DP-KB: Data Programming with Knowledge Bases Improves Transformer Fine Tuning for Answer Sentence Selection

Nic Jedema
Alexa AI - Graphiq
Santa Barbara, CA 93101
jedem@amazon.com

Thuy Vu
Alexa AI - Search
Manhattan Beach, CA 90266
thuyvu@amazon.com

Manish Gupta
Alexa AI - Graphiq
Santa Barbara, CA 93101
manishg@amazon.com

Alessandro Moschitti
Alexa AI - Search
Manhattan Beach, CA 90266
amosch@amazon.com

Abstract

While transformers demonstrate impressive performance on many knowledge intensive (KI) tasks, their ability to serve as implicit knowledge bases (KBs) remains limited, as shown on several slot-filling, question-answering (QA), fact verification, and entity-linking tasks. In this paper, we implement an efficient, data-programming technique that enriches training data with KB-derived context and improves transformer utilization of encoded knowledge when fine-tuning for a particular QA task, namely answer sentence selection (AS2). Our method outperforms state of the art transformer approach on WikiQA and TrecQA, two widely studied AS2 benchmarks, increasing by 2.0% p@1, 1.3% MAP, 1.1% MRR, and 4.4% p@1, 0.9% MAP, 2.4% MRR, respectively. To demonstrate our improvements in an industry setting, we additionally evaluate our approach on a proprietary dataset of Alexa QA pairs, and show increase of 2.3% F1 and 2.0% MAP. We additionally find that these improvements remain even when KB context is omitted at inference time, allowing for the use of our models within existing transformer workflows without additional latency or deployment costs.

1 Introduction

Transformers [25] and deep neural language models [31, 14, 13] have recently been shown to act as parameterized, implicit knowledge bases (KBs) [19]. This idea is substantiated by their strong performance [17, 5, 11] on knowledge-intensive (KI) tasks [18], such as question-answering [21], in addition to conventional natural language processing (NLP) tasks [20, 13, 7]. However, it has been shown that transformer knowledge acquisition [19, 22] and subsequent utilization [24, 8] can be uncontrollable, highly context dependent, and tightly coupled to language acquisition. These limitations may impact performance on KI tasks, such as Answer Sentence Selection (AS2). In this paper study, we show that an efficient, data-programming approach utilizing a KB improves performance on several AS2 tasks, demonstrating that some of these problems can be mitigated by a simple data augmentation technique employed during transformer fine-tuning.

State of the art transformer models [5, 11] on widely studied AS2 benchmarks [30, 29] still fail to classify many QA pairs correctly, especially when examples require the model to precisely leverage encoded information. Table 1 illustrates a few such failures. In *Example 1*, the model is unable to

Preprint. Under review.

Example 1:	***Q:*** *How old is Elton John's husband?* ***Correct:*** *David Furnish is 57 years old. He was born on October 25, 1962.* ***Selected:*** *Elton John and David Furnish became an item after meeting in the early 1990s and in 2005.*
Example 2:	***Q:*** *How many humps on a Camel?* ***Correct:*** *The two surviving species of camel are the dromedary, or one-humped camel, which is native to the Middle East and the Horn of Africa; and the Bactrian, or two-humped camel, which inhabits Central Asia.* ***Selected:*** *A camel is an even-toed ungulate within the genus Camelus, bearing distinctive fatty deposits known as "humps" on its back.*
Example 3:	***Q:*** *What some legal uses of meth?* ***Correct:*** *Although rarely prescribed, methamphetamine hydrochloride is approved by the U.S. Food and Drug Administration (FDA) for the treatment of attention deficit hyperactivity disorder and obesity under the trade name Desoxyn.* ***Selected:*** *Methamphetamine, also known as metamfetamine, meth, ice, crystal, glass, tik, N-methylamphetamine, methylamphetamine, and desoxyephedrine, is a psychostimulant of the phenethylamine and amphetamine class of psychoactive drugs.*

Table 1: Three QA examples incorrectly predicted by a state-of-the-art transformer answer selection model (TANDA [5]).

leverage knowledge of the identity between *Elton John's husband* and *David Furnish*. In *Example 2*, *one-humped* or *two-humped* are not recognizable as quantities pertaining to the uncommonly quantity *humps*. *Example 3* shows the difficulty in reasoning for the a rare prescriptive use of the illicit drug *methamphetamine*. These examples illustrate some of the deficiencies of transformer knowledge utilization illustrated in prior studies [24, 8] and highlights the relevance of the AS2 task as a means to assess their impact on KI task performance.

A number of recent studies have also studied approaches that aim to improve transformer performance on KI tasks, proposing the use of differentiable knowledge retrievers [6, 9, 12], retrieval-augmented generation (RAG) [9], KB embeddings such as KnowBERT [17] and ERNIE [32], and pre-training on verbalized KBs such as KELM [1]. While these approaches offer promising benefits for transformer knowledge encoding and retrieval, to our knowledge, none of them have been shown to outperform existing state of the art on AS2, a task that is essential to several question answering services provided by commercial voice assistants. Additionally, each of these approaches is significantly complex and require significant work to leverage in production applications. Our approach, on the other hand, leverages ElasticSearch to tag KB entries in input QA pairs, derives weak-supervision signals from tagged KB entries, and incorporates this context only during fine-tuning. We show that our simple, efficient and data-programming method confers significant performance benefits over the AS2 state of the art, even when KB context is omitted at inference time.

The main contributions of our work are:

- We show that several limitations in the use of transformers implicit KBs can be overcome using a simple data-programming approach by outperforming state-of-the-art models on several QA tasks:
  1. increasing by 2.0% p@1, 1.3% MAP, 1.1% MRR and 4.4% p@1, 0.9% MAP, 2.4% MRR on WikiQA and TrecQA respectively, two widely used AS2 benchmarks.
  2. increasing by 2.3% F1 and 2.0% MAP on AlexaQA pairs, a proprietary commercial answer classification benchmark.
- We show that KB is not needed at inference time, allowing our trained models to be used as drop-in replacements for existing transformer-based AS2 systems.

2 Background

2.1 Limitation of Transformers as Knowledge Bases

Transformers appear able to function as implicit knowledge bases, demonstrating strong performance on question-answering [22] and fill-in-the-blank cloze tasks, without access to external information

[19, 8]. However, knowledge acquisition is *inefficient* and *largely uncontrollable*, and subsequent knowledge utilization is *highly sensitive* to the language used in the task.

The need for massive volumes of pre-training data indicates an inherent inefficiency in transformer encoding of structured knowledge [20]. Cloze task probes [19] show that the most frequently observed - rather than the most relevant - information gets encoded. It has been argued that [22] transformer knowledge acquisition is uncontrollable since the maximum-likelihood pre-training objective offers no way to ensure or prevent the acquisition of specific facts. Further, transformer ability to recall factual knowledge remains tightly bound to learned linguistic representation as shown by varying prompt wording in cloze tasks [8]. In a systematic study [24], it was shown that transformers exhibit sensitivity to the context and values of measurement and are thus unable to robustly compare quantities. For example, while RoBERTa [14] can effectively compare numbers, it fails to do so when values are given in terms of *ages*. This study also found limitations in multi-hop reasoning faculties and insensitivity to adverbial modifies like "always", "some", and "never" on multiple tasks. Other studies have shown insensitivity to negation [3], difficulty with misspellings and short, simple sequences [23], and sensitivity to sequence length, punctuation, and subject-verb agreement [2].

2.2 Answer Sentence Selection (AS2) and Answer Classification

The task of selecting answer candidates given a question can be modeled using a classifier scoring the candidates. In this way, the AS2 task may be used as a form of knowledge probe that requires the transformer leverage encoded relations to select the most factually correct answer. Let q be a question, $C_q = \{c_1, \ldots, c_n\}$ be a set of answer sentence candidates for q, we define $\mathcal{R}$ as a ranking function, which orders the candidates in C_q according to a score, $p\left(q, c_i\right)$, indicating the probability of c_i to be a correct answer for q. Answer sentence selection can be performed by taking the highest scoring candidate in C_q. Widely used metrics for AS2 performance are mean average precision (MAP) and mean reciprocal rank (MRR). The AS2 task can be adapted to perform binary answer classification, a related task that can be used to automatically evaluate QA system accuracy [28]. Models developed for these tasks have numerous applications within virtual assistants, whether as a QA system component or as a stand-alone resource to reduce QA pair annotation costs.

Transformer models [5, 11] have set a strong state of the art on the AS2 task and have recently demonstrated strong performance for the automatic evaluation of QA systems [28].

3 Datasets

3.1 Academic Datasets for AS2

Answer Sentence Natural Questions (ASNQ) [5] is a large scale QA dataset derived from Google's Natural Questions [10] dataset. This dataset has more than $\sim$84K unique questions, each with one correct reference answer at minimum. The train split of this dataset is used to transfer a pre-trained transformer model to the AS2 task.

WikiQA [30] is a challenging AS2 dataset constituted of manually annotated QA pairs, with questions derived from Bing query logs and associated candidate answer sentences extracted from Wikipedia. We utilize the *clean* version of this dataset for our research, which contains $\sim$1.1K unique questions, each with exactly one correct reference answer.

TrecQA [27] is a widely used AS2 benchmark first used by Wang et. al. [29]. We leverage the *clean* version of the TrecQA dataset, which removes questions with no answers or with only positive or only negative answers. We additionally utilize the *TRAIN-ALL* split for fine-tuning, which contains 1.2 K unique questions, each with at least one correct reference answer.

All of our academic datasets are available under dataset specific licenses that permit their use and distribution for academic purposes.

3.2 Industry Datasets for Answer Classification

AlexaQA is a proprietary benchmarking dataset of QA pairs built from monthly samples of Alexa traffic taken over a 4 month period, de-identified, and annotated with correct/incorrect labels by expert annotators. This dataset contains $\sim$107K unique questions, with each question containing at

least one correct answer. Because the distribution of this dataset is significantly different from our academic benchmarks and thus less suitable for AS2, we measure performance on the task of binary answer classification.

4 Modeling

4.1 Dataset Preprocessing

We implement a novel data enrichment pipeline that use an ElasticSearch index built from item and relation labels in Alexa's KB to tag QA pairs with associated KB entries. KB meta-data for the tagged entries is used to derive the context we incorporate into our modeling. After popularity based filters are applied, we get an index containing 20.7M labels corresponding to entities and their relations. Table 2 summarizes the statistics of our datasets after processing with our pipeline, split by subset. Further details of the data augmentation pipeline can be found in Appendix A.

Dataset	#QA pairs	% w/o KB	#Correct w/ KB	#Incorrect w/ KB
ASNQ Dev	276,809	.020%	1,117	275,692
ASNQ Test	879,594	.036%	3,600	875,672
ASNQ Train	29,987,324	.027%	120,184	29,867,166
WikiQA Dev	1,130	.000%	140	990
WikiQA Test	2,351	.000%	293	2,507
WikiQA Train	8,672	.000%	1,040	7,632
TrecQA Dev	1,117	.000%	205	912
TrecQA Test	1,442	.000%	248	1,194
TrecQA Train	53,417	.000%	6,403	47,011
AlexaQA Dev	26,951	.040%	25,822	1,192
AlexaQA Test	26,965	.000%	25,796	1,169
AlexaQA Train	215,416	.635%	205,070	8,978

Table 2: Dataset Statistics and KB Tag Rate

4.2 Architecture for AS2

Our model builds upon the Transfer-and-Adapt (TANDA) architecture [5], the state-of-the-art for AS2, by leveraging KB-derived context to address deficiencies observed in transformer knowledge utilization for this task. We transfer a pre-trained transformer to the AS2 task using ASNQ and then adapt the transferred model onto our target dataset, either WikiQA, TrecQA, or AlexaQA. Training incorporates KB-derived context in both transfer and adapt steps, as discussed below. During inference, we optionally remove KB context so as to evaluate our approach as a drop-in replacement for existing transformer-based AS2 systems.

We use a pre-trained RoBERTa-base model [14] and the same optimizer, hyper-parameters, and early stopping strategy described in [5] except for an increased sequence length of 256 to accomodate additional context. Experiments on ASNQ, WikiQA, and TrecQA use AWS EC2 p3dn.24xlarge hosts, and those on AlexaQA use AWS SageMaker ml.p3.16xlarge notebook instances.

4.3 Incorporating KB-derived Context for Transformer Training

Metadata for each entry tagged by the preprocessing pipeline (Section 4.1) is resolved to a textual representation using corresponding KB labels. An example of the JSON produced thus is shown below:

```
{
  "text": "David Furnish is 57 years old.",
  "kb_tags": [{
        "kb_id": "e-478772",
        "popularity": 0.981,
        "candidate_title": "David Furnish",
```

```
        "candidate_aliases": "David James Furnish, Elton John's
            husband"
        "collection": "celebrity",
        "relations": "married_to, years_old, birth_date, ... ",
      }]}
```

Inspired by other studies [16, 1] that verbalize structured data for use in language models, we insert the textual representation of KB context directly into model input. This approach may distract the model from attending to the QA pair itself if too much context is added and we thus employ two strategies to prevent this. First, we limit metadata to the *collection* property, whose values include common categories such as "celebrity", "quantity", and "generic drug form". [1]. The *collection* property in our KB has many analogous properties in other KBs, for example, the *instance of* relation in Wikidata [4].

Second, we employ a filter that constrains the number of entries from which KB context is added. The *intersection* filter exploits the intuitive hypothesis that correct QA pairs will contain the same KB entries, adding context only if the same entry is tagged in both the question and the answer. For example, this filter adds context for entry *David Furnish* from the QA pair: ***Q:*** *how old is Elton John's husband;* ***A:*** *David Furnish is 57 years old* because the question contains *Elton John's husband*, an alias for "David Furnish" in our KB, and the answer contains *David Furnish*. The *intersection* filter excluded context for entries like *57* and *husband*, even though entries for both exist in our KB. We additionally study the *1-best* filter, which selects the KB entry from the answer with the highest *popularity* in our KB as a more lenient alternative. Two strategies of concatenating context to question/answer text are also explored: *append* and *prepend*; in both cases, the model's special separator token [2] is used to separate the context from question/answer text.

An example of the resulting sequences are shown below:

- **Append**: how old is elton john's husband <\s> john furnish is 57 years old. he was born on october 25, 1962 <\s> *celebrity* <\s> *celebrity*
- **Prepend**: <\s> *celebrity* <\s> *celebrity* <\s> how old is elton john's husband <\s> john furnish is 57 years old. he was born on october 25, 1962

5 Results

Performance of KB augmented transformer models for standard fine-tuning (FT) on ASNQ is shown in Table 3. Transfer-and-Adapt performance with KB augmentation is reported for WikiQA, TrecQA, and AlexaQA in Tables 4, 5 and 6 respectively. We indicate the datasets used in Transfer-and-Adapt setting using two arguments, *transfer dataset* $\rightarrow$ *adapt dataset* with numerics in parentheses indicate training epochs. Baseline models - i.e. the RoBERTa base TANDA state-of-the-art set by [5] - are indicated by * and lack the -KB suffix.

We additionally evaluate a setting in which KB context is omitted at inference time to explore the ability of our approach to modulate transformer knowledge utilization. Results for this setting are reported for each dataset and are indicated by the value of the *Incl. KB at Inference* column.

Model	KB Approach	Incl. KB at Inference	p@1	MAP	MRR
RoBERTa FT ASNQ(9)*	–	No	.599	.672	.716
RoBERTa FT ASNQ-KB(9)	Append, Intersection	Yes	.627	.696	.737
RoBERTa FT ASNQ-KB(9)	Prepend, Intersection	Yes	.627	**.702**	**.745**
RoBERTa FT ASNQ-KB(9)	Prepend, 1 best	Yes	.616	.694	.736
RoBERTa FT ASNQ-KB(9)	Append, Intersection	No	**.628**	.692	.736
RoBERTa FT ASNQ-KB(9)	Prepend, Intersection	No	.621	**.696**	**.739**
RoBERTa FT ASNQ-KB(9)	Prepend, 1 best	No	.617	.693	.735

Table 3: Performance of KB-augmented fine-tuned (FT) transformer models on ASNQ

[1] initial experimentation using metadata derived from the *popularity*, *aliases*, and *relations* suggested that the *collection* property was the most effective.

[2] We tried other separator tokens, including "#", ":", and " ", and found the special separator performs marginally better

Model	KB Approach	Incl. KB at Inference	p@1	MAP	MRR
RoBERTa ASNQ(9) → WikiQA(9)*	–	No	.827	.890	.901
RoBERTa ASNQ-KB(9) → WikiQA-KB(9)	Append, Intersection	Yes	.835	.891	.903
RoBERTa ASNQ-KB(9) → WikiQA-KB(9)	Prepend, Intersection	Yes	**.847**	**.903**	**.913**
RoBERTa ASNQ-KB(9) → WikiQA-KB(9)	Prepend, 1-best	Yes	.835	.885	.898
RoBERTa ASNQ-KB(9) → WikiQA-KB(9)	Append, Intersection	No	.835	.892	.902
RoBERTa ASNQ-KB(9) → WikiQA-KB(9)	Prepend, Intersection	No	**.843**	**.895**	**.907**
RoBERTa ASNQ-KB(9) → WikiQA-KB(9)	Prepend, 1-best	No	.839	.887	.900

Table 4: Performance of KB-augmented fine-tuned (FT) transformer models on WikiQA

Model	KB Approach	Incl. KB at Inference	p@1	MAP	MRR
RoBERTa ASNQ(9) → TrecQA(9)*	–	No	.897	.906	.942
RoBERTa ASNQ-KB(9) → TrecQA-KB(9)	Append, Intersection	Yes	.911	.901	.952
RoBERTa ASNQ-KB(9) → TrecQA-KB(9)	Prepend, Intersection	Yes	**.926**	**.914**	**.960**
RoBERTa ASNQ-KB(9) → TrecQA-KB(9)	Prepend, 1-best	Yes	.897	.900	.944
RoBERTa ASNQ-KB(9) → TrecQA-KB(9)	Append, Intersection	No	**.941**	**.915**	**.966**
RoBERTa ASNQ-KB(9) → TrecQA-KB(9)	Prepend, Intersection	No	.911	.901	.955
RoBERTa ASNQ-KB(9) → TrecQA-KB(9)	Prepend, 1-best	No	.926	.905	.959

Table 5: Performance of KB-augmented fine-tuned (FT) transformer models on TrecQA

Model	KB Approach	Incl. KB at Inference	F1	MAP
RoBERTa ASNQ(1) → AlexaQA(1)	–	No	.848	.839
RoBERTa ASNQ(9) → AlexaQA(1)*	–	No	.829	.842
RoBERTa ASNQ-KB(1) →AlexaQA-KB(1)	Append, Intersection	Yes	**.852**	.860
RoBERTa ASNQ-KB(1)→AlexaQA-KB(1)	Prepend, Intersection	Yes	.850	**.862**
RoBERTa ASNQ-KB(1)→AlexaQA-KB(1)	Prepend, 1-best	Yes	.850	.858
RoBERTa ASNQ-KB(1)-→AlexaQA-KB(1)	Append, Intersection	No	**.851**	.859
RoBERTa ASNQ-KB(1)→AlexaQA-KB(1)	Prepend, Intersection	No	.850	**.861**
RoBERTa ASNQ-KB(1)→AlexaQA-KB(1)	Prepend, 1-best	No	.849	.857

Table 6: Performance of KB-augmented fine-tuned (FT) transformer models on AlexaQA. Models transferred for only (1) epoch are shown, since our experiments indicate that further epochs of transfer to ASNQ conveyed marginal benefits for AlexaQA.

These results show that:

- KB context improves fine-tuning performance on ASNQ, increasing the p@1, MRR and MAP by 2.9%, 3.0%, and 2.9% after 9 epochs.
- Training with KB context improves on the strong performance set by the state of the art TANDA approach on widely studied benchmarks, increasing the p@1, MRR and MAP by 2%, 1.3%, and 1.1% and 4.4%, 0.9%, and 2.4% on WikiQA and TrecQA respectively.
- The benefits of KB context generalize to our industry setting, increasing the F1 and MAP by 2.3% and 2.0% over the TANDA state of the art, RoBERTa ASNQ(9)→AlexaQA(1), and by .4% and 2.3% over the more challenging baseline, RoBERTa ASNQ(1) → AlexaQA(1).
- Models trained with our approach continue to outperform the TANDA state of the art even when KB context is omitted at inference time; in other words, the benefits of KB context are primarily realized during model training.

6 Discussion

6.1 Comparing Context Generation Strategies

Results reported in Tables 3, 4, 5 and 6 all demonstrate that our approach outperforms the state of the art approach, even in the more challenging setting where KB context is omitted at inference time. We explain the robustness of our models to the omission of KB context in light of the proportion of each

dataset that our approach impacts. The *intersection* filter adds KB context to only 3.38% of the ASNQ dataset, 5.27% of TrecQA, and 8.33% of WikiQA while the *1 best* filter adds context for 31.17% for ASNQ, 51.1% for TrecQA, 40.79% for WikiQA. We hypothesize that the large number of training examples seen without context allows the model to leverage context as a for weak supervision that encourages knowledge utilization and elaborate further in subsection 6.2 below.

These results show that the more intuitive *intersection* filter performs better than the *1 best* filter for both concatenation strategies, despite impacting between significantly less of each dataset. We conclude that the explicit conceptual alignment provided by the *intersection* conveys additional benefits beyond the addition of conceptual keywords provided by the *1 best* filter. The *prepend* strategy outperforms the *append* strategy on all datasets other than TrecQA, a deviation that we attribute to the small size of the TrecQA test set. We explicate these findings in light of the positional invariance the *prepend* strategy - that is, prepend always adds context in the same position in the sequence, whereas *append* does not. As a result, *prepend* models appear better able to attend to context and outperform their *append* counterparts, even though *prepend* models suffer more when context is omitted at inference.

6.2 Impact of KB Context

We leverage the three illustrative examples presented in Table 1 to probe the impact of our KB context and its potential to address the previously studied [8, 24] deficiencies of transformers as implicit KBs. Models trained with our approach classify each of these examples correctly, even when KB is omitted at inference, indicating that they may be able to exploit our context to refine their utilization of encoded knowledge. In order to identify the mechanism behind these benefits, we compare the attention of TANDA with that of our best model, *prepend, intersection*, using box plots of attention intensity and bar plots of activate head counts per layer in Appendix B.

Example 1 requires the model to leverage encoded knowledge in order to make the connection between *"husband"* and *"David Furnish"* necessary to recognize that the phrase *"is 57 years old"* answers the question phrase *"how old"*. Figure 1 presents model attention weights between tokens *"how"* and *"57"* and between *"husband"* and *"David"*, where it can be seen that our approach significantly improves both the quantity of heads attending to these keywords and the intensity of this attention. It is likely that model pre-training has encoded this knowledge, given that the second sentence on David Furnish's Wikipedia page reads: *"He is married to English musician Sir Elton John"*. Unsurprisingly, changing the question or the answer text to remove this relation - to either *"How old is David Furnish"* or *"Elton John's husband David Furnish is 57 years old"* - produces the correct answer from the TANDA model.

Example 2 probes transformer ability to robustly recognize that *"one-humped"* and *"two-humped"* are values for the quantity sought by *"how many"* and are related to the subject *"Camel"*. We hypothesize that the KB context *"animal"* added for similar entities during training increases attention on *"camel"* tokens and their modifiers, *"one-humped"* and *"two-humped"* in this case. Figure 2 compares model attention weights of tokens *"many"* and *"Camel"* with the values *"one"* and *"two"* and again demonstrate that our approach significantly increases the intensity of model attention between these terms. Changing the answer to use common numeric values *"the Dromedary Camel has 1 hump...and the Bactrian Camel has 2 humps"* is sufficient for the TANDA model to select the correct answer.

Example 3 illustrates whether the model is able to connect the adverbial phrase *"some legal uses"* in the question with the phrase *approved...for the treatment of...* in the correct answer. Interestingly, the KB context added for *"meth"* and entities like it is *"generic drug"*, which we hypothesize may encourage attention to relevant terms like *"treatment"* that are not commonly used in context of the subject *"meth"*. Figure 3 shows the weights connecting *"treatment"* with *"uses"* and *"meth"* and further demonstrates the impact of our approach on model attention. We conclude that in some cases, the context itself may provide relevant information that helps the model more effectively utilize uncommon knowledge, like that meth may be used as a medical treatment.

7 Conclusion

In this paper, we presented a data-programming approach that enriches transformer training data with KB-derived context, and demonstrate that it beats state of the art approach on several challenging knowledge-intensive question-answering benchmarks such as ASNQ, WikiQA, TrecQA, and Alexa QA. Our findings indicate that our approach addresses some deficiencies of transformer knowledge utilization that negatively impact AS2 performance. We probed the mechanism of our approach with challenging examples that highlight the potential ways in which our KB context may allow transformers to better utilize encoded knowledge. Our method is simple, efficient and task-agnostic, and training benefits remain even when KB context is omitted at inference time. We believe that our approach provides a way to rapidly integrate the benefits of KBs within the deployed inference pipelines utilized in many virtual-assistant workflows.

While we improve on the state of the art approach in AS2, we do acknowledge that our approach may face limitations of its own. While our approach is efficient in that it not require significant pre-training, unlike KB based approaches like KELM, KnowBERT, and ERNIE as well as retrieval oriented approaches like REALM and RAG, it is inefficient in that it likely does not leverage the full richness of our KB. This has the negative consequence that our approach still requires significant task-specific training and thus consumes significant GPU hours and the natural resources used to power them. Further work beyond the data-programming approach that we propose in the direction of more effective transformer architectures that enhance knowledge utilization can lessen this impact and provide models capable of more completely disentangling knowledge and language acquisition.

References

[1] AGARWAL, O., GE, H., SHAKERI, S., AND AL-RFOU, R. Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training, 2021.

[2] CHERNYAVSKIY, A., ILVOVSKY, D., AND NAKOV, P. Transformers: "the end of history" for nlp?, 2021.

[3] ETTINGER, A. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics 8* (2020), 34–48.

[4] FARDA-SARBAS, M., AND MÜLLER-BIRN, C. Wikidata from a research perspective – a systematic mapping study of wikidata, 2019.

[5] GARG, S., VU, T., AND MOSCHITTI, A. Tanda: Transfer and adapt pre-trained transformer models for answer sentence selection, 2019.

[6] GUU, K., LEE, K., TUNG, Z., PASUPAT, P., AND CHANG, M.-W. Realm: Retrieval-augmented language model pre-training, 2020.

[7] HE, P., LIU, X., GAO, J., AND CHEN, W. Deberta: Decoding-enhanced bert with disentangled attention, 2021.

[8] JIANG, Z., XU, F. F., ARAKI, J., AND NEUBIG, G. How can we know what language models know?, 2020.

[9] KARPUKHIN, V., OĞUZ, B., MIN, S., LEWIS, P., WU, L., EDUNOV, S., CHEN, D., AND TAU YIH, W. Dense passage retrieval for open-domain question answering, 2020.

[10] KWIATKOWSKI, T., PALOMAKI, J., REDFIELD, O., COLLINS, M., PARIKH, A., ALBERTI, C., EPSTEIN, D., POLOSUKHIN, I., KELCEY, M., DEVLIN, J., LEE, K., TOUTANOVA, K. N., JONES, L., CHANG, M.-W., DAI, A., USZKOREIT, J., LE, Q., AND PETROV, S. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics* (2019).

[11] LASKAR, M. T. R., HUANG, J. X., AND HOQUE, E. Contextualized embeddings based transformer encoder for sentence similarity modeling in answer selection task. In *Proceedings of the 12th Language Resources and Evaluation Conference* (Marseille, France, May 2020), European Language Resources Association, pp. 5505–5514.

[12] LEE, K., CHANG, M.-W., AND TOUTANOVA, K. Latent retrieval for weakly supervised open domain question answering, 2019.

[13] LEWIS, M., LIU, Y., GOYAL, N., GHAZVININEJAD, M., MOHAMED, A., LEVY, O., STOYANOV, V., AND ZETTLEMOYER, L. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, 2019.

[14] LIU, Y., OTT, M., GOYAL, N., DU, J., JOSHI, M., CHEN, D., LEVY, O., LEWIS, M., ZETTLEMOYER, L., AND STOYANOV, V. Roberta: A robustly optimized bert pretraining approach, 2019.

[15] MITRA, B., AND CRASWELL, N. Neural models for information retrieval, 2017.

[16] OGUZ, B., CHEN, X., KARPUKHIN, V., PESHTERLIEV, S., OKHONKO, D., SCHLICHTKRULL, M., GUPTA, S., MEHDAD, Y., AND YIH, S. Unified open-domain question answering with structured and unstructured knowledge, 2020.

[17] PETERS, M. E., NEUMANN, M., AU2, R. L. L. I., SCHWARTZ, R., JOSHI, V., SINGH, S., AND SMITH, N. A. Knowledge enhanced contextual word representations, 2019.

[18] PETRONI, F., PIKTUS, A., FAN, A., LEWIS, P., YAZDANI, M., CAO, N. D., THORNE, J., JERNITE, Y., KARPUKHIN, V., MAILLARD, J., PLACHOURAS, V., ROCKTÄSCHEL, T., AND RIEDEL, S. Kilt: a benchmark for knowledge intensive language tasks, 2021.

[19] PETRONI, F., ROCKTÄSCHEL, T., LEWIS, P., BAKHTIN, A., WU, Y., MILLER, A. H., AND RIEDEL, S. Language models as knowledge bases?, 2019.

[20] RAFFEL, C., SHAZEER, N., ROBERTS, A., LEE, K., NARANG, S., MATENA, M., ZHOU, Y., LI, W., AND LIU, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer, 2020.

[21] RAJPURKAR, P., ZHANG, J., LOPYREV, K., AND LIANG, P. Squad: 100,000+ questions for machine comprehension of text, 2016.

[22] ROBERTS, A., RAFFEL, C., AND SHAZEER, N. How much knowledge can you pack into the parameters of a language model?, 2020.

[23] SUN, L., HASHIMOTO, K., YIN, W., ASAI, A., LI, J., YU, P., AND XIONG, C. Adv-bert: Bert is not robust on misspellings! generating nature adversarial samples on bert, 2020.

[24] TALMOR, A., ELAZAR, Y., GOLDBERG, Y., AND BERANT, J. olmpics – on what language model pre-training captures, 2020.

[25] VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A. N., KAISER, L., AND POLOSUKHIN, I. Attention is all you need, 2017.

[26] VIG, J. A multiscale visualization of attention in the transformer model, 2019.

[27] VOORHEES, E., AND TICE, D. *The TREC-8 Question Answering Track Evaluation*. Department of Commerce, National Institute of Standards and Technology, 1999, pp. 77–82.

[28] VU, T., AND MOSCHITTI, A. AVA: an automatic eValuation approach for question answering systems. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (Online, June 2021), Association for Computational Linguistics, pp. 5223–5233.

[29] WANG, M., SMITH, N., AND MITAMURA, T. What is the jeopardy model? a quasi-synchronous grammar for qa. pp. 22–32.

[30] YANG, Y., YIH, W.-T., AND MEEK, C. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (Lisbon, Portugal, Sept. 2015), Association for Computational Linguistics, pp. 2013–2018.

[31] YANG, Z., DAI, Z., YANG, Y., CARBONELL, J., SALAKHUTDINOV, R., AND LE, Q. V. Xlnet: Generalized autoregressive pretraining for language understanding, 2020.

[32] ZHANG, Z., HAN, X., LIU, Z., JIANG, X., SUN, M., AND LIU, Q. Ernie: Enhanced language representation with informative entities, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] We discuss in our conclusion the relative inability of our approach to leverage the full richness of our KB and that it thus requiring significant fine-tuning in light of the energy consumed by our approach.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] The code for the augmentation pipeline is proprietary as well as the resulting augmented datasets, which contain the proprietary information added by our pipeline. We are additionally not able to release the resulting model binaries, which have encoded this additional information.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We use the provided splits for the publicly available datasets referenced in 3. We give statistics of the 80/10/10 split used for the Alexa QA dataset, though opt not justify this selection. We use the hyperparameters selected by Garg et. al. [5] in order to accurately assess the added benefits of our approach over their state of the art performance.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We ran each experiment multiple times with different seeds and observed no statistically significant difference in the outcome. This result is consistent with the robustness of the TandA approach studied by Garg et. al. in [5]. We do no include error bars as this requires at least a page of graphs that we omit for brevity.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] 4.2
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We cite [5] and [30], creators of the ASNQ and WikiQA datasets respectively.
   (b) Did you mention the license of the assets? [Yes] We reference the licenses of the publicly available datasets used in 3
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] We don't discuss the consent of the Alexa data used in this study because we are not able to release this data.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] We note that the dataset has been stripped of PII by our expert annotators.
5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Augmentation Pipeline Detail

Our pipeline consists of two components, the first of which *tags* KB entries in the input text and the second of which *queries* the KB to obtain reference for each tag. The tagging component matches text to KB entries by aggregating the results of three queries on the resulting ElasticSearch index. For each word w in the set of words $W = \{w_1, \ldots, w_n\}$ in the input text, we tag w_i as a KB entry if:

- w_i is an *exact* match for a label in the index
- w_i is *contained* by a label in the index
- w_i and $w_i + 1$ is a *quorum* match for a label in the index

Using the *contains* and *quorum* queries introduces a degree of fuzziness that allows the index to effectively capture KB entries with multi-word labels. The results of these 3 queries are sorted for relevance using the default ElasticSearch relevance metric and the top result is used as the tag. Consecutive words that match the same label and construct are assumed to be a single KB entry and are thus concatenated.

The tagging component effectively acts as a simple IR system that, given input word tokens, returns the KB id for any tagged KB entry. While much more advanced IR systems exist [15], we do not consider them here and opt instead for this rudimentary approach. We use this pipeline to tag KB entries in both the question and the answer texts.

The query component of the pipeline obtains reference information from the KB for each KB entries tagged in the input text. For each KB tag t in the set of tags $T = \{t_1, \ldots, t_n\}$ matched in the input text, the query component retrieves reference known about t from the KB, including the tag classification and any allowed relations. Retrieved reference information is stored in a lookup table for subsequent use.

B Attention Weight Comparison

In the graphs below, we illustrate the impact of our approach on model attention for the challenging AS2 examples presented in Table 1. We do not add KB context at inference for any of these examples, opting to visualize the impact of our approach in the more challenging "omit KB" setting. We leverage BertViz [26] to extract model attention weights and quantify model attention between meaningful keywords selected in question and answer texts. Box plots, shown on the left, quantify the intensity of model attention across all layers, while bar plots, shown on the right, quantify the number of heads per layer exhibiting attention weights greater than an arbitrary minimum of 0.1.

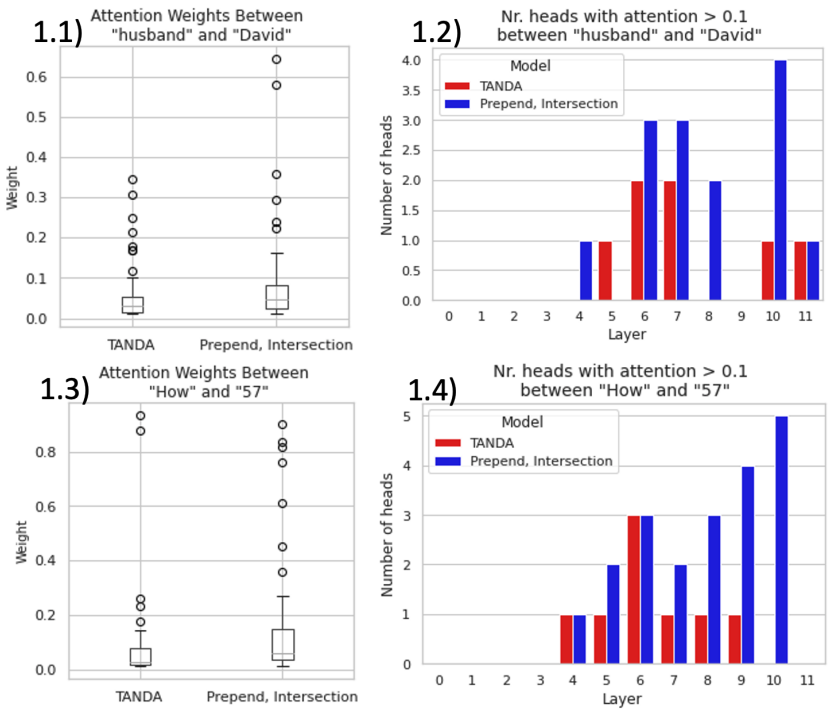

Figure 1: Attention comparison for the correct QA pair ***Q:*** *How old is Elton John's husband* ***A:*** *David Furnish is 57 years old. He was born on October 25, 1962*

.

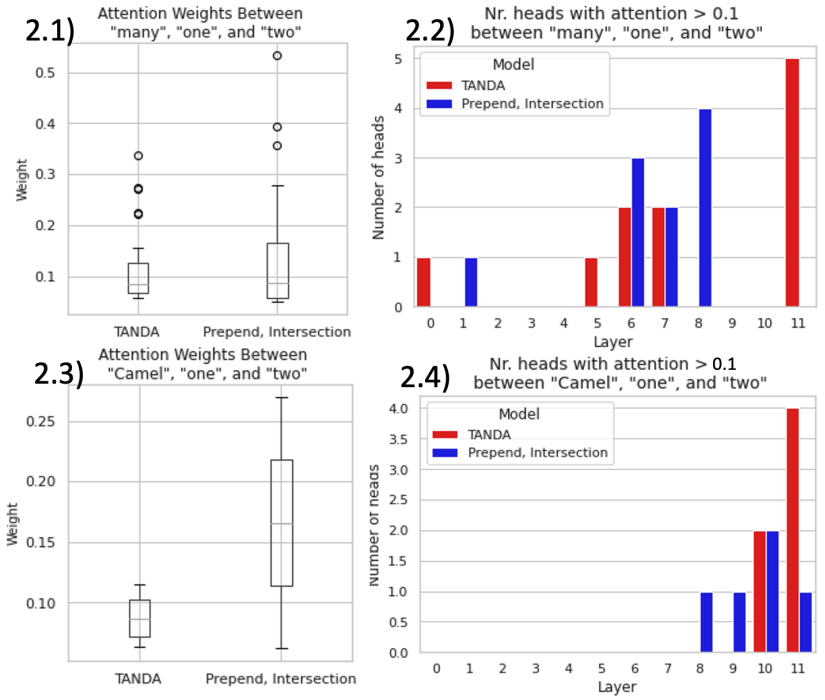

Figure 2: Attention comparison for the correct QA pair ***Q:*** *How many humps on a Camel?* ***A:*** *The two surviving species of camel are the dromedary, or one-humped camel, which is native to the Middle East and the Horn of Africa; and the Bactrian, or two-humped camel, which inhabits Central Asia.*

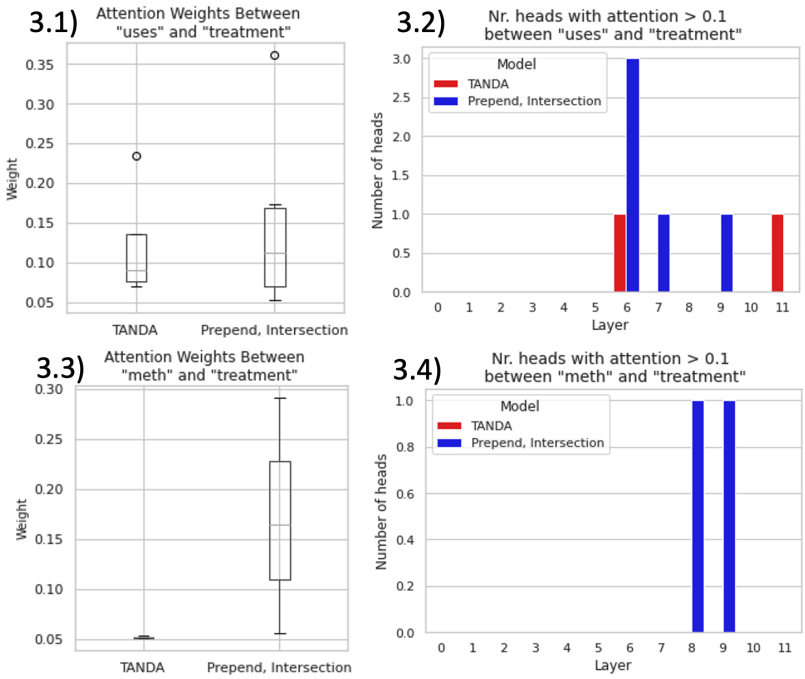

Figure 3: Attention comparison for the correct QA pair ***Q:*** *What some legal uses of meth?* ***A:*** *Although rarely prescribed, methamphetamine hydrochloride is approved by the U.S. Food and Drug Administration (FDA) for the treatment of attention deficit hyperactivity disorder and obesity under the trade name Desoxyn.*