Multi-Task Retrieval for Knowledge-Intensive Tasks

Jean Maillard* **Vladimir Karpukhin*** **Fabio Petroni**
Wen-tau Yih **Barlas Oğuz** **Veselin Stoyanov** **Gargi Ghosh**
Facebook AI
{jeanm,vladk,fabiopetroni,scottyih,barlaso,ves,gghosh}@fb.com

Abstract

Retrieving relevant contexts from a large corpus is a crucial step for tasks such as open-domain question answering and fact checking. Although neural retrieval outperforms traditional methods like tf-idf and BM25, its performance degrades considerably when applied to out-of-domain data. Driven by the question of whether a neural retrieval model can be *universal* and perform robustly on a wide variety of problems, we propose a multi-task trained model. Our approach not only surpasses previous methods in the few-shot setting, but also rivals specialised neural retrievers, even when in-domain training data is abundant. With the help of our retriever, we improve existing models for downstream tasks and closely match or improve the state of the art on multiple benchmarks.

1 Introduction

Knowledge-intensive tasks is the common designation for a class of real-world NLP problems which, because of their nature, require large amounts of knowledge about the world (Petroni et al., 2020). For example, *open-domain question answering* requires producing answers to general factoid questions; *fact checking* involves determining the veracity of claims based on a database of trusted evidence. Practical solutions to these tasks usually involve an efficient *retrieval* component that, given an input query, selects a limited subset of relevant information from a large knowledge source. Sophisticated downstream models then consider the input only in the context of the retrieved information, and perform the final task.[1]

The standard retrieval component in many systems (e.g., Thorne et al., 2018; Wang et al., 2018; Chen et al., 2017) has long relied on term-matching methods, such as tf-idf or BM25 (Robertson and Zaragoza, 2009). These methods rely on efficient algorithms and usually perform reasonably well regardless of the problem. In contrast, recent neural retrieval models, such as ICT (Lee et al., 2019), DPR (Karpukhin et al., 2020) and RAG (Lewis et al., 2020b) achieve better results by learning directly from task-specific training data and going beyond simple keyword matching. While task specialisation results in improved task performance, researchers have observed that a retriever trained for one specific domain will typically achieve low out-of-domain performance, and even lower performance on entirely different tasks (Petroni et al., 2020). This has two implications. First, unlike tf-idf or BM25, neural retrieval models are unsuitable for low data regimes such as few- and zero-shot settings. Second, task-specific retrievers complicate practical applications where multiple knowledge-intensive tasks may need to be performed using the same supporting database or over the same input text. It may not be practical to deploy multiple separate specialised models due to computational performance or memory concerns.

In this work, we ask the following question: can we develop a *universal* neural retriever? Namely, we target a retriever which can perform well on a wide variety of problems *without* domain-specific training, but which – if additional in-domain labelled data is available – can be further fine-tuned to improve its performance. We perform a large experimental study to attempt to build such a universal retrieval model. We find that, by jointly training on an extensive selection of retrieval tasks, we obtain a model which is not only more robust than previous approaches, but also can lead to better performance on the downstream knowledge-

* Equal Contribution.

[1] While large pre-trained neural models have been shown to incorporate real-world knowledge in their parameters and thus may skip retrieval (Petroni et al., 2019), they still have limited capacity and suffer from a lack of explainability.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 1098–1111
August 1–6, 2021.

intensive tasks when plugged into an existing system. Our approach combines the benefits from IR-based models with those of task-specific neural retrievers – namely, good performance when no (or not enough) training data is available and high task performance due to its ability to learn highly specialised representations.

Our contributions can be summarised as follows.

- We propose a single general-purpose "universal" retrieval model, able to perform comparably or better than specialised retriever approaches in both zero-shot (leave-one-out) and few-shot retrieval. We investigate several model variants, shedding light on what are the aspects of the architecture that affect its performance.
- We show that, with in-domain training, our model's gains in terms of retrieval directly translate into performance gains for a variety of downstream knowledge-intensive tasks.
- We will share the implementation as well as our best model. This is in the form of a readily available BERT checkpoint which, as we will show, can be used by NLP practitioners as a strong out-of-the-box retrieval system, and can also undergo further in-domain training for even higher performance.

2 Background

In this section, we first give an overview of retrieval methods based on sparse and dense representations. We then discuss a wide range of knowledge-intensive NLP tasks, where retrieval plays a crucial role in solving the problems.

2.1 Retrieval methods

Given a large collection of unstructured text passages, information retrieval (IR) can be broadly defined as finding a small set of passages that satisfies an information need, often presented in the form of a short text query (Manning et al., 2008). Traditional IR methods, such as tf-idf and BM25 (Robertson and Zaragoza, 2009), match keywords efficiently with an inverted index. Such methods can be seen as representing queries and passages in high-dimensional, *sparse* vectors, where each dimension corresponds to a term in the vocabulary and the weight indicates its importance.

In contrast to tf-idf and BM25, dense retrieval methods encode text as a latent semantic vector of a fixed, much smaller dimensionality. Whether a passage is relevant to a given query is determined by the distance of their vectors (Deerwester et al., 1990). Although dense representations do not encode tokens explicitly and can potentially map paraphrases of completely different tokens to close vectors, performance of early dense retrieval methods was often inferior to term-matching approaches, except when large labelled data is available (Yih et al., 2011; Gao et al., 2011; Huang et al., 2013). Thanks to success of large pre-trained models (Devlin et al., 2019; Liu et al., 2019b), however, recent dense retrieval methods have shown to outperform the sparse counterparts, when fine-tuned on a small set of in-domain labelled data (Karpukhin et al., 2020; Lewis et al., 2020b; Xiong et al., 2020). Efficient index and search of dense vectors are made possible by maximum inner product search (MIPS) algorithms (e.g., Shrivastava and Li, 2014; Guo et al., 2016), as well as tools like FAISS (Johnson et al., 2019).

Our work is built upon the Dense Passage Retriever (DPR) architecture of Karpukhin et al. (2020), which was initially proposed for the task of open-domain question answering. DPR is a neural bi-encoder model which embeds queries with an encoder $\boldsymbol{f}(\cdot)$ and passages with a separate encoder $\boldsymbol{g}(\cdot)$. Given an input query x and a target passage y, we have

$$\mathrm{p}(x \mid y) \propto \mathrm{sim}(x, y),$$

where the similarity score $\mathrm{sim}(x, y)$ is defined as the inner product of the embeddings of its arguments, $\boldsymbol{f}(x) \cdot \boldsymbol{g}(y)$. Given a query at inference time, calculating its similarity with every possible passage would be prohibitive for large knowledge sources. Therefore, DPR makes use of the FAISS library (Johnson et al., 2019) to perform fast approximate nearest neighbour search in sub-linear time.

Training of DPR is based on a contrastive loss. Given a query x, a relevant passage y, and a set of n irrelevant passages y_i^-, we train the model by optimising the following negative log likelihood:

$$\mathcal{L} = -\log \frac{\exp(\mathrm{sim}(x, y))}{\exp(\mathrm{sim}(x, y)) + \sum_{i=1}^{n} \exp(\mathrm{sim}\left(x, y_i^-\right))}.$$

As the set of irrelevant passages, we use the relevant passages for other queries within the same batch, as well as a specially selected "hard" confounder. This is a passage which has high lexical

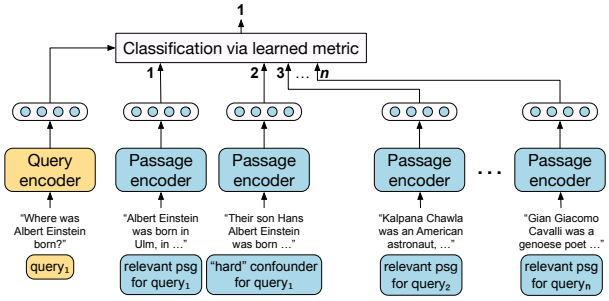


Figure 1: Training of DPR (Karpukhin et al., 2020), a bi-encoder model for open-domain question answering. Queries and passages are encoded as vectors, and retrieval is performed as a maximum inner product search.

overlap with the query (high BM25 score), but is not among the set of relevant passages for the given data point. Karpukhin et al. (2020) have shown that the inclusion of such "hard" confounders leads to substantially improved training results. This training process is illustrated in Figure 1.

2.2 Knowledge-intensive Tasks

For the training and evaluation of all models in the paper we make use of KILT, a benchmark and library of datasets (Petroni et al., 2020). KILT consists of a selection of datasets spanning five varied classes of knowledge-intensive tasks (i.e., question answering, slot filling, fact checking, dialogue, entity linking), with the aim to cover many different ways of seeking knowledge. Input queries can vary wildly from one task to the other, and include classic examples of open-domain retrieval tasks such as natural language questions and claims to be verified, as well as more unusual examples like conversation fragments and long chunks of annotated text. Crucially, all datasets distributed in KILT have been re-aligned such that they are all grounded in the same snapshot of Wikipedia, which the authors distribute. The knowledge required to answer any of the queries in the library of tasks can thus be found within the same unified knowledge source.

To illustrate the variety of ways in which the input queries for different tasks can be formulated, we provide a few simple examples in Table 1. In spite of the differences between query formulations, all these tasks share one crucial aspect: they all require a retriever to fetch the relevant passages from the knowledge source, in order to support the final downstream task.

3 Methods

3.1 Universal retrieval

Using task-specific models to tackle our collection of retrieval tasks would involve completely separate models, one per dataset. Following the definitions of §2.1, for a family of tasks $i = 1, \ldots, n$ this would require n query encoders $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_n$, and n corresponding passage encoders $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_n$. As illustrated in Figure 2, this would lead to a proliferation of models and data, down to separate indexed copies of the knowledge source itself. This fully specialised setup will form one of our baselines.

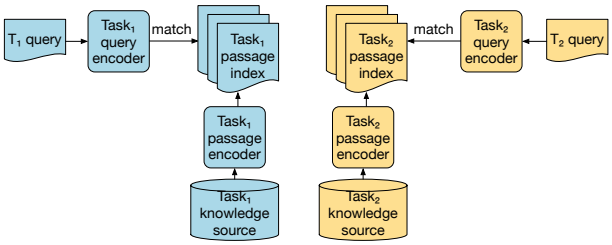
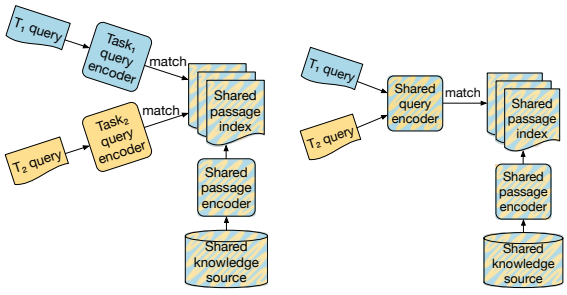


Figure 2: Two retrieval tasks T_1 and T_2 performed by two fully-specialised models.

Multi-task training has been successfully used to allow models to leverage cross-task data, as well as to provide a regularisation effect leading to better generalisation ability (Liu et al., 2019a). We apply this concept to neural retrievers, with the aim of improving performance by jointly leveraging multiple different retrieval datasets.



(a) Separate query encoders. (b) A single retrieval model.

Figure 3: Parameter sharing between neural retrievers.

Our base setup is illustrated in Figure 3b and involves using, across all tasks, a shared passage encoder $\boldsymbol{g}$ — so that a single index of encoded passages can be used — as well as a shared query encoder $\boldsymbol{f}$. In essence, in this setup a single DPR model is used to perform all retrieval tasks.

Task	Example query	Answer	Relevant doc.
Question Answering	Who is playing the Halftime Show at Super Bowl 2016?	Coldplay	The Super Bowl 50 Halftime Show took place on February 7, 2016 ... It was headlined by the British rock group Coldplay.
Fact Checking	Bermuda Triangle is in the western part of the Himalayas	REFUTES	The Bermuda Triangle ... is a loosely defined region in the western part of the North Atlantic Ocean
Slot Filling	Piner Creek [sep] mouth of the watercourse	Santa Rosa Creek	Piner Creek discharges to Santa Rosa Creek which in turn ...
Entity Linking	Leicestershire take over at top after innings victory. London. [start_ent] West Indian [end_ent] all-rounder Phil Simmons ...	West Indies cricket team	The West Indies cricket team is a multinational men's cricket team representing the Anglophone Caribbean region
Dialogue	I am a big fan of Star Trek [sep] I don't know much about it. When did the first episode air? [sep] It debuted in .. [sep] What is the plot of the show?	William Shatner plays the role of Captain Kirk	It followed the interstellar adventures of Captain James T. Kirk (William Shatner) and his crew ...

Table 1: Illustrative examples of some of the tasks within KILT, and how varied their query formulations can be.

3.2 Model variants

Due to the complexity of training and evaluating retrievers (which involves training the model, embedding all of Wikipedia, and indexing it), our main experiments are all based on the configuration of Figure 3b, which was found to work well.

We did, however, also investigate other more complex model variants in a set of preliminary experiments. As these were not found to be beneficial, we leave them in the appendix, but mention the variants' architecture for completeness:

- **Task-specific query encoder**. A different query encoder $\boldsymbol{f}_i$ is used for each family of tasks. For example, all question answering tasks use the same query encoder. This is meant to allow for potentially different needs in processing queries, given the fundamentally diverse nature of the tasks at hand. This setup configuration is illustrated in Figure 3a.
- **Task markers**. This is a variant of the base model in which specialised tokens are inserted at the beginning of each query. Their aim is to help the model distinguish between the different tasks, by marking them. We use one task marker for each of the five task classes of KILT, such that all question answering tasks share the same marker.

Experimental results comparing these variants to the base model can be found in Appendix B.

Dataset	Task class	#Train
FEVER	Fact Checking	71 k
AIDA-YAGO2	Entity Linking	18 k
T-REx	Slot Filling	2,284 k
Zero Shot RE	Slot Filling	132 k
Natural Questions	QA	77 k
HotpotQA	QA	69 k
TriviaQA	QA	53 k
Wizard of Wikipedia	Dialogue	64 k

Table 2: KILT datasets used in this work, and the size of our converted training sets for each.

4 Experiments

4.1 Experimental settings

Dataset selection For our experiments we select the eight KILT datasets listed in Table 2, which cover all five task classes and include a training split, a validation split and a held-out test split each.

Preprocessing Starting from the KILT data, we split each Wikipedia article into disjoint 100-token chunks which form our basic retrieval units, following Wang et al. (2019) and Karpukhin et al. (2020). To maintain the same language introduced in §3, we will simply call these chunks *passages*.

This preprocessing results in a knowledge source of 36 million passages. In order to harmonise all datasets to the same knowledge source, KILT used a mapping strategy based on the BLEU metric to

map relevant passages in the original versions of its datasets to passages in its own shared knowledge source (Petroni et al., 2020). Entries included in the KILT training sets which have a mapping BLEU score below 0.5 are likely to be noise, and we exclude them from training and validation (resulting in a 18% reduction on average for the validation sets).

Multi-tasking Training is performed on the union of all data. Since two training sets are vastly larger, we downsample them to the same order of magnitude as the others. Preliminary experiments with more complex sampling methods, like resampling all datasets so that each epoch would see an equal number of samples from each, found that they had no measurable effect compared to this simpler approach.

Encoders Our query and passage encoders are initialised as distinct BERT-base uncased encoders (Devlin et al., 2019), trained separately. As pooling mechanism we find it effective to simply take the `[CLS]` token representation at the topmost layer.

Training We train our models for up to 80 epochs. To select the best checkpoint, we evaluate the retrieval performance on the validation set at regular intervals. We optimise with Adam (Kingma and Ba, 2015) with a learning rate of $2 \cdot 10^{-5}$, warmup, a linear decay schedule, and a dropout rate of 0.1. The batch size is set to 128 samples, and in preliminary experiments we found no benefit in increasing this further. We use an additional "hard" confounder per batch, selected based on BM25 score as in Karpukhin et al. (2020).

Downstream evaluation When evaluating our retriever within a larger architecture to perform a knowledge-intensive task, we replicate a setup analogous to DPR + BART of Petroni et al. (2020). This uses our multi-task model to retrieve and prepend the top 3 passages to the query, which is then processed by a task-specific fine-tuned BART model to generate the final answer for the end task.

Baselines For our retrieval experiments, we include as baselines a BM25 model as well as a task-specific DPR model for each of the training datasets. For the downstream evaluations, we compare against three strong representative models trained by Petroni et al. (2020): a task-specific DPR model combined with BART (Lewis et al., 2020a), RAG (Lewis et al., 2020b), and T5 (Raffel et al., 2020).

4.2 Universal retrieval

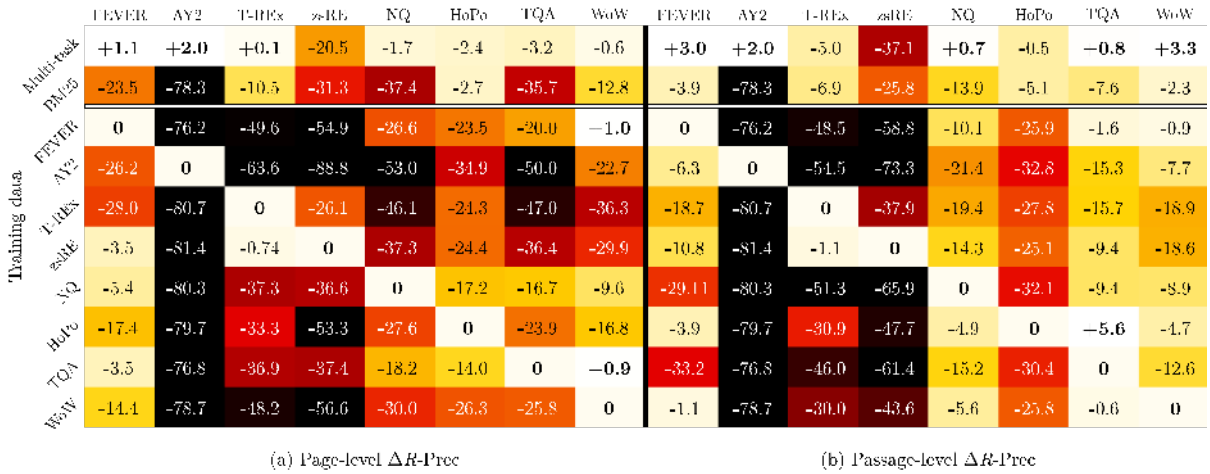
The results of the evaluations reported in (Petroni et al., 2020) show that retrievers trained for question answering have poor performance outside of their domain. We would like to understand if it is possible to design a single model which can accurately satisfy the information needs of a wide variety of knowledge-intensive tasks. In short: *Can a neural retriever be universal?*

We perform a comprehensive evaluation of several models on the 8 tasks of Table 2. We evaluate 8 task-specific models (one trained on each of the 8 datasets), for which we measure both in-domain and out-of-domain performance, and a BM25 baseline. Additionally, we include a multi-task trained model – as described in §3.1 – with the hope that it can learn to perform all tasks satisfyingly. This amounts to 10 models evaluated on 8 tasks each, for a total of 80 evaluations. To measure retrieval performance, we adopt the main metric used for the KILT benchmark, R-precision. This is calculated as r/R, where R is the total number of relevant passages for a given query, and r is the number of relevant passages returned among the top-R retrieval results. For the case of $R = 1$ this is therefore equivalent to precision@1.

This experiment is of a very large scale, amounting to 10 models evaluated on 8 tasks, each repeated at the page and passage level – for a total of 160 figures to report. Due to this complexity, we report the results in Table 3 via a heatmap showing, for each evaluation task, the difference in R-precision between a given model and the task-specific model that was trained on the relevant task only. This is to highlight how each approach stacks up against a specialised model.

While the KILT evaluation focuses on retrieval at the level of Wikipedia pages (thereby marking as "hits" any results that lie within the correct page), we are also interested in performing an evaluation at a more fine-grained level. We therefore also evaluate our models at the passage level, using a modified version of the official KILT evaluation scripts. These are shown at the right side of Table 3. For full context, we also provide the full absolute results in Appendix A.

We straight away notice that task-specific models tend to achieve high performance on their respective tasks, often taking one of the top two spots. Interestingly, we also note that these neural retrievers consistently outperform the BM25 baseline, show-



(a) Page-level ΔR-Prec (b) Passage-level ΔR-Prec

Table 3: Difference in retrieval R-precision (at page- and passage-level) with respect to a task-specific model, on KILT validation data. The rows show our proposed multi-task retriever, the BM25 baseline, and a series of task-specific models trained on each of the tasks. For the AIDA-YAGO2 dataset, due to the nature of the task, page- and passage-level results coincide.

ing that the result achieved by Karpukhin et al. (2020) for open-domain question answering also holds for other knowledge-intensive tasks.

The results reveal a strong performance for the multi-task model, confirming the hypothesis that a single model can be trained to perform well on a wide variety of retrieval tasks. With the exception of one dataset, the multi-task model achieves the best retrieval performance or is within a few points of the top score. We note that the one exception, the Zero-shot RE task (Levy et al., 2017), is a trivial task in which the query will always contain the title of the page to be retrieved. Indeed, the model specific to this task achieves a near-perfect score (see full results in Appendix A).

Another task which stands out for being markedly different in formulation is AIDA-YAGO 2 (Hoffart et al., 2011). As shown in Table 3, models that were not trained on AIDA-YAGO 2 do very poorly on it. Entity linking is normally better performed by models which are explicitly designed for it (De Cao et al., 2020). We nevertheless include it to showcase the ability of neural retrievers to adapt to a variety of tasks, and note how well the multi-task retriever performs on it in spite of its unusual nature.

4.3 Downstream performance

We saw that our proposed approach achieves strong performance across a variety of retrieval tasks. However, our interest in neural retrievers stems from their use as components within larger systems, to perform tasks such as question answering. Our next experimental question is therefore: *Can a universal retriever lead to better downstream performance in knowledge-intensive tasks?*

We perform a downstream evaluation of our approach used in conjunction with BART (Lewis et al., 2020a) as the generative component, adopting a setup identical to that of Petroni et al. (2020). The results are reported in Table 4, with bold and underline marking the best and second best scores respectively.

The *DPR + BART* line refers to a setup similar to ours, but with the simpler retriever of Karpukhin et al. (2020) as trained in Petroni et al. (2020), which lacked the multi-task aspect. Therefore, comparing to its performance gives us a clear indication of the contribution of multi-task training on the overall performance on knowledge-intensive tasks.

Our proposed model achieves significantly better performance than this baseline in AY2, zsRE and HoPo; while for the other tasks, the discrepancy is always below two points.

This fact is reflected in the last column, showing that on average multi-task training leads to better downstream performance. The model also compares favourably to RAG (Lewis et al., 2020b), a more advanced system in which the query encoder is fine-tuned on the end task.

[2]Performing this evaluation required retrieving relevant documents for all training sets. Due to the very large size of T-REx, this particular dataset could not be included in this section.

Model	Fact Check. **FEV**	Ent. L. **AY2**	Slot Fill. **zsRE**	Open Domain QA **NQ**	**HoPo**	**TQA**	Dial. **WoW**	**Avg.**
Multi-task + BART	86.32	**82.61**	**57.95**	39.75	**31.77**	59.60	15.12	**53.30**
DPR + BART	**86.74**	75.49	30.43	41.27	25.18	58.55	**15.19**	47.55
RAG	86.31	72.62	44.74	**44.39**	26.97	**71.27**	13.11	51.34
T5	76.30	74.05	9.02	19.60	12.64	18.11	13.53	31.89

Table 4: KILT *test* scores on the downstream evaluation. Results in the bottom section are as reported in Petroni et al. (2020). The score metrics are accuracy for fact checking, entity linking and slot filling; exact match for QA; and F1 score for dialogue.[2]

4.4 Zero- and few-shot performance

Task-specific neural retrievers can achieve higher performance than IR-based methods, but they are not suitable for cases where no training data (or not enough) is available. In those cases, tf-idf and BM25 are the better choice. To evaluate the performance of a multi-task retriever as a suitable replacement for them in this scenario, we run a series of experiments in the low data regimes (few-shot and zero-shot).

We start by training a set of multi-task retrievers (with the base setup) in the leave-one-out setting for each dataset, in order to see how a neural retriever will perform when trained on all domains except for the one it is to be evaluated on.

The results of these zero-shot experiments are reported in the second line of Table 5 (again, bold and underline indicate best and second best overall performance, respectively). They show that, even in the zero-shot setting, the multi-task neural retriever achieves performance that is competitive to BM25, with retrieval being 10 points higher at the page level and 5 points lower at the passage level on average.

The advantage of neural retrievers over BM25 lies in their ability to improve with training. We therefore look at few-shot training for each task, and create two smaller copies for each of the original training sets with a random sample of 128 and 1,024 examples respectively. In order to evaluate the suitability of a multi-task trained retriever as a starting checkpoint for few-shot training, we take the various leave-one-out models and fine-tune them on our few-shot training sets. To check whether multi-task pre-training is effective, we also compare these to DPR models, which are just initialised with BERT weights and then fine-tuned on the same data.

The bottom two sections of Table 5 report the results. The most dramatic gains from fine-tuning are seen for AY2, an "outlier" task whose formulation differs from that of the other tasks, and which seems to benefit the most from being trained on in-domain data. The zsRE performance does not seem to improve from fine-tuning on the smaller dataset, but sees a very big jump when switching to the larger dataset. As a reminder, in this trivial task the title of the page to be retrieved always appears at the start of the query. It is therefore not surprising that models specifically fine-tuned on it can achieve near-perfect scores, as long as enough training data is provided.

In spite of the fine-tuning, we note that both DPR and the multi-task model fail to improve on their performance for T-REx, suggesting that large amounts of training data are required to learn this task. Nevertheless, the multi-task model proves itself more robust, and achieves the top performance on it.

Finally, we note for 2 out of 8 tasks, namely zsRE and WoW, DPR achieves lower page-level retrieval scores than the multi-task model, but performs better at the passage level. This shows that fine-grained and coarse-grained retrieval performance are not always perfectly correlated.

Overall, the experiments show strong results for the multi-task model, with the average zero-shot performance being competitive to BM25, and the average few-shot performance being markedly better than the alternatives. The discrepancy in performance between a vanilla DPR model and the leave-one-out multi-task model is especially noticeable when using the smaller of the two datasets, in which case average performance for the latter is more than double that of vanilla DPR.

Model	FEV	AY2	T-REx	zsRE	NQ	HoPo	TQA	WoW	Avg.
BM25	50.13 / 40.06	3.47	58.60 / **51.64**	66.43 / 52.98	25.83 / 14.20	**43.95** / **38.38**	29.44 / 16.16	27.50 / 18.41	38.17 / 33.12
Leave-one-out multi-task models									
Zero-shot	74.11 / 37.09	4.16	**67.54** / 44.84	73.42 / 32.65	47.23 / **21.50**	34.72 / 16.52	49.08 / **28.06**	36.92 / 16.19	48.40 / 28.12
Finetune (128)	**75.95** / 32.75	32.38	**67.54** / 44.84	73.41 / 32.65	47.48 / 14.98	34.72 / 27.82	54.71 / 19.82	**48.36** / 17.46	54.23 / 27.19
Finetune (1k)	73.08 / 40.83	70.40	**67.54** / 44.84	**93.04** / 58.67	**51.00** / 19.90	39.19 / 35.43	**59.08** / 20.22	47.65 / 19.75	**62.62** / **34.23**
Vanilla DPR models									
Finetune (128)	37.99 / 25.31	26.23	0.20 / 0.02	0.16 / 0.00	20.92 / 9.52	14.46 / 14.08	26.85 / 10.54	30.31 / 17.20	19.64 / 10.95
Finetune (1k)	70.87 / **47.82**	**72.49**	0.20 / 0.02	90.33 / **80.20**	43.43 / 19.81	30.75 / 30.50	52.50 / 17.33	44.70 / **24.92**	50.66 / 31.51

Table 5: Page- and passage-level R-Precision in the zero-shot setting and with additional fine-tuning of 128 and 1,024 examples. We also compare to a BM25 retriever and a DPR model initialised with BERT weights.

5 Related work

The approach most closely related to ours is DPR (Karpukhin et al., 2020), upon which we built all our retrievers. It is covered in detail, along with historical context, in § 2.1. Another closely related approach is the Retrieval-Augmented Generation (RAG) model of Lewis et al. (2020b). In its base configuration it augments DPR with a generative reader, and trains the query encoder end-to-end (differing from traditional retriever-reader architectures, which treat the two steps as disjoint). A natural extension of our work would be to combine RAG with the multi-task learning approach, to study whether it can lead to further gains in performance or robustness.

A number of promising techniques to boost retrieval performance have been proposed recently. These are orthogonal to our work, and as such they could be combined with it. Amongst these, pre-training methods form one class. Inverse Cloze Task (Lee et al., 2019) and its extensions (Chang et al., 2020) are self-supervised pre-training methods designed for retrieval in open-domain question answering. Whether such specific pre-training is beneficial to tasks other than question answering remains an open question. CERT (Fang et al., 2020) is an alternative pre-training approach, inspired by some recent advances in computer vision. While to our knowledge this has not been applied to retrieval problems, we believe it might be promising due to its focus on sentence-level semantics (as opposed to the more standard masked language modelling pre-training, which focuses on the token level).

Another class of orthogonal improvements to dense retrieval involves models which embed passages into multiple fixed-size vectors. Of these, ColBERT (Khattab and Zaharia, 2020) and ME-BERT (Luan et al., 2020) are two representative examples. One further approach is ColBERT-QA (Khattab et al., 2020), which additionally uses a data augmentation strategy closely related to our own approach described in Appendix D.

Retrieval does not strictly have to be performed with a model which contains an explicit memory. Large-scale pre-trained models have been shown to store knowledge directly into their parameters. A model which demonstrates this ability is T5 (Raffel et al., 2020) – which we used as a baseline in § 4.

Regarding the multi-task aspect of our approach, a related strategy has been demonstrated by Aghajanyan et al. (2021). In this recent work, the authors multi-task train a pre-trained model on around 50 datasets, before performing the final fine-tuning. While they do not focus on retrieval, their results are consistent with ours and show that multi-task training leads to improved performance and increased sample efficiency.

On the topic of question answering, Lewis et al. (2021) show in a recent notable paper that, for several popular QA datasets, a portion of questions in the test set has near-duplicates in the training sets, and the same holds true for an even larger set of answers. To our knowledge, similar analyses have yet to be performed on the other KILT tasks.

Finally two entity linkers, GENRE (De Cao et al., 2020) and BLINK (Wu et al., 2020), are worth mentioning. Being trained specifically for entity linking, these models will generally outperform retrieval-based approaches on that task. While they are not comparable to retrieval models and will not generally be applicable to information retrieval tasks, we cite them here to provide readers with a fuller context of the existing literature on related tasks.

6 Conclusions

We have conducted a large-scale experimental study on knowledge-intensive tasks, and how re-

trieval models that tackle them seek the required information from knowledge bases like Wikipedia.

The study started with the question of whether the way in which information is embedded for retrieval purposes is universal. §4.2 provided evidence that to a large extent it is, with a single "universal" retriever, trained jointly on 8 datasets, often performing comparably to task-specific models.

Armed with this knowledge, in §4.3 we plugged our single model in a larger pipeline, in order to see its contribution to the downstream performance on a wide range of knowledge-intensive tasks. This led to an overall improvement in downstream performance, setting new top results for a number of tasks in the KILT benchmark.

Next, in §4.4, we evaluated the model's performance in the zero-shot and few-shot settings. By evaluating on a wide range of tasks, we were able to show that our proposed approach performs comparably to BM25 in the zero shot setting, and quickly overtakes it even with minimal in-domain training.

In the appendices, readers interested in getting a fuller picture will find further experiments. Namely, in Appendix B we test two more complex variants of the model involving task specialisation, but fail to see clear performance improvements. In Appendix D we show how a simple iterative approach to data augmentation, easily applied to our base approach, can lead to better performance throughout.

We provide a pre-trained snapshot of our best-performing model, in the form of a BERT checkpoint.[3] As shown, this model will be useful in zero-shot and few-shot settings as a better performing alternative to both IR-based approaches such as BM25, as well as task-specific models. The multi-task training approach demonstrated here can also be useful in industry settings where several retrieval operations may need to be performed on the same piece of content,[4] and the deployment of multiple task-specific models might not be possible due to space or computational performance concerns.

References

Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. 2021. Muppet: Massive multi-task representations with pre-finetuning. *ArXiv*, abs/2101.11038.

Wei-Cheng Chang, Felix X. Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2020. Pre-training tasks for embedding-based large-scale retrieval. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2020. Autoregressive entity retrieval. *ArXiv*, abs/2010.00904.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Hongchao Fang, Sicheng Wang, Meng Zhou, Jiayuan Ding, and Pengtao Xie. 2020. CERT: Contrastive self-supervised learning for language understanding. *ArXiv*, abs/2005.12766.

Jianfeng Gao, Kristina Toutanova, and Wen-tau Yih. 2011. Clickthrough-based latent semantic models for web search. In *Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2011, Beijing, China, July 25-29, 2011*, pages 675–684. ACM.

Ruiqi Guo, Sanjiv Kumar, Krzysztof Choromanski, and David Simcha. 2016. Quantization based fast inner product search. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016, Cadiz, Spain, May 9-11, 2016*, volume 51 of *JMLR Workshop and Conference Proceedings*, pages 482–490. JMLR.org.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 782–792, Edinburgh, Scotland, UK. Association for Computational Linguistics.

[3] https://github.com/facebookresearch/DPR/

[4] E.g., fact checking and hate speech detection.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry P. Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013*, pages 2333–2338. ACM.

J. Johnson, M. Douze, and H. Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Omar Khattab, Christopher Potts, and Matei Zaharia. 2020. Relevance-guided supervision for OpenQA with ColBERT. *ArXiv*, abs/2007.00814.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 39–48. ACM.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, Vancouver, Canada. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. 2021. Question and answer test-train overlap in open-domain question answering datasets. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1000–1008, Online. Association for Computational Linguistics.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496, Florence, Italy. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2020. Sparse, dense, and attentional representations for text retrieval. *ArXiv*, abs/2005.00181.

Christopher D Manning, Hinrich Schütze, and Prabhakar Raghavan. 2008. *Introduction to information retrieval*. Cambridge university press.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2020. KILT: a benchmark for knowledge intensive language tasks. In *arXiv:2009.02252*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

Anshumali Shrivastava and Ping Li. 2014. Asymmetric LSH (ALSH) for sublinear time maximum inner product search (MIPS). In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2321–2329.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Shuohang Wang, Mo Yu, Xiaoxiao Guo, Z. Wang, Tim Klinger, Wei Zhang, S. Chang, G. Tesauro, Bowen Zhou, and Jing Jiang. 2018. R3: Reinforced ranker-reader for open-domain question answering. In *AAAI*.

Zhiguo Wang, Patrick Ng, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. 2019. Multi-passage BERT: A globally normalized BERT model for open-domain question answering. 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)*, pages 5878–5882, Hong Kong, China. Association for Computational Linguistics.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Scalable zero-shot entity linking with dense entity retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6397–6407, Online. Association for Computational Linguistics.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *ArXiv*, abs/2007.00808.

Wen-tau Yih, Kristina Toutanova, John C. Platt, and Christopher Meek. 2011. Learning discriminative projections for text similarity measures. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 247–256, Portland, Oregon, USA. Association for Computational Linguistics.

A Full retrieval results

The heatmap in Table 3 showed a full comparison of task-specific models to our multi-task model and the BM25 for the experiments of § 4.2. In order to aid in the interpretation of a very large set of results, the heatmap showed, for each task, the difference in R-precision to the respective task-specific model. Here, for full context, we also provide in Table 6 the full set of absolute R-precisions for the experiments of § 4.2.

B Model variants

We compare our base multi-task model with the two variants described in § 3.2. Due to the high memory consumption of the "task-specific encoders" variant (requiring one full query encoder per task family, in addition to the passage encoder), it was only possible to perform these evaluations in a restricted setting of three datasets. The results in Table 7 do not reveal a clear winner, suggesting that the base architecture might be the better choice due to its simplicity and generally good performance. Not included in this table and in any other experiments, due to very poor performance in preliminary evaluations, are two further variants: a base model with a single encoder for both queries and passages, and a base model trained from scratch without BERT pre-training.

C Task learning curve

One of the initial studies we conducted involved computing the learning curve of the multi-task model for each task, using the full validation metrics. This is particularly expensive, as it involves embedding the whole of Wikipedia for each evaluation, indexing it, and performing a full retrieval. Figure 4 shows this for one of our preliminary models, trained on six tasks (excluding the abnormally large T-REx and the outlier AY2). We note the unstable behaviour of zsRE, whose unusual nature was already remarked upon in §4.2.

D Adversarial confounder selection

We saw in § 2.1 how "hard" confounder passages are collected using a BM25 baseline, following the standard approach in DPR. However, any other retriever can be used to select such confounders, including the very retriever being trained, leading to an iterative, self-adversarial training. Concretely, this amounts to following steps: (1) a first version of the retriever is trained with BM25 confounders; (2) new confounders are selected with the trained model, by retrieving high-ranking passages which are not among the set of relevant ones; (3) a second version of the model is trained using the additional new confounders.

Intuitively, it is expected that this approach should lead to higher quality confounders compared to those selected by BM25 based on simple keyword matching. Based on our own experience as well as relevant literature (Khattab et al., 2020), this adversarial approach has been shown to work well for question answering.

As a way of further pushing the performance of the model, we experiment with this adversarial confounder selection on two datasets, Natural Questions (Kwiatkowski et al., 2019) and TriviaQA (Joshi et al., 2017). We selected these two datasets since, out of all of the tasks we are considering, they have an easy way of checking whether a certain passage is relevant or not for a given query – namely, by checking whether the answer is present in the passage. This enabled us to automatically build sets of confounders, ensuring relevant passages would be excluded.[5]

The performance of this approach is reported in Table 8, showing an overall improvement across multiple tasks. While this approach is demonstrated here on our multi-task model, it is in fact orthogonal to it, and could be applied to any other neural retrievers trained with a contrastive loss.

[5] Strictly speaking, assuming a passage to be irrelevant because of the absence of the answer span is not formally correct. However, experiments show a good correlation between this simple check and the overall model quality.

model	Fact Check. **FEV**	Ent. L. **AY2**	Slot Filling **T-REx**	**zsRE**	Open Domain QA **NQ**	**HoPo**	**TQA**	Dial. **WoW**
Multi-task	**74.72** / **46.96**	**83.78**	**69.18** / 53.54	77.23 / 41.70	61.51 / **28.80**	44.21 / 38.42	61.95 / 24.56	39.70 / **24.07**
BM25	50.13 / 40.06	3.47	58.60 / 51.64	66.43 / 52.98	25.83 / 14.20	43.95 / 38.38	29.44 / 16.16	27.50 / 18.41
Task-specific models								
FEVER	73.60 / 43.92	5.62	19.50 / 10.02	42.88 / 19.98	36.69 / 18.05	23.18 / 17.59	45.08 / 22.24	**41.27** / 19.85
AY2	47.36 / 37.58	81.77	5.52 / 4.08	8.94 / 5.50	10.22 / 6.77	11.69 / 10.71	15.11 / 8.47	17.59 / 13.08
T-REx	45.63 / 25.22	1.05	69.08 / **58.54**	71.64 / 40.95	17.10 / 8.71	22.31 / 15.63	18.10 / 8.06	4.02 / 1.83
zsRE	70.10 / 33.12	0.42	68.34 / 57.40	**97.74** / **78.81**	25.98 / 13.81	22.23 / 18.35	28.68 / 14.44	10.40 / 2.09
NQ	68.16 / 14.81	1.44	31.78 / 7.20	61.12 / 12.92	**63.24** / 28.13	29.39 / 11.33	48.39 / 14.42	30.77 / 11.81
HoPo	56.18 / 40.03	2.07	35.76 / 27.62	44.44 / 31.15	35.60 / 23.26	**46.63** / **43.47**	41.18 / **29.37**	23.51 / 16.02
TQA	70.06 / 10.68	4.95	32.22 / 12.52	60.37 / 17.43	45.01 / 12.97	32.62 / 13.05	**65.12** / 23.79	41.17 / 8.11
WoW	59.16 / 42.79	3.11	20.92 / 18.52	41.14 / 35.26	33.27 / 22.52	20.36 / 17.66	39.37 / 23.15	40.32 / 20.73

Table 6: Page- and passage-level R-precision on KILT validation data. For the AIDA-YAGO 2 dataset, due to the nature of the task, only page-level retrieval is defined.

variant	**FEV**	**NQ**	**TQA**
Base	**76.38** / 40.76	60.91 / 24.50	**64.77** / **21.75**
Task markers	75.84 / **40.79**	**62.31** / 25.10	64.04 / 20.86
Task-spec. enc.	73.53 / 40.02	61.05 / **25.52**	64.17 / 21.23

Table 7: Multi-task model variants evaluated on a subset of tasks (R-precision on validation data at page/passage level).

confounders	Fact Check. **FEV**	Ent. L. **AY2**	Slot Filling **T-REx**	**zsRE**	Open Domain QA **NQ**	**HoPo**	**TQA**	Dial. **WoW**
BM25	74.72 / 46.96	83.78	69.18 / 53.54	77.23 / 41.70	**61.51** / 28.80	**44.21** / 38.42	**61.95** / 24.56	39.70 / 24.07
BM25 + adv	**74.79** / **52.12**	**84.86**	**71.36** / **61.40**	**80.04** / **54.08**	59.25 / **40.11**	44.08 / **41.04**	59.19 / **34.17**	**41.04** / **24.62**

Table 8: Comparison of two confounder selection methods for the multi-task model: simple BM25, and BM25 augmented with adversarial confounders (R-precision on validation data at page/passage level).

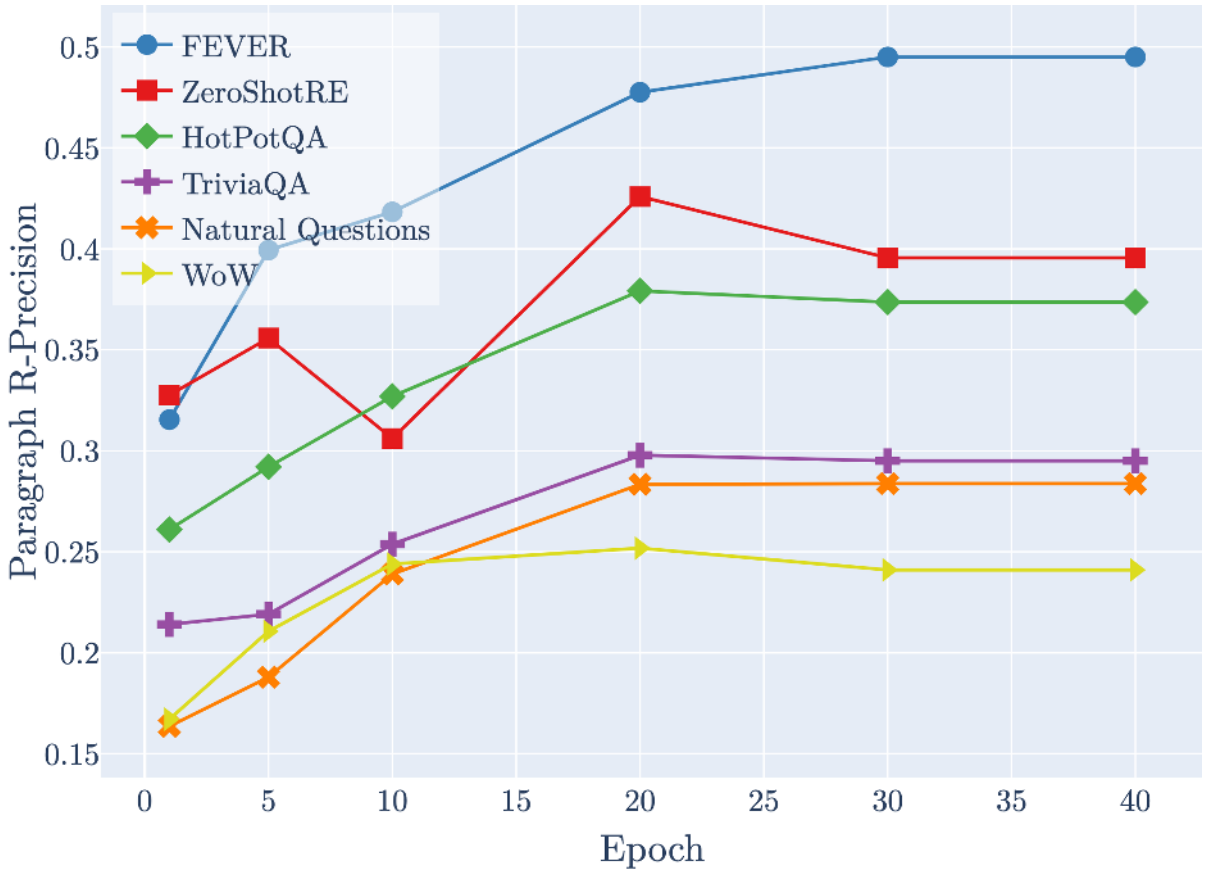


Figure 4: Retrieval R-precision versus training epoch for a multi-task model, on KILT validation data.