2022

KSAT: Knowledge-infused Self Attention Transformer - Integrating Multiple Domain-Specific Contexts

Kaushik Roy

Yuxin Zi

Vignesh Narayanan

Manas Gaur

Amit P. Sheth

KSAT: Knowledge-infused Self Attention Transformer - Integrating Multiple Domain-Specific Contexts

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

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

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

Manas Gaur
Dept of Computer Science and Engineering
University of Maryland, Baltimore County
manas@umbdc.edu

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

Abstract

Domain-specific language understanding requires integrating multiple pieces of relevant contextual information. For example, we see both suicide and depression-related behavior (multiple contexts) in the text "*I have a gun and feel pretty bad about my life, and it wouldn't be the worst thing if I didn't wake up tomorrow*". Domain specificity in self-attention architectures is handled by fine-tuning on excerpts from relevant domain specific resources (datasets and external knowledge - medical textbook chapters on mental health diagnosis related to suicide and depression). We propose a modified self-attention architecture *Knowledge-infused Self Attention Transformer* (KSAT) that achieves the integration of multiple domain-specific contexts through the use of external knowledge sources. KSAT introduces knowledge-guided biases in dedicated self-attention layers for each knowledge source to accomplish this. In addition, KSAT provides mechanics for controlling the trade-off between learning from data and learning from knowledge. Our quantitative and qualitative evaluations show that (1) the KSAT architecture provides novel human-understandable ways to precisely measure and visualize the contributions of the infused domain contexts, and (2) KSAT performs competitively with other knowledge-infused baselines and significantly outperforms baselines that use fine-tuning for domain-specific tasks.

1 Motivation

Solving domain-specific tasks such as mental health diagnosis (MHD), and triaging, requires integrating relevant contextual information from data and knowledge sources. Self-Attention based Language Models (SAMs) capture an aggregated broader context from domain-agnostic, voluminous training corpora [1]. Fine-tuning SAMs on domain-specific corpora achieves domain-specific context capture [2, 3]. However, SAM architectures are black-box in nature [4]. Consequently, fine-tuned

Preprint. Under review.

SAM architectures do not lend themselves to the robust evaluation of the open research aims: (**R1**) Relevant domain-specific context coverage, and (**R2**) The influence of knowledge context traded-off against the data context in downstream tasks [5, 6]. We propose a modified self-attention architecture *Knowledge-infused Self Attention Transformer* (KSAT) to address these aims. KSAT performs well on select domain-specific tasks (see Task Description, Data, and External Knowledge Sources) while lending itself to a robust human-understandable evaluation of **R1** and **R2**. Thus KSAT provides a substantial step towards fostering AI-user trust, and satisfaction [7, 8].

2 Background

2.1 Related Work

Prior approaches that are relevant to **R1** and **R2** and incorporate multiple knowledge contexts can be broadly categorized based on the knowledge-infusion technique as (**1**) knowledge modulated SAMs and (**2**) knowledge infused input embedding-based SAMs [9, 10]. The former uses knowledge to guide the self-attention mechanism in SAMs, and the latter embeds the knowledge into a vector space before passing the inputs into SAMs. Here, we briefly summarize their contributions towards **R1** and **R2**. Both Category (**1**), and Category (**2**) methods' domain coverage is evaluated through performance on domain-specific task descriptions (**R1**). These methods' ablations highlight contributions of knowledge context (**R2**). However, inspecting the numerical outputs from the model components (projection matrices and vectors) does not easily lend themselves to human-understandable scrutiny. Explainable AI techniques (post-processing of the numerical outputs that transform them into human-understandable information) are required to confirm the author(s) perspectives [11]. Post-processing-based explanations are local approximations of the SAM reasoning for particular inputs and therefore do not present the global picture, casting doubts on the SAM evaluation validity. KSAT presents a SAM architecture whose numerical outputs lend themselves to robust human-understandable evaluations of **R1** and **R2**.

2.2 Task Description, Data, and External Knowledge Sources

Although the KSAT architecture broadly applies to any domain-specific task, we choose the specific task of Mental Health Diagnostic Assistance for Suicidal Tendencies by Gaur et al. [12]. We denote this dataset as MHDA. The data contains high-quality expert annotations on Reddit posts from suicide-related subreddits. The annotation method ensures minimal noise from measurement artifacts and high agreement among the expert annotators. We use the clinically established diagnostic process information contained in the *Columbia Suicide Severity Rating Scale* (CSSRS) for knowledge contexts. Figure 1 (a, b) illustrate the various contexts (each tree path represents a context) under which suicidal patterns can arise. The task is to predict the suicidal patterns, namely - *indication*, *ideation1*, *ideation2*, and *behavior or attempt*. Figure 1 (c) shows examples from the MHDA dataset, augmented with knowledge context annotations. We denote the augmented dataset as k-MHDA. k-MHDA contains knowledge context annotations at the post and sentence level (see Figure 1 (c)). We defer construction details of k-MHDA from the CSSRS knowledge and the MHDA data to the appendix Section Constructing k-MHDA as it is not the main focus of the paper[1].

3 KSAT - Proposed Architecture

For an input x in the vanilla SAM layer l, a CLS token is introduced that encodes the information in the data for downstream tasks. For example, for classification, Z_{CLS}, the representation of the CLS token from the final layer L is passed through a softmax layer which outputs class probabilities. In the KSAT layer, we introduce a Z_{KCLS} token that encodes the knowledge for that layer. The Z_{KCLS} values are determined by the graph context between the input x sentences and the concepts

[1]We will release the k-MHDA dataset, along with code to construct it along with the KSAT code for reproducibility of results.

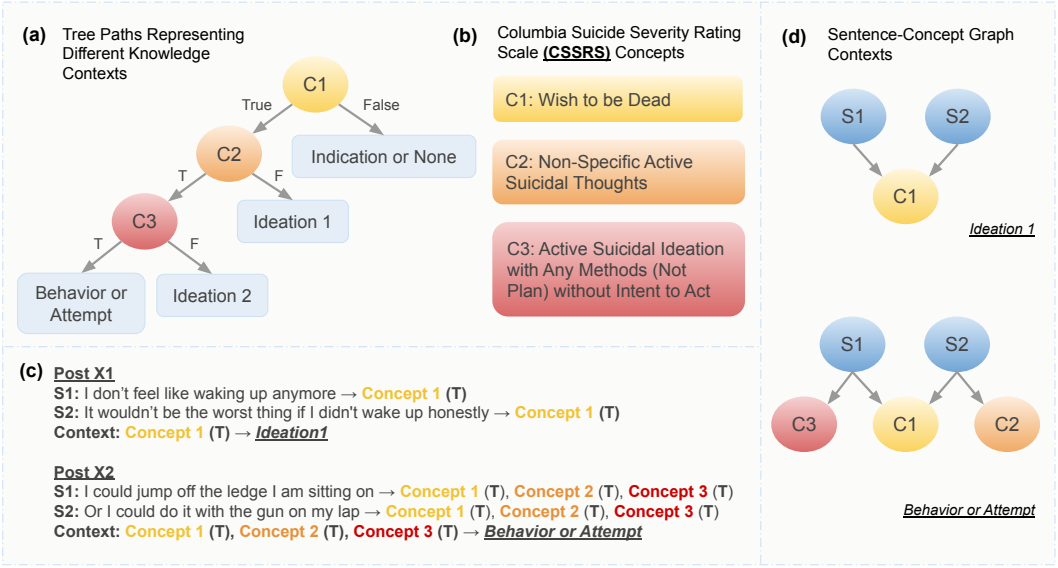

Figure 1: The knowledge source we utilize for our experiments is the *Columbia Suicide Severity Rating Scale* (CSSRS) and the data are posts from Reddit subreddits related to suicide. (**a**) Shows the knowledge contexts (tree-path represents a context) present in the CSSRS leading to varying suicidal patterns. (**b**) Shows the different concepts listed in the CSSRS. (**c**) Shows the posts and sentences from the data annotated with relevant knowledge concepts and contexts from the CSSRS. (**d**) Shows how sentences from the input posts (**Post X1** and **Post X2**) and concepts from the CSSRS are graphically connected in different contexts (In this case, the *Ideation1* and the *Behavior or Attempt* contexts).

in the knowledge context annotations for input x. As an example, consider the inputs x to be like the posts **Post X1** and **Post X2** shown in Figure 1(c), then the graphs in Figure 1 (d), illustrate the graph contexts that show sentence-concept connections. One would expect that the Z_{KCLS} encoding for the sentences **S1** and **S2** in both posts would be similar as they have similar graph contexts. The $\mathbf{KG_{bias}}$ term ensures that the values in Z_{KCLS} captures this behavior (see Figure 2 (b)).

3.1 The KSAT layer

In contrast with the vanilla SAM layer, every KSAT layer has the following key differences:

3.1.1 Layer Parameters and Outputs

Let Y denote the set of suicidal outcomes pertaining to each context: {*Indication or None*, *Ideation1*, *Ideation2*, *Behavior or Attempt*} (see Figure 1 (a)). Let x denote an input post (see Figure 1 (c) for example posts). For each input x, every KSAT layer l outputs a vector of probabilities for every outcome $y \in Y$ given by **Equation** 1:

$$P_l^{ksat}(y|x) = \sigma(W^T(\alpha_l Z_{KCLS}(x) + (1-\alpha_l)Z_{CLS}(x)) + \mathbf{KG_{bias}}(x)) \quad (1)$$

W, $Z_{CLS}(x)$, and $Z_{KCLS}(x)$ are of dimension size 384×1 as we use the sentence-transformer published by Reimers et al. fine-tuned on the MHDA corpus, for embedding inputs [13] . Recall that the term $\mathbf{KG_{bias}}$ is used to ensure that $Z_{CLS}(x)$ encodes the knowledge context represented as a graph (details in Encoding Graph Contexts in Layers). The dimension size of $\mathbf{KG_{bias}}$ is 1×1. The scalar term α_l is used as a trade-off factor modulating the contributions between data and knowledge contexts using a convex combination (see Equation 1). We contrast KSAT layer outputs against the vanilla SAM layer, where a similar probability vector is output only in the final layer and not at every layer (see Figure 2). The parameters within the vanilla SAM layer (the query, key, and value projection matrices) used to compute the self-attention matrix are retained in the KSAT layer [1]. Unlike in the vanilla SAM layer, where the parameters may or may not be shared across layers, the

parameters are not shared across different KSAT layers as different layers encode different knowledge contexts.

3.1.2 Encoding Graph Contexts in Layers

Even though every KSAT layer l outputs a vector of probabilities for every outcome $y \in$ {*Indication or None*, *Ideation1*, *Ideation2*, *Behavior or Attempt*}, representing each context, it encodes only a single context in Z_{KCLS}. We show how to compute the $\mathbf{KG_{bias}}$ term by referencing the posts **Post X1** and **Post X2** and the graph contexts *Ideation1* and *Behavior or Attempt* from Figure 1 (c,d).

Computing Sentence-Concept Connection Vectors: We first compute a sentence-concept connection vector for each sentence in the posts. For **Post X1**, both sentence **S1** and **S2** are connected to the concept C1 in the concept set {C1,C2,C3} (see Figure 1 (b)). Therefore both their sentence-concept connection vectors are computed as: [1, 0, 0]. Similarly for **Post X2**, the sentence-concept connection vectors for both **S1** and **S2** are computed as: [1, 1, 1].

Computing Graph Context Distances: Recall that KSAT layers take as input a single post (x in Equation 1). Denoting the sentence-concept vectors for a sentence **S1** as C_{S1}, to compute $\mathbf{KG_{bias}}(x)$, we first need to compute graph context distances for the posts **Post X1** and **Post X2**: $d(C_{S1}, C_{S2})$. If the posts had more than two sentences the graph context distances would include all pairs (Eg: $d(C_{S1}, C_{S2})$, $d(C_{S2}, C_{S3})$, $d(C_{S1}, C_{S3})$ for three sentences). The term $d(C_{Si}, C_{Sj})$ for a pair of sentences (S_i, S_j) in post x captures the graph context-based distance between the sentences. Intuitively, sentences that have equivalent graph contexts should have $d(S_i, S_j) = 0$. We use hamming distance in our experiments.

The $\mathbf{KG_{bias}}$ term for an input x is thus given by Equation 2 as:

$$\mathbf{KG_{bias}}(x) = -\sum_{(S_i,S_j)\in x} \frac{(Z_{KCLS}(x)[S_i] - Z_{KCLS}(x)[S_j])^2}{d(S_i, S_j) + \epsilon} \tag{2}$$

The formulation for the $\mathbf{KG_{bias}}(x)$ term in Equation 2 encourages the Z_{KCLS} representations of sentences that are in the same graph context to be similar. The ε term is to prevent dividing by zero errors.

3.2 Aggregating KSAT Layer Outputs

Combining KSAT layer probabilities given by Equation 1 is application domain dependent. For the suicidal outcomes in the set: {*Indication or None*, *Ideation1*, *Ideation2*, *Behavior or Attempt*}, it is reasonable to expect that suicidal ideation (both *Ideation1* and *Ideation2*) precedes the act of attempting suicide (*Behavior or Attempt*). In other application domains, the contexts could be independent of each other.

In our experiments, we stack four KSAT layers corresponding to the outcomes *Indication or None*, *Ideation1*, *Ideation2*, *Behavior or Attempt*, in that order (the order is derived from the tree structure in Figure 1 (a)). Typically, for dependent probability outcomes, X, and Y, where X precedes Y, the probability $P(X, Y)$ would be modeled as $P(Y \mid X)P(X)$. However, since we stack KSAT layers in a particular order, we use a product approximation ($P(X, Y) = P(X)P(Y)$) as information is propagated upwards through the KSAT layers. Thus, the final layer probabilities from the KSAT layers is computed using Equation 3 as:

$$P^{ksat}_{final}(y|x) = \prod_l P^{ksat}_l(y|x), \tag{3}$$

where $P^{ksat}_l(y|x)$ is given by Equation 1

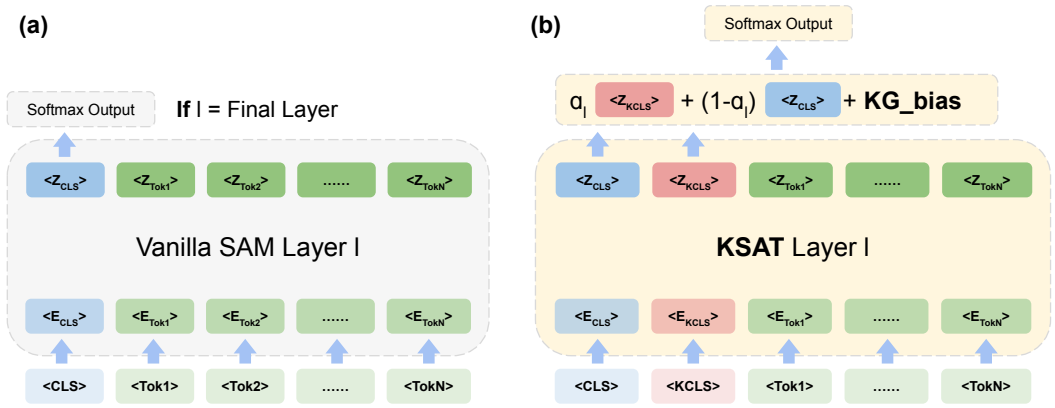

Figure 2: The comparative architecture of KSAT with existing vanilla SAMs. (a) Shows the vanilla SAM layer where the Z_{CLS} token from the final layer is passed as input to the softmax function for prediction. (b) Shows the introduction of the additional Z_{KCLS} token in a KSAT layer. The convex combination added with the knowledge bias as input to the softmax function is used to balance the use of data and knowledge contexts selectively.

4 KSAT - Results and Analysis

Recall the research aims that KSAT addresses when integrating multiple contexts - (**R1**) Relevant domain-specific context coverage, and (**R2**) The influence of knowledge context traded-off against the data context in downstream tasks (see **Section** Motivation).

4.1 KSAT - Quantitative Results (Addresses R1)

Table 1 shows the accuracy / AUC-ROC scores (rounded-of) for KSAT vs two best-performing fine-tuned transformer models, and knowledge infused baseline models **K-type(i)** and **K-type(ii)** (see section Related Work). Due to space concerns we describe details of the models **K-type(i)** and **K-type(ii)** in the Appendix Section Construction of the **K-type(i)** and **K-type(ii)** baseline models. **KSAT** outperforms the fine-tuned transformer models and performs comparably with models **K-type(i)** and **K-type(ii)** on the k-MHDA dataset (see section Constructing k-MHDA).

Dataset	KSAT	K-type(i)	K-type(ii)	XLNET	RoBERTa
k-MHDA	83% / %78	84% / 71%	84% / 72%	68% / 57%	68% / 63%

Table 1: Shows the accuracy / AUC-ROC scores (rounded-of) for KSAT vs fine-tuned transformer models, and knowledge infused baseline models **K-type(i)** and **K-type(ii)**. **KSAT** outperforms all fine-tuned transformer models and performs comparably with models **K-type(i)** and **K-type(ii)** on the k-MHDA dataset.

4.2 KSAT-Qualitative Results (Addresses R2)

Figure 3 illustrates the final KSAT layer representations (the Z_{KCLS} vectors) and α_l values of sample test posts to visualize data and knowledge contexts (see section **Layer Parameters and Outputs**).

5 Conclusion

We proposed KSAT that integrates multiple contexts and data and shows its utility in the domain-specific use-case of MHDA. Although we test KSAT on MHDA, KSAT applies to other domain-specific tasks that require contextualization from multiple knowledge sources. The architecture of KSAT allows for precise measurement and visualization of the contributions from the different knowledge contexts and the data, thus addressing the research aims **R1** and **R2**. In future work, we will apply KSAT to other downstream domain-specific knowledge-intensive tasks such as conversational

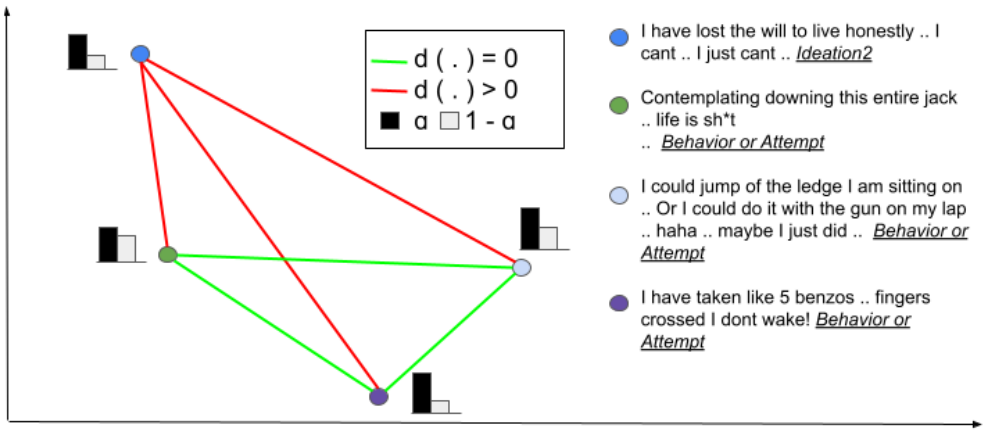

Figure 3: Shows that the first two *Behavior or Attempt* posts in the list of posts are far off based on the Z_{CLS} representations (false), where as KSAT representations depict them as closer i.e d(.) < ε (true). The α shows the data and knowledge contributions for the samples.

question answering that requires drawing context from multiple knowledge sources. We will replicate KSAT's qualitative evaluation in MHDA for robust human-understandable evaluation on future tasks.

References

[1] 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.

[2] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune bert for text classification? In *China national conference on Chinese computational linguistics*, pages 194–206. Springer, 2019.

[3] Laila Rasmy, Yang Xiang, Ziqian Xie, Cui Tao, and Degui Zhi. Med-bert: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. *NPJ digital medicine*, 4(1):1–13, 2021.

[4] Manas Gaur, Keyur Faldu, and Amit Sheth. Semantics of the black-box: Can knowledge graphs help make deep learning systems more interpretable and explainable? *IEEE Internet Computing*, 25(1):51–59, 2021.

[5] Jerome R Bellegarda. Statistical language model adaptation: review and perspectives. *Speech communication*, 42(1):93–108, 2004.

[6] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. Don't stop pretraining: adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*, 2020.

[7] Amit Sheth, Manas Gaur, Kaushik Roy, and Keyur Faldu. Knowledge-intensive language understanding for explainable ai. *IEEE Internet Computing*, 25(5):19–24, 2021.

[8] Amit Sheth, Manas Gaur, Kaushik Roy, Revathy Venkataraman, and Vedant Khandelwal. Process knowledge-infused ai: Toward user-level explainability, interpretability, and safety. *IEEE Internet Computing*, 26(5):76–84, 2022.

[9] Matthew E Peters, Mark Neumann, Robert L Logan IV, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. Knowledge enhanced contextual word representations. *arXiv preprint arXiv:1909.04164*, 2019.

[10] Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Guihong Cao, Daxin Jiang, Ming Zhou, et al. K-adapter: Infusing knowledge into pre-trained models with adapters. *arXiv preprint arXiv:2002.01808*, 2020.

[11] Feiyu Xu, Hans Uszkoreit, Yangzhou Du, Wei Fan, Dongyan Zhao, and Jun Zhu. Explainable ai: A brief survey on history, research areas, approaches and challenges. In *CCF international conference on natural language processing and Chinese computing*, pages 563–574. Springer, 2019.

[12] Manas Gaur, Amanuel Alambo, Joy Prakash Sain, Ugur Kursuncu, Krishnaprasad Thirunarayan, Ramakanth Kavuluru, Amit Sheth, Randy Welton, and Jyotishman Pathak. Knowledge-aware assessment of severity of suicide risk for early intervention. In *The world wide web conference*, pages 514–525, 2019.

[13] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[14] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26, 2013.

[15] 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.

A Constructing k-MHDA

There are 500 Reddit posts in the MHDA dataset. The knowledge contexts in the CSSRS can be illustrated as a tree (see Figure 1 (**d**)). We can construct a probabilistic decision tree that takes input post x and outputs an outcome y from among the leaves. We can write the tree in algebraic form as shown in **Equation** 4.

$$P(y \mid x, \{\theta_i\}) = \sum_{y \in Outcomes} p_y \prod_{i=1}^{3} \sum_{x_{sub} \in x} \left(cos\_sim\left(x_{sub}^R, q_i^R \right) \geq \theta_i \right) \geq 0.5 \tag{4}$$

p_y is the ground truth probability for each outcome. Index i iterates through the 3 concepts in Figure 1 (d). x_{sub} denotes a sub-fragment of the input post (1 sentence, 2 sentence, etc.). q_i denotes the concept texts from the 3 concepts that i indexes. x_{sub}^R and q_i^R are representations of the post sub-fragment and the concept texts using the sentence-transformer published by Reimers et al. [13].

Equation 5 determines the presence or absence of concept q_i in a post sub-fragment x_{sub}. First, we compute the cosine similarity between their sentence-transformer representations x_{sub}^R and q_i^R. If the resulting value is $\geq \theta_i$, we determine that the concept q_i is present in x_{sub}, else we determine that the concept q_i is absent in x_{sub}.

$\sum_{x_{sub} \in x}(.) \geq 0.5$ in Equation 4 is the algebraic form of the $\vee$ operation as we determine that concept q_i is present in the post x, if any of the post fragments $x_{sub} \in x$ show presence of concept q_i.

$$\left(cos\_sim\left(x_{sub}^R, q_i^R \right) \geq \theta_i \right) \tag{5}$$

We can then evaluate the Bernoulli Loss $\mathcal{L}$ given an input, outcome pair (x, y) and parameters $\{\theta_i\}$ as:

$$\begin{aligned} \mathcal{L}(x, y, \{\theta_i\}) = P(y \mid x, \{\theta_i\}) log(P(y \mid x, \{\theta_i\})) + \\ (1 - P(y \mid x, \{\theta_i\})) log(1 - P(y \mid x, \{\theta_i\})) \end{aligned} \tag{6}$$

We use grid-search to find a configuration of parameters $\{\theta_i\}$ and post sub-fragment x_{sub} that has the maximum value for $\prod_{(x,y)\in\mathbf{MHDA}} \mathcal{L}(x, y, \{\theta_i\})$. We vary each individual θ_i in the range -1 to 1 (the *range of the cosine function*) and x_{sub} takes values from the set $\{1, 2, 3\}$.

Inference is carried out as it is in a decision tree classifier with the concept presence or absence at each branch, evaluated using **Equation** 5.

Knowledge Context Annotation with outputs from grid-search: The grid-search yielded outputs $\{\theta_i\} = \{0.3, 0.5, 0.3\}$, and post sub-fragment size $|x_{sub}| = 1$ (one sentence). Therefore the post "I don't feel like waking up and have a gun. Oh well." is annotated with the knowledge context: **(Concept 1 (T)), Concept 2 (T), Concept 3 (T) =** *Behavior or Attempt*, as evaluation of **Equation** 5 determines absence of **Concept 1**, **Concept 2**, and **Concept 3** in the post sentence"I don't feel like waking up and have a gun". The evaluation uses the grid-search outputs of $\{\theta_i\}$. The second sentence "Oh well' is not necessary to evaluate as we determine a concept's presence or absence in the post if any of the post fragments x_{sub} (one sentence) show the presence of the concept.

B Construction of the K-type(i) and K-type(ii) baseline models

For our baseline implementations we adapt the state-of-the-art models for each of the model types **K-type(i)** and **K-type(ii)** in Section 2.1 for our task description (see Section 2.2. Specifically, we use the model KnowBERT and K-Adapter as **K-type(i)** and **K-type(ii)** model instances respectively [9, 10]. For knowledge graph embeddings in the KnowBERT case, we utilize TransE embeddings. Similar to their work, we use the RoBERTa model for the task-specific adapter module. We retain all hyperparameters from the original implementations [14, 15].