

**3.2.2 Data Efficiency at K (DE@k).** : This metric evaluates the performance of the KSAT model after training it with only k% of the total available training data. The rationale behind this is that effective knowledge infusion should result in improved data sufficiency. In other words, the additional context provided by high-quality knowledge should compensate for the need for a large volume of data to achieve good performance.

#### 4 KNOWLEDGE GRAPHS, GRAPH ENCODER NETWORKS, AND TRANSFORMER MODELS USED IN EXPERIMENTATION

We use the knowledge graphs ConceptNet and WordNet in our experiments. For the graph encoder network to obtain graph node embeddings (see Figure 2 (a)), we use ConceptNet Numberbatch embeddings and wise embeddings for ConceptNet and WordNet, respectively [8, 9]. We sum the graph node embeddings from both ConceptNet and WordNet for every input token to obtain a single graph node embedding per token. For the transformer models using which we test knowledge infusion, we use the language models - BERT, XLNET, RoBERTa, ELECTRA, and Longformer [10–14].

### 5 EXPERIMENTS

#### 5.1 Traditional Performance Tests

For all our experiments, we use the large version of the models. All experiments are run on a single A100 GPU. We use the standard configurations of the models (e.g., 12 heads per block and  $N=12$  in BERT) Out of all the transformer-based models we experiment with from among those listed in Section 4, XLNET performs the best, both as a baseline and when used with various knowledge infusion techniques. Table 1 shows this result.

#### 5.2 Performance Tests using Newly Introduced Metrics

For the DE@K metric, we want to see if higher performance can be achieved using lesser data. So we plot results averaged across the

GLUE tasks for each transformer-based model with the different infusion techniques using only 50% of the training data, i.e., DE@50. We also want to measure the combined accuracies on link prediction of the embeddings from ConceptNet and WordNet(summed) with the average performance across the GLUE tasks. The method of link prediction is as described in Section 3.2.1, i.e., to predict a link between a subject and object in a (subject, predicate, object) triple, we check if the cosine similarity between the sum of subject and predicate vectors, and the object vectors is greater than 0.5 (we tuned this number from the set {0, 0.25, 0.5, 0.75}). Table 2 shows the results.

| Model      | CGKA      | DE@50    | DE@75    | DE@100   |
|------------|-----------|----------|----------|----------|
| BERT       | 80/71/67/ | 81/79/70 | 81/81/73 | 82/81/73 |
| RoBERTa    | 79/67/65/ | 81/74/71 | 81/76/73 | 82/79/73 |
| ELECTRA    | 80/67/60/ | 75/72/72 | 77/73/71 | 81/75/71 |
| XLNET      | 81/71/65/ | 80/79/69 | 82/80/71 | 83/81/73 |
| Longformer | 78/67/61/ | 79/75/73 | 79/76/72 | 78/76/74 |

**Table 2: Here, we see the performance (accuracy) of different models with deep-infusion/semi-deep infusion/shallow-infusion using the metrics introduced in Section 3. We see that both metrics improve with the addition of external knowledge using the infusion methods, with deep infusion performing the best. Furthermore, we see that the accuracy numbers are already in the 70-80s with just 50% of the data points used (this is average accuracy measured across all GLUE tasks)**

### 6 CONCLUSION AND FUTURE WORK

This paper introduces a systematic approach to knowledge infusion in transformer-based models. The findings indicate that incorporating external knowledge indeed enhances the performance of the models on language understanding tasks. Even more so as the infusion techniques operate on both the latent representations and inductive biases of the model across all transformer layers. This improvement is observed through the evaluation of both traditional metrics and newly introduced evaluation metrics, validating the effectiveness of knowledge infusion. For future research, we plan to explore hybrid knowledge infusion, which involves selectively

choosing the blocks where knowledge infusion occurs and determining which knowledge to infuse. This differs from the current setup where graph node embeddings from WordNet and ConceptNet are summed. The analysis presented in this paper aims to lay the foundation for more principled approaches to external knowledge-augmented language models in the future.

## 7 ACKNOWLEDGEMENT

This work is built on prior work [15–30], and supported by the National Science Foundation under Grant 2133842, “EAGER: Advancing Neuro-symbolic AI with Deep Knowledge-infused Learning” [31–33].

## REFERENCES

- [1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- [2] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.
- [3] Chenguang Zhu, Yichong Xu, Xiang Ren, Bill Yuchen Lin, Meng Jiang, and Wenhao Yu. Knowledge-augmented methods for natural language processing. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 1228–1231, 2023.
- [4] Xuhui Zhou, Yue Zhang, Leyang Cui, and Dandan Huang. Evaluating common-sense in pre-trained language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 9733–9740, 2020.
- [5] R Thomas McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *arXiv preprint arXiv:1902.01007*, 2019.
- [6] Emily M Bender and Alexander Koller. Climbing towards nlu: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 5185–5198, 2020.
- [7] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.
- [8] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.
- [9] Sawan Kumar, Sharmistha Jat, Karan Saxena, and Partha Talukdar. Zero-shot word sense disambiguation using sense definition embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5670–5681, 2019.
- [10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [11] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.
- [12] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*, 2020.
- [13] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.
- [14] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.
- [15] Kaushik Roy, Yuxin Zi, Manas Gaur, Jinendra Malekar, Qi Zhang, Vignesh Narayanan, and Amit Sheth. Process knowledge-infused learning for clinician-friendly explanations. *arXiv preprint arXiv:2306.09824*, 2023.
- [16] Kaushik Roy, Vedant Khandelwal, Raxit Goswami, Nathan Dolbir, Jinendra Malekar, and Amit Sheth. Demo alleviate: Demonstrating artificial intelligence enabled virtual assistance for telehealth: The mental health case. *arXiv preprint arXiv:2304.00025*, 2023.
- [17] Kaushik Roy, Usha Lokala, Vedant Khandelwal, and Amit Sheth. " is depression related to cannabis?": A knowledge-infused model for entity and relation extraction with limited supervision. *arXiv preprint arXiv:2102.01222*, 2021.
- [18] Kaushik Roy, Qi Zhang, Manas Gaur, and Amit Sheth. Knowledge infused policy gradients for adaptive pandemic control. *arXiv preprint arXiv:2102.06245*, 2021.
- [19] Parth Asawa, Manas Gaur, Kaushik Roy, and Amit Sheth. Covid-19 in spain and india: comparing policy implications by analyzing epidemiological and social media data. *arXiv preprint arXiv:2010.14628*, 2020.
- [20] Kaushik Roy, Yuxin Zi, Vignesh Narayanan, Manas Gaur, and Amit Sheth. Ksat: Knowledge-infused self attention transformer—integrating multiple domain-specific contexts. *arXiv preprint arXiv:2210.04307*, 2022.
- [21] Revathy Venkataramanan, Kaushik Roy, Kanak Raj, Renjith Prasad, Yuxin Zi, Vignesh Narayanan, and Amit Sheth. Cook-gen: Robust generative modeling of cooking actions from recipes. *arXiv preprint arXiv:2306.01805*, 2023.
- [22] Kaushik Roy, Tarun Garg, Vedant Palit, Yuxin Zi, Vignesh Narayanan, and Amit Sheth. Knowledge graph guided semantic evaluation of language models for user trust. *arXiv preprint arXiv:2305.04989*, 2023.
- [23] Manas Gaur, Kaushik Roy, Aditya Sharma, Biplav Srivastava, and Amit Sheth. “who can help me?”: Knowledge infused matching of support seekers and support providers during covid-19 on reddit. In *2021 IEEE 9th International Conference on Healthcare Informatics (ICHI)*, pages 265–269. IEEE, 2021.
- [24] Kaushik Roy, Manas Gaur, Misagh Soltani, Vipula Rawte, Ashwin Kalyan, and Amit Sheth. Proknow: Process knowledge for safety constrained and explainable question generation for mental health diagnostic assistance. *Frontiers in big Data*, 5:1056728, 2023.
- [25] Adam Tsakalidis, Jenny Chim, Iman Munire Bilal, Ayah Zirikly, Dana Atzil-Slonim, Federico Nanni, Philip Resnik, Manas Gaur, Kaushik Roy, Becky Inkster, et al. Overview of the clpsych 2022 shared task: Capturing moments of change in longitudinal user posts. In *Proceedings of the Eighth Workshop on Computational Linguistics and Clinical Psychology*, pages 184–198, 2022.
- [26] Shrey Gupta, Anmol Agarwal, Manas Gaur, Kaushik Roy, Vignesh Narayanan, Ponnurangam Kumaraguru, and Amit Sheth. Learning to automate follow-up question generation using process knowledge for depression triage on reddit posts. *arXiv preprint arXiv:2205.13884*, 2022.
- [27] Nathan Dolbir, Triyasha Dastidar, and Kaushik Roy. Nlp is not enough—contextualization of user input in chatbots. *arXiv preprint arXiv:2105.06511*, 2021.
- [28] Vipula Rawte, Megha Chakraborty, Kaushik Roy, Manas Gaur, Keyur Faldu, Prashant Kikani, Hemang Akbari, and Amit Sheth. Tdlr: Top (semantic)-down (syntactic) language representation. *UMBC Faculty Collection*, 2022.
- [29] Usha Lokala, Francois Lamy, Triyasha Ghosh Dastidar, Kaushik Roy, Raminta Daniulaityte, Srinivasan Parthasarathy, and Amit Sheth. edarktrends: Harnessing social media trends in substance use disorders for opioid listings on cryptomarket. *arXiv preprint arXiv:2103.15764*, 2021.
- [30] Yuxin Zi, Kaushik Roy, Vignesh Narayanan, Manas Gaur, and Amit Sheth. Ierl: Interpretable ensemble representation learning—combining crowdsourced knowledge and distributed semantic representations. 2023.
- [31] Amit Sheth, Kaushik Roy, and Manas Gaur. Neurosymbolic ai—why, what, and how. *arXiv preprint arXiv:2305.00813*, 2023.
- [32] Amit Sheth, Manas Gaur, Kaushik Roy, and Keyur Faldu. Knowledge-intensive language understanding for explainable ai. *IEEE Internet Computing*, 25(5):19–24, 2021.
- [33] Amit Sheth, Manas Gaur, Kaushik Roy, Revathy Venkataramanan, and Vedant Khandelwal. Process knowledge-infused ai: Toward user-level explainability, interpretability, and safety. *IEEE Internet Computing*, 26(5):76–84, 2022.