Re²G: Retrieve, Rerank, Generate

Anonymous ACL submission

Abstract

As demonstrated by GPT-3 and T5, transformers grow in capability as parameter spaces become larger and larger. However, for tasks that require a large amount of knowledge, non-parametric memory allows models to grow dramatically with a sub-linear increase in computational cost and GPU memory requirements. Recent models such as RAG and REALM have introduced retrieval into conditional generation. These models incorporate neural initial retrieval from a corpus of passages. We build on this line of research, proposing Re²G, which combines both neural initial retrieval and reranking into a BART-based sequence-to-sequence generation. Our reranking approach also permits merging retrieval results from sources with incomparable scores, enabling an ensemble of BM25 and neural initial retrieval. To train our system end-to-end, we introduce a novel variation of knowledge distillation to train the initial retrieval, reranker and generation using only ground truth on the target sequence output. We find large gains in four diverse tasks: zero-shot slot filling, question answering, fact checking and dialog, with relative gains of 9% to 34% over the previous state-of-the-art on the KILT leaderboard. We make our code available as open source[1].

1 Introduction

GPT-3 [Brown et al., 2020] and T5 [Raffel et al., 2020] are arguably the most powerful members in a family of deep learning NLP models called transformers. Such models store surprising amount of world knowledge. They have been shown to produce good performance on a range of demanding tasks, especially in generating human like texts. However, such large transformers' capability is tied to the increasingly larger parameter spaces on which they are trained.

[1] https://github.com/anonymous

Recently, there has been work towards transformers that make use of non-parametric knowledge. REALM (Retrieval Augmented Language Model) [Guu et al., 2020] and RAG (Retrieval Augmented Generation) [Lewis et al., 2020b] both use an indexed corpus of passages to support conditional generation. By using the corpus as a source of knowledge these models can extend the information available to the model by tens or even hundreds of gigabytes with a sub-linear scaling in computation cost.

These recent advancements, in turn, have been inspired by BART (Bidirectional and Auto-Regressive Transformer) [Lewis et al., 2020a] that combines a Bidirectional Encoder (e.g. BERT [Devlin et al., 2019]) with an Autoregressive decoder (e.g. GPT [Brown et al., 2020]) into one sequence-to-sequence model.

We build on this line of research, pioneered by REALM and RAG, and propose a new approach that we call Re²G (**Re**trieve, **Re**rank, **G**enerate), which combines both neural initial retrieval and reranking into a BART-based sequence-to-sequence generation.

There are two particular aspects on which our approach is different from the previous works. Firstly, our reranking approach permits merging retrieval results from sources with incomparable scores, e.g. enabling an ensemble of BM25 and neural initial retrieval. Secondly, to train our system end-to-end, we introduce a novel variation of knowledge distillation to train the initial retrieval, reranker and generation using only ground truth on the target sequence output.

The KILT benchmark [Petroni et al., 2021] has been recently introduced to evaluate the capabilities of pre-trained language models to address NLP tasks that require access to external knowledge. We evaluate on four diverse tasks from KILT: slot filling, question answering, fact checking and dialog. Figure 1 shows examples of these tasks. Re²G makes significant gains on all four tasks, reaching the top of the KILT leaderboards and establishing a new state-of-the-art.

The contributions of this work are as follows:

- We introduce Re^{2}G, demonstrating the effectiveness of reranking for generative language models that incorporate retrieval.
- We further extend Re^{2}G by ensembling initial retrieval methods, combining neural and traditional keyword-based approaches.
- Re^{2}G improves the current state-of-the-art of 9%, 31%, 34%, 22% and 10% relative gains on the headline KILT metrics for T-REx (slot filling), Natural Questions (question answering), TriviaQA (question answering), FEVER (fact checking), and Wizard of Wikipedia (dialog), respectively.
- We publicly release our code as open source to support continued development.

2 Related Work

The KILT benchmark and public leaderboard[2] combines eleven datasets across five tasks. The main advantage of the KILT distribution of these datasets is that the provenance information from each dataset is realigned to reference the same snapshot of Wikipedia. A unified evaluation script and set of metrics is also provided. In this work, we focus on four tasks, such as Slot Filling [Levy et al., 2017, Elsahar et al., 2018], Question Answering [Kwiatkowski et al., 2019, Joshi et al., 2017], Fact Checking [Thorne et al., 2018a,c], and Dialog [Dinan et al., 2019] (see Figure 1).

A set of baseline methods have been proposed for KILT. GENRE [Cao et al., 2021] is trained on BLINK [Wu et al., 2020] and all KILT tasks jointly using a sequence-to-sequence language model to generate the title of the Wikipedia page where the answer can be found. This method is a strong baseline to evaluate the retrieval performance, but it does not address the downstream tasks. On the other hand, generative models, such as BART [Lewis et al., 2020a] and T5 [Raffel et al., 2020], show interesting performance when fine-tuned on the downstream tasks relying only on the implicit knowledge stored in the weights of the neural networks, without the use of any explicit retrieval component.

RAG [Lewis et al., 2020b], an end-to-end retrieval-based generative model, is the best performing baseline in KILT and it incorporates DPR [Karpukhin et al., 2020] to first retrieve relevant passages for the query, then it uses a model initialized from BART [Lewis et al., 2020a] to perform a sequence-to-sequence generation from each evidence passage concatenated with the query in order to generate the answer. Figure 2 shows the architecture of RAG.

Multi-task DPR [Maillard et al., 2021] exploits multi-task learning by training both DPR passage and query encoder on all KILT tasks. DensePhrases [Lee et al., 2021] addresses the knowledge intensive tasks with a short answer, such as slot filling. It indexes the phrases in the corpus that can be potential answers. The extracted phrases are represented by their start and end token vectors from the final layer of a transformer initialized from SpanBERT [Joshi et al., 2020].

Knowledge Graph Induction (KGI) [Glass et al., 2021] combines DPR and RAG models, both trained with task and dataset specific training. KGI employs a two phase training procedure: first training the DPR model, i.e. both the query and context encoder, using the KILT provenance ground truth. Then, KGI trains the sequence-to-sequence generation and further trains the query encoder using only the target output as the objective. This results in large improvements in retrieval performance and, as a consequence, in the downstream tasks.

Multi-stage or cascade approaches to retrieval have received ample attention in Information Retrieval (IR) research. The multi-stage approach begins with the initial retrieval phase, where an initial set of documents or passages form the pool of candidates to be considered for ranking. Then one or more phases of increasingly computationally demanding rerankers are applied. Early approaches in learning to rank [Liu, 2009] used features and linear classifiers. Pre-trained language models, especially BERT [Devlin et al., 2019], have shown state-of-the-art performance when applied to the task of relevance ranking. Transformers may be applied as classifiers to each query and passage pair independently [Nogueira and Cho, 2019] or as generators to produce labels for passages in a sequence-to-sequence model [Nogueira et al., 2020].

[2] https://eval.ai/web/challenges/challenge-page/689/leaderboard

T-REx

Input: Dracula [SEP] narrative location
Output: Transylvania
Provenance: 7923-2

Natural Questions

Input: when did bram stoker's dracula come out
Output: 1897
Provenance: 7923-1

FEVER

Input: Dracula is a novel by a Scottish author.
Output: REFUTES
Provenance: 7923-1

Dracula (7923)

Dracula is an 1897 Gothic horror novel by Irish author Bram Stoker. It introduced the character of Count Dracula, and established many conventions of subsequent vampire fantasy.

The novel tells the story of Dracula's attempt to move from Transylvania to England so that he may find new blood and spread the undead curse, and of the battle between Dracula and a small group of men and a woman led by Professor Abraham Van Helsing.

Wizard of Wikipedia

Input:

- I really like vampires!!
- Vampires are intense and based on European folklore. Do you have any favorite vampires?
- I think dracula is the best one!!!

Output: He's one of the best! He's based on the character from the 1897 horror book of the same name.
Provenance: 7923-1

Figure 1: KILT tasks of slot filling, question answering, fact checking and dialog

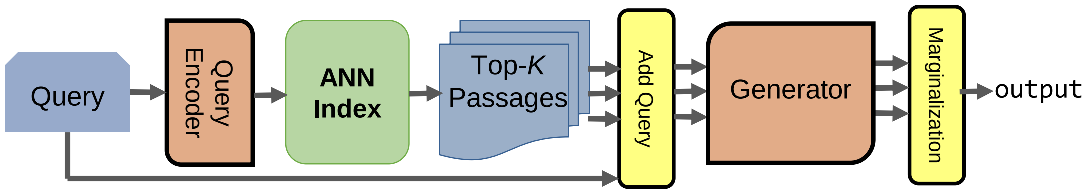

Figure 2: RAG Architecture

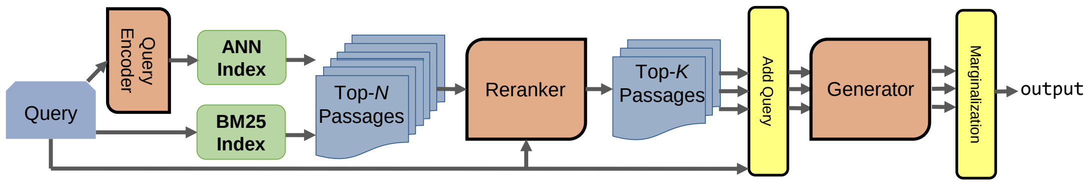

Figure 3: Re^{2}G Architecture

3 Methodology

The approach of RAG, Multi-DPR, and KGI is to train a neural IR (Information Retrieval) component and further train it end-to-end through its impact in generating the correct output. Figure 2 illustrates the end-to-end RAG system.

It has been previously established that results from initial retrieval can be greatly improved through the use of a reranker [Liu, 2009, Wang et al., 2011]. Therefore we hypothesized that natural language generation systems incorporating retrieval can benefit from reranking.

In addition to improving the ranking of passages returned from DPR, a reranker can be used after merging the results of multiple retrieval methods with incomparable scores. For example, the scores returned by BM25 [Robertson and Zaragoza, 2009] are not comparable to the inner products from DPR.

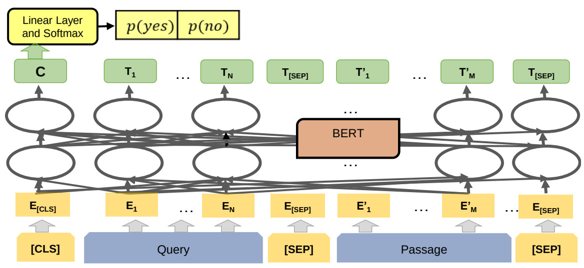

Figure 4: Interaction Model Reranker

Using the scores from a reranker, we can find the top-k documents from the union of DPR and BM25 results. Figure 3 illustrates our extension of RAG with a reranker. We call our system Re^{2}G (**Re**trieve, **Re**rank, **G**enerate).

3.1 Reranker

The reranker we use is based on the sequence-pair classification of Nogueira and Cho [2019]. This

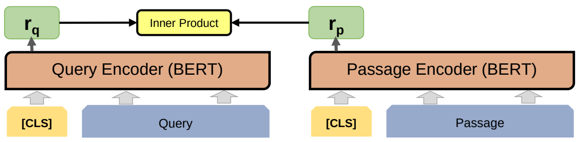

Figure 5: Representation Model for Initial Retrieval

model is shown in Figure 4. The query and passage are input together to a BERT [Devlin et al., 2019] transformer. Cross attention is applied over the tokens of both sequences jointly. This is called an interaction model.

This model contrasts with the representation model used for initial retrieval. Figure 5 shows the bi-encoder representation model for DPR. The representation vectors for the query and passage are produced independently. This allows for efficient retrieval by pre-computing vectors for all passages in the corpus and indexing them with an ANN (Approximate Nearest Neighbors) index. By using an interaction model to rerank the top-N passages from the representation model, we can get the advantages of both model types: accuracy and scalability.

We initialize the reranker from the BERT model trained on MS MARCO [Nguyen et al., 2016] by NBoost [Thienes and Pertschuk, 2019] and available through Hugging Face[3].

3.2 Training

As Figure 1 illustrates, KILT tasks are provided with two types of ground truth: the target output sequence and the provenance information indicating the passage or passages in the corpus that support the output.

Our training is carried out in four phases: DPR training, generation training, reranking training, and full end-to-end training. The initial DPR and reranking phases make use of the provenance ground truth. The generation and full end-to-end training make use of only the target output.

Formally:

- The original KILT instances are a tuple: $\langle q, t, \mathbf{Prov} \rangle$ where q is the input or prompt, t is the target output, and $\mathbf{Prov}$ is the set of provenance passages that support the target output.
- DPR training is a tuple: $\langle q, p^+, p^- \rangle$ where $p^+ \in \mathbf{Prov}$ and p^- where $p^- \in \text{BM25}(q) \wedge p^- \notin \mathbf{Prov}$
- Reranking training begins with the application of DPR and BM25, producing tuples: $\langle q, \mathbf{P}, \mathbf{Prov} \rangle$ where $\mathbf{P} = \text{BM25}(q) \cup \text{DPR}(q)$
- Generation and end-to-end training instances are pairs of query and target: $\langle q, t \rangle$

The first two phases, DPR and generation, are identical to KGI, specifically KGI_0. We use the codes from Glass et al. [2021][4].

DPR Stage 1 training is the same training used by Karpukhin et al. [2020]. The triplets of query, positive passage and "hard negative" passages from BM25 are put into batches of 128 instances. The positives and hard negatives from other instances form the "batch negatives" for each instance. The DPR bi-encoder model gives each query a probability distribution over the positive, hard negative, and batch negatives. The loss is the negative log-likelihood for the positive. After DPR Stage 1 training the passages from the corpus are indexed with a Hierarchical Navigable Small World (HNSW) [Malkov and Yashunin, 2018] using FAISS [Johnson et al., 2017].

Generation training extends the training of the query encoder and trains the $\text{BART}_{\text{LARGE}}$ sequence-to-sequence model on the target sequence output. This training is the same as that described by Lewis et al. [2020b].

3.3 Reranking Training

The next phase, training the reranking in isolation, begins with gathering the initial retrieval results from DPR and BM25 on the training set. These results are merged and used as training data for the reranker.

In some datasets there are multiple positive passages. Therefore, we use the negative of the summed log-likelihood for the positive passages as the loss function. The logits given by the reranker are $\mathbf{z_r}$ and the indices for the correct passages (from the ground truth provenance) are $\mathbf{Prov}$.

$$loss = -\sum_{i \in \mathbf{Prov}} log(softmax(\mathbf{z_r})_i)$$

[3] https://huggingface.co/nboost/pt-bert-base-uncased-msmarco

[4] https://github.com/IBM/kgi-slot-filling

	T-REx					(Slot Filling)
	R-Prec	**Recall@5**	**Accuracy**	**F1**	**KILT-AC**	**KILT-F1**
Re^{2}G (ours)	**80.70**	**89.00**	**87.68**	**89.93**	**75.84**	**77.05**
KGI$_1$ [Glass et al., 2021]	74.36	83.14	84.36	87.24	69.14	70.58
KILT-WEB 2 (anonymous)	71.86	84.76	82.20	85.28	62.92	64.60
KGI$_0$ [Glass et al., 2021]	59.70	70.38	77.90	81.31	55.54	56.79
DensePhrases [Lee et al., 2021]	37.62	40.07	53.90	61.74	27.84	32.34

	Natural Questions					(Question Answering)
	R-Prec	**Recall@5**	**Accuracy**	**F1**	**KILT-AC**	**KILT-F1**
Re^{2}G (ours)	**70.78**	**76.63**	**51.73**	**60.97**	**43.56**	**49.80**
RAG [Petroni et al., 2021]	59.49	67.06	44.39	52.35	32.69	37.91
BERT+DPR [Petroni et al., 2021]	60.66	46.79	38.64	47.09	31.99	37.58
BART+DPR [Petroni et al., 2021]	54.29	65.52	41.27	49.54	30.06	34.72
MultiDPR [Maillard et al., 2021]	59.42	68.24	39.75	48.43	29.09	34.70

	TriviaQA					(Question Answering)
	R-Prec	**Recall@5**	**Accuracy**	**F1**	**KILT-AC**	**KILT-F1**
Re^{2}G (ours)	**72.68**	**74.23**	**76.27**	**81.40**	**57.91**	**61.78**
MultiDPR [Maillard et al., 2021]	61.49	68.33	59.60	66.53	42.36	46.19
RAG [Petroni et al., 2021]	48.68	57.13	71.27	75.88	38.13	40.15
BERT+DPR [Petroni et al., 2021]	43.40	31.45	70.38	74.41	34.48	36.28
BART+DPR [Petroni et al., 2021]	44.49	56.99	58.55	67.79	31.40	35.34

	FEVER			(Fact Checking)
	R-Prec	**Recall@5**	**Accuracy**	**KILT-AC**
Re^{2}G (ours)	**88.92**	**92.52**	**89.55**	**78.53**
KGI$_0$ [Glass et al., 2021]	75.60	84.95	85.58	64.41
MultiDPR [Maillard et al., 2021]	74.48	87.52	86.32	63.94
RAG [Petroni et al., 2021]	61.94	75.55	86.31	53.45
GENRE [Cao et al., 2021]	83.64	88.15	0.00	0.00

	Wizard of Wikipedia					(Dialog)
	R-Prec	**Recall@5**	**Rouge-L**	**F1**	**KILT-RL**	**KILT-F1**
Hindsight [Paranjape et al., 2021]	56.08	74.27	**17.06**	**19.19**	**11.92**	**13.39**
Re^{2}G (ours)	60.10	**79.98**	16.76	18.90	11.39	12.98
KGI$_0$ [Glass et al., 2021]	55.37	78.45	16.36	18.57	10.36	11.79
RAG [Petroni et al., 2021]	57.75	74.61	11.57	13.11	7.59	8.75
MultiDPR [Maillard et al., 2021]	41.06	67.13	13.27	15.12	5.91	6.96
GENRE [Cao et al., 2021]	**62.88**	77.74	0.00	0.00	0.00	0.00

Table 1: KILT leaderboard top systems

3.4 End-to-End Training

Training end-to-end poses a special challenge. In RAG, the gradient propagates to the query encoder because the inner product between the query vector and the passage vector is used to weight the influence of each sequence, a process RAG calls marginalization. The inputs to the BART model are sequences ($s_j = p_j$ [SEP] q) that comprise a query q plus retrieved passage p_j. The probability for each sequence is determined from the softmax over the retrieval (or reranker) scores for the passage. The probability for each target token t_i given the sequence s_j is a softmax over BART's token prediction logits. The loss therefore is a negative log-likelihood summed over all target tokens and sequences, weighted by each sequence's probability.

Consider that in Re^{2}G the score from the reranker, not the initial retrieval, is used to weight the impact of each sequence in generation. This allows the reranker to be trained through the ground truth on target output, but it means the gradient for the query encoder will be zero since the marginalization no longer depends on the inner product from the query and passage representation vectors.

$$P(s_j) = softmax(\mathbf{z_r})_j$$
$$P(t_i|s_j) = softmax(\text{BART}(s_j)_i)_{t_i}$$
$$loss = -\sum_{i,j} log\left(P(t_i|s_j) \cdot P(s_j)\right)$$

We consider three possible resolutions to this issue.

- Combine the DPR and reranker scores
- Freeze the query encoder
- Online Knowledge Distillation

The first candidate solution is tempting but fatally flawed. By adding the log softmax from DPR and the reranker we can ensure that both systems are trained through impact in generation. However, if the DPR score is added to the reranker score, then the DPR score is being trained to provide a complementary signal to the reranker. Therefore, when DPR is used to gather the candidate passages, it does not give the highest scores to the passages that are most likely to be relevant, but instead gives the highest scores to the passages the reranker is most likely to underrate. We find that this theoretical concern is also a practical concern, as DPR performance (and overall system performance) declines greatly when trained in this way.

The simplest solution is to freeze the parameters of the query encoder, training only the reranker and generation components. We find this is indeed the best solution for one of our datasets, Wizard of Wikipedia. Note that DPR has already been trained in two phases, first from the provenance ground truth and then again in generation training in the RAG model.

The third solution is our novel application of knowledge distillation [Hinton et al., 2015]. We use the reranker as a teacher model to provide labels to the DPR student model. We distill the knowledge across architectures: from an interaction model to a representation model. Further, this knowledge distillation occurs online, while the reranker is being trained. The loss for the initial retrieval is therefore the KL-divergence between the probability distribution it gives over the retrieved passages and the reranker's probability distribution over the same passages. A temperature hyperparameter T smooths these distributions to prevent excessive loss and stabilize training.

$$loss = D_{KL}\left(softmax\left(\frac{\mathbf{z_s}}{T}\right)\middle\|softmax\left(\frac{\mathbf{z_t}}{T}\right)\right) \cdot T^2$$

The knowledge distillation has the usual advantage of providing signal not only of positive and negative instances, but degrees of negativeness. In addition, since we retrieve $n = 12$ passages from DPR but only use the top-k ($k = 5$) for generation, the knowledge distillation loss is providing a (soft) label for more passages.

3.5 Inference

At inference time the query is encoded using the DPR query encoder and the top-12 passages from the HNSW index are returned. The query is also passed to BM25 search, specifically Anserini[5], gathering the top-12 BM25 results. Both sets of passages are passed to the reranker and scored. The top-5 passages are then joined with the query and passed to $\text{BART}_{\text{LARGE}}$ to generate the output. The five output sequences are weighted according to the softmax over the reranker scores to produce the final output.

4 Experiments

We test our model on five datasets, over four distinct tasks in the KILT benchmark: slot filling, question answering, fact checking and dialog. Figure 1 shows an example of these four tasks.

The slot filling dataset, T-REx [Elsahar et al., 2018], provides as input a head entity and relation, and expects as output the entity or term that fills the slot, also called the tail entity. The T-REx dataset contains 2.3M instances. We use only 370k training instances by downsampling the relations that occur more than 5000 times. This reduces the training time required while keeping state-of-the-art performance. The development and test sets each have 5k instances.

The question answering datasets are "open" versions of Natural Questions [Kwiatkowski et al., 2019] and TriviaQA [Joshi et al., 2017]. Unlike the original versions, the relevant Wikipedia page must be found by a retrieval step. The training sets for Natural Questions and TriviaQA contain 87k and 62k questions, with another 3k and 5k for the development and 1.4k and 6.5k for test.

The fact checking dataset in KILT is FEVER (Fact Extraction and VERification). It is a combination of the two FEVER versions [Thorne

[5]https://github.com/castorini/anserini

et al., 2018b, 2019] omitting the NOTENOUGHINFO class. There are approximately 10k instances in the development and test sets, and 100k for training. FEVER is a classification task, but we cast it as a generation task by training the model to generate either the token "SUPPORTS" or "REFUTES".

Wizard of Wikipedia [Dinan et al., 2018] is the dialog dataset. The input is a short dialog history ending with the information seeker's turn. The expected output is a fact presented conversationally or just an utterance or question mentioning content from a relevant Wikipedia page. It is the smallest dataset with approximately 3k instances in development and test and 64k in train.

For all tasks, systems are expected to produce the target output as well as justify it with provenance information from the KILT knowledge source. The metrics of R-Precision and Recall@5 measure the correctness of the provenance. R-Precision measures what fraction of the R documents in the ground truth provenance ($|\mathbf{Prov}| = R$) are present in the top-R documents returned by the system. Accuracy and (token-level) F1 measure the correctness of the generated output. For Wizard of Wikipedia, Rouge-L [Lin, 2004] is used instead of accuracy, since systems are very unlikely to generate the exact target output. The metrics of KILT-Accuracy, KILT-F1 and, for Wizard of Wikipedia, KILT-Rouge-L are the underlying metric (e.g. Accuracy) for instances where R-Precision is one, otherwise zero. These metrics indicate output correctness when provenance is also correctly supplied.

Table 1 shows the performance of Re^2G on the KILT leaderboard. Except for Wizard of Wikipedia where it is now second best, Re^2G is the best on all metrics across all five datasets attempted. We achieve 9%, 31%, 34%, 22% and 10% relative gains over the previous state-of-the-art on the headline KILT metrics for T-REx, Natural Questions, TriviaQA, FEVER, and Wizard of Wikipedia, respectively.

The closest competition in retrieval metrics is GENRE. This system, as described in Section 2, uses a Wikipedia-specific approach to retrieval: generating the title of the Wikipedia page as in an entity-linking task. In contrast our system can be applied to any corpus and provides passage-level granularity.

Since our submission to the KILT leaderboard for the Wizard of Wikipedia, a new system called Hindsight [Paranjape et al., 2021] achieved even better results on the generation metrics on that particular task.

4.1 Ablation Study

To understand the impact of different components we ran ablations of Re^2G over each of the five datasets. We considered a variant that eliminates the online knowledge distillation, and a variant that removes results from BM25, using 24 DPR results rather than 12 from both DPR and BM25.

These variants performed worse in four out of five datasets. Online knowledge distillation failed to improve for Wizard of Wikipedia and ensembling with BM25 failed to improve for Natural Questions. More details are found in the appendix.

Table 2 examines how the retrieval improves through each step of training. In the first half of the table we consider the initial retrieval alone. DPR Stage 1 is the DPR training described earlier - training only from the provenance ground truth with batch negatives and hard negatives from BM25. KGI_0 further trains the query encoder of DPR Stage 1 through its impact in generating the target output. Finally Re^2G extends the training of DPR with online knowledge distillation from the reranker. This step is beneficial in two of the three datasets, while the previous steps improve performance across all datasets.

In the second half of the table we examine the improvement in reranking. The baseline of KGI_0 DPR+BM25 merges the results of KGI_0's DPR and BM25 by scoring each passage by the sum of the inverse rank from each method. For both T-REx and FEVER, even this simple approach to ensembling DPR and BM25 improves Recall@5, although not R-Precision. Following reranker training using the provenance ground truth (Reranker Stage 1), we find improvement over DPR across all five datasets on both retrieval metrics. The reranker's improvement following end-to-end training is mixed. In FEVER and Wizard of Wikipedia there is substantial gain in R-Precision, approximately 2%. T-REx and Natural Questions are flat. However, there is a sharp decline in the performance of TriviaQA, in retrieval metrics. This is true despite the fact that retrieving these passages greatly improves answer accuracy and F1. This suggests some incompleteness in the provenance ground truth for TriviaQA.

4.2 Analysis

More details about all the analysis described below can be found in Appendix E.

	T-REx		NQ		TriviaQA		FEVER		WoW	
	R-Prec	**R@5**	**R-Prec**	**R@5**	**R-Prec**	**R@5**	**R-Prec**	**R@5**	**R-Prec**	**R@5**
BM25	46.88	69.59	24.99	42.57	26.48	45.57	42.73	70.48	27.44	45.74
DPR Stage 1	49.02	63.34	56.64	64.38	60.12	64.04	75.49	84.66	34.74	60.22
KGI_0 DPR	65.02	75.52	64.65	69.60	60.55	63.65	80.34	86.53	**48.04**	**71.02**
Re^2G DPR	**67.16**	**76.42**	**65.88**	**70.90**	**62.33**	**65.72**	**84.13**	**87.90**	47.09	69.88
KGI_0 DPR+BM25	60.48	80.06	36.91	66.94	40.81	64.79	65.95	90.34	35.63	68.47
Reranker Stage 1	81.22	87.00	70.78	73.05	**71.80**	**71.98**	87.71	92.43	55.50	**74.98**
Re^2G Reranker	**81.24**	**88.58**	**70.92**	**74.79**	60.37	70.61	**90.06**	**92.91**	**57.89**	74.62

Table 2: Development Set Results for Retrieval

4.2.1 Analysis of gains

Since the Re^2G model differs from the KGI model only in the retrieval phase, we hypothesized that its gains in output quality are driven by its better retrieval quality. To test this hypothesis we considered all cases where the Re^2G model produces better output than the KGI_0 model and calculated the fraction of such cases where Re^2G's rank for the first correct passage is lower than KGI_0's.

We find that for T-REx, NQ, and FEVER the fractions of output gains that could be attributed to improved retrieval and ranking are 67.73%, 61.08% and 66.86% respectively. While for TriviaQA and Wizard of Wikipedia only 36.86% and 27.74% of output improvements were accompanied by improved ranking for the correct passage. It is important to note that in Wizard of Wikipedia, many of these improved outputs have only a small gain in token-level F1.

While much of the gain in output quality is attributable to improved recall, at least a third is not. This reinforces an observation of Glass et al. [2021], that models trained with better retrieval can produce better output even when the retrieved passages are equivalent at test time.

4.2.2 Slot filling error analysis

To understand the types of errors Re^2G makes we sampled 50 instances of the development set of the T-REx dataset where the Accuracy and token-level F1 score was zero.

Interestingly, the most common class of error (33/50) was due to the incompleteness of the ground truth. Often the head entity is ambiguous (19/50), or the relation has multiple fillers (16/50). As an example, consider the following where there are two *Joe O'Donnell* notable for sports in the passages retrieved, and each played for at least two different teams.

Joe O'Donnell [SEP] member of sports team

Target: Buffalo Bills
Re^2G: Dumbarton F.C.

- Joe O'Donnell (footballer) / Joe O'Donnell (footballer) Joseph 'Joe' O'Donnell (born 3 March 1961) was a Scottish footballer who played for **Dumbarton** and Stranraer.
- Joe O'Donnell (American football) / ... fullback, guard and tackle for the University of Michigan from 1960 to 1963. He also played professional football as a guard and tackle for eight seasons for the **Buffalo Bills**...

When Re^2G produces genuine errors it is usually because it has selected some entity as a filler related in a different way (6/17) or it has failed to retrieve the necessary passage (9/17).

5 Conclusions

Relative to previous work, such as RAG or KGI, Re^2G substantially improves both in retrieval and end-to-end performance on slot filling, question answering, fact checking, and dialog. The reranker alone improves performance and enables the inclusion of multiple sources of initial retrieval. This architecture permits us to integrate results from BM25, further improving in accuracy. Our online knowledge distillation is able to improve the performance of DPR in four of the five datasets, despite the loss in end-to-end training not depending on the DPR scores. We have directed our efforts towards improving the retrieval of relevant knowledge. This also enables improvement in end-to-end performance by supplying better passages to the generation component. Further experiments on domain adaptation of Re^2G on tasks like question answering or dialog might provide useful insight on the application of this technology to real world use cases. We are releasing our source code as open source (Apache 2.0 license) to enable further research.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *NeurIPS*, 2020.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. Autoregressive entity retrieval. In *International Conference on Learning Representations*. OpenReview.net, 2021. URL https://openreview.net/forum?id=5k8F6UU39V.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. Wizard of wikipedia: Knowledge-powered conversational agents. In *International Conference on Learning Representations*, 2018.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. Wizard of wikipedia: Knowledge-powered conversational agents. In *ICLR (Poster)*. OpenReview.net, 2019.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. T-REx: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA). URL https://aclanthology.org/L18-1544.

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, and Alfio Gliozzo. Robust retrieval augmented generation for zero-shot slot filling. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021. URL https://aclanthology.org/2021.emnlp-main.148.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Realm: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*, 2020.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015. URL http://arxiv.org/abs/1503.02531.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*, 2017.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 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, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020. doi: 10.1162/tacl_a_00300. URL https://aclanthology.org/2020.tacl-1.5.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 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, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.550. URL https://aclanthology.org/2020.emnlp-main.550.

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. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, March 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026.

Jinhyuk Lee, Mujeen Sung, Jaewoo Kang, and Danqi Chen. Learning dense representations of phrases at scale. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6634–6647, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.518. URL https://aclanthology.org/2021.acl-long.518.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 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, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/K17-1034. URL https://aclanthology.org/K17-1034.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 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, July 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://aclanthology.org/2020.acl-main.703.

Patrick 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. Retrieval-augmented generation for knowledge-intensive nlp tasks. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc., 2020b.

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

Tie-Yan Liu. Learning to rank for information retrieval. *Information Retrieval*, 3(3):225–331, 2009.

Jean Maillard, Vladimir Karpukhin, Fabio Petroni, Wen-tau Yih, Barlas Oguz, Veselin Stoyanov, and Gargi Ghosh. Multi-task retrieval for knowledge-intensive tasks. In *ACL/IJCNLP (1)*, pages 1098–1111. Association for Computational Linguistics, 2021.

Yu A Malkov and Dmitry A Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*, 42(4):824–836, 2018.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. Ms marco: A human generated machine reading comprehension dataset. In *CoCo@ NIPS*, 2016.

Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*, 2019.

Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. Document ranking with a pre-trained sequence-to-sequence model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 708–718, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.63. URL https://aclanthology.org/2020.findings-emnlp.63.

Ashwin Paranjape, Omar Khattab, Christopher Potts, Matei Zaharia, and Christopher D Manning. Hindsight: Posterior-guided training of retrievers for improved open-ended generation. *arXiv preprint arXiv:2110.07752*, 2021.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.200. URL https://aclanthology.org/2021.naacl-main.200.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2020.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, April 2009. ISSN 1554-0669. doi: 10.1561/1500000019. URL http://dx.doi.org/10.1561/1500000019.

Cole Thienes and Jack Pertschuk. Nboost: Neural boosting search results. https://github.com/koursaros-ai/nboost, 2019.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and verification. In *NAACL-HLT*, pages 809–819. Association for Computational Linguistics, 2018a.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 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, 2018b.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. The fact extraction and verification (FEVER) shared task. *CoRR*, abs/1811.10971, 2018c.

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

Lidan Wang, Jimmy Lin, and Donald Metzler. A cascade ranking model for efficient ranked retrieval. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 105–114, 2011.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 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, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.519. URL https://aclanthology.org/2020.emnlp-main.519.

Appendix

A Hyperparameters

We have not done hyperparameter tuning for DPR Stage 1, Generation, or Reranking training. Instead we used hyperparameters similar to the original works on training DPR, BERT reranking and RAG. Table 3 shows the hyperparameters used in our experiments.

For knowledge distillation we used the same hyperparameter settings as Generation. For the additional hyperparameters in online knowledge distillation: temperature and KD learn rate scaling, we experimented with temperatures of 10 and 40 and KD learn rate scaling of 1.0 and 0.1. For our reported results we used a temperature of 10.0 and a learn rate scaling of 1.0.

When training using online knowledge distillation, there is a separate optimizer for the query encoder while training generation. This optimizer uses the same hyperparameter settings.

Table 5 shows the settings for retrieval and generation used for all datasets.

All results are from a single run. The random seed for python, numpy and pytorch was 42.

B Software Details

We used the following software versions:

- Ubuntu 18
- Pytorch 1.7
- Transformers 4.3.2
- Anserini 0.4.1
  (commit
  3a60106fdc83473d147218d78ae7dca7c3b6d47c)

C Model Details

Number of parameters Re^{2}G uses three BERT$_{BASE}$ transformers: query encoder, passage encoder and reranker. Each has $110M$ parameters. The generation component is a BART$_{LARGE}$ model with $400M$ parameters. There are $730M$ parameters in total.

Computing infrastructure Using a single NVIDIA V100 GPU DPR training of two epochs takes approximately 24 hours for T-REx and less than 12 hours for FEVER and WoW.

Using a two NVIDIA P100 GPUs generation training for 370k T-REx instances takes two days, while FEVER and WoW training completes in half a day.

The FAISS index on the KILT knowledge source requires a machine with large memory, we use machines with 128GB of memory.

D Ablations

Table 6 explores ablations of the Re^{2}G system. The point estimates and 95% confidence intervals are reported. Re^{2}G-KD excludes the online knowledge distillation, instead freezing the query encoder when training the reranker and generator during end-to-end training. Re^{2}G-BM25 excludes BM25 results, fetching 24 passages from DPR rather than 12 from DPR and 12 from BM25. The passages are still reranked. KGI$_0$ is the baseline system, without a reranker and therefore also without BM25 results or online knowledge distillation during training.

The FEVER dataset shows the simplest pattern where each component: reranking, including BM25 results, and online knowledge distillation all produce gains, although these gains do not reach significance for online knowledge distillation. In T-REx and Wizard of Wikipedia the impact of reranking and including BM25 results is still clear, but the online knowledge distillation has mixed and non-significant impact on the metrics. For FEVER and Wizard of Wikipedia most of the gain comes from including the reranker on DPR results. However, for T-REx, incorporating BM25 produces the largest gain.

E Generation Analysis

We examined 20 instances coupled with 3 output texts: the baseline KGI$_0$, Re^{2}G, and the target text in the ground-truth. The three output texts were presented unlabeled and in random order to

Hyperparameter	DPR	Reranker	Generation
learn rate	5e-5	3e-5	3e-5
batch size	128	32	128
epochs	2	1	1*
warmup instances	0	10%	10%
learning schedule	linear	triangular	triangular
max grad norm	1	1	1
weight decay	0	0	0
Adam epsilon	1e-8	1e-8	1e-8

Table 3: Re^2G hyperparameters

avoid bias. For each instance, we read the conversation history and then mark each text either `GOOD`, `OK` or `INCONSISTENT` generation. To our surprise, 5/20 ground-truth target texts are `INCONSISTENT` which indicates the WoW benchmark might have limitations in annotation quality. Both the systems have similar results (`GOOD/OK/INCONSISTENT` - Re^2G: 8/2/10; KGI_0: 9/2/9).

Second, we checked a set of 20 WoW instances where Re^2G's F1 score was in the bottom quintile. The conversation history was presented along with Re^2G generated text and the passages retrieved. Manual examination showed 8/20 as `INCONSISTENT` and in 4/8 cases supporting ground-truth passages were not retrieved. Below is one of the 12/20 cases where Re^2G generated text was found `CONSISTENT` with respect to the conversation history, although it has low F1 and Rouge-L scores.

Conversation History:

- My favorite color is red.
- Red is at the end of the spectrum of light, its with orange and opposite of violet.
- I didn't know that. What else do you know about red?

Target: It's actually a primary color for the RGB and CMYK color model.
Re^2G: It has a dominant wavelength of approximately 625-740 nanometres.

Hyperparameter	Value
type	IndexHNSWSQ
m	128
ef search	128
ef construction	200
index batch size	100000
scalar quantizer	8

Table 4: FAISS index hyperparameters

Hyperparameter	Value
DPR passages	12
BM25 passages	12
BART sequences	5
BART beam size	6
BART length penalty	1.0
BART minimum length	2
BART maximum length	64

Table 5: Inference hyperparameters

E.1 Generation Quality

Table 7 shows couple of examples that were part of the set of randomly selected instances from WoW dataset and used for manual inspection. We choose these two particular instances to show when we thought the ground truth (i.e. target) is not coherent with respect to the corresponding conversation history.

In the first example, the system generated outputs were judged as coherent. We found that both Re^2G and KGI_0 retrieved the following passage which might have helped generation of the above output -

Horseshoe Falls / Horseshoe Falls Horseshoe Falls, also known as Canadian Falls, is the largest of the three waterfalls that collectively form Niagara Falls on the Niagara River along the Canada–United States border. Approximately 90% of the Niagara River, after diversions for hydropower generation, flows over Horseshoe Falls. The remaining 10% flows over American Falls and Bridal Veil Falls. It is located between Terrapin Point on Goat Island in the US state of New York, and Table Rock in the

			T-REx			(Slot Filling)
	R-Prec	**Recall@5**	**Accuracy**	**F1**	**KILT-AC**	**KILT-F1**
Re^{2}G	81.24±1.08	88.58±0.84	86.60±0.94	89.20±0.81	75.66±1.19	77.08±1.15
Re^{2}G-KD	81.08±1.09	88.84±0.83	87.00±0.93	89.46±0.80	75.72±1.19	77.00±1.15
Re^{2}G-BM25	71.92±1.25	78.67±1.10	79.48±1.12	82.52±1.00	66.58±1.31	67.93±1.28
KGI$_0$	65.02±1.32	75.52±1.16	77.52±1.16	80.91±1.03	60.18±1.36	61.38±1.34

			Natural Questions			(Question Answering)
	R-Prec	**Recall@5**	**Accuracy**	**F1**	**KILT-AC**	**KILT-F1**
Re^{2}G	70.92±1.67	74.79±1.27	46.70±1.84	62.44±1.65	39.23±1.80	50.90±1.76
Re^{2}G-KD	69.72±1.69	73.73±1.30	46.56±1.84	61.68±1.67	38.24±1.79	49.93±1.76
Re^{2}G-BM25	70.88±1.67	74.39±1.28	46.70±1.84	61.98±1.66	39.41±1.80	50.91±1.76
KGI$_0$	64.65±1.76	69.60±1.39	40.50±1.81	55.07±1.71	32.96±1.73	42.87±1.75

			TriviaQA			(Question Answering)
	R-Prec	**Recall@5**	**Accuracy**	**F1**	**KILT-AC**	**KILT-F1**
Re^{2}G	72.01±1.20	73.16±0.98	74.01±1.17	80.86±0.99	56.04±1.33	60.91±1.27
Re^{2}G-KD	72.01±1.20	73.16±0.98	73.80±1.18	80.62±1.00	56.04±1.33	60.84±1.28
Re^{2}G-BM25	71.10±1.21	68.60±1.03	68.59±1.24	76.68±1.08	52.85±1.34	58.37±1.29
KGI$_0$	61.13±1.31	63.12±1.08	60.68±1.31	66.61±1.20	44.00±1.33	47.35±1.31

			FEVER			(Fact Checking)
	R-Prec	**Recall@5**	**Accuracy**		**KILT-AC**	
Re^{2}G	90.06±0.53	92.91±0.47	91.05±0.55		80.56±0.76	
Re^{2}G-KD	89.85±0.54	92.48±0.48	90.78±0.55		80.14±0.77	
Re^{2}G-BM25	88.36±0.57	88.46±0.59	90.63±0.56		78.74±0.78	
KGI$_0$	80.34±0.73	86.53±0.63	87.84±0.63		70.06±0.88	

			Wizard of Wikipedia			(Dialog)
	R-Prec	**Recall@5**	**Rouge-L**	**F1**	**KILT-RL**	**KILT-F1**
Re^{2}G	56.48±1.76	74.00±1.56	17.29±0.52	19.35±0.57	11.37±0.58	12.75±0.63
Re^{2}G-KD	57.89±1.75	74.62±1.54	17.26±0.52	19.39±0.57	11.61±0.58	13.14±0.64
Re^{2}G-BM25	55.83±1.76	72.72±1.58	17.15±0.51	19.17±0.56	11.13±0.57	12.52±0.63
KGI$_0$	48.04±1.77	71.02±1.61	16.75±0.48	19.04±0.53	9.48±0.53	10.74±0.59

Table 6: Development Set Results for Re^{2}G Variations

> *Canadian province of Ontario. Section: International border.*

As for the ground truth, we marked it (factually) inconsistent based on the following retrieved passage -

> *Niagara Falls / Located on the Niagara River, which drains Lake Erie into Lake Ontario, **the combined falls** have the highest flow rate of any waterfall in North America that has a vertical drop of more than . During peak daytime tourist hours, more than 168,000 m (**six million cubic feet**) of water goes over the crest of **the falls** every minute. Horseshoe Falls is the most powerful waterfall in North America, as measured by flow rate.*

In the second example, all three texts were marked inconsistent. Interestingly, all the items in the conversation contains subjective opinion. Consequently, all the three candidate texts also contains subjective opinion. The problem is both the systems generated texts that are almost repetition of earlier conversation. In case of the ground truth, we find that the text is semantically incoherent.

We have also submitted files that contain all instances that were used to generate the different analysis reported in Section 4.2 of the paper. These files also contains our annotations/remarks where applicable.

Conversation history: • Have you ever been to Niagra Falls? I want to go there, but I know it's very far. Those beautiful 3 waterfalls are right between New York and Ontario, thousands of miles from me! • I never have but I would love to. It looks beautiful. • Yes, they do! They are different sized waterfalls, named Horseshoe, American and Bridal Veil Falls. • I had no idea. I've never heard that before. Very cool. • Yes, and Horseshoe Falls in actually the most powerful waterfall in all of the US! • I never new any details about it. I have just seen it in pictures and on TV and it looks very neat. **Re^{2}G:** Horseshoe falls, also known as Canadian Falls, is the largest of the three waterfalls that collectively form Niagara Falls on the Niagara River along with American Falls and Bridal Veil Falls. `[GOOD]`
KGI$_0$: Horseshoe falls is also known as Canadian Falls, which straddles the international border between Canada and the U.S. `[GOOD]`
Ground truth: It's really quite interesting and beautiful. There's over six million cubic feet of water that crosses the fall's crest every 60 seconds. Can you imagine? `[INCONSISTENT - 6M cubic feet of water crosses for the 3 waterfalls combined, not just Horseshoe falls.]`
Conversation history: • I love the taste of pasta mixed with tomato sauce and shredded cheese all over it • I just love mines with chunky tomatoes mixed with meatballs and mushrooms. • Olive Garden is the place that makes the best pastas. I try to visit the place as much I can with my friends • I love Olive Garden especially the original one in Orlando they opened in 1982. • I think Italians have special genes that makes them so adept at preparing pastas • I believe so as well I would love to visit Argentina that is where Italian-born people reside great pasta I bet. • I don't know how they do it, but no matter how hard I try to prepare one at home, mine ends up with some peculiar taste **Re^{2}G:** I don't know how they do it but I love the taste of pasta mixed with tomato sauce and shredded cheese all over it `[INCONSISTENT - Dialogue repetition]`
KGI$_0$: I just love the taste of pasta mixed with tomato sauce and shredded cheese all over it `[INCONSISTENT - Dialogue repetition]`
Ground truth: The taste is so full of an abundance of taste it just can't be copied lol. `[INCONSISTENT - Semantically incoherent text "taste is full of .. taste".]`

Table 7: Some of the randomly selected instances from WoW benchmark that were evaluated by a human evaluator. The comments inside `[..]` are the feedback provided by the evaluator.