Efficient Token Sparsification Through the Lens of Infused Knowledge

Bohua Peng[a,b], Bin Chen[b], Wei He[b], William Thorne[b], Visakan Kadirkamanathan[b]
[a] Northwestern Polytechnical University, Xian, China
[b] The University of Sheffield, Sheffield, The UK
marvinbp1119@gmail.com, {bin.chen, wei.he1, wthorne1, visakan}@sheffield.ac.uk

***Abstract*—Leveraging large language models (LLMs) to fuse heterogeneous knowledge is an exciting emerging field. However, with billions of parameters, these pretrained language models are prohibitively computationally expensive at inference time. Token sparsification methods can proactively accelerate inference by selecting important features from the sequence but often require task-dependent retraining. To address this, we propose Bilevel Token prUniNg wiTh Infused kNowledGe (Bunting), an interpretable token pruning method that leverages task-level knowledge encoded in prefixes to guide token sparsification, eliminating the need for task-specific retraining. Bunting performs Bayesian Token Sparsification, where the inner loop learns a joint representation to perform the task, and the outer loop learns adaptive attention masks for sparse representations, thus pruning redundant tokens layer-by-layer without compromising the pretrained abilities of LLMs. Additionally, we introduce an innovative antiphrasis evaluation protocol to test model adaptivity on rhetorical relations. Furthermore, we demonstrate that precomputed prefixes can effectively guide token sparsification in different knowledge-intensive tasks, maintaining task-level knowledge to identify important tokens and reduce the finetuning burden. Experimental results demonstrate that our method achieves over 0.3x wall-clock speed-up with only 0.14× learnable parameters in knowledge-intensive tasks. Our findings suggest that token pruning can improve out-of-distribution detection, with sarcasm being more challenging to detect than immorality.**

***Index Terms*—Knowledge graphs, large language models, out-of-distribution detection**

I. Introduction

Large language models (LLMs) have recently become powerful information fusion tool in realworld AI applications, such as vision [1], speech [2] and robotics [3]. These models leverage their reasoning abilities from extremely energy consuming pretraining. However, finetuning full model parameters to specific downstream tasks is prohibitively challenging due to low-resource finetuning datasets that require expensive expert knowledge. Within this context, parameter-efficient finetuning (PEFT) [4] can address such learning challenges in training memory and sample size by infusing knowledge as plug-in modules to Transformers. While PEFT modules have prospered and grown in the dimensions of adapter [5], prompts [6] and low-rank reparameterization [7], these additional modules add to the burden of inference latency, especially for latency-sensitive applications.

To accelerate inference, model sparsification [8] or pruning [9] is a straightforward approach, which can date back to the early age of machine learning. Sparsification can be roughly categorized as structured and unstructured methods. Unstructured sparsification makes it hard to achieve wall-clock time speed-up due to the dense computation nature of GPUs. By contrast, token sparsification has shown effectiveness in computer vision and language processing. Nevertheless, sparsification with infused knowledge has not kept pace with the advances of LLMs. For example, Diff pruning [10] proposes a task-specific sparsification paradigm that extends the original pretrained parameters but only focuses on classification with BERT-style models.

In this paper, we demonstrate that prefixes can guide token sparsification of transformer models as prior knowledge in link prediction and textual entailment. Within this context, we propose Bilevel token prUniNg wiTh Infused kNowledGe, Bunting, to accelerate knowledge-intensive inference (**RQ1**). Our work merges the gap between Token Sparsification [11] and Parameter Efficient Finetuning [5]. Specifically, Bunting employs two levels of optimization, with the inner level learning joint knowledge representations, modeling predicates or relations, and the outer level evolving sparsification through learnable masks. In particular, the learnable attention masks are part of additional parameters and hence they converge simultaneously and the pruning masks do not require retraining.

Inspired by representation probing [12], we extend the generalizability analysis to evaluate whether a model can handle *antiphrasis* (**RQ2**). Antiphrasis, as a rhetorical device [13], describes nearly contradictory situations from the original relation. For example, changing ”score a goal” to ”score on own goal” reflects a shift from ”wining” to ”messing up”. Despite being literally similar, antiphrasis can significantly shift the underlying semantics, posing a great challenge on the generalizability of LLMs. Based on zero-shot task generalization [14], a fully transferrable LLMs should output other reasonable predictions but not necessarily the original entity, given shifted antiphrasis. Therefore, we hypothesize a performance decline may be an indicator for generalization. To validate this idea, we further analyze the performance of LLMs on antiphrasis with a human-in-the-loop data collection process (**RQ3**).

To summarize, the contributions of this paper are threefold:

- We propose an interpretable token sparsification method, namely **Bunting**, to accelerate inference.
- We introduce an innovative antiphrasis evaluation protocol to evaluate adaptivity on antiphrasis relations.

- We demonstrate that precomputed prefixes can effectively guide token sparsification in different knowledge-intensive tasks.

II. Related Works

This section covers several model compression studies and knowledge-intensive tasks that involve information fusion.

A. Knowledge Graph Completion

Knowledge graphs are data structures that represent knowledge as concepts and relationships between them as facts. Due to their inherent incompleteness, automatic knowledge graph completion [15] is a fundamental challenge in knowledge-intensive tasks, such as textual entailment [16] and question answering [17]. Open domain KGC methods fuse text description into knowledge representations, which generally subscribe to embedding aggregation [18] and message-passing network aggregation. Embedding-based methods such as FuAlign [19] and MvTuckER [20] fuse neighborhood information into vectors encoding relation properties such as reflexivity and symmetry. To transfer to unseen nodes, Graph Neural Networks (GNNs) such as InGram [21] and Neural Bellman-Ford Nets [22] iteratively aggregate information from neighborhood.

B. LLM based Data Fusion

Large language models (LLMs) are language models that can achieve general purpose language understanding and generation by learning statistical relations from computationally intensive self-supervised learning. As AI applications expand to new domains, LLMs become popular tools for fusing open domain knowledge for NLP tasks. Methods such as BLP [23] and KEPLER [24] incorporate Bidirectional Transformers (BERT) [25] into the translating embedding framework [26], fuse text description and other context information into first-order predicate logics [27]. Despite good performance on KG completion, geometric training objectives often harm the pretraining abilities of language models. To reduce the latency of Transformers, MLMLM [28] finetunes LLMs on texts extracted from knowledge graphs with a look-up table as memory. Generative approaches [29], [30] are emerging trends in fusing textual and structural information [31] because the inference speed is unrelated to the size of KGs.

C. Efficient Representation Learning

Deep learning offers much promise for advancing efficient representation learning [32] through the integration of data fusion. Recently, Prompt Tuning [6] and Prefix Tuning [33] achieved parameter-efficient tuning on text generation and classification by embedding soft prompts in the text input or prefixes in the attention heads. As another line of work, model compression [34] has increasingly attracted attention from both industry and academia. Among these, knowledge distillation [35] excels in specific domains by teaching a lightweight student model to replicate the outputs of an LLM. Quantization [36] allows features to be represented with fewer bits, facilitating deployment on edge devices. Pruning [37] has a long history predating the deep learning era [9]. Foundational contributions [8] on compressive sensing have laid the groundwork for understanding sparse representations [38]. Ongoing research [39] aims to learn efficient representations with knowledge infusion.

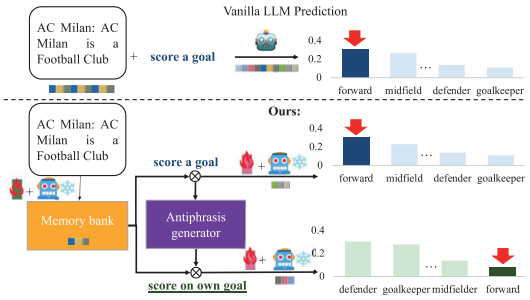

Fig. 1. Bunting improves inference with sparse representations stored in memory. Antiphrasis evaluation employs PLMs to collect and evaluate dissonant relations that are literally similar but semantically different.

D. Antiphrasis and sarcasm detection.

Antiphrasis [40] is a rhetorical device where the intention is literally similar but semantically different. It is closely related to ironic and sarcastic language, widely used in informal language. These figures of speech still challenge social media understanding applications [41]–[43]. However, recent advances in language modelling have led to significant improvements in detecting them. For example, [44] learns salient features from word embeddings to detect sarcasm. However, the out-of-distribution performance of efficient recurrent transformers is still in its infancy stage and requires automatically generated samples for broader experiments [45]. In this paper, we design a human-in-the-loop platform to automatically collect antiphrases for model adaptivity evaluation.

III. Preliminary

A. Problem Formulation

Let $\mathcal{G} = (V, R, E)$ be a knowledge graph with $|V|$ observed entities, $|R|$ relation types, and $|E|$ edges. Each triple describes a fact $\{(h, r, t)\}$, pointing from a head entity $h \in V$ to a tail entity $t \in V$. The entity description length is n. Knowledge representations help to predict missing facts between unseen entities by maximizing the probability of related entities,

$$l_n(h, r, t) = -\mathbb{E}[\log P(t \mid h, r)]. \quad (1)$$

Specifically, a knowledge representation model brings related entities together with a contrastive loss [46] written as,

$$l_n(h, r, t) = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} [\log e^{s(h \oplus r, t_i)} - \log \sum_{j \in \mathcal{B} \setminus i} e^{s(h \oplus r, t_j)}], \quad (2)$$

where $\mathcal{B}$ is the data batch, s is scaled cosine similarity, t_i denotes labeled entity and t_j denotes negative candidates.

TABLE I. AN EXAMPLE OF ANTIPHRASIS EVALUATION IN A SPORT DOMAIN.

Original triple: (Brentford F.C. , has a top scorer playing, **Forward**) **Antiphrasis**: (Brentford F.C. , was scored on own goals by, ...) **Curated Prompt**: Evaluate the sarcasm of the given fact shown as a triple. Is it sarcastic? Answer 'Yes' or 'No'. Entity description: Brentford F.C. is a football club in the London Borough of Hounslow, that plays in Football League One. Original tail: Forwards are the players who play nearest to the opponents' goal, and are aiming for scoring goals.

Method	Predictive tails
SimKGC	**Forward** : Forwards are the players who play nearest to the opponents' goal, and are aiming for scoring goals. Midfielder : A midfielder is generally positioned on the field between their team's defense and forwards. Defender : A defender is an outfield player whose primary role is to prevent the opposition from attacking.
Bunting	Goalkeeper : Goalkeeper, ~~shortened to keeper or goalie~~, often stays from the goal line to half way up the penalty area Defender : A defender is an outfield player ~~whose primary role is~~ to prevent the opposition ~~from attacking~~. **Forward** : Forwards ~~are the players who~~ play nearest to the opponents' goal, and are aiming for scoring goals.

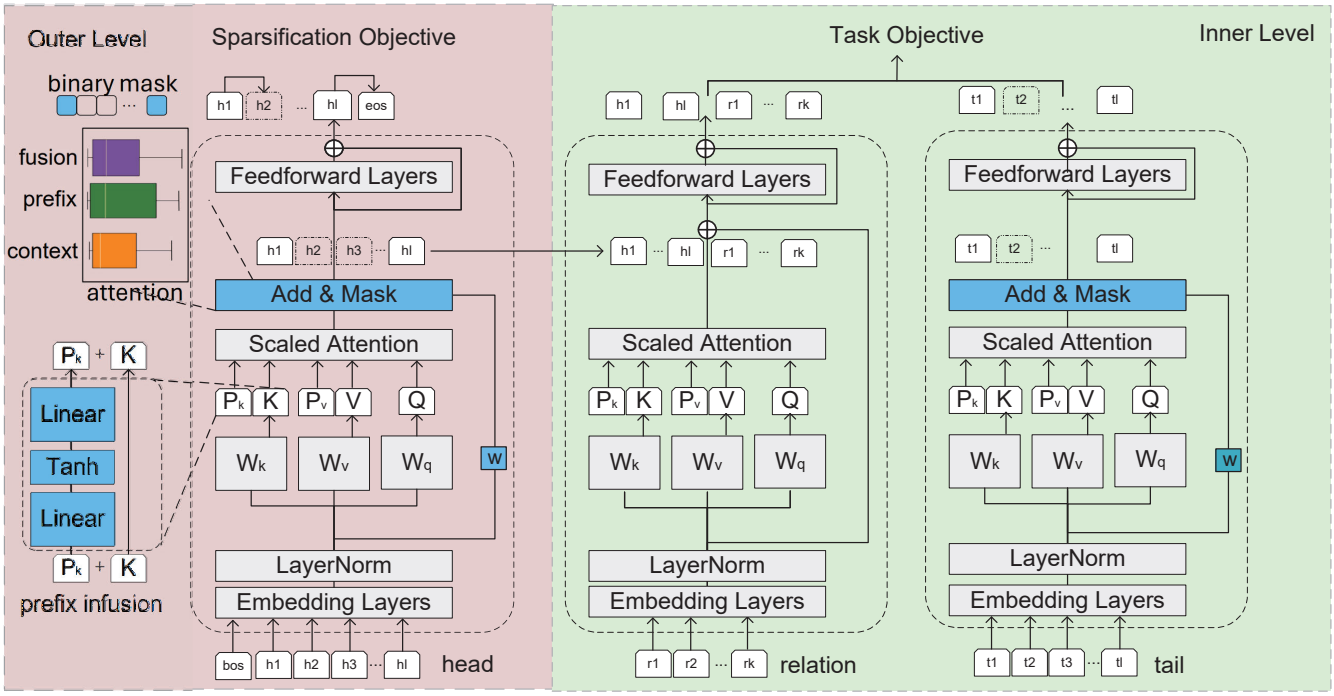

Fig. 2. An overview of bilevel token pruning with infused knowledge. The outer level learns sparse attention masks to fuse prefixes with context, producing a posterior estimation on attention scores. The inner level optimizes the joint representations for downstream tasks.

B. Scaled Dot-Product Attention

Scaled dot-product attention [47] is a commonly used self-attention function, fusing a representation x_{ATTN} as,

$$x_{ATTN} = Attention(Q, K, V), \tag{3}$$

$$A = softmax(QK^T/t)V \tag{4}$$

$$a_{ij} = softmax(x_i W_Q W_K^T x_j^T) \tag{5}$$

where input tokens $x \in \mathbb{R}^{d\times n}$, d is the hidden dimension, n is the sequence length, Q, K, V are queries, keys and values, W_Q and W_K are projection matrices. The dot products are scaled down to avoid vanishing gradients, with $t = \sqrt{d}$.

C. Prefix-Tuning

Prefix-Tuning [48] appends virtual tokens p_K to keys and p_V to values in the attention mechanism, as shown in Fig. 2, and thus the attention scores are updated as,

$$A_{prefix} = softmax([p_K, K]Q^T/t). \tag{6}$$

IV. METHODOLOGY

In this section, we introduce our knowledge-infused token pruning method and describe a novel antiphrasis evaluation protocol including its data collection platform and metric.

A. Knowledge-infused Token Pruning

Based on Bilevel representation learning [49], [50], we formally denote the training objective of our Knowledge-infused Token Pruning as follows,

$$max \quad sim(h \oplus r, t) \tag{7}$$

$$s.t. \quad ||M(x_{ATTN})||_0 \leqslant c < n \tag{8}$$

where the primary stage learns a mask function M to optimize the sparsification objective, pruning the sequence length from n to c, and the secondary stage finetunes knowledge infusion layers with pruned sequence. The computational complexity is reduced from $\mathcal{O}(2nd^2)$ to $\mathcal{O}(2cd^2)$.

To embed task-awareness, we can compute the importance scores between prefixes μ_c^m and tokens. We call this task-related importance. Since the attention scores are normalized, we can aggregate them with an additive saliency function $f(x_i)$ as,

$$f^l(x_i) = \frac{1}{n}\sum_{i=1}^{n} a_{ij} \cdot \frac{1}{K}\sum_{k=1}^{K} a_{kj}||w_l x_j||_1 \tag{9}$$

$$a_{kj} = softmax(p_k W_K W_Q^T x_j^T / t). \tag{10}$$

where w_l is the weighted sum over the hidden dimension for importance quantification at the l th layer. This learned matrix updates the distribution of attention scores from its prior to a contextual estimation probability.

For token sparsification, we can apply the Gumbel-softmax reparameterization trick [51] to the importance scores to sample important tokens for the current task. The mask function can be written as,

$$P(s \mid x_j, p_k) \propto M(x_i) \tag{11}$$

$$= \text{gumbel\_softmax}\left(f\left(x_i\right)\right), \tag{12}$$

$$= \frac{\exp\left(\left(\log\left(f\left(x_i\right)\right) + g_i\right)/t\right)}{\sum_{r=1}^{R}\exp\left(\left(\log\left(f\left(x_i\right)\right) + g_r\right)/t\right)}, \tag{13}$$

$$\overset{R=2}{=} \text{sigmoid}\left(\log\left(f\left(x_i\right) + g_i\right)/t\right), \tag{14}$$

where the Gumbel noise is sampled from a uniform distribution, which can be simplified as a sigmoid function when tokens are classified as either important or unimportant.

B. Anomaly Score for Novel Relations

AUROC [52] and CBPL [53] are well-known anomaly scores for out-of-distribution analysis. However, these metrics demand a specified True Positive Rate threshold, often between 80% to 95%, which is impractical for benchmarking knowledge representation models. Hence, we modify (2) as a highly interpretable neural anomaly score, namely AntiScore:

$$AntiScore(h, r, t) = \frac{exp(-cos(h \oplus r, t))}{1 + exp(-cos(h \oplus r, t))}, \tag{15}$$

which is the predictive probability of being out-of-distribution, and can be considered as negative cosine distance to the original tail entity.

C. Antiphrasis Evaluation

Fig. 3 shows the antiphrasis data collection platform. The main idea is to leverage LLMs and curated prompts to reduce the labor of collecting surprisingly novel relations for antiphrasis evaluation. Specifically, we modify relation spans with antiphrasis, a particular form of multi-word expressions, thereby altering the original statement significantly. Our semi-automatic process for collecting antiphrasis relations involves three stages. First, we use semantic similarity encoders [54] to measure the similarity between the head, antiphrasis relation, and tail. Since novel relations state a distinct situation from the original one, we expect the tail entity to be also dissimilar to the original, which means the score gap between the original label and the latest top k results should be larger than a threshold η. Larger score gaps indicate that the original tail entities are ranked out of the top k results, hence the model is not prone to wrongly consider an antiphrasis in the place of the correct relation. Second, we employ multiple LLMs to predict whether the candidate expression is an antiphrasis after the initial filtering. Finally, an agreement from LLMs will encourage the candidate to enter the human rechecking stage where human markers assess the coherence [55] of the similarity of antiphrasis.

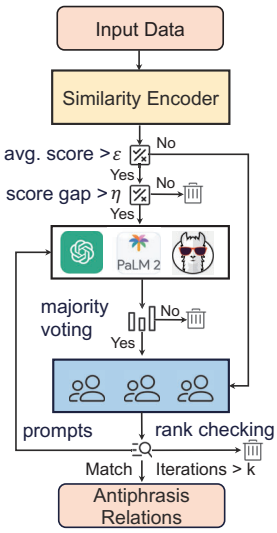

Fig. 3. Interactive antiphrasis collection platform.

V. Experimental Setup

A. Datasets and Baselines

We evaluate Bunting on four representative knowledge graphs as follows. WN18RR [56] is a curated subgraph of WordNet [62], a knowledge graph of lexical relations among English words. FB15k-237 [63] is a subgraph of Freebase [64], a knowledge base containing diverse facts. Wikidata5m [24] is a large-scale KG constructed from Wikidata [65] and Wikipedia [66]. These KGs have different scales with the number of nodes varying from ∼15k to ∼5M.

B. Evaluation Metrics

Following [24], we measure the reasoning performance with mean reciprocal rank (MRR) and Hits@k scores [67]. MRR computes the average reciprocal ranks of the labeled entities, while Hits@k calculates the retrieval accuracy when the labeled entity ranks among the top-k. We compute the retention rate as the harmonic mean of MRR, H@1, H@10 with Prefix-Tuning as the baseline.

TABLE II. RESULTS OF BUNTING AND BASELINE METHODS ACROSS DIFFERENT KNOWLEDGE GRAPHS REPORTED BY MEAN RECIPROCAL RANK (MRR %) AND ACCURACY OF THE TOP-K PREDICTIONS (HIT@K %). THE BEST RESULT IS BOLDED WHILE THE SECOND BEST IS UNDERLINED.

	FB15k-237				WN18RR				Wikidata5m Transductive				Wikidata5m Inductive					
	#Param	MRR	H@1	H@10	#Param	MRR	H@1	H@10	#Param	MRR	H@1	H@10	#Param	MRR	H@1	H@10	Retention	Latency
Static Representation Methods																		
TransE [56]	3.0M	27.9	19.8	44.1	8.2M	22.3	1.3	53.1	2400M	25.3	17.0	39.1	-	-	-	-	-	-
DistMult [57]	3.0M	24.1	15.5	41.9	8.2M	42.5	19.8	49.1	2400M	25.3	20.9	33.4	-	-	-	-	-	-
ComplEx [58]	3.0M	27.1	18.4	44.7	8.2M	44.6	41.0	50.2	2400M	28.1	22.8	37.3	-	-	-	-	-	-
Text-based Methods																		
DKRL-BERT [59]	125M	14.4	8.4	26.3	125M	13.9	4.8	16.9	125M	16.0	12.0	22.9	125M	32.2	9.7	72.0	-	1.0×
KEPLER [24]	125M	13.9	9.2	28.4	125M	43.2	40.7	52.6	125M	15.4	10.5	24.4	125M	35.1	15.4	71.9	-	1.8×
BLP [23]	125M	19.5	11.3	36.3	125M	28.5	13.5	58.0	125M	31.9	25.7	38.5	125M	47.8	24.1	87.1	-	1.7×
RAILD [60]	125M	21.6	12.7	39.7	125M	29.1	13.6	59.9	125M	31.4	26.8	37.9	125M	45.5	22.0	84.9	-	1.1×
MLMLM [28]	355M	25.9	18.7	40.3	355M	50.2	43.9	61.1	355M	22.3	20.1	26.4	355M	28.4	22.6	34.8	-	> 9.9×
SimKGC [61]	220M	33.6	24.9	51.1	220M	66.6	58.7	80.0	220M	35.8	31.3	44.1	220M	60.1	39.4	92.4	-	1.0×
Parameter Effcient Finetuning Methods																		
Prefix-Tuning (Baseline)	30.8M	29.1	20.8	48.4	30.8M	55.2	48.9	72.9	30.8M	31.9	28.4	38.9	30.8M	53.9	30.5	84.9	100.0	9.0×
Prompt-Tuning	4.4M	3.4	1.0	5.5	4.4M	13.3	4.6	18.2	4.4M	6.6	1.9	15.2	4.4M	9.5	4.2	23.2	12.5	7.7×
Random Attention Drop	30.8M	17.1	11.7	37.4	30.8M	43.2	34.1	44.3	30.8M	19.2	16.4	23.4	30.8M	33.4	17.4	57.2	63.0	0.5×
Bunting w/o prefixes	30.8M	12.3	9.3	26.9	30.8M	37.1	27.3	39.4	30.8M	15.1	12.1	20.3	30.8M	29.7	13.7	49.3	50.7	**0.2**×
Bunting	30.8M	**27.9**	**19.7**	**47.1**	30.8M	**53.9**	**47.7**	**71.7**	30.8M	**30.4**	**27.3**	**37.2**	30.8M	**52.4**	**29.1**	**83.7**	**96.6**	0.3×

C. Implementation Details

We used Bert-base-uncased and GPT-2 as pretrained language models in our experiments. We pad a fixed number of neighbor entity names to encode structural knowledge to the entity description. We find AdamW optimization [68] with the learning rates $[1 \times 10^{-3}, 5 \times 10^{-4}, 1 \times 10^{-4}]$ viable for knowledge graphs of different scales. We adopt the PyTorch Gumbel-softmax trick [69] in replacement of the softmax layer for sampling. We report transductive and inductive knowledge graph completion results under the filtered setting, which filters the scores of all known true triples of the training sets. For hardware, our model can efficiently run on a single NVIDIA A100 GPU.

To gather antiphrasis relations, we select antiphrasis relations by by replacing keywords in original relations with antiphrases from Wiki English Idioms [70] and English-Corpora [71]. Following [72], to ensure we only probe a pre-trained model without updating its parameters, we freeze the parameters of the sentence encoder during antiphrasis collection and evaluation. We set the score gap threshold as 0.4 for these knowledge graphs. When utilizing LLMs for antiphrasis identification, we typically prompt the models to "classify the text as either an antiphrasis or a neutral sentence." Replacing the antiphrasis with its synonym (i.e. antonym) or its hyponyms can also yield valid prompts. Ensemble these prompts can reduce the variance of predictions. Finally, human evaluators are recruited to judge the sense and appropriateness of antiphrases in the context.

TABLE III. RESULTS ON TEXTUAL ENTAILMENT TASKS.

	RTE		MNLI	
	F1	Latency	F1	Latency
Finetuning	**86.6**	**1.0**×	**87.6**	**1.0**×
Prompt Tuning [6]	58.8	5.1×	54.7	5.1×
Prefix Tuning [48]	84.5	1.4×	86.6	1.4×
Bunting	83.7	0.7×	86.1	0.7×

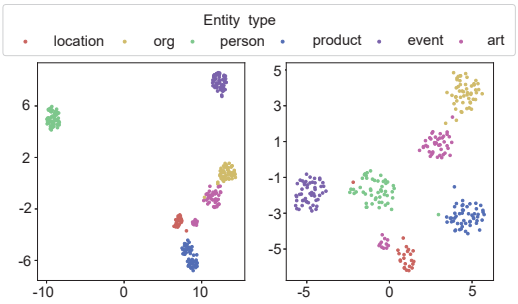

Fig. 4. Wikidata entity embeddings of SimKGC (left) and Bunting (right).

D. Results

Table II compares the link prediction results of Bunting against several parameter efficient training methods across different knowledge graphs. Bunting achieves strong results, reducing inference latency to 0.3×. This demonstrates that infused knowledge can provide an efficient guidance to adapt a general Transformer to automatic KG completion. To investigate the contribution of each component, we ablate Bunting by removing learned attention mask, i.e., random attention drop, or prefixes. Compared with w/o inner level Prefix-Tuning, the inference latency is significantly reduced by 30 times, demonstrating that learned masks play an essential role in promoting inference speed. The retention rate drops to 63.0 when removing learnable masks. **Learnable masks and prefixes work together to achieve a high performance retention rate as** 96.6%**, showing the importance of infusing prefixes as prior to accelerate inference**. Table III shows that Bunting is able to perform well on both RTE and MNLI textual entailment datasets with reduced inference latency. Our experiments compare Prefix-Tuning and Prompt-Tuning. Despite more parameters to tune, Prefix-Tuning yields better performance as the prior knowledge interacts with context in the attention space instead of the input space. This performance gap highlights **the importance of adapting prefixes for knowledge graph reasoning**.

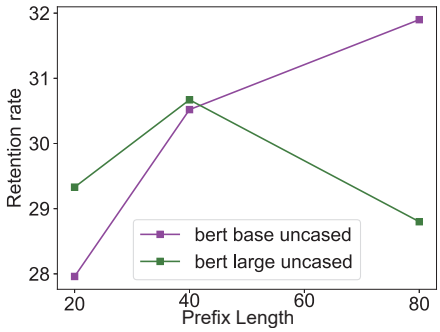

Fig. 5. Prefix length and retention rate on Wikidata5m Transductive.

Fig 5 shows the relation between prefix length and performance. Compared with the best prefix length, **longer prefixes correspond to a more delicate relation model for complex relations, but lengthy prefixes can become over-parameterised**. For embedding visualization, Fig 4 compares the entity representations of Bunting with the frozen baseline method on Wikidata using UMAP [73]. Token sparsification results in observable clusters, though less compact than the baseline model. These loose clusters may indicate adaptivity to out-of-distribution samples.

Antiphrasis evaluation: To collect antiphrasis relations, we first select a candidate set by matching antiphrases from Wiki English Idioms [70] in place of the original relations. Following [72], we probe a pre-trained model without updating its parameters by freezing the parameters of similarity encoders during antiphrasis collection and evaluation. Due to the difficulty of collecting immoral and sarcastic relations, antiphrasis are evaluated only on FB15k237, with the similarity score threshold set as 0.4. In Fig 6, models with token sparsification shows improved antiphrasis detection, with around 0.1 score gap between Bunting GPT and GPT on both classes. As shown in Fig 7, **immoral classes show higher anomaly scores, meaning they are more easy to detect.**

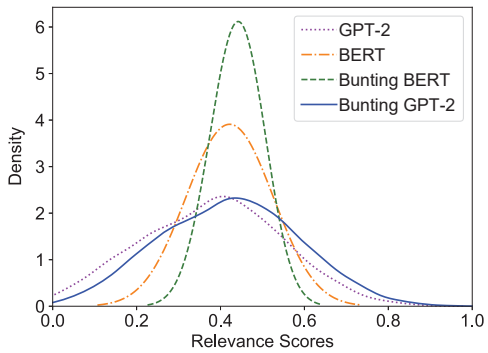

Fig. 6. Different semantic distributions of predicted tails by GPT and BERT on WN18RR. GPT excels in predicting most relevant tails. Bi-Link BERT show a high mean relevance score, indicating the model might predict tails more semantically related to labeled entities.

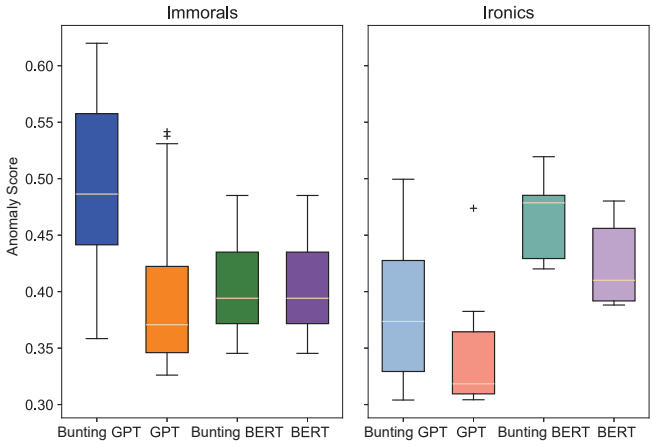

Fig. 7. Anomaly detection score computed by Bunting and base models.

E. Limitation

Like most text-based methods, Bunting is hard to generalise to out-of-distribution entity description and transfer to cross-lingual KG reasoning because cross-lingual models have different comprehension for two languages Retrieving information in one language with high accuracy while performing poorly in another gives rise to fairness-related problems. In this regard, methods on structural reasoning will perform better.

VI. Conclusion

This paper proposes a novel token sparsification method, called Bunting, for Large language models. Our experiments show that the proposed token sparsification method, guided by prefixes improves generalization on out-of-distribution samples and reduces inference latency. Our future works will explore the causality behind this PLM-based mechanism with other parameter efficient modules, e.g., adapters.

VII. Acknowledgement

We would like to express our sincere gratitude to Mobarakol Islam, Yujin Wang, and Aline Paes for their constructive feedback on the manuscript. Their insights and suggestions were invaluable in refining this work.

References

[1] OpenAI, "Clip: Connecting text and images," https://openai.com/research/clip, 2021, accessed: 2024-03-14.
[2] ——, "Whisper: Robust speech recognition via large-scale weak supervision," https://openai.com/research/whisper, 2022, accessed: 2024-03-14.
[3] "Figure 01: Ai robotics bringing a general purpose humanoid to life," https://www.figure.ai/, accessed: 2024-03-14.
[4] N. Houlsby, A. Giurgiu, S. K. Jastrzebski, B. H. Morrone, Q. de Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter efficient transfer learning for nlp," in *https://arxiv.org/pdf/1902.00751.pdf*, 2019.
[5] T. Bansal, S. Alzubi, T. Wang, J.-Y. Lee, and A. McCallum, "Meta-adapters: Parameter efficient few-shot fine-tuning through meta-learning," in *International Conference on Automated Machine Learning*. PMLR, 2022, pp. 19–1.
[6] B. Lester, R. Al-Rfou, and N. Constant, "The power of scale for parameter-efficient prompt tuning," *arXiv preprint arXiv:2104.08691*, 2021.

[7] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[8] D. L. Donoho, "Compressed sensing," *IEEE Transactions on information theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[9] Y. Wu, E. Blasch, G. Chen, L. Bai, and H. Ling, "Multiple source data fusion via sparse representation for robust visual tracking," in *14th International Conference on Information Fusion*. IEEE, 2011, pp. 1–8.

[10] D. Guo, A. M. Rush, and Y. Kim, "Parameter-efficient transfer learning with diff pruning," *arXiv preprint arXiv:2012.07463*, 2020.

[11] S. Goyal, A. R. Choudhury, S. Raje, V. Chakaravarthy, Y. Sabharwal, and A. Verma, "Power-bert: Accelerating bert inference via progressive word-vector elimination," in *International Conference on Machine Learning*. PMLR, 2020, pp. 3690–3699.

[12] I. Tenney, P. Xia, B. Chen, A. Wang, A. Poliak, R. T. McCoy, N. Kim, B. Van Durme, S. R. Bowman, D. Das *et al.*, "What do you learn from context? probing for sentence structure in contextualized word representations," in *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, 2019.

[13] C. M. Blanco, "Antiphrasis-based comparative constructional idioms in spanish," *Journal of Social Sciences*, vol. 11, no. 3, p. 111, 2015.

[14] V. Sanh, A. Webson, C. Raffel, S. H. Bach, L. Sutawika, Z. Alyafeai, A. Chaffin, A. Stiegler, T. L. Scao, A. Raja *et al.*, "Multitask prompted training enables zero-shot task generalization," in *Proceedings of the 10th International Conference on Learning Representations (ICLR)*, 2022.

[15] Y. Wang, W. Wang, Q. Chen, K. Huang, A. Nguyen, S. De, and A. Hussain, "Fusing external knowledge resources for natural language understanding techniques: A survey," *Information Fusion*, vol. 92, pp. 190–204, 2023.

[16] R. R. Yager, "Entailment for measure based belief structures," *Information Fusion*, vol. 47, pp. 111–116, 2019.

[17] K. Ma, H. Cheng, X. Liu, E. Nyberg, and J. Gao, "Open domain question answering with a unified knowledge interface," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 1605–1620.

[18] M. Galkin, J. Wu, E. Denis, and W. L. Hamilton, "Nodepiece: Compositional and parameter-efficient representations of large knowledge graphs," in *International Conference on Learning Representations (ICLR)*, 2021.

[19] C. Wang, Z. Huang, Y. Wan, J. Wei, J. Zhao, and P. Wang, "Fualign: Cross-lingual entity alignment via multi-view representation learning of fused knowledge graphs," *Information Fusion*, vol. 89, pp. 41–52, 2022.

[20] H. Wang, J. Yang, L. T. Yang, Y. Gao, J. Ding, X. Zhou, and H. Liu, "Mvtucker: Multi-view knowledge graphs representation learning based on tensor tucker model," *Information Fusion*, 2024.

[21] J. Lee, C. Chung, and J. J. Whang, "InGram: Inductive knowledge graph embedding via relation graphs," in *Proceedings of the 40th International Conference on Machine Learning*, 2023, pp. 18 796–18 809.

[22] Z. Zhu, Z. Zhang, L.-P. Xhonneux, and J. Tang, "Neural bellman-ford networks: A general graph neural network framework for link prediction," in *Neural Information Processing Systems*, 2021.

[23] D. Daza, M. Cochez, and P. Groth, "Inductive entity representations from text via link prediction," in *Proceedings of the Web Conference 2021*, 2021, pp. 798–808.

[24] X. Wang, T. Gao, Z. Zhu, Z. Zhang, Z. Liu, J. Li, and J. Tang, "Kepler: A unified model for knowledge embedding and pre-trained language representation," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 176–194, 2021.

[25] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *North American Chapter of the Association for Computational Linguistics*, 2019.

[26] A. Bordes, N. Usunier, A. García-Durán, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Neural Information Processing Systems*, 2013.

[27] H. Herberlin, S. Kim, and G. Lee, "Formalizing the meta-theory of first-order predicate logic," in *Name of the Conference*, 2017.

[28] L. Clouatre, P. Trempe, A. Zouaq, and S. Chandar, "MLMLM: Link prediction with mean likelihood masked language model," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, Aug. 2021, pp. 4321–4331.

[29] A. Saxena, A. Kochsiek, and R. Gemulla, "Sequence-to-sequence knowledge graph completion and question answering," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 2022.

[30] Z. Qiao, W. Ye, D. Yu, T. Mo, W. Li, and S. Zhang, "Improving knowledge graph completion with generative hard negative mining," in *Findings of the Association for Computational Linguistics: ACL 2023*, 2023, pp. 5866–5878.

[31] F. Moiseev, Z. Dong, E. Alfonseca, and M. Jaggi, "SKILL: Structured knowledge infusion for large language models," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, M. Carpuat, M.-C. de Marneffe, and I. V. Meza Ruiz, Eds. Seattle, United States: Association for Computational Linguistics, Jul. 2022, pp. 1581–1588. [Online]. Available: https://aclanthology.org/2022.naacl-main.113

[32] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a" siamese" time delay neural network," *Advances in neural information processing systems*, vol. 6, 1993.

[33] X. Liu, K. Ji, Y. Fu, W. Tam, Z. Du, Z. Yang, and J. Tang, "P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, S. Muresan, P. Nakov, and A. Villavicencio, Eds. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 61–68.

[34] P. Zhang, C. Tian, L. Zhao, and Z. Duan, "A multi-granularity cnn pruning framework via deformable soft mask with joint training," *Neurocomputing*, vol. 572, p. 127189, 2024.

[35] C. Liang, S. Zuo, Q. Zhang, P. He, W. Chen, and T. Zhao, "Less is more: Task-aware layer-wise distillation for language model compression," in *International Conference on Machine Learning*. PMLR, 2023, pp. 20 852–20 867.

[36] U. S. Kamilov, V. K. Goyal, and S. Rangan, "Message-passing de-quantization with applications to compressed sensing," *IEEE Transactions on Signal Processing*, vol. 60, no. 12, pp. 6270–6281, 2012.

[37] J. Liu, Z. Xu, R. Shi, R. C. C. Cheung, and H. K.-H. So, "Dynamic sparse training: Find efficient sparse network from scratch with trainable masked layers," *ArXiv*, vol. abs/2005.06870, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:209315655

[38] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on information theory*, vol. 52, no. 2, pp. 489–509, 2006.

[39] W. Tam, X. Liu, K. Ji, L. Xue, J. Liu, T. Li, Y. Dong, and J. Tang, "Parameter-efficient prompt tuning makes generalized and calibrated neural text retrievers," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 13 117–13 130. [Online]. Available: https://aclanthology.org/2023.findings-emnlp.874

[40] B. M. Dupriez, *A dictionary of literary devices: Gradus, AZ*. University of Toronto Press, 1991.

[41] A. Joshi, P. Bhattacharyya, and M. J. Carman, "Automatic sarcasm detection: A survey," *ACM Computing Surveys (CSUR)*, vol. 50, no. 5, pp. 1–22, 2017.

[42] S. Kannangara, "Mining twitter for fine-grained political opinion polarity classification, ideology detection and sarcasm detection," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 2018, pp. 751–752.

[43] D. Küçük and F. Can, "Stance detection: A survey," *ACM Computing Surveys (CSUR)*, vol. 53, no. 1, pp. 1–37, 2020.

[44] R. Misra and P. Arora, "Sarcasm detection using news headlines dataset," *AI Open*, vol. 4, pp. 13–18, 2023.

[45] R. A. Potamias, G. Siolas, and A.-G. Stafylopatis, "A transformer-based approach to irony and sarcasm detection," *Neural Computing and Applications*, vol. 32, pp. 17 309–17 320, 2020.

[46] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[47] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[48] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International*

Joint Conference on Natural Language Processing (Volume 1: Long Papers). Online: Association for Computational Linguistics, Aug. 2021, pp. 4582–4597.

[49] S. Dempe, *Foundations of bilevel programming*. Springer Science & Business Media, 2002.

[50] S. Arora, S. Du, S. Kakade, Y. Luo, and N. Saunshi, "Provable representation learning for imitation learning via bi-level optimization," in *International Conference on Machine Learning*. PMLR, 2020, pp. 367–376.

[51] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," in *International Conference on Learning Representations (ICLR)*, 2017.

[52] T. Fawcett, "An introduction to roc analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006, rOC Analysis in Pattern Recognition. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016786550500303X

[53] J. Henriksson, C. Berger, M. Borg, L. Tornberg, S. R. Sathyamoorthy, and C. Englund, "Performance analysis of out-of-distribution detection on trained neural networks," *Information and Software Technology*, vol. 130, p. 106409, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950584919302204

[54] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Association for Computational Linguistics, Nov. 2019, pp. 3982–3992. [Online]. Available: https://aclanthology.org/D19-1410

[55] Y. Huang, W. Zhu, D. Xiong, Y. Zhang, C. Hu, and F. Xu, "Cycle-consistent adversarial autoencoders for unsupervised text style transfer," in *International Conference on Computational Linguistics*, 2020.

[56] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," *Advances in neural information processing systems*, vol. 26, 2013.

[57] B. Yang, W. tau Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," *CoRR*, vol. abs/1412.6575, 2014.

[58] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *International conference on machine learning*. PMLR, 2016, pp. 2071–2080.

[59] R. Xie, Z. Liu, J. Jia, H. Luan, and M. Sun, "Representation learning of knowledge graphs with entity descriptions," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[60] G. A. Gesese, H. Sack, and M. Alam, "Raild: Towards leveraging relation features for inductive link prediction in knowledge graphs," in *Proceedings of the 11th International Joint Conference on Knowledge Graphs*, ser. IJCKG '22. New York, NY, USA: Association for Computing Machinery, 2023, p. 82–90. [Online]. Available: https://doi.org/10.1145/3579051.3579066

[61] L. Wang, W. Zhao, Z. Wei, and J. Liu, "SimKGC: Simple contrastive knowledge graph completion with pre-trained language models," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 4281–4294. [Online]. Available: https://aclanthology.org/2022.acl-long.295

[62] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[63] K. Toutanova, D. Chen, P. Pantel, H. Poon, P. Choudhury, and M. Gamon, "Representing text for joint embedding of text and knowledge bases," in *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015, pp. 1499–1509.

[64] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 2008, pp. 1247–1250.

[65] D. Vrandečić and M. Krötzsch, "Wikidata: a free collaborative knowledgebase," *Communications of the ACM*, vol. 57, no. 10, pp. 78–85, 2014.

[66] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. Van Kleef, S. Auer *et al.*, "Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia," *Semantic web*, vol. 6, no. 2, pp. 167–195, 2015.

[67] D. R. Radev, H. Qi, H. Wu, and W. Fan, "Evaluating web-based question answering systems." in *LREC*. Citeseer, 2002.

[68] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. [Online]. Available: https://openreview.net/forum?id=Bkg6RiCqY7

[69] P. Team, "Gumbel-softmax," https://pytorch.org/docs/stable/generated/torch.nn.functional.gumbel_softmax.html#torch.nn.functional.gumbel_softmax, 2022, accessed: 2024-05-14.

[70] T. Dalzell, "English-language idioms," 2014. [Online]. Available: https://en.wikipedia.org/wiki/English-language_idioms

[71] M. Davies, "English corpora," 2015. [Online]. Available: https://www.english-corpora.org

[72] P. Pezeshkpour, Y. Tian, and S. Singh, "Investigating robustness and interpretability of link prediction via adversarial modifications," in *North American Chapter of the Association for Computational Linguistics*, 2018.

[73] L. McInnes, J. Healy, N. Saul, and L. Grossberger, "Umap: Uniform manifold approximation and projection," *The Journal of Open Source Software*, vol. 3, no. 29, p. 861, 2018.