KILT: a Benchmark for Knowledge Intensive Language Tasks

Fabio Petroni[1] **Aleksandra Piktus**[1] **Angela Fan**[1,3] **Patrick Lewis**[1,2]
Majid Yazdani[1] **Nicola De Cao**[6] **James Thorne**[4] **Yacine Jernite**[5] **Vladimir Karpukhin**[1]
Jean Maillard[1] **Vassilis Plachouras**[1] **Tim Rocktäschel**[1,2] **Sebastian Riedel**[1,2]
[1]Facebook AI Research [2]University College London [3]LORIA
[4]University of Cambridge [5]HuggingFace [6]University of Amsterdam

Abstract

Challenging problems such as open-domain question answering, fact checking, slot filling and entity linking require access to large, external knowledge sources. While some models do well on individual tasks, developing general models is difficult as each task might require computationally expensive indexing of custom knowledge sources, in addition to dedicated infrastructure. To catalyze research on models that condition on specific information in large textual resources, we present a benchmark for knowledge-intensive language tasks (KILT). All tasks in KILT are grounded in the same snapshot of Wikipedia, reducing engineering turnaround through the reuse of components, as well as accelerating research into task-agnostic memory architectures. We test both task-specific and general baselines, evaluating downstream performance in addition to the ability of the models to provide provenance. We find that a shared dense vector index coupled with a seq2seq model is a strong baseline, outperforming more tailor-made approaches for fact checking, open-domain question answering and dialogue, and yielding competitive results on entity linking and slot filling, by generating disambiguated text. KILT data and code are available at https://github.com/facebookresearch/KILT.[1]

1 Introduction

There has been substantial progress on natural language processing tasks where the inputs are short textual contexts such as a sentences, paragraphs, or perhaps a handful of documents. Critically, we have seen the emergence of general-purpose architectures and pre-trained models that can be applied to a wide range of such tasks (Devlin et al., 2019). However, for many real world problems, processing at this local level is insufficient. For example, in open-domain question answering (Chen et al., 2017) models need to find answers within a large corpus of text. Fact checking a claim (Thorne et al., 2018a) requires models to find evidence, often on the web. In knowledgeable open dialogue (Dinan et al., 2019), models need access to knowledge from large corpora to sustain informed conversations.

In general, solving *knowledge-intensive* tasks requires–even for humans–access to a large body of information. Like in Information Retrieval (IR) this involves satisfying an information need leveraging large collections of text (Manning et al., 2008). However, while IR focuses of finding relevant material (usually documents), the tasks we consider focus on more fine-grained behavior, such as producing specific answers to queries. For such knowledge-intensive tasks, general infrastructure and architectures across tasks have yet to emerge, and fundamental research questions remain open. For example, while it was long assumed that non-parametric and explicit memory accessed through retrieval is strictly required for competitive results (Chen et al., 2017), recent large pre-trained sequence-to-sequence models such as T5 (Raffel et al., 2019a) and BART (Lewis et al., 2019) store all knowledge in their parameters while performing remarkably well (Petroni et al., 2019). Likewise, while the classical approach of information extraction for populating a Knowledge Base (KB, Riedel et al., 2013; Surdeanu and Ji, 2014) seems out-of-fashion, recent results show that they remain contenders (Fan et al., 2019a; Xiong et al., 2019).

While there are numerous datasets for knowledge-intensive tasks (e.g. Thorne et al., 2018a; Dinan et al., 2019; Kwiatkowski et al., 2019, to name just a few), it is difficult to answer the above questions generally across them. Each dataset comes in a different format, is pre-processed with different assumptions, and requires different loaders, evaluations, and analysis

[1]and at https://huggingface.co/datasets?search=kilt

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 2523–2544
June 6–11, 2021.

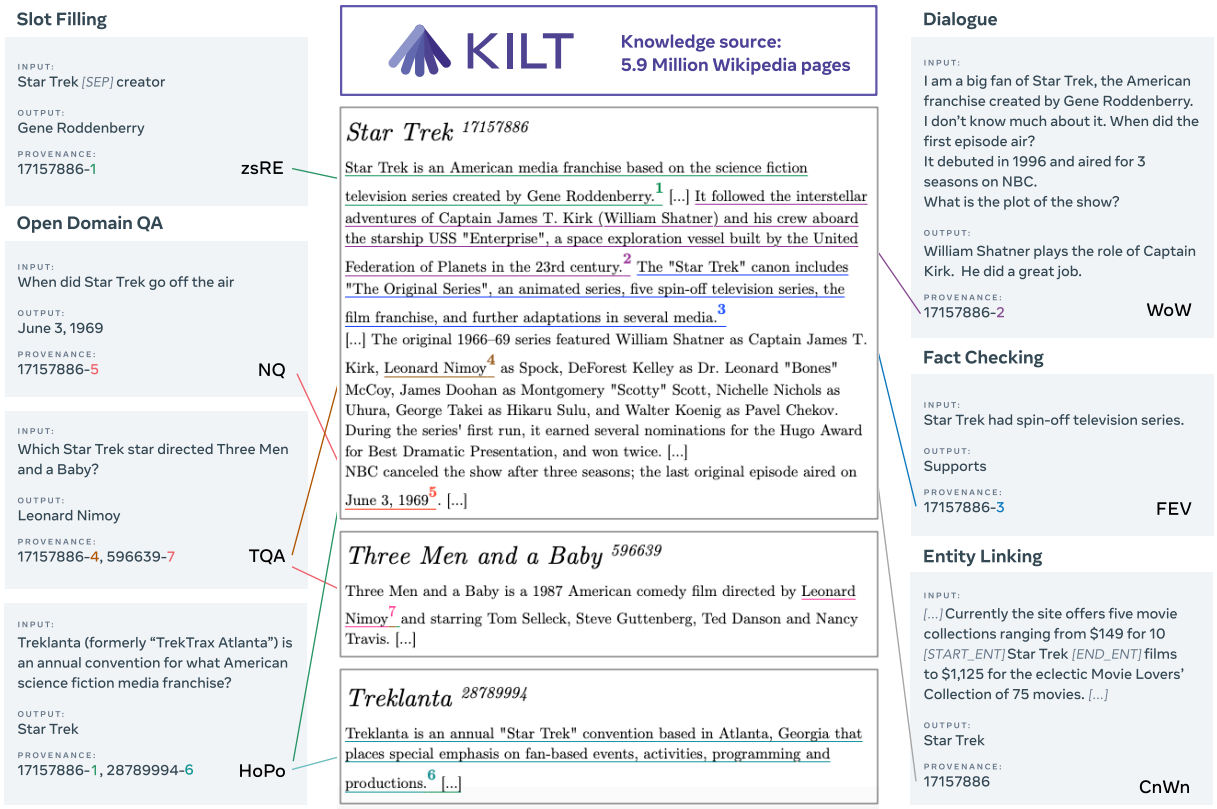


Figure 1: Common KILT interface for knowledge intensive language tasks: each instance consists of *input* and *output* with a *provenance* (text span) from the common KILT knowledge source. Source: `https://en.wikipedia.org/wiki/{Star_Trek,Three_Men_and_a_Baby,Treklanta}`

tools. Critically, they all use different knowledge sources, from different versions of Wikipedia to entirely different corpora. This makes task-to-task comparisons difficult and substantially increases computational overhead. For example, one cannot easily assess whether the same knowledge representation can be re-used if each dataset is tied to a different source. Moreover, if one decides to work with different sources across different tasks, many approaches require re-indexing and re-encoding large numbers of documents. If a language model is pre-trained on one snapshot of Wikipedia to capture its knowledge, tasks that use other snapshots might require re-training.

To facilitate research on models that must access specific information in a knowledge source, we introduce **KILT**, a benchmark and library for ***K***nowledge ***I***ntensive ***L***anguage ***T***asks. KILT aims to lower the entry barrier for such research by formulating several knowledge-intensive NLP tasks with respect to a common interface and the same unified knowledge source—a single Wikipedia snapshot. The KILT benchmark consists of eleven datasets spanning five distinct tasks, and includes the test set for all datasets considered.[2] An important aim of KILT is to cover many different ways of seeking knowledge. For this reason, we select tasks that provide a variety of ways to formulate both the input query (e.g., a claim to verify, a text chunk to annotate, a structured query, a natural question or a conversation) and the expected output (e.g., discrete, extractive, or abstractive). Moreover, while some tasks are factoid in nature (e.g., slot filling), others require using background knowledge to answer more complex questions (e.g, ELI5) or to sustain a conversation (e.g,. Wizard of Wikipedia). The format of the KILT benchmark is model-agnostic, so any system capable of producing a textual output given a textual input is eligible to participate. KILT is an *in-KB* resource (Petroni et al., 2015), i.e., the evidence required to answer each of the ~3.2M instances in KILT is present somewhere in the knowledge source. Hence there are no unanswerable instances in KILT. Although recognizing unanswerable instances is important, we believe the in-KB setting already poses an hard

[2] A brand new portion of the Natural Question (NQ) dataset, originally held out, is used as the KILT test set for NQ.

challenge to current state-of-the-art techniques, and thus leave unanswerable instances as future work.

KILT enables researchers to develop general-purpose models and evaluate them across multiple domains, testing hypotheses around task-agnostic memory and knowledge representations without indexing different large-scale textual corpora or writing new IO routines. Furthermore, the KILT library provides general building blocks to ease research on knowledge intensive NLP. We provide various state-of-the-art information retrieval systems (both neural and non-neural) coupled with different models that read text in the knowledge source and make predictions for different tasks.

We evaluate several state-of-the-art models that represent diverse approaches to knowledge-intensive NLP, and find that a hybrid approach combining a neural retriever with a pretrained sequence-to-sequence model outperforms most task-specific solutions when trained end-to-end. We additionally evaluate whether systems can provide evidence for their predictions. With this aim, we augment every instance in KILT with *provenance* information in the form of textual spans in specific Wikipedia pages to corroborate the output. We additionally perform an annotation campaign via Amazon Mechanical Turk to increase the provenance coverage. Lastly, in addition to evaluating downstream performance with popular metrics we formulate novel KILT variants for those that award points only if systems find provenance Wikipedia pages for the output given the input. The poor absolute performance of our baselines for those metrics indicates the need for focused research on systems able to explain their decisions.

In summary, we contribute:

1. a publicly-available benchmark of knowledge-intensive tasks aligned to a single Wikipedia snapshot, to spur the development of general-purpose models and enable their comparison;
2. an open-source library to facilitate the development of new architectures for knowledge-intensive tasks;
3. a provenance indication for all instances in KILT, made more comprehensive with an annotation campaign, which allows to jointly assess output accuracy and ability to provide supporting evidence in the knowledge source;
4. a comparative performance of various modeling approaches, showing promising results for general baselines across all tasks.

2 Knowledge Source

A main feature of the KILT benchmark is the use of a unified knowledge source that contains all information necessary for all tasks. Defining a unified knowledge source is a challenging problem — although all tasks use Wikipedia, they consider different snapshots. As Wikipedia pages are constantly modified, added, and removed, the knowledge can differ drastically from snapshot to snapshot. Concretely, the KILT knowledge source is based on the 2019/08/01 Wikipedia snapshot and contains 5.9M articles. We describe how each dataset is represented in KILT, and our mapping strategy for aligning data to our chosen snapshot. Additional details are in the appendix.

Mapping Datasets to a Fixed Snapshot The main challenge in defining a unified knowledge source is ensuring the knowledge for all task examples is available. We assume tasks provide an *input* (e.g. a question in question answering, or a conversation in dialogue) needed to produce an *output* (e.g. an answer or a subsequent utterance). In addition, tasks provide *provenance*, defined as a set of textual spans in Wikipedia that contain evidence for producing an output given a specific input. These provenance spans range from single entities, short answers, sentences, paragraphs, to whole articles. The idea of our mapping strategy is to identify provenance spans in the KILT knowledge source—if we find all the provenance spans for an input-output pair, the knowledge needed to produce the output is available in our snapshot. The provenance can be a span of any size, from a single token to a paragraph to an entire document.

Concretely, the mapping strategy operates as follows.[3] First, we try to match Wikipedia pages in each dataset to our snapshot, relying on Wikipedia URL redirections for pages that changed title. Second, we look for the provenance span in the matched page. We scan the whole page and return the span with the highest BLEU (Papineni et al., 2002) with the given provenance span.[4] Third, we replace the original provenance in a task's input-output pair with the span from the KILT knowledge source, and we report the BLEU score between the two. Finally, we remove from the dev and test sets all outputs for which the BLEU score is lower than a threshold for at least one provenance span (we

[3] Scripts for the mapping algorithm available on GitHub.

[4] We return the shortest span if there's a tie in BLEU score.

use 0.5 as threshold) — this is meant to ensure high quality mappings in the evaluation sets — discarding on average 18% of test and dev data (for all tasks except entity linking). We keep all input-output pairs in the train sets (see Figure 5 in the appendix for more details).

3 Tasks

We consider five tasks that use Wikipedia as a knowledge source for KILT: fact checking, open domain question answering, slot filling, entity linking, and dialogue. The diversity of these tasks challenge models to represent knowledge flexibly. Some tasks require a discrete prediction (e.g., an entity), others, such as extractive question answering, can copy the output directly from a Wikipedia page, while still other tasks must synthesize multiple pieces of knowledge in an abstractive way to produce an output. KILT also provides a variety of ways to seek knowledge, from a claim to verify to a text chunk to annotate, from a structured or natural question to a conversation (see Table 1 for details). We are able to include the test set for all datasets in KILT, either because the test set is public, or because we were able to obtain the test set from the authors of the original dataset. These test sets are not publicly released, but are used for the KILT challenge on EvalAI (Yadav et al., 2019) where participants can upload their models' predictions and be listed on the public leaderboard.[5]

To facilitate experimentation, we define a consistent interface for all datasets in the KILT Benchmark. Each dataset is represented in JSON Line format , where each record contains three fields: *id*, *input*, *output*. The *input* is a natural language string and the *output* a non-empty list of equally-valid outputs (e.g. if multiple answers to a question are valid in a question answering dataset). Each output is a string and it is accompanied by a non-empty list of complementary provenance spans (all should be used to acquire the knowledge needed to provide a valid output). Figure 1 displays an example for all considered tasks (Figure 3 in the appendix contains further details on the common interface).

3.1 Fact Checking

Fact checking verifies a claim against a collection of evidence. It requires deep knowledge about the claim and reasoning over multiple documents. We consider the claim as *input* and the classification label as *output*. Each label is accompanied by a set of provenance spans that corroborate the classification label. We model multiple equally-valid provenance sets per label.

FEVER (Thorne et al., 2018a) is a large dataset for claim veracity that requires retrieving sentence-level evidence to support if a claim is supported or refuted. Additional details are in the appendix.

3.2 Entity Linking

Entity Linking (EL) assigns a unique Wikipedia page to entities mentioned in text. Each KILT record for EL has text in the *input* (max 256 tokens) where a single entity mention is tagged with two special tokens (i.e., *[START_ENT]* and *[END_ENT]*—see Figure 1 for an example). The *output* is the title of the Wikipedia page for the entity mention plus *provenance* pointing to the entire page (through a unique identifier). Since Wikipedia associates unambiguous titles to entities[6], finding the correct output is enough to link entity mention and Wikipedia page. The *provenance* mimics the canonical approach to EL, that is to produce an identifier for each mention (Wu et al., 2019). To map the provenance (whole Wikipedia page), we simply match Wikipedia pages specified in various datasets to the KILT knowledge source. We consider three popular EL datasets in KILT, two of which do not contain a train set but should be assessed in a zero-shot fashion. Note that, in addition to the AY2 train set, the whole knowledge source can be used as training data by exploiting hyperlinks. To facilitate experimentation, we release such data in KILT format (9M train instances), following the splits of Wu et al. (2019).

AIDA CoNLL-YAGO (Hoffart et al., 2011b) supplements the CoNLL 2003 dataset (Sang and De Meulder, 2003) with Wikipedia URL annotations for all entities using the YAGO2 system (Hoffart et al., 2011a). The original data is split into three parts: *train*, *testa*, *testb*. Following Hoffart et al. (2011b) we consider *testa* as dev and *testb* as test.

WNED-WIKI (Guo and Barbosa, 2018) is a dataset automatically created by sampling document from the 2013/06/06 Wikipedia dump, and balancing the difficulty of linking each mention (using a baseline as proxy). We randomly split the dataset into dev and test.

[5] available at https://evalai.cloudcv.org/web/challenges/challenge-page/689.

[6] Wikipedia uses explicit text in titles to disambiguate.

Label	Dataset	Reference	Task	Input Format	Output Format
FEV	FEVER	Thorne et al. (2018a)	Fact Checking	Claim	Classification
AY2	AIDA CoNLL-YAGO	Hoffart et al. (2011b)	Entity Linking	Text Chunk	Entity
WnWi	WNED-WIKI	Guo and Barbosa (2018)	Entity Linking	Text Chunk	Entity
WnCw	WNED-CWEB	Guo and Barbosa (2018)	Entity Linking	Text Chunk	Entity
T-REx	T-REx	Elsahar et al. (2018)	Slot Filling	Structured	Entity
zsRE	Zero Shot RE	Levy et al. (2017)	Slot Filling	Structured	Entity
NQ	Natural Questions	Kwiatkowski et al. (2019)	Open Domain QA	Question	Extractive
HoPo	HotpotQA	Yang et al. (2018)	Open Domain QA	Question	Short Abstractive
TQA	TriviaQA	Joshi et al. (2017)	Open Domain QA	Question	Extractive
ELI5	ELI5	Fan et al. (2019b)	Open Domain QA	Question	Long Abstractive
WoW	Wizard of Wikipedia	Dinan et al. (2019)	Dialogue	Conversation	Long Abstractive

Table 1: Datasets and tasks considered in KILT.

WNED-CWEB (Guo and Barbosa, 2018) is a dataset created with the same strategy as WNED-WIKI, but sampling from the ClueWeb 2012 corpora annotated with the FACC1 system.[7] Similarly, we randomly split into dev and test.

3.3 Slot Filling

The goal of the Slot Filling (SF) is to collect information on certain relations (or slots) of entities (e.g., subject entity *Albert Einstein* and relation *educated_at*) from large collections of natural language texts. A potential application is structured Knowledge Base Population (KBP Surdeanu and Ji, 2014). SF requires (1) disambiguation of the input entity and (2) acquiring relational knowledge for that entity. For KILT, we model the *input* as a structured string *subject entity [SEP] relation*, the *output* as a list of equally-valid object-entities, each one accompanied with *provenance* where the subject-relation-object fact manifests.

Zero Shot RE (Levy et al., 2017) is a dataset designed to translate relation extraction into a reading comprehension problem. We consider the open-domain version of this dataset and align the input/output with the KILT interface. Additional details are in the appendix.

T-REx (Elsahar et al., 2018) provides a large-scale collection of facts aligned to sentences in Wikipedia abstracts through distant supervision. We consider each sentence as *provenance* and formulate the input as above (details in the appendix).

3.4 Open Domain Question Answering

Open domain Question Answering (Chen et al., 2017) is the task of producing the correct answer for a question, without a predefined location for the answer. Standard tasks such as SQuAD (Rajpurkar et al., 2016) provide an evidence document, but in open domain tasks, models must reason over an entire knowledge source (such as Wikipedia). We consider the question as *input* and the answer as *output* with dataset-specific *provenance*.

Natural Questions (Kwiatkowski et al., 2019) is a corpus of real questions issued to the Google search engine. Each question comes with an accompanied Wikipedia page with an annotated long answer (a paragraph) and a short answer (one or more entities). We consider the open-version of the dataset and use both long and short answers spans as *provenance*. We collaborated with the authors of Natural Questions to access a held out, unpublished portion of the original dataset to form a new test set for KILT. By construction each QA pair is associated with a single Wikipedia page, although other pages might contain enough evidence to answer the question. To increase the provenance coverage we perform an Amazon Mechanical Turk campaign for the dev and test sets and increase the average number of provenance pages per question from 1 to 1.57 (details in section 4).

HotpotQA (Yang et al., 2018) requires multi-hop reasoning over multiple Wikipedia pages to answer each question. For each question-answer pair, a set of supporting sentences are provided, and we consider these as *provenance*. We focus on the *fullwiki* setting, where systems are required to retrieve and reason over the whole Wikipedia.

TriviaQA (Joshi et al., 2017) is a collection of question-answer-evidence triples. Evidence documents are automatically gathered from Wikipedia or the Web. We consider only the Wikipedia case. We use the answer span as *provenance* and consider the full version of the dev and test set.

[7] http://lemurproject.org/clueweb12

ELI5 (Fan et al., 2019b)[8] is a collection of question-answer-evidence triples where the questions are complex, and the answers are long, explanatory, and free-form. For dev and test, we collect annotations using Amazon Mechanical Turk, asking evaluators to select which supporting documents from Wikipedia can be used to answer the question. We treat these as gold provenance annotations for evaluation (details in section 4).

3.5 Dialogue

Chitchat dialogue is the task of developing an engaging chatbot that can discuss a wide array of topics with a user, which often relies on topical, factual knowledge. For example, it would be difficult to have a conversation about "grayhounds" without any information about that dog breed. We consider the conversation history as *input* and the next utterance as *output*.

Wizard of Wikipedia (Dinan et al., 2019) is a large dataset of conversation grounded with knowledge retrieved from Wikipedia. One speaker in the conversation must ground their utterances in a specific knowledge sentence, chosen from a Wikipedia page. The chosen sentence forms the *provenance* for KILT.

4 Provenance Annotation Campaign

We perform an Amazon Mechanical Turk campaign on the NQ and ELI5 datasets for the dev and test splits. While for the NQ our aim is to increase the provenance coverage (i.e., we already have a provenance page for each qa pair) for ELI5 we want to collect provenance information from scratch. For each question we ask annotators to indicate if four pre-determined passages contain enough evidence to answer the question and additionally highlight a salient span in them. We select the passages to annotate using our baseline retrieval models, namely Tf-idf, DPR, RAG and BLINK + flair (details in the Appendix).[9] We only consider passages with some tokens overlap with the gold answers (at least 10%).

For NQ, we additionally include gold passages among those to annotate, with the twofold objective of controlling the quality of the annotation process and filter out questions that can't be answered given the KILT Wikipedia snapshot.[10] If no passage is selected by an annotator we ask to provide either another one from Wikipedia or an explanation. We collect three annotations for each passage, and insert the passage as new provenance for the question if at least two annotators found enough evidence to answer in it. The average inter-annotator agreement is 0.3 and 0.1 Cohen's kappa for NQ and ELI5 respectively. Note that ELI5 questions are in general more complex than NQ ones, the required answer is not an extracted span from a page but a free-form explanation that not always can be grounded in Wikipedia.

To make ELI5 data more robust we computed the overlap between provenance passages and answers for each instance using ROUGE-L and manually annotate instances with low overlap (ROUGE-L < 0.15). Overall, we were able to collect provenance information for 1507 dev instances (3000 annotated) and 600 test instances (2000 annotated) for ELI5, with an average of 1.18 Wikipedia pages as provenance per instance. For NQ, we filter out on average 8% of data (258 dev and 110 test instances) and include on average 1.57 Wikipedia pages as provenance per instance. Additional details in the Appendix, table 6.

5 Evaluation Metrics

Various tasks in the KILT Benchmark need to be evaluated differently, which can make task-wide comparison challenging. Further, there are multiple aspects of each system that we want to assess, namely (1) downstream results, (2) performance in retrieving relevant evidence to corroborate a prediction and (3) a combination of the two. We report different metrics to capture these aspects.[11]

Downstream performance. We consider different metrics to capture the uniqueness of the different tasks in KILT and mimic the typical way to assess performance for each dataset. We use *Accuracy* for tasks that require a discrete output (e.g., an entity); *Exact Match* (EM) for tasks with extractive (i.e., Natural Questions, TriviaQA) or short abstractive output format (i.e., HotpotQA); finally, for tasks with long abstractive output format, we use *ROUGE-L* (Lin, 2004) for ELI5 and F1-score for Wizard of Wikipedia. For EM and F1-score we follow standard post-processing to lowercase,

[8] https://yjernite.github.io/lfqa.html

[9] for Tf-idf and BLINK + flair we consider the first passage in the retrieved page

[10] we present passages in random order to the annotator to exclude biases.

[11] evaluation scripts available in GitHub.

Model	#Parameters
Trans MemNet (Dinan et al., 2019)	15.5M
BERT (base) (Devlin et al., 2019)	110M
NSMN (Nie et al., 2019)	199M +93M nt
T5 (base) (Raffel et al., 2019b)	220M
DPR (Karpukhin et al., 2020)	220M +15B idx
BERT (large) (Devlin et al., 2019)	340M
BART (large) (Lewis et al., 2019)	406M
RAG (Lewis et al., 2020b)	626M +15B idx
BLINK (Wu et al., 2019)	680M +6B idx

Table 2: Baselines considered and total number of their trainable parameters. Non trainable (nt) parameters and index (idx) sizes are also reported.

strip articles, punctuation, and duplicate whitespace from gold and predicted output (Rajpurkar et al., 2016). Note that Accuracy is equivalent to strict exact match, without post-processing. We report additional metrics for some datasets in the appendix (Table 7-17).

Retrieval. We adopt a page-level formulation and measure the ability of a model to provide a set of Wikipedia pages as evidence for a prediction.[12] For most datasets in KILT a single page is enough to provide complete evidence, with the exception of FEVER (~12% which requires more than one page) and HotpotQA (two pages are always required). We consider the following retrieval metrics in KILT:

R-precision, calculated as $\frac{r}{R}$, where R is the number of Wikipedia pages inside each provenance set and r is the number of relevant pages among the top-R retrieved pages. For most of the datasets $R = 1$ and this formulation is equivalent to *Precision@1*. Concretely, R-precision=1 if all Wikipedia pages in a provenance set are ranked at the top. We report the maximum value among all provenance sets for any given input.

Recall@k, calculated as $\frac{w}{n}$, where n is the number of distinct provenance sets for a given input and w is the number of complete provenance sets among the top-k retrieved pages. For datasets that require more than one page of evidence (e.g., FEVER and HotpotQA), we use the lowest ranked page in each provenance set to determine its position and remove the other pages in the set from the rank. For both metrics, we report the mean over all test datapoints.

[12] our evaluation scripts allow to evaluate retrieval performance at a more fine-grained level (e.g., paragraph).

KILT scores. We propose a KILT version for downstream metrics that, inspired by the FEVER-score (Thorne et al., 2018a), takes into account the provenance supporting the output. For each datapoint, we only award Accuracy, EM, ROUGE-L, and F1 points to *KILT-AC*, *KILT-EM*, *KILT-RL* and *KILT-F1* respectively, if the R-precision is 1. This is equivalent to awarding points if the system finds (and ranks at the top) a complete set of provenance Wikipedia pages for at least one ground truth output given the input. We choose this metric to emphasize that systems must be able to explain their output with proper evidence, not simply answer.

6 Baselines

The KILT tasks provide a dual challenge of retrieving information and conditioning upon that to create an output. Various directions could be applied to these. For example, the Wikipedia knowledge could be represented *explicitly*, as natural language or in a structured form, or represented *implicitly*, as knowledge stored in model parameters. Models could be *discriminative*, *extractive*, where a specific span is selected as output, or *generative*, where the model writes an output. We consider retrieval, task-specific, and general baselines for KILT (see Table 2). Additional details are in the appendix.

7 Results

We summarize the main results in three tables: downstream performance in Table 3, retrieval in Table 4 and KILT scores in Table 5. Additional results, as well as comparisons with recent works reported numbers, can be found in the appendix. It's possible to get the performance of a system for the KILT test sets by uploading its predictions to our EvalAI challenge.[5]

When considering downstream performance (Table 3), although pre-trained sequence-to-sequence models can embed knowledge implicitly in their parameters to some extent (Petroni et al., 2019; Roberts et al., 2020), they clearly lag behind models with explicit knowledge access in almost all datasets. The BART+DPR baseline that incorporates an explicit retrieval step in addition to the generative pretraining, works well. It outperforms some of the task-specific solutions, and gets close to others. Performance are even stronger when the retriever and reader components are trained end-to-end, as in the case of RAG. We find this a promising direction for knowledge intensive tasks.

		Fact Check.	Entity Linking			Slot Filling		Open Domain QA				Dial.
	model	**FEV**	**AY2**	**WnWi**	**WnCw**	**T-REx**	**zsRE**	**NQ**	**HoPo**	**TQA**	**ELI5**	**WoW**
		Accuracy						Exact Match			RL	F1
ts	NSMN	66.1	-	-	-	-	-	-	-	-	-	-
	BERT + DPR	69.68	-	-	-	-	6.93	38.64	11.29	70.38	-	-
	BLINK	-	**81.54**	**80.24**	**68.77**	-	-	-	-	-	-	-
	Trans MemNet	-	-	-	-	-	-	-	-	-	-	11.5
im	BART (large)	78.93	77.55	45.91	49.16	45.06	9.14	21.75	15.37	32.39	**20.55**	12.96
	T5 (base)	76.3	74.05	47.13	49.29	43.56	9.02	19.6	12.64	18.11	19.08	13.49
ex	BART + DPR	**86.74**	75.49	45.2	46.87	59.16	30.43	41.27	25.18	58.55	17.41	**15.55**
	RAG	86.31	72.62	48.07	47.61	**59.2**	**44.74**	**44.39**	**26.97**	**71.27**	14.05	13.22

Table 3: Downstream performance on the test data. Baselines are grouped by task-specific (*ts*) and general with implicit (*im*) or explicit (*ex*) knowledge access. Task-specific solutions cannot be generally applied to all datasets in KILT, hence there are empty cells in the top part of the table. We report the typical metric to assess performance for each dataset, specified in the first row.

	Fact Check.	Entity Linking			Slot Filling		Open Domain QA				Dial.
model	**FEV**	**AY2**	**WnWi**	**WnCw**	**T-REx**	**zsRE**	**NQ**	**HoPo**	**TQA**	**ELI5**	**WoW**
	R-Precision										
DPR + BERT	72.93	-	-	-	-	40.11	**60.66**	25.04	43.4	-	-
DPR	55.33	1.81	0.3	0.51	13.26	28.96	54.29	25.04	44.49	10.67	25.48
Multi-task DPR	**74.48**	26.48	4.91	1.86	**69.46**	**80.91**	59.42	42.92	61.49	**15.5**	41.07
Tf-idf	50.85	3.74	0.24	2.09	44.74	60.83	28.12	34.14	46.37	13.67	49.01
RAG	61.94	72.62	48.07	47.61	28.68	53.73	59.49	30.59	48.68	11.0	**57.78**
BLINK + flair	63.71	**81.54**	**80.24**	**68.77**	59.56	78.78	24.52	**46.12**	**65.58**	9.5	38.21

Table 4: Page-level R-Precision on test data. For DPR, we additionally report the performance after the BERT-based classifier (for FE) or reader (for NQ,HP,TR) re-ranked relevant pages (i.e., DPR + BERT). R-Precision is equivalent to Precision@1 for all datasets except FEV and HoPo that require multi-hop.

By formulating Entity Linking within KILT, we can evaluate the ability of seq2seq models at this task. They perform surprisingly well, even without any explicit access to knowledge (i.e., BART and T5). These solutions are able to link entity mentions by either leaving them untouched (if they match the correct Wikipedia title), completely altering mention text (e.g., "European Cup" → "UEFA Champions League"), or adding disambiguation tokens (e.g., "Galatasaray" → "Galatasaray S.K. (football)"). We report an example in Figure 4.

When considering retrieval alone (Table 4) there is no clear winner—entity-centric tasks (Entity Linking and Slot Filling) clearly benefit from entity-based retrieval, while DPR works better for NQ, FEV and ELI5, that require more fine grained passages supervision. We believe that combining all these ingredients (i.e., dense representations, fine grained supervision, entity awareness) will be necessary for general task-agnostic memories. Moreover, jointly training a single DPR model on all KILT training data (Multi-task DPR) led to strong performance gains on all datasets compared with the original model (DPR), that considers only NQ and TQA as training data (Karpukhin et al., 2020). This suggests synergies between KILT datasets that are beneficial in terms of model performance.

Finally, the KILT scores formulation allows us to systematically assesses the performance for output and provenance jointly (Table 5). We don't report results for BART and T5 since answers are generated solely from the input with no explicit retrieval and there is no straightforward way to access provenance for each prediction. The relative performance of the other baselines with respect to KILT scores is consistent with downstream results. However, the generally low absolute numbers leave a large room for improvement for systems able to provide the correct output but also successfully justify their decision.

	model	Fact Check.	Entity Linking			Slot Filling		Open Domain QA				Dial.
		FEV	**AY2**	**WnWi**	**WnCw**	**T-REx**	**zsRE**	**NQ**	**HoPo**	**TQA**	**ELI5**	**WoW**
		KILT-AC						KILT-EM			-RL	-F1
ts	NSMN	41.88	-	-	-	-	-	-	-	-	-	-
	BERT + DPR	**58.58**	-	-	-	-	4.47	31.99	0.74	34.48	-	-
	BLINK	-	**81.54**	**80.24**	**68.77**	-	-	-	-	-	-	-
	Trans MemNet	-	-	-	-	-	-	-	-	-	-	2.23
ex	BART + DPR	47.68	75.49	45.2	46.87	11.12	18.91	30.06	1.96	31.4	**1.9**	4.52
	RAG	53.45	72.62	48.07	47.61	**23.12**	**36.83**	**32.69**	**3.21**	**38.13**	1.69	**9.1**

Table 5: KILT scores on the test data. We do not report KILT scores for baselines with implicit knowledge access since no provenance information is returned by them. We report the KILT version of donwstream metrics, specified in the first row (to save space we abbreviate KILT-RL and KILT-F1). KILT scores are computed by awarding points only if provenance pages are found (i.e., R-Precision = 1).

8 Discussion

There are custom solutions that can easily simplify the slot filling task. For instance, subject entities can be used for lookups by title in Wikipedia to retrieve knowledge (this heuristic will always work for zsRE), and structured human-curated resources (such as Wikidata[13]) could be used to get all answers right. Nevertheless, we are interested in testing if a general model can extract attributes about specific entities from a large body of text.

The provenance to justify each system prediction can come from anywhere, including a different system, and this is difficult to detect. Moreover our provenance might not be exhaustive—given the redundancy of information in Wikipedia there could be other pages with the knowledge needed to solve a KILT instance. We conduct an annotation campaign to mitigate the problem.

9 Related Work

Several natural language benchmarks have been introduced to track and support NLP progress, including natural language understanding (Wang et al., 2018, 2019), multitask question answering (McCann et al., 2018), reading comprehension (Dua et al., 2019), question understanding (Wolfson et al., 2020), and dialogue (Shuster et al., 2019). We focus on multi-domain tasks that need to seek knowledge in a large body of documents to produce an output. Although there exist several tasks and resources that define large-scale external knowledge sources—including the TAC-KBP challenges (McNamee and Dang, 2009; Ji et al., 2010; Surdeanu, 2013; Surdeanu and Ji, 2014), ARC (Clark et al., 2018), TriviaQA-web (Joshi et al., 2017), Quasar-T (Dhingra et al., 2017), WebQuestions (Berant et al., 2013) and ComplexWebQuestions (Talmor and Berant, 2018)—in KILT we exclusively consider publicly available Wikipedia-based datasets in order to merge and unify the knowledge source.

10 Conclusion

We introduce KILT, a benchmark for assessing models that need to condition on specific knowledge in a defined snapshot of Wikipedia to solve tasks spanning five domains. The goal is to catalyze and facilitate research towards general and explainable models equipped with task-agnostic representations of knowledge. Our experiments show promising results for a general solution combining dense retrieval and seq2seq generations, although there is large room for improvements. In particular, we find that provenance of current models is generally low.

11 Acknowledgment

The authors would like to greatly thank the team behind Natural Questions[14] for the held out data, that defines our NQ test set; FEVER[15], HotpotQA[16] and TriviaQA[17] teams for sharing official test data for the KILT leaderboard; Luke Zettlemoyer and Scott Wen-tau Yih for helpful discussions; Rishabh Jain for the help in setting up the EvalAI challenge.

[13] https://www.wikidata.org

[14] https://ai.google.com/research/NaturalQuestions

[15] https://fever.ai

[16] https://hotpotqa.github.io

[17] https://nlp.cs.washington.edu/triviaqa

References

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. Flair: An easy-to-use framework for state-of-the-art nlp. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1533–1544.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. *ACL*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

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.

Bhuwan Dhingra, Kathryn Mazaitis, and William W Cohen. 2017. Quasar: Datasets for question answering by search and reading. *arXiv preprint arXiv:1707.03904*.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of wikipedia: Knowledge-powered conversational agents. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Dheeru Dua, Ananth Gottumukkala, Alon Talmor, Sameer Singh, and Matt Gardner. 2019. Orb: An open reading benchmark for comprehensive evaluation of machine reading comprehension. *arXiv preprint arXiv:1912.12598*.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Elena Simperl, and Frederique Laforest. 2018. T-rex: A large scale alignment of natural language with knowledge base triples. *LREC*.

Angela Fan, Claire Gardent, Chloé Braud, and Antoine Bordes. 2019a. Using local knowledge graph construction to scale Seq2Seq models to multi-document inputs. 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 4186–4196, Hong Kong, China. Association for Computational Linguistics.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019b. ELI5: long form question answering. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3558–3567. Association for Computational Linguistics.

Paolo Ferragina and Ugo Scaiella. 2011. Fast and accurate annotation of short texts with wikipedia pages. *IEEE software*, 29(1):70–75.

Zhaochen Guo and Denilson Barbosa. 2018. Robust named entity disambiguation with random walks. *Semantic Web*, 9(4):459–479.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training.

Johannes Hoffart, Fabian M Suchanek, Klaus Berberich, Edwin Lewis-Kelham, Gerard De Melo, and Gerhard Weikum. 2011a. Yago2: exploring and querying world knowledge in time, space, context, and many languages. In *Proceedings of the 20th international conference companion on World wide web*, pages 229–232.

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

Heng Ji, Ralph Grishman, Hoa Trang Dang, Kira Griffitt, and Joe Ellis. 2010. Overview of the tac 2010 knowledge base population track. In *Third text analysis conference (TAC 2010)*, volume 3, pages 3–3.

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 Oğuz, Sewon Min, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin,

Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. *arXiv preprint arXiv:1906.00300*.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. *CoNLL*.

Mike Lewis, Marjan Ghazvininejad, Gargi Ghosh, Armen Aghajanyan, Sida Wang, and Luke Zettlemoyer. 2020a. Pre-training via paraphrasing. *arXiv preprint arXiv:2006.15020*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *ArXiv*, abs/1910.13461.

Patrick Lewis, Ethan Perez, Aleksandara 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.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

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

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

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*.

Paul McNamee and Hoa Trang Dang. 2009. Overview of the tac 2009 knowledge base population track. In *Text Analysis Conference (TAC)*, volume 17, pages 111–113. National Institute of Standards and Technology (NIST) Gaithersburg, Maryland ....

Alexander H Miller, Will Feng, Adam Fisch, Jiasen Lu, Dhruv Batra, Antoine Bordes, Devi Parikh, and Jason Weston. 2017. Parlai: A dialog research software platform. *arXiv preprint arXiv:1705.06476*.

Yixin Nie, Haonan Chen, and Mohit Bansal. 2019. Combining fact extraction and verification with neural semantic matching networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6859–6866.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Fabio Petroni, Luciano del Corro, and Rainer Gemulla. 2015. Core: Context-aware open relation extraction with factorization machines. In *EMNLP*. Assoc. for Computational Linguistics.

Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2020. How context affects language models' factual predictions. *AKBC*.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *EMNLP*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019a. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019b. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *EMNLP*.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 74–84.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? *arXiv preprint arXiv:2002.08910*.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

Kurt Shuster, Da Ju, Stephen Roller, Emily Dinan, Y-Lan Boureau, and Jason Weston. 2019. The dialogue dodecathlon: Open-domain knowledge and image grounded conversational agents. *arXiv preprint arXiv:1911.03768*.

Mihai Surdeanu. 2013. Overview of the tac2013 knowledge base population evaluation: English slot filling and temporal slot filling. In *TAC*.

Mihai Surdeanu and Heng Ji. 2014. Overview of the english slot filling track at the tac2014 knowledge base population evaluation. In *Proc. Text Analysis Conference (TAC2014)*.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 641–651.

James Thorne and Andreas Vlachos. 2020. Avoiding catastrophic forgetting in mitigating model biases in sentence-pair classification with elastic weight consolidation. *arXiv preprint arXiv:2004.14366*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018a. FEVER: a large-scale dataset for fact extraction and verification. In *NAACL-HLT*.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018b. The fact extraction and verification (fever) shared task. *EMNLP*.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2019. The fever2. 0 shared task. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, pages 1–6.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, pages 3261–3275.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Thomas Wolf, L Debut, V Sanh, J Chaumond, C Delangue, A Moi, P Cistac, T Rault, R Louf, M Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv, abs/1910.03771*.

Tomer Wolfson, Mor Geva, Ankit Gupta, Matt Gardner, Yoav Goldberg, Daniel Deutch, and Jonathan Berant. 2020. Break it down: A question understanding benchmark. *Transactions of the Association for Computational Linguistics*, 8:183–198.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2019. Zero-shot entity linking with dense entity retrieval. *arXiv preprint arXiv:1911.03814*.

Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. 2019. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. *arXiv preprint arXiv:1912.09637*.

Deshraj Yadav, Rishabh Jain, Harsh Agrawal, Prithvijit Chattopadhyay, Taranjeet Singh, Akash Jain, Shiv Baran Singh, Stefan Lee, and Dhruv Batra. 2019. Evalai: Towards better evaluation systems for ai agents. *arXiv preprint arXiv:1902.03570*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380.

A Appendix

Wikipedia Representation We represent the KILT knowledge source as a collection of JSON records, one per Wikipedia page. Each record is assigned: (i) a unique Wikipedia id; (ii) a unique Wikipedia title; (iii) a text field containing a list of strings, one for each paragraph, bulleted list item, and section header (for which we preserve the hierarchical structure); (iv) a list of anchors elements, one for each hyperlink in the original page text, with span reference in the text field and page linked; (v) a list of categories; (vi) a url redirecting to the original html for the page, with timestamp of the last page revision before the considered snapshot.

Datasets Mapping Details In *FEVER*, often multiple pieces of knowledge must be combined to produce an output. For example, 30% of claims have more than one equally-valid provenance and 16% require the combination of multiple evidence spans. The second iteration (*FEVER2.0*, Thorne et al., 2019) introduces a collection of adversarial instances. For KILT, we merge the two versions of FEVER into a single resource and consider only supported refuted claims. We exclude all claims classified as not having enough information since these instances have no evidence to assess the claim and cannot be mapped to the KILT knowledge source. Therefore we cannot asses whether such label is still appropriated given our snapshot. Moreover, we design KILT as an in-KB resource where each instance can be answered and corroborated by information in the knowledge source.

In the *Zero Shot RE* dataset a set crowd-sourced template questions are defined for each relation — for example, *What is Albert Einstein's alma mater?*. Each datapoint reports a Wikipedia sentence expressing the fact that we take as *provenance*. Some examples in the dataset are negative, obtained by matching a valid question and a random sentence, that likely does not contain the answer. To consider an open-domain version of this dataset and align the input/output with the KILT interface we reformatted this dataset, as follows: (i) exclude neagative pairs - since we consider the whole knowledge source (as opposite to a single sentence) as text all questions can be answered; (ii) group template questions by the subject-relation pair, and create a single datapoint for each (*input* as above); (iii) randomly split the set of relations, in line with the original dataset, into three disjoint sets train (with 84 relations), dev (12 relations) and test (24 relations)—systems are tested on relations never seen during training; (iv) use the subject entity as the query against Wikipedia titles for the first step of the mapping strategy, and (v) include all template questions in a *meta* field.

For *T-REx*, We filter out facts with more than 20 provenances, relations with less than 1000 facts, and merge all the facts for the same subject-relation pair (i.e., for 1-N and M-N relations there could be multiple valid answers), resulting in 113 relations and 2.3M facts. We include object aliases as equally valid answers and report in a *meta* field subject aliases as well as all surface mentions for the subject, relation and object. We randomly select 5k facts for both dev and test set.

To define an open-version of the *Natural Questions* dataset we follow Lee et al. (2019) and (1) keep only questions with short answers and (2) discard all answers with more than five tokens.

To find answers in *TriviaQA*, the original work used distant supervision: (1) find Wikipedia entities in the question with the TAGME entity linked (Ferragina and Scaiella, 2011); (2) search for the answer (and all Wikipedia aliases) in the corresponding page; (3) if the answer is found, add the page in the evidence documents. Therefore, the documents are not guaranteed to contain evidence for the question-answer pair (but the authors estimate that they do 79.7% of the time).

In *ELI5* Evidence documents are automatically gathered, and we focus on the case where evidence documents are extracted from Wikipedia. However, as the original work first collected question-answer pairs from the subreddit *Explain Like I'm Five*, the documents are not guaranteed to contain evidence.

For *Wizard of Wikipedia* we discard cases where the dataset does not contain provenance. Moreover, we consider a full open-domain setting where no topic is provided for the conversation and the model must search over all of Wikipedia for knowledge at each dialogue turn (rather than the provided knowledge candidates for each turn in the original dataset). We use the unseen split for dev and test.

Performance Impact Of The Mapping Strategy We want to assess if the performance we obtain after mapping each dataset to a unified Wikipedia snapshot are in line with what reported in previous work. Thorne and Vlachos (2020) report a 2-way accuracy of 79.09 for the FEVER dev set when considering purely claims in input to a RoBERTa-

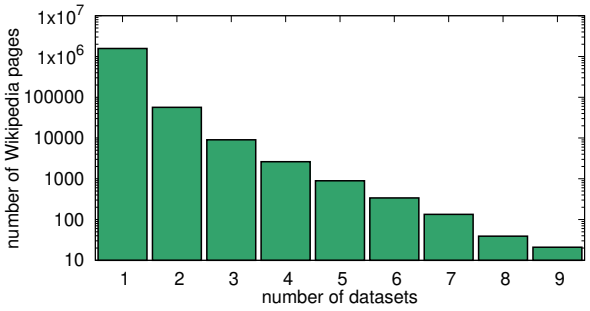


Figure 2: Number of pages vs number of dataset with knowledge in a page. 1,642,311 pages contains knowledge needed for KILT (~28% of the knowledge source).

based classifier (Liu et al., 2019). Our dev set includes also the adversarial examples of FEVER 2.0, nevertheless the performance of BART are in line (80.67 dev, 78.93 test). Karpukhin et al. (2020) report 41.5 for EM on the open domain version of the NQ dev set[18]. With our setting, DPR achieves an on-par performance on the dev set, with a 42.58 EM (50.43 F1-score). Results on our brand new NQ test set are 3/4 points lower for EM and F1-score than dev results. We don't evaluate multi-hop specific baselines on KILT but the current best F1-score for HotpotQA is 75.43 according to the official leaderboard[19], that is quite far from what achieved by our general solutions. BLINK results are in line with what reported in the GitHub repository[20] for all three entity linking datasets. The Tranformer MemNet of Dinan et al. (2019) achieves a F1-score of 14.3 on the original version of the WW dataset while 11.5 in our setting, probably because in KILT we consider an harder open-domain setting.

Retrieval Baselines The ability to retrieve relevant documents from Wikipedia given an input is an important aspect we assess in KILT. A system should select only the relevant knowledge needed for the task, without redundant or excess information. A way to surface such knowledge is using a dedicated retrieval system. We consider three off-the-shelf retrievers and investigate drastically different retrieval paradigms: (i) *Tf-idf* with the DrQA Document Retriever (Chen et al., 2017)—traditional page-level sparse vector space retrieval model; (ii) *DPR* (Karpukhin et al., 2020)—a modern passage-level retrieval solution using dense representations; (iii) A combination of *BLINK* (Wu et al., 2019) and *flair* (Akbik et al., 2019)—retrieval solution that ranks pages according to entities in the input.

The DrQA Document Retriever combines bigram hashing and TF-IDF matching to return relevant Wikipedia pages given an input. DPR splits each Wikipedia page into disjoint 100-word passages[21] and encodes passages and inputs with a BERT-based bi-encoder to perform dense Maximum Inner Product Search. The BLINK entity linking system uses a BERT-based bi-encoder to encode each Wikipedia page as well as each input, where a single entity mention is tagged. Final results are refined with a BERT-based cross-encoder. To use BLINK for retrieval, we look for entity mentions in each input with flair, then use BLINK to return a ranked list of Wikipedia pages for each entity mention. When multiple entities are identified in the input, we merge results and sort by score. The input string might not contain tags. For all systems, we use the index created on the KILT knowledge source.

We also experiment with multi-tasking, by jointly training a single DPR model on all KILT training data. We use uniform sampling to balance the datasets. In particular, the Multi-task variant of DPR is a single dense passage retriever, trained jointly on the union of TQA, NQ, HoPo, FEV, zsRE, AY2, T-REx and WoW. In order to avoid large datasets, such as T-REx, from having an oversize effect, we resample all datasets uniformly, such that every training epoch contains 150k samples from each task. Batches are formed from a single dataset at a time, iterating through the various datasets in a round-robin fashion.

Task-specific Baselines Approaches to the KILT Benchmark should be able to generalize to many different tasks, as developing model architectures that can represent knowledge generally is a valuable direction. However, several tasks may benefit from dedicated architectures designed for them.

For *fact checking*, we consider *NSMN* (Nie et al., 2019), the highest scoring system from the FEVER shared task (Thorne et al., 2018b). We use the public model[22] pre-trained on FEVER, and consider not enough information predictions as false.

[18] Reported as test results in (Karpukhin et al., 2020)

[19] https://hotpotqa.github.io

[20] https://github.com/facebookresearch/BLINK

[21] 22,220,793 passages in the KILT knowledge source. Following Karpukhin et al. (2020) we don't consider Wikipedia bulleted lists in the text.

[22] available at https://github.com/easonnie/combine-FEVER-NSMN

Moreover, we develop a fact checking baseline that combines a BERT-base classifier with passages returned from DPR where the claim and retrieved passage are input. The classifier is trained to label the claim-passage pair as supported or refuted with an additional neutral class for negative-sampled unrelated passages. Unrelated passages are sampled from two sources: (1) DPR-retrieved passages from pages that are not in the list of pages in the instance's provenance and (2) passages sampled uniformly at random from pages in the instance's provenance. At inference, we classify the first sentence of the Wikipedia pages retrieved by the top-100 DPR passages against the claim. Using pages labelled as supported or refuted, we label the claim through majority voting. For claim provenance, we re-rank passages by probability according to this label.

For *Open Domain QA* and Slot Filling, we use DPR combined with the pre-trained BERT-based extractive reading comprehension model of Karpukhin et al. (2020). We use the model pre-trained on TriviaQA for HotpotQA and the model pre-trained on Natural Questions for Zero Shot RE. We reduce the slot filling problem to question answering, by using the specified template questions. We consider a single random template question per subject-relation during inference.

For *Entity Linking*, we consider *BLINK*.

For *Dialogue*, we consider the Generative *Transformer MemNet* (Dinan et al., 2019) that encodes the dialogue history and knowledge to generates the next utterance. We use the pre-trained version available in ParlAI (Miller et al., 2017). Finally, to test the performance of combining BART and DPR on FEVER, we develop a classifier that uses these—full description in the appendix.

General Baselines A main motivation of the KILT Benchmark is to enable a unified approach towards a wide range of knowledge-intensive tasks. We analyze existing general architectures that can be used as a baseline for multiple tasks in KILT.

Large pre-trained sequence-to-sequence models such as BART (Lewis et al., 2019) and T5 (Raffel et al., 2019a) implicitly store a surprising amount of knowledge in their parameters (Petroni et al., 2019). We treat all KILT tasks as generative, relying on the knowledge accumulated by the model while pre-training, with no retrieval (similarly to Roberts et al. (2020)). We finetune pre-trained variants on all KILT tasks, using `fairseq` (Ott et al., 2019) for BART and Huggingface's Transformer (Wolf et al., 2019) for T5.

A natural way to boost performance is to incorporate an explicit knowledge mechanism. For our BART+DPR baseline, we follow Petroni et al. (2020) to retrieve and prepend the top-3 passages from DPR for each input sample and use context-enhanced training data to fine-tune a BART model. We use the DPR rank when reporting provenance for all except entity linking tasks. For entity linking, we report the Wikipedia id of the page whose title exactly matches the predicted string.

Recently, state-of-the-art results on a wide range of NLP tasks have been achieved by combining a trainable retrieval step with language modeling or generation (Guu et al., 2020; Lewis et al., 2020a). We experiment with fine-tuning RAG (Lewis et al., 2020b) on KILT tasks, establishing a strong baseline on all of them. RAG combines a DPR retriever with a BART generator, however, unlike in the case of our previous baseline, RAG back-propagates to the retriever's input encoder, learning to adapt the input embedding to retrieve more relevant results. At every generation step we retrieve top-5 passages and use them as provenance.

Dataset Label	Multi-hop	Average Provenance Size (APS)	Average Provenance Number (APN)	Average Provenance Pages (APP)	Average Answers Number (AAN)	Train Size	Dev Size	Test Size
FEV	x	1.12	1.35	1.13	1	104,966	10,444	10,100
AY2		1	1	1	1	18,395	4,784	4,463
WnWi		1	1	1	1	-	3,396	3,376
WnCw		1	1	1	1	-	5,599	5,543
T-REx		1	1.68	1.26	5.29	2,284,168	5,000	5,000
zsRE		1	1	1	1	147,909	3,724	4,966
NQ		1	3.22	1.57	2.08	87,372	2,837	1,444
HoPo	x	2.4	1	2	1	88,869	5,600	5,569
TQA		1	3.39	1.68	28.67	61,844	5,359	6,586
ELI5		1	1.21	1.18	4.69	272,634	1,507	600
WoW		1	1	1	1	94,577	3,058	2,944
			Total			3,160,734	51,464	50,736

Table 6: Datasets statistics. APS refers to the average number of textual spans in each provenance set—for most of the datasets a single span is sufficient to provide enough evidence while FEV and HoPo might require more (hence they require multi-hop reasoning). APN indicates the average number of equally valid provenance sets for each instance while APP the average number of Wikipedia pages overall in the provenance (note that multiple spans might refer to the same Wikipedia page). Finally AAN reports the average number of equally valid gold answers per instance. We additionally report the size of the train, dev and test split for each dataset.

```
{'id': # original data point id if available otherwise unique id
 'input': # question / claim / sentence / etc
 'output': [ # each element might contain an answer, a provenance or both
    {
     'answer': # answer in textual form
     'provenance': [
         # evidence set for the answer from the KILT knowledge source
         {
             'wikipedia_id':  # *mandatory*
             'title':
             'section':
             'start_paragraph_id':
             'start_character':
             'end_paragraph_id':
             'end_character':
             'bleu_score': # wrt original evidence
             'meta': # dataset/task specific
         }
         ]
       }
    ]
 'meta': # dataset/task specific
 }
```

Figure 3: KILT datasets' interface. Each dataset is represented as a JSON Line file. The Figure shows the pseudo-JSON structure for each record in the files.

model	**R-Precision**	**Recall@5**	**Accuracy**	**KILT-AC**
		test		
BART	0.0	0.0	78.93	0.0
T5	0.0	0.0	76.3	0.0
NSMN	49.24	70.16	66.1	41.88
BART + DPR	55.33	74.29	86.74	47.68
RAG	61.94	75.55	86.31	53.45
BERT + DPR	72.93	73.52	69.68	58.58
		dev		
BART	0.0	0.0	80.67	0.0
BART + DPR	55.46	73.84	88.11	48.25
RAG	63.5	76.1	87.7	55.47

Table 7: FEVER

model	**R-Precision**	**Recall@5**	**Accuracy**	**KILT-AC**
		test		
RAG	72.62	72.62	72.62	72.62
T5	74.05	74.05	74.05	74.05
BART + DPR	75.49	75.49	75.49	75.49
BART	77.55	77.55	77.55	77.55
BLINK	81.54	94.73	81.54	81.54
		dev		
RAG	77.4	77.47	77.4	77.4
T5	81.84	81.84	81.84	81.84
BART	86.62	86.62	86.62	86.62

Table 8: AIDA CoNLL-YAGO

model	**R-Precision**	**Recall@5**	**Accuracy**	**KILT-AC**
		test		
BART + DPR	45.2	45.2	45.2	45.2
BART	45.91	45.91	45.91	45.91
T5	47.13	47.13	47.13	47.13
RAG	48.07	48.07	48.07	48.07
BLINK	80.24	91.47	80.24	80.24
		dev		
BART + DPR	44.96	44.96	44.96	44.96
T5	47.35	47.35	47.35	47.35
BART	47.91	47.91	47.91	47.91
RAG	49.0	49.0	49.0	49.0

Table 9: WNED-WIKI

model	**R-Precision**	**Recall@5**	**Accuracy**	**KILT-AC**
		test		
BART + DPR	46.87	46.87	46.87	46.87
RAG	47.61	47.61	47.61	47.61
BART	49.16	49.16	49.16	49.16
T5	49.29	49.29	49.29	49.29
BLINK	68.77	81.78	68.77	68.77
		dev		
BART + DPR	45.7	45.7	45.7	45.7
T5	46.58	46.58	46.58	46.58
RAG	46.7	46.7	46.7	46.7
BART	48.01	48.01	48.01	48.01

Table 10: WNED-CWEB

model	**R-Precision**	**Recall@5**	**Accuracy**	**F1**	**KILT-AC**	**KILT-F1**
			test			
BART	0.0	0.0	45.06	49.24	0.0	0.0
T5	0.0	0.0	43.56	50.61	0.0	0.0
BART + DPR	13.26	17.04	59.16	62.76	11.12	11.41
RAG	28.68	33.04	59.2	62.96	23.12	23.94
			dev			
BART	0.0	0.0	43.84	48.25	0.0	0.0
T5	0.0	0.0	47.24	51.73	0.0	0.0
BART + DPR	13.62	16.93	56.7	60.19	11.56	11.87
RAG	29.26	33.69	61.48	65.03	25.4	26.22

Table 11: T-REx

model	**R-Precision**	**Recall@5**	**Accuracy**	**F1**	**KILT-AC**	**KILT-F1**
			test			
BART	0.0	0.0	9.14	12.21	0.0	0.0
T5	0.0	0.0	9.02	13.52	0.0	0.0
BERT + DPR	40.11	40.11	6.93	37.28	4.47	27.09
BART + DPR	28.9	39.21	30.43	34.47	18.91	20.32
RAG	53.73	59.52	44.74	49.95	36.83	39.91
			dev			
BART	0.0	0.0	3.03	12.61	0.0	0.0
T5	0.0	0.0	1.58	10.8	0.0	0.0
BART + DPR	45.6	58.49	34.96	44.79	29.08	32.85
RAG	65.36	73.07	47.42	57.98	42.64	48.35

Table 12: Zero Shot RE

model	**R-Precision**	**Recall@5**	**EM**	**F1**	**KILT-EM**	**KILT-F1**
			test			
BART	0.0	0.0	21.75	28.69	0.0	0.0
T5	0.0	0.0	19.6	27.73	0.0	0.0
BART + DPR	54.29	65.52	41.27	49.54	30.06	34.72
BERT + DPR	60.66	46.79	38.64	47.09	31.99	37.58
RAG	59.49	67.06	44.39	52.35	32.69	37.91
			dev			
BART	0.0	0.0	26.15	32.06	0.0	0.0
T5	0.0	0.0	25.2	31.88	0.0	0.0
BART + DPR	54.25	64.99	45.05	52.98	31.62	35.84
BERT + DPR	60.03	45.06	42.58	50.43	35.32	39.84
RAG	60.31	65.47	48.78	56.1	36.31	40.64

Table 13: Natural Questions

model	**R-Precision**	**Recall@5**	**EM**	**F1**	**KILT-EM**	**KILT-F1**
			test			
BART	0.0	0.0	15.37	21.97	0.0	0.0
T5	0.0	0.0	12.64	19.57	0.0	0.0
BERT + DPR	25.04	10.4	11.29	17.35	0.74	1.26
BART + DPR	25.04	10.4	25.18	34.07	1.96	2.53
RAG	30.59	12.59	26.97	36.03	3.21	4.1
			dev			
BART	0.0	0.0	16.86	23.81	0.0	0.0
T5	0.0	0.0	12.66	19.74	0.0	0.0
BERT + DPR	24.62	10.7	10.82	16.96	0.96	1.34
BART + DPR	24.62	10.7	25.75	35.2	1.96	2.46
RAG	30.76	12.29	27.68	37.37	3.14	3.87

Table 14: HotpotQA

model	**R-Precision**	**Recall@5**	**EM**	**F1**	**KILT-EM**	**KILT-F1**
			test			
BART	0.0	0.0	32.39	39.85	0.0	0.0
T5	0.0	0.0	18.11	27.83	0.0	0.0
BART + DPR	44.49	56.99	58.55	67.79	31.4	35.34
BERT + DPR	43.4	31.45	70.38	74.41	34.48	36.28
RAG	48.68	57.13	71.27	75.88	38.13	40.15
			dev			
BART	0.0	0.0	32.54	39.58	0.0	0.0
T5	0.0	0.0	25.79	33.72	0.0	0.0
BERT + DPR	40.87	29.96	70.24	74.21	32.9	34.48
BART + DPR	45.36	56.72	59.28	68.31	32.56	36.36
RAG	49.26	56.93	61.73	67.12	36.13	38.71

Table 15: TriviaQA

model	**R-Precision**	**Recall@5**	**Rouge-L**	**F1**	**KILT-RL**	**KILT-F1**
			test			
T5	0.0	0.0	19.08	16.1	0.0	0.0
BART	0.0	0.0	20.55	19.23	0.0	0.0
RAG	11.0	22.92	14.05	14.51	1.69	1.79
BART + DPR	10.67	26.92	17.41	17.88	1.9	2.01
			dev			
T5	0.0	0.0	21.02	18.36	0.0	0.0
BART	0.0	0.0	22.69	22.19	0.0	0.0
RAG	16.39	27.27	16.11	17.24	2.65	2.88
BART + DPR	16.32	21.11	18.53	18.75	2.87	2.89

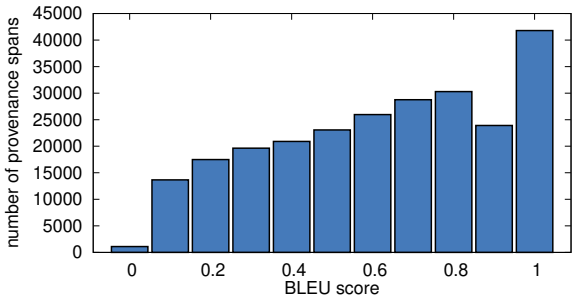
Table 16: ELI5

model	**R-Precision**	**Recall@5**	**Rouge-L**	**F1**	**KILT-RL**	**KILT-F1**
			test			
BART	0.0	0.0	11.84	12.96	0.0	0.0
T5	0.0	0.0	12.58	13.49	0.0	0.0
TransMemNet	18.38	18.38	9.92	11.5	1.83	2.23
BART + DPR	25.48	55.1	13.56	15.55	3.88	4.52
RAG	57.78	74.63	11.83	13.22	8.04	9.1
			dev			
BART	0.0	0.0	12.05	13.35	0.0	0.0
T5	0.0	0.0	12.8	13.28	0.0	0.0
BART + DPR	0.0	0.0	13.23	15.03	0.0	0.0
RAG	46.73	66.61	12.03	13.42	7.01	7.69

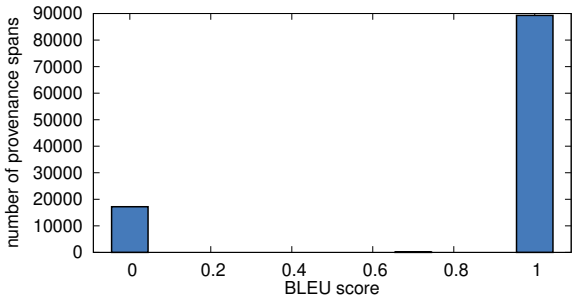
Table 17: Wizard of Wikipedia

```
input: 'SOCCER - UNCAPPED PLAYERS CALLED TO FACE MACEDONIA . '[SE0]'BUCHAREST'[EE0]' 1996-12-06 '[SE1]'Romania'[EE1]' trainer '[SE2]'Anghel Iordanescu'[EE2]' called up three uncapped players on Friday in his squad to face '[SE3]'Macedonia'[EE3]' next week in a '[SE4]'World Cup'[EE4]' qualifier . Midfielder Valentin Stefan and striker '[SE5]'Viorel Ion'[EE5]' of Otelul Galati and defender '[SE6]'Liviu Ciobotariu'[EE6]' of National Bucharest are the newcomers for the '[SE7]'European'[EE7]' group eight clash in '[SE8]'Macedonia'[EE8]' on December 14 . Iordanescu said he had picked them because of their good performances in the domestic championship in which National Bucharest are top and Otelul Galati third . " I think it s fair to give them a chance , " he told reporters . League title-holders Steaua Bucharest , who finished bottom of their Champions League group in the '[SE9]'European Cup'[EE9]', have only two players in the squad . Attacking midfielder '[SE10]'Adrian Ilie'[EE10]', who recently moved from Steaua to Turkish club '[SE11]'Galatasaray'[EE11]', is ruled out after two yellow-card offences . Squad : Goalkeepers - '[SE12]'Bogdan Stelea'[EE12]', '[SE13]'Florin Prunea'[EE13]'. Defenders - '[SE14]'Dan Petrescu'[EE14]', '[SE15]'Daniel Prodan'[EE15]', Anton Dobos , Cornel Papura , '[SE16]'Liviu Ciobotariu'[EE16]', Tibor Selymess , '[SE17]'Iulian Filipescu'[EE17]'. Midfielders - '[SE18]'Gheorghe Hagi'[EE18]', '[SE19]'Gheorghe Popescu'[EE19]', '[SE20]'Constantin Galca'[EE20]' , Valentin Stefan , '[SE21]'Basarab Panduru'[EE21]', '[SE22]'Dorinel Munteanu'[EE22]', Ovidiu Stinga . Forwards - Ioan Vladoiu , '[SE23]'Gheorghe Craioveanu'[EE23]', '[SE24]'Ionel Danciulescu'[EE24]', '[SE25]'Viorel Ion'[EE25]' . REUTER'

BART predictions:
v E0: 'Bucharest' -> https://en.wikipedia.org/wiki/Bucharest
x E1: 'Romania' (gold:'Romania national football team')
x E2: 'Anghel Iordanescu' (gold:'Anghel Iordanescu')
v E3: 'North Macedonia national football team' -> https://en.wikipedia.org/wiki/North_Macedonia_national_football_team
x E4: '1998 FIFA World Cup' (gold:'FIFA World Cup')
v E5: 'Viorel Ion' -> https://en.wikipedia.org/wiki/Viorel_Ion
v E6: 'Liviu Ciobotariu' -> https://en.wikipedia.org/wiki/Liviu_Ciobotariu
v E7: 'Europe' -> https://en.wikipedia.org/wiki/Europe
v E8: 'North Macedonia' -> https://en.wikipedia.org/wiki/North_Macedonia
v E9: 'UEFA Champions League' -> https://en.wikipedia.org/wiki/UEFA_Champions_League
v E10: 'Adrian Ilie' -> https://en.wikipedia.org/wiki/Adrian_Ilie
v E11: 'Galatasaray S.K. (football)' -> https://en.wikipedia.org/wiki/Galatasaray_S.K._(football)
v E12: 'Bogdan Stelea' -> https://en.wikipedia.org/wiki/Bogdan_Stelea
v E13: 'Florin Prunea' -> https://en.wikipedia.org/wiki/Florin_Prunea
v E14: 'Dan Petrescu' -> https://en.wikipedia.org/wiki/Dan_Petrescu
v E15: 'Daniel Prodan' -> https://en.wikipedia.org/wiki/Daniel_Prodan
v E16: 'Liviu Ciobotariu' -> https://en.wikipedia.org/wiki/Liviu_Ciobotariu
v E17: 'Iulian Filipescu' -> https://en.wikipedia.org/wiki/Iulian_Filipescu
v E18: 'Gheorghe Hagi' -> https://en.wikipedia.org/wiki/Gheorghe_Hagi
v E19: 'Gheorghe Popescu' -> https://en.wikipedia.org/wiki/Gheorghe_Popescu
x E20: 'Constantinos Galca' (gold:'Constantin Galca')
v E21: 'Basarab Panduru' -> https://en.wikipedia.org/wiki/Basarab_Panduru
v E22: 'Dorinel Munteanu' -> https://en.wikipedia.org/wiki/Dorinel_Munteanu
v E23: 'Gheorghe Craioveanu' -> https://en.wikipedia.org/wiki/Gheorghe_Craioveanu
x E24: 'Ion Danciulescu' (gold:'Ionel Danciulescu')
v E25: 'Viorel Ion' -> https://en.wikipedia.org/wiki/Viorel_Ion

F1-score = 87.52
KILT-F1-score = 21/26 = 80.77
EM = 21/26 = 80.77
KILT-EM-score = 21/26 = 80.77
```

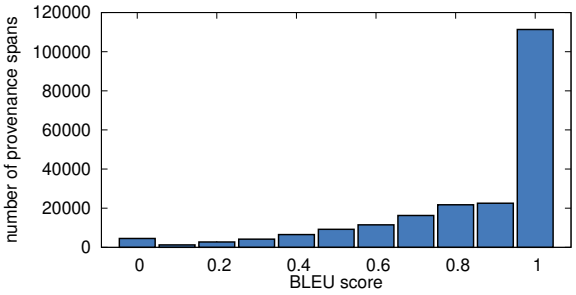
Figure 4: Entity linking BART predictions, schematic of 25 input-output pairs condensed, in each one a single entity in tagged.



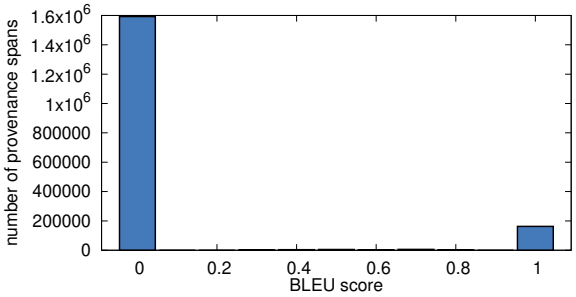
(a) FEVER, dev data discarded 26.03% (3675), test data discarded 27.7% (3869).



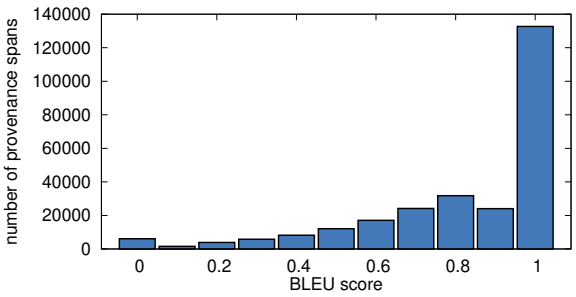
(b) Natural Questions, dev data discarded 16.12% (595), test data discarded 15.59% (287).



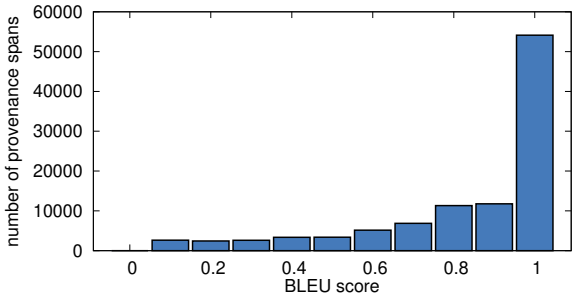
(c) HotpotQA, dev data discarded 22.76% (1650), test data discarded 23.43% (1704).



(d) TriviaQA, dev data discarded 15.06% (950), test data discarded 14.41% (1109).



(e) Zero Shot RE, dev data discarded 15.42% (679), test data discarded 13.38% (767).



(f) Wizard of Wikipedia, dev data discarded 12.06% (469), test data discarded 11.39% (427).

Figure 5: BLEU score distribution in train data per provenance. For TriviaQA, we try to map all object aliases for the answer. FEVER has the oldest Wikipedia snapshot. We discards on average 17.9% dev and 17.65% test data. For TriviaQA there are a large number of 0 scores because we try to map all aliases for the answer and most of the aliases are not found in a Wikipedia page. Note that we consider a QA pair valid if we match at least one alias.