Knowledge Transfer from Answer Ranking to Answer Generation

Matteo Gabburo[1*], **Rik Koncel-Kedziorski**[2], **Siddhant Garg**[2],
Luca Soldaini[3†], **Alessandro Moschitti**[2]
[1]University of Trento , [2]Amazon Alexa AI, [3]Allen Institute for AI
matteo.gabburo@unitn.it
{rikdz,sidgarg,amosch}@amazon.com
lucas@allenai.org

Abstract

Recent studies show that Question Answering (QA) based on Answer Sentence Selection (AS2) can be improved by generating an improved answer from the top-k ranked answer sentences (termed GenQA). This allows for synthesizing the information from multiple candidates into a concise, natural-sounding answer. However, creating large-scale supervised training data for GenQA models is very challenging. In this paper, we propose to train a GenQA model by transferring knowledge from a trained AS2 model, to overcome the aforementioned issue. First, we use an AS2 model to produce a ranking over answer candidates for a set of questions. Then, we use the top ranked candidate as the generation target, and the next k top ranked candidates as context for training a GenQA model. We also propose to use the AS2 model prediction scores for loss weighting and score-conditioned input/output shaping, to aid the knowledge transfer. Our evaluation on three public and one large industrial datasets demonstrates the superiority of our approach over the AS2 baseline, and GenQA trained using supervised data.

1 Introduction

In recent times, extractive QA research can be categorized into two broad directions for the task of producing the final answer for a question: (i) Answer Sentence Selection (AS2), which, given a question and a set of answer-sentence candidates, selects sentences (e.g., retrieved by a search engine) that correctly answer the question; and (ii) Machine Reading (MR), e.g., (Chen et al., 2017), which, given a question and a reference text, involves finding an exact text span that answers the question. AS2 models can perform more efficiently with large text databases (as they originated from the TREC-QA track (Voorhees, 1999)), and there seems a renewed research interest in these models for applications to personal assistants, e.g., Alexa (Garg et al., 2020; Matsubara et al., 2020a; Garg and Moschitti, 2021).

Both approaches (AS2 and MR) when applied for QA over unstructured web text, while effective, may have certain drawbacks. Arbitrary web sentences may not contain all the information needed to answer a question, or may contain distracting extraneous information. Moreover, they may have a particular sentiment or style that is not suited to QA, or be too structurally reliant on longer discourse context to serve as a standalone answer. In light of this, researchers have been exploring text generation systems for writing 'better' answers. For example, in MR, RAG (Lewis et al., 2020b) generates an answer from a set of documents selected by dense passage retrieval models.

For AS2 systems, research has focused on learning to summarize answers from relevant paragraphs (Lewis et al., 2020a), or to synthesize information from the top ranked candidates of an AS2 system (Hsu et al., 2021). The latter approach, termed as GenQA, has shown improvements in terms of both answer accuracy and style suitability. A distinctive characteristic of GenQA over a generation-based approach for MR is the length of the answer: the former uses an entire sentence as the target, while the latter in practice uses a short text (primarily targeting entity names). In this work, we focus on GenQA as we are interested to generate complete answer sentences from precise information selected by AS2 models.

A challenge for training effective GenQA models is the difficulty of obtaining large-scale, high-quality training data. Producing such data for GenQA typically requires human annotators to read questions and paragraphs of relevant background information, and then author a self-contained, natural answer (typically a sentence). This fairly involved procedure highly diminishes the veloc-

[*]Work done as an intern at Amazon Alexa AI
[†]Work completed at Amazon Alexa AI

ity of annotation. Existing datasets in research works either offer limited coverage of all domains, where GenQA can be applied (Bajaj et al., 2018), or are too small to be used as supervised training data (Muller et al., 2021). Generally, collecting a human-authored answer to a question when given a context is significantly more expensive compared to annotating the correctness of an extracted web sentence as an answer for the same question. Consequently, there are a large number of annotated datasets (Wang et al., 2007; Yang et al., 2015; Garg et al., 2020) available for the latter type, aimed at training answer sentence selection (AS2) systems.

In this work, we propose a training paradigm for transferring the knowledge learned by a discriminative AS2 ranking model to train an answer generation QA system. Towards this, we learn a GenQA model using weak supervision provided by a trained AS2 model on a unlabeled data set comprising of questions and answer candidates. Specifically, for each question, the AS2 model is used to rank a set of answer candidates without having any label of correctness/incorrectness for answering the question. The top ranked answer is used as the generation target for the GenQA model, while the question along with the next k top-ranked answers are used as the input for the GenQA model.

We supplement the ranking order of answer candidates with the prediction confidence scores provided by the AS2 model for each answer candidate. This is done by modifying our knowledge transfer strategy in two ways. First, we weight the loss of each training instance (question + context, comprised of k answer candidates) using the AS2 model score of the top ranked answer, which is to be used as the GenQA target. This allows the GenQA model to selectively learn more from 'good' quality target answers in the weakly supervised training data (AS2 models are calibrated to produce higher confidence scores for correct answers). However, this loss weighting only considers the score of the output target, and does not exploit the scores of the input candidates. To overcome this limitation, we discretize and label the AS2 scores into l confidence buckets, add these bucket labels to the GenQA vocabulary, and finally prepend the corresponding label to each answer candidate in the input and/or the output. This confidence bucket label provides the GenQA model with an additional signal about the answer quality of each candidate as assigned by the AS2 model. We show that both these techniques improve the QA accuracy, and can be combined to provide additional improvements.

We empirically evaluate [1] our proposed knowledge transferring technique from AS2 to GenQA on three popular public datasets: MS-MARCO NLG (Bajaj et al., 2018), WikiQA (Yang et al., 2015), TREC-QA (Wang et al., 2007); and one large scale industrial QA dataset. Our results show that the GenQA model trained using our paradigm of weak supervision from an AS2 model can surprisingly outperform both the AS2 model that was used for knowledge transfer (teacher), as well as a GenQA model trained on fully supervised data. On small datasets such as WikiQA and TREC-QA, we show that AS2 models trained even on small amounts of labeled data can be effectively used to weakly supervise a GenQA model, which then can outperform its teacher in QA accuracy. Additionally, on MS-MARCO NLG, where fully supervised GenQA training data is available, we show that an initial round of training with our weakly supervised methods yields additional performance improvements compared to the standard supervised training of GenQA. Qualitatively, the answers generated by our model are often more directly related to the question being asked, and stylistically more natural-sounding and suitable as responses than answers from AS2 models, despite being trained only on extracted sentences from the web.

2 Related Work

Our work builds upon recent research in AS2, answer generation for QA, and transfer learning.

Answer Sentence Selection Early approaches for AS2 use CNNs (Severyn and Moschitti, 2015) or alignment networks (Shen et al., 2017; Tran et al., 2018; Tay et al., 2018) to learn and score question and answer representations. Compare-and-aggregate architectures have also been extensively studied (Wang and Jiang, 2017; Bian et al., 2017; Yoon et al., 2019) for AS2. Tayyar Madabushi et al. (2018) exploited fine-grained question classification to further improve answer selection. Garg et al. (2020) achieved state-of-the-art results by fine-tuning transformer-based models on a large-scale QA dataset first, and then adapting to smaller AS2 datasets. Matsubara et al.

[1] We will release code and all trained models checkpoints at https://github.com/amazon-research/wqa-genqa-knowledge-transfer

(2020b) combine multiple heterogeneous systems for AS2 to improve a QA pipeline, similar in spirit to GenQA. Several follow-up works have further improved the performance of AS2 using transformer models, using multiple answer candidates (Zhang et al., 2021) and document-aware pre-training strategies (Di Liello et al., 2022a,b).

Answer Generation for QA Answer generation for MR has been studied by Izacard and Grave (2021); Lewis et al. (2020b), while Iida et al. (2019); Goodwin et al. (2020); Deng et al. (2020) have studied question-based summarization (QS). Asai et al. (2022) incorporate the evidentiality of retrieved passages for training a generator, evaluated for the QA task of open-domain MR span-extraction. Xu et al. (2021) obtain extractive answer spans from a generative model by leveraging the decoder cross-attention patterns. Fajcik et al. (2021) combine a generative reader with an extractive reader to aggregate evidence from multiple passages for open-domain span-extraction.

All the previously described approaches focus on identifying short answer spans for answering questions. Research on generating complete sentences as answers (similar to answer sentences produced by extractive AS2 systems) is rarer, but includes Hsu et al. (2021), that propose a QA pipeline for GenQA (refer Fig 1). This pipeline starts with an AS2 model that selects 'good' answer candidates that are then used for generating the answer. Hsu et al. learn to generate natural responses to questions using the top ranked candidates from the AS2 model as input context to the GenQA model. GenQA has also been explored for multilingual QA (Muller et al., 2021) by extending the answer generation approach to a multilingual setting, where the answer candidates (that are used as input to the GenQA model) can be from a mix of different languages.

In all these works, a major challenge is finding training data for effectively training GenQA models, which requires annotator-authored natural responses. In this work, we alleviate this problem by showing that it is possible to use AS2 ranked candidates to create the input context and output target for training GenQA, achieving state-of-the-art results.

Transfer Learning Transfer learning is well studied in NLP, including pre-training (Devlin et al., 2019; Liu et al., 2019), multi-task learning (Luong et al., 2016), cross-lingual transfer (Schuster et al., 2019) and domain adaptation (Gururangan et al., 2020). Our work is squarely located in this space: our underlying language models are based on pre-training for text generation (Radford et al., 2019; Raffel et al., 2020); our main contribution is to show that knowledge can be transferred sequentially from a ranking (discriminative) task to a generation task. Recently Wang et al. (2021) propose a new domain adaptation method leveraging large unlabeled datasets and a query generator model. Izacard and Grave used retrieved text passages containing evidences to train a generative model for open domain QA.

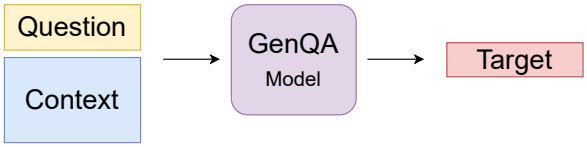

Figure 1: A GenQA model (Hsu et al., 2021) is a seq2seq model that takes in input a question and k answer candidates, and generates an answer.

3 Knowledge Transfer: AS2 → GenQA

Previous works on GenQA require the use of labeled data for effectively training the GenQA model. To reduce the need of expensive large-scale training data for GenQA, we propose a training paradigm that uses unlabeled data while being weakly-supervised by a discriminative AS2 model (as shown in Fig. 2).

3.1 Answer Sentence Selection (AS2)

AS2 is a popular task in QA, defined as follows: Given a question q, and a set of answer candidates $C = \{c_1, \ldots, c_n\}$ (retrieved using a web-index, KB, etc), find the answer candidate $c_q \in C$ that best answers q. This is typically modeled as a binary classifier $\mathcal{M}$ over QA pairs, labeled as correct or incorrect. At inference, the scores assigned by $\mathcal{M}$ can be used to produce a ranking over C, with $c_q = \text{argmax}_i \, \mathcal{M}(q, c_i)$.

3.2 Generative QA (GenQA)

Generative QA refers to using a text generation model for generating an answer for a question. More specifically, when provided with a question q and context $\bar{c}$, the GenQA model $\mathcal{M}_G$ should generate a natural sounding answer $c_q = \mathcal{M}_G(q, \bar{c})$ that correctly answers q. Following Hsu et al. (2021), we consider a set of k answer candidates as the context $\bar{c}$ to be provided to $\mathcal{M}_G$.

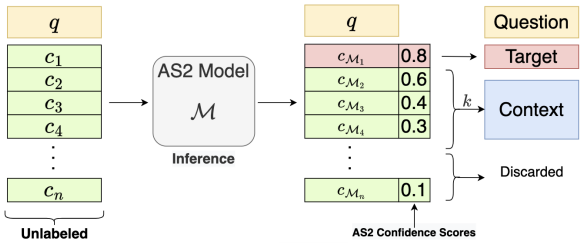

Figure 2: Our pipeline for creating weakly supervised training examples for training GenQA models. The AS2 model assigns a confidence score to each answer candidate sentence. These scores are used to select the inputs and the target sequences for the GenQA model.

3.3 Training GenQA using an AS2 model

We aim at training a GenQA model, $\mathcal{M}_G$, using a trained AS2 model, $\mathcal{M}$, which predicts correctness/incorrectness of answer candidates for a given question. Specifically, we use an unsupervised dataset, $\mathcal{U}$, comprising of a set of questions along with their retrieved answer candidates, i.e, $(q, C = \{c_1, \ldots, c_n\})$. Note that there are *no* human annotations of correctness/incorrectness for the answer candidates in C for the question q.

For each question $q \in \mathcal{U}$, we denote the ranking of answer candidates by $\mathcal{M}$ in decreasing order of prediction scores by $C_M = \{c_{M_1}, c_{M_2}, \ldots, c_{M_n}\}$. We create weakly supervised examples for training the GenQA model by using $\left(q, \bar{c} = \{c_{M_2}, c_{M_3}, \ldots, c_{M_{k+1}}\}\right)$ as the input, and setting the generation target to be the top ranked answer candidate from $\mathcal{M}$, i.e, c_{M_1}. For seq2seq transformer-based text generation models such as T5 (Raffel et al., 2020) and BART (Lewis et al., 2020a), we concatenate the question and k answer candidates: "q [SEP] c_{M_2} [SEP] ... [SEP] $c_{M_{k+1}}$" to be provided as input to $\mathcal{M}_G$ and use the negative log probability of predicting each token of the target c_{M_1} given the previous tokens as the training loss.

The resulting GenQA model $\mathcal{M}_G$ is trained on the unsupervised dataset only using *weak supervision* from the discriminative AS2 model $\mathcal{M}$. For the rest of the paper, we denote this training paradigm for GenQA by **WS**. This approach is related to knowledge distillation (KD) (Hinton et al., 2015) wherein the predictions of a teacher model are used for guiding the learning of a student model. The novelty of our proposed approach from standard distillation techniques stems from the fact that the teacher (AS2) and student (GenQA) belong to different paradigms of training, the former being a discriminative classifier model while the latter being a generative model. Furthermore, standard KD techniques (Hinton et al., 2015; Sanh et al., 2019) use a combination of supervision from the teacher (KL divergence) and supervision from the labeled data (Cross Entropy) for teaching the student, while in our case, we only use the supervision signal from the teacher without any access to labeled data.

3.4 Weighting GenQA Loss with AS2 scores

The binary cross-entropy loss used for training discriminative AS2 models typically calibrates their prediction w.r.t answer correctness (Kamath et al., 2020; Garg and Moschitti, 2021). This means that the top ranked answer to a question from $\mathcal{M}$ that receives a high prediction probability is more likely to be correct than the answer to another question that receives a lower prediction probability. We exploit this in addition to the ranking order generated by $\mathcal{M}$ to improve the learning of the GenQA model $\mathcal{M}_G$. Intuitively, we want the GenQA model to learn more from 'good' quality target answers (having higher prediction scores) than from lower quality answers.

To this end, we propose to modify our **WS** cross-entropy loss by incorporating the AS2 scores provided by the AS2 model $\mathcal{M}$ when performing the knowledge transfer. Specifically, we use the prediction score $\mathcal{M}(q, c_{M_1})$ (normalized in $[0, 1]$) of $\mathcal{M}$ on the top ranked answer candidate c_{M_1} to weight the loss term for $\mathcal{M}_G$ corresponding to that instance (question q). Formally, the loss for each last generated word, y_r of the generated output y is:

$$\mathcal{L}_{\mathcal{M}_G}(q, c_{M_1}) = \frac{1}{Z}\mathcal{M}(q, c_{M_1}) \times \mathcal{L}_G(y_r, c_{M_1}), \quad (1)$$

where Z is the normalizing constant for AS2 scores computed on the training dataset, $\mathcal{L}_G$ is the standard loss for generating y_r, and c_{M_1} is assumed to be the gold standard output. $\mathcal{L}_G$ is defined as:

$$\mathcal{L}_G(y_r, c_{M_1}) = -\sum_{v \in V} log \frac{e^{y_r(v)}}{\sum_{h \in V} e^{y_r(h)}} c_{M_1}(r, v),$$

where V is the vocabulary, $y_r(v)$ is the score of generating the word v at position r, and $c_{M_1}(r, v)$ is 1 if the r^{th} word of c_{M_1} is v, otherwise it is -1. We refer to the model trained with Eq. 1 as **LW**.

3.5 AS2 Score Conditioned I/O Shaping

In the previous section, we described how to use $\mathcal{M}(q, c_{M_1})$ – the AS2 prediction score of c_{M_1} – to weight the training loss for question q, since this candidate is used as the target for q in $\mathcal{M}_G$. However **LW** ignores the AS2 scores for the other answer candidates $c_{M_2} \ldots c_{M_k+1}$, and does not explicitly provide this AS2 score as context to the

GenQA model. To overcome this, we propose a method for labeling each candidate in the input of $\mathcal{M}_G$ with a representation of its AS2 score. This method can also be applied to the model output, which results in an improved performance (as shown in Section 5).

We define a bucketing function $\mathcal{F}$ over the normalized interval $[0, 1]$ that operates on the AS2 prediction score $\mathcal{M}(q, a)$. For a QA pair (q, a), $\mathcal{F}(q, a)$ assigns a confidence bucket label $b_i \in [b_1, \ldots, b_l]$ based on $\mathcal{M}(q, a)$. Here $\mathcal{F}(q, a)$ is assigned b_i if $\mathcal{M}(q, a)$ is in the interval $\left[\frac{i-1}{l}, \frac{i}{l}\right)$. For our experiments, we set the value of l=5. We add the bucket labels b_i as special tokens to the vocabulary of $\mathcal{M}_G$.[2] We use $\mathcal{F}$ to modify the input and output of the GenQA model as follows:

- **AS2 Score Conditioned Input (SCI):** We prepend the bucket label $b_j = \mathcal{F}(q, c_{M_j})$ to each of the $j \in \{2, \ldots, k{+}1\}$ answer candidates to be provided as input to $\mathcal{M}_G$, so that the new input is formatted as: "q [SEP] b_2 c_{M_2} [SEP] ... [SEP] b_{k+1} $c_{M_{k+1}}$".
- **AS2 Score Conditioned Output (SCO):** We prepend the bucket label $b_1 = \mathcal{F}(q, c_{M_1})$ to the target answer candidate c_{M_1}, so that the output target of $\mathcal{M}_G$ is: "b_1 c_{M_1}".

SCI and **SCO** can be used independently as well as jointly for training the GenQA model $\mathcal{M}_G$ using $\mathcal{M}$. For simplicity, we will use the acronym **SC** when these two techniques are used together.

We propose **SCI** to make the knowledge transfer more effective. Intuitively, labeling each input candidate with a special token correlated with its AS2 score helps the GenQA model: during training the model can focus more on the answer candidates associated with higher quality (more correct answers), thereby improving the model performance.

While **SCO** is related to **LW** presented in Sec 3.4, it differs in the fact that the former allows the model to "know" the score of the target when designing internal representations of the text in its input and output. We hypothesize that this knowledge allows the model to organize its internal representations differently in the presence of bad targets, rather than just be less influenced by them as in **LW**. Another advantage of **SCO** is that during inference time, we can use the generated bucket label token as a confidence score for the GenQA model's answer. Calibrating confidence scores for text generation models, e.g., using sequence likelihood, etc. is challenging, especially when decoding is constrained as in real world applications. Finally, we can force decoding to start from any one of the **SCO** bucket tokens in order to exploit its influence on the model's output. We empirically explore this in Appendix E.

[2]We experimented with using existing tokens from the vocabulary as bucket labels (e.g. "Probably","Maybe") in hopes of reusing model knowledge about the semantics of these words, but obtained worse empirical results.

4 Datasets and Models

For training and evaluating our knowledge transfer techniques (**WS**, **LW**, **SC**) described above, we categorize the data that we use for each experiment into the following four sources/types:

- **AS2:** Labeled (q, a) pairs with correctness and incorrectness annotation for training $\mathcal{M}$
- **Transfer:** Unlabeled (q, a) pairs that are ranked by $\mathcal{M}$, and used for knowledge transfer to $\mathcal{M}_G$
- **Fine-tuning:** Labeled data (human written answers / answers with correctness labels) for fine-tuning $\mathcal{M}_G$, *whenever available*
- **Evaluation:** Evaluation data for $\mathcal{M}$ and $\mathcal{M}_G$

In Section 5, we vary the sources of different types of the data described above, to demonstrate the robustness and generality of our knowledge transfer method. Below, we provide details about the data sources we use, along with a summary of the underlying models.

4.1 Unlabeled Data

MS-MARCO QA A popular MR dataset released by (Bajaj et al., 2018). We use the training split which contains ∼800k unique user queries from the Bing search engine along with ∼ 10 passages retrieved for each question.[3] We split the original dataset into individual sentences using the BlingFire tokenizer[4] to be used as the **Transfer** data. Note that this dataset is used as unlabeled data for our experiments.

AQAD-U A large scale internal industrial QA dataset containing *non-representative de-identified* user questions from Alexa virtual assistant. This unlabeled Alexa QA Dataset (AQAD-U) contains ∼50 million questions, and ∼400 answer candidates retrieved for each question using a large scale web index that contains over 100M web documents. We use this dataset as **Transfer** data for experiments in the industrial setting.

[3]MS-MARCO v2.1: https://huggingface.co/datasets/ms_marco

[4]https://github.com/microsoft/BlingFire

4.2 Labeled Data

ASNQ A large-scale AS2 corpus (Garg et al., 2020) derived from Google Natural Questions (NQ) dataset (Kwiatkowski et al., 2019). It consists of ∼60K questions with labeled answer sentences. We use this as **AS2** training data.

MS-MARCO NLG A split of MS-MARCO (Bajaj et al., 2018) that contains manually generated answers along with retrieved passages for ∼150k user queries, which we use for **Fine-tuning**. We sub-sample 1k questions from the development set, along with their answer candidates extracted from the associated passages, to be used as **Evaluation** data in our experiments. (We do not use the entire development set of ∼ 100k questions for evaluation due to the expensive cost of human annotations).

TREC-QA A popular QA benchmark (Wang et al., 2007) used to evaluate AS2 models. For our experiments, we use the filtering and splits proposed in (Zhang et al., 2022), where all questions have at least one positive and one negative candidate, and the test split is larger. The resulting dataset contains 816, 204 and 340 unique questions respectively for the training, dev. and test sets.

WikiQA A popular AS2 dataset (Yang et al., 2015) containing questions from Bing search logs and answer candidates from Wikipedia. We use a 'clean' setting for training by retaining questions with at least one positive answer candidate in the train and validation splits. This results in training/dev./test sets of WikiQA having 2118/296/236 questions, respectively.

AQAD-L The labeled counterpart of the industrial dataset AQAD-U as described in Section 4.1 above, where answer candidates additionally have human annotations of correctness/incorrectness. We use AQAD-L, comprising of ∼ 5k questions, as **Evaluation** data for experiments in the industrial setting. Results on AQAD-L are presented relative to the baseline AS2 model due to the data being internal.

For data statistics, please refer to Appendix A.2.

4.3 Modeling Details

We use T5 (Raffel et al., 2020) as the model for GenQA $\mathcal{M}_G$. For the AS2 models, we use a RoBERTa-Large (Liu et al., 2019) or ELECTRA-Base (Clark et al., 2020) trained using the TANDA approach (Garg et al., 2020), depending on the experimental setting. For our experiments, we set the value of k=5, i.e, the number of answer candidates to be provided as input to the GenQA model.

We train our models using $fp16$ precision, Adam (Kingma and Ba, 2015) as optimizer with a $lr = 1e{-}4$ and a batch size of 256. We trained each model for 25 epochs on both the versions of MS-MARCO (QA and NLG), and for 50 epochs on WikiQA and TrecQA. We select the best model by maximizing the average AS2 score on the development set of each dataset instead of minimizing the validation loss (see the details in Appendix B).

4.4 Evaluation and Metrics

We perform human evaluation of our generated answers: for each question/answer pair, we collect the annotations from five annotators (corresponding to the answer being correct/incorrect) using Amazon MTurk (see Appendix C for details). We use accuracy as the primary metric for all our experiments and models. Given a set of questions, this is computed as the fraction of correct answers divided by the number of incorrect answers as judged by the annotators. Note that: (i) each QA pair receives an average score from five annotators, and (ii) for the AS2 model, the accuracy is the same as Precision@1, which is the precision of the top ranked answer candidate.

5 Experiments and Results

We perform experiments in three data settings to evaluate different features of our method. On the MS-MARCO datasets, we show that weak supervision can augment strong models trained on in-domain data. On WikiQA and TREC, we show that weak supervision on large data improves over direct supervision on small data for this QA task. We also present an experiment on a very large industrial dataset to measure the contribution of each of our proposed techniques for using unlabeled training data at scale.

5.1 Comparison with the State of the art

These experiments aim at (i) understanding the effectiveness of our weakly supervised methods (**WS**, **LW** and **SC**), and (ii) comparing them with a GenQA state-of-the-art model, which is trained using fully supervised training data. We create weakly supervised data with a RoBERTa-Large AS2 model trained on ASNQ by applying it to the MS-MARCO QA dataset. It is important to note that, in this setting, during the training, both the student and the teacher models do not have any knowledge of the original labels of MS-MARCO QA as we consider this dataset to be unlabeled. We compare our approach against a supervised GenQA model baseline (Hsu et al., 2021), which is trained

Approach	Model	Unlabeled: MS-MARCO QA		Labeled: MS-MARCO NLG		Accuracy (%)
		Used	Training Strategy (Ours)	Used	Training Strategy	
AS2	RoBERTa-Large	✗	-	✗	-	79.3
GenQA	T5 - Large	✓	**WS**	✗	-	79.9
	T5 - Large	✓	**WS + LW**	✗	-	81.5
	T5 - Large	✓	**WS + SCI**	✗	-	82.0
	T5 - Large	✓	**WS + SCO**	✗	-	82.5
	T5 - Large	✓	**WS + LW + SC**	✗	-	83.7
	T5 - Large	✗	-	✓	(Hsu et al., 2021)	82.6
	T5 - Large	✓	**WS + LW + SC**	✓	(Hsu et al., 2021)	**85.3**

Table 1: Results on the test split of MS MARCO-NLG for different training paradigms of GenQA models. The weak supervision is provided by a RoBERTa-Large AS2 model trained on ASNQ. We compare with a fully supervised GenQA baseline (Hsu et al., 2021) trained on the train split of MS MARCO-NLG.

on MS-MARCO NLG. The MS-MARCO NLG dataset is much smaller than MS-MARCO QA but has higher quality answers as the targets since they are manually written. We also investigate a two-stage training strategy, by first applying our knowledge transfer approach, followed by the supervised training to understand if the two approaches are complementary or essentially capture similar information. All models are evaluated by manual annotators on the same MS-MARCO NLG test set.

Results: We present the results on the MS-MARCO NLG test set in Table 1. The baseline zero-shot accuracy of the AS2 model on this data is 79.3 and the baseline accuracy of the fully supervised GenQA model (Hsu et al., 2021) is 82.6. Our weak supervision (**WS**) transfer technique, which does not use any answers written by annotators as targets, shows improvements over the AS2 baseline (0.6%). This shows that our approach can transfer information learned by an AS2 model from ASNQ (a large labeled dataset) into a GenQA model.

Ablating each of the approaches (**LW**,**SCI**,**SCO**) individually in addition to **WS**, we observe consistent improvements (+1.6, +2.1 and +2.6% respectively) over the performance of **WS**, indicating that the AS2 scores can help in the knowledge transfer. Additionally, combining all the approaches with **WS** significantly improves the performance, and surprisingly can even outperform the supervised GenQA baseline (by 1.1% = 83.7-82.6). This shows that the knowledge transferred by our approach from ASNQ exceeds what can be learned from MS-MARCO NLG.[5]

Finally, when we combine our weak supervised training techniques with the supervised training in a two stage pipeline, we observe very significant performance gains, e.g., 2.7% over the supervised approach. This shows that (i) the information in MS-MARCO NLG is complementary to the knowledge transferred from ASNQ, and (ii) our approach is effective in transferring knowledge from a discriminative ranker to a downstream GenQA model.

[5]MS-MARCO NLG is larger than ASNQ, but the latter is a much higher quality dataset in terms of diversity and complexity of questions and answer annotations

Approach	Unlabeled (MS-MARCO QA)		Labeled (WikiQA Train)		Accuracy (%)
	Used	Training Strategy	Used	Training Strategy	
AS2	✗	-	✓	Fine-tuning	78.3
GenQA	✓	**WS + LW + SC**	✗	-	78.7
	✗	-	✓	(Hsu et al., 2021)	72.9
	✓	**WS + LW + SC**	✓	(Hsu et al., 2021)	**79.8**

(a) WikiQA

Approach	Unlabeled (MS-MARCO QA)		Labeled (TREC-QA)		Accuracy (%)
	Used	Training Strategy	Used	Training Strategy	
AS2	✗	-	✓	Fine-tuning	85.9
GenQA	✓	**WS + LW + SC**	✗	-	**90.5**
	✗	-	✓	(Hsu et al., 2021)	80.7
	✓	**WS + LW + SC**	✓	(Hsu et al., 2021)	89.8

(b) TREC-QA

Table 2: Results on the test split of WikiQA and TREC-QA. The weak supervision is provided by RoBERTa-Large AS2 models trained respectively on WikiQA and TREC-QA. We compare with a fully supervised GenQA baseline (Hsu et al., 2021) trained respectively on the train split of WikiQA and TREC-QA using ground truth correct answers as the target for generation. We use T5-Large for all GenQA models.

Due to brevity of space, we present a qualitative ablation of the generated examples in Appendix E.

5.2 Scarce Data Setting

In this experiment, we measure the quality of our weak supervision approaches, by evaluating their performance on two popular AS2 benchmark datasets: WikiQA and TREC-QA. We train the AS2 teacher model on this data and still use the unlabeled data from MS-MARCO QA for performing the knowledge transfer. This way, we test if our approach is applicable in real scenarios, where data can be scarce and no large labeled data dataset is available (such as ASNQ or MS-MARCO NLG). Additionally, we verify if our approach works for other domains and if fine-tuning GenQA on the target domain data can help knowledge transfer in that domain, even in case of data scarcity.

We compare our weakly-supervised approaches with an AS2 baseline and a GenQA model trained

on the target datasets, using their ground-truth labels. We used the original test splits of the datasets. Note that for these experiments, (i) we use the best performing strategy for our transfer learning, i.e., **WS** along with **LW** and **SC**, and (ii) the AS2 baseline is the same model that we use to transfer knowledge on the MS-MARCO QA.

Results: From our results in Table 2, we make the following observations:
(i) AS2 accuracy evaluated with our human annotations is around 10% lower than results from previous works, e.g., (Zhang et al., 2021). As we use the same model,[6] the difference is due to the fact that we use the 'raw' test setting which includes questions with no correct answer candidates.
(ii) Our transfer learning techniques have better performance than both the AS2 model and the supervised GenQA baselines. For WikiQA, our knowledge transfer approach, which has only seen unlabeled MS-MARCO data and no labeled training data from WikiQA, gets higher accuracy than both the AS2 baseline (+0.4%) and a fully supervised GenQA baseline (+5.8%), which uses the ground truth labels from the target datasets.
(iii) We observe the same trend for TREC-QA: our weakly supervised models improve over both AS2 (4.6%) and supervised GenQA (9.8%) baselines.
(iv) In contrast to our observations from Table 1, the supervised GenQA baseline for WikiQA and TREC-QA is less accurate than the AS2 baseline. We explain this with two reasons: (a) the small size of these datasets (only few thousands training questions) might be insufficient to train a large T5 model for GenQA, and (b) the usage of extracted answers as the target for generation instead of a human-written and natural sounding answer affects the quality of answer generation.
(v) Finally, supervised fine-tuning applied after our transfer learning only improves performance on WikiQA. The WikiQA dataset has several questions with no correct answers (∼40%). Fine-tuning on the supervised dataset reinforces the training of the generator on questions having actual positive labels, thereby helping to reduce noise, and improving the final accuracy on the entire test set.

5.3 Industrial Setting

In this experiment, we aim to show that our experimental findings extend to very large-scale and real-world data, i.e., *non-representative de-identified* customer questions from Alexa virtual assistant. We use the 50M question AQAD-U corpus as the unlabeled QA corpus for training the GenQA model, transferring the knowledge of an AS2 teacher model (no human-authored answer is used for training fully-supervised GenQA models on this data). We compare our methods for weak supervision against the AS2 teacher model on the labeled test split: AQAD-L.

[6]Starting from the https://huggingface.co/roberta-large checkpoint

Approach	Model	Unlabeled: AQAD-U		Accuracy (%)
		Used	Training Strategy	
AS2	ELECTRA-Base	✗	-	Baseline
GenQA	T5 - Base	✓	**WS**	+1.34%
	T5 - Base	✓	**WS + LW**	+4.08%
	T5 - Base	✓	**WS + LW + SC**	**+7.35%**

Table 3: Results on AQAD-L for different training paradigms of GenQA models. All results are reported in absolute % changes w.r.t the AS2 baseline.

Results: We present the results in Table 3 relative to the AS2 baseline, due to the data being internal, which is used as the 'teacher' for transferring knowledge to train the GenQA model. For these experiments we use T5-Base as the GenQA model (due to the large size of AQAD-U), and our results show that knowledge transfer from the AS2 model using the unlabeled data surprisingly improves the accuracy of the baseline by 1.34%. This indicates that the weak supervision provided by the AS2 model is able to train a GenQA model that performs better than the AS2 teacher itself. Furthermore, using loss weighting (**LW**) and input/output shaping (**SC**) significantly improves our weak supervision approach. The T5-Base model trained using a combination of **LW** and **SC** on the unlabeled AQAD-U corpus achieves an impressive 7.35% gain in accuracy over the baseline AS2 model (which has been trained on labeled data with annotations for answer correctness).

6 Analysis and Ablation Studies

Automatic Evaluation: We consider whether automatic evaluation metrics correlate with human evaluation for our task. In Table 4, we compare BERT-Score (Zhang* et al., 2020) and BLEURT (Sellam et al., 2020) with the human evaluation of various models on MS MARCO-NLG test data. We find that despite their good performance for other NLG tasks neither metric has a particularly strong Pearson correlation with the human evaluation results for the task of answer sentence generation (GenQA): BLEURT has a correlation 0.622; BERT-Score has a correlation of 0.447. Neither automatic metric is able to correctly identify

Approach/Strategy	Human Evaluation	BERT-Score PREC	REC	F1	BLEURT AVG	STDEV
AS2	0.793	0.876	0.905	0.890	0.509	0.225
WS	0.799	0.884	0.903	0.893	0.520	0.217
WS+LW	0.815	0.993	0.994	0.993	0.517	0.216
WS+SCI	0.820	0.885	0.905	0.895	0.525	0.218
WS+SCO	0.825	0.884	0.905	0.894	0.521	0.221
WS+LW+SC	0.837	0.883	0.901	0.891	0.512	0.211
(Hsu et al., 2021)	0.826	0.907	0.911	0.908	0.555	0.224
WS+LW+SC+ (Hsu et al., 2021)	0.853	0.994	0.996	0.995	0.559	0.221
Correlation (with Human Evaluation)			**0.447**			**0.622**

Table 4: Results on the the testset of MS MARCO-NLG comparing human evaluation to well-known automatic evaluation metrics: BERT-Score and BLEURT. The last row shows the correlation between the human evaluation and the two automatic metrics (BERT-Score, and BLEURT), indicating that these metrics do not correlate strongly with human evaluation.

Starting **SCO** Token	Accuracy %
'[_YES_]'	78.6
'[_PROBABLY_]'	71.2
'[_MAYBE_]'	69.1
'[_DOUBT_]'	66.0
'[_NO_]'	60.5

Table 5: Accuracy on generated answers clustered according to the starting **SCO** bucket token generated by the GenQA model on the WikiQA test set

the system ranking effected by human evaluation as presented in Table 1. Additional analysis using the BLEU score is presented in Appendix D.

Structured Output and Accuracy: In the **SCO** approach we propose, we prepend a special bucket token $b_1 = \mathcal{F}(q, c_{M_1})$ corresponding to the AS2 confidence score of the target answer candidate c_{M_1}, so that the output target for the GenQA model $\mathcal{M}_G$ is: "b_1 c_{M_1}". We denote the bucket tokens for l=5 with the following set: ['[_YES_]','[_PROBABLY_]','[_MAYBE_]', '[_DOUBT_]','[_NO_]'] corresponding to the confidence intervals $[0.8, 1]$, $[0.6, 0.8)$, $[0.4, 0, 6)$, $[0.2, 0.4)$ and $[0, 0.2)$ for $\mathcal{M}$'s score respectively. In this section, we analyze the role of the **SCO** bucket tokens as a confidence measure for the GenQA model's output during inference. For this, we cluster the answers generated by the GenQA model (a T5-Large model trained on MS-MARCO QA using both **LW** and **SC**) based on the generated **SCO** bucket token. We then manually evaluate the answers and show the accuracy for each cluster in Table 5. Here we observe an evident difference in the correctness of the generated answers in the **'[_YES_]'** cluster from those in the **'[_NO_]'** cluster (78.6% v/s 60.5%). The accuracy of generated answers monotonically decreases with the confidence interval of the bucket token (**'[_YES_]'** $\rightarrow$ **'[_NO_]'**). Additional analysis of model outputs is presented in Appendix E.

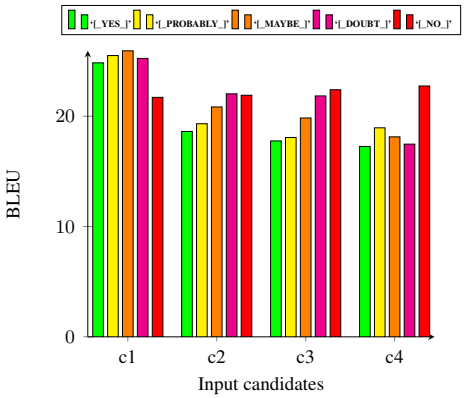

Figure 3: Correlation between the generated answer (with different starting **SCO** bucket tokens) with each input candidate. We observe that generated answers starting with high confidence bucket tokens (e.g., **'[_YES_]'** and **'[_PROBABLY_]'**) tend to copy more from the top ranked answer candidate, than the generated answers starting with lower confidence bucket tokens (e.g., **'[_NO_]'**).

Generated Answer v/s Input Candidates: We compare the generated answers (for each **SCO** bucket token) with the input answer candidates to understand how the model copies from the input candidates, and if there is a correlation with the **SCO** bucket tokens. In Fig. 3, we present the similarity between the generated answer with the top 4 ranked input candidates using BLEU score. This analysis shows that generated answers starting with a high confidence **SCO** bucket token (e.g., **'[_YES_]'** or **'[_PROBABLY_]'**) are more similar to the first candidate (higher ranked), while answers starting with lower confidence **SCO** bucket tokens (e.g., **'[_NO_]'**) are on average equally distant from all the input candidates.

7 Conclusion

In this paper, we have presented a novel approach for transferring knowledge from a discriminative AS2 model to an answer generation model, only using unlabeled data. We use the ranking produced by the AS2 model for training a GenQA model using the top answer as the target output, and the next k top ranked answers along with the question as the input. We also propose input/output shaping and loss weighting techniques during knowledge transfer to improve the performance of GenQA. Our experimental results on three public and one large industrial datasets show that GenQA models trained with knowledge transfer from AS2 models achieve higher answering accuracy than both the AS2 teacher and supervised GenQA trained with in-domain data. We are releasing our code and trained models to support future research.

Limitations

Our approach of training GenQA models requires access to large GPU resources for training large pre-trained language models such as T5-Large, etc. For the experiments in this paper, we only consider datasets from the English language, however we conjecture that our techniques should work similarly for languages with a similar morphology. The evaluations for all experiments performed in this paper are done using human annotations on MTurk, which is time consuming and expensive. Currently, automatic evaluation of correctness and style suitability for question answering is extremely challenging, and we hope that research advances in this domain further encourages broader research in answer generation systems.

Acknowledgements

We thank the anonymous reviewers and the meta-reviewer for their valuable suggestions and comments. We would like to thank Thuy Vu for developing and sharing the AQAD dataset.

References

Akari Asai, Matt Gardner, and Hannaneh Hajishirzi. 2022. Evidentiality-guided generation for knowledge-intensive NLP tasks. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2226–2243, Seattle, United States. Association for Computational Linguistics.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. Ms marco: A human generated machine reading comprehension dataset.

Weijie Bian, Si Li, Zhao Yang, Guang Chen, and Zhiqing Lin. 2017. A compare-aggregate model with dynamic-clip attention for answer selection. *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*.

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

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

Yang Deng, Wai Lam, Yuexiang Xie, Daoyuan Chen, Yaliang Li, Min Yang, and Ying Shen. 2020. Joint learning of answer selection and answer summary generation in community question answering. In *AAAI*.

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.

Luca Di Liello, Siddhant Garg, Luca Soldaini, and Alessandro Moschitti. 2022a. Paragraph-based transformer pre-training for multi-sentence inference. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2521–2531, Seattle, United States. Association for Computational Linguistics.

Luca Di Liello, Siddhant Garg, Luca Soldaini, and Alessandro Moschitti. 2022b. Pre-training transformer models with sentence-level objectives for answer sentence selection.

Martin Fajcik, Martin Docekal, Karel Ondrej, and Pavel Smrz. 2021. R2-D2: A modular baseline for open-domain question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 854–870, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Siddhant Garg and Alessandro Moschitti. 2021. Will this question be answered? question filtering via answer model distillation for efficient question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7329–7346, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Siddhant Garg, Thuy Vu, and Alessandro Moschitti. 2020. Tanda: Transfer and adapt pre-trained transformer models for answer sentence selection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7780–7788.

Travis Goodwin, Max E. Savery, and Dina Demner-Fushman. 2020. Towards zero shot conditional summarization with adaptive multi-task fine-tuning. *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, 2020:3215–3226.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey,

and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network.

Chao-Chun Hsu, Eric Lind, Luca Soldaini, and Alessandro Moschitti. 2021. Answer generation for retrieval-based question answering systems. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4276–4282, Online. Association for Computational Linguistics.

Ryu Iida, Canasai Kruengkrai, Ryo Ishida, Kentaro Torisawa, Jong-Hoon Oh, and Julien Kloetzer. 2019. Exploiting background knowledge in compact answer generation for why-questions. In *AAAI*.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Amita Kamath, Robin Jia, and Percy Liang. 2020. Selective question answering under domain shift. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5684–5696, Online. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference of Learning Representations*.

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

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

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA. Curran Associates Inc.

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.

Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Yoshitomo Matsubara, Thuy Vu, and Alessandro Moschitti. 2020a. Reranking for efficient transformer-based answer selection. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 1577–1580. ACM.

Yoshitomo Matsubara, Thuy Vu, and Alessandro Moschitti. 2020b. Reranking for efficient transformer-based answer selection. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 1577–1580, New York, NY, USA. Association for Computing Machinery.

Benjamin Muller, Luca Soldaini, Rik Koncel-Kedziorski, Eric Lind, and Alessandro Moschitti. 2021. Cross-lingual genqa: A language-agnostic generative question answering approach for open-domain question answering. *CoRR*, abs/2110.07150.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In *5th Workshop on Energy Efficient Machine Learning and Cognitive Computing @ NeurIPS 2019*.

Sebastian Schuster, Sonal Gupta, Rushin Shah, and Mike Lewis. 2019. Cross-lingual transfer learning for multilingual task oriented dialog. 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 3795–3805, Minneapolis, Minnesota. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to rank short text pairs with convolutional deep neural networks. *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Noam M. Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. *ArXiv*, abs/1804.04235.

Gehui Shen, Yunlun Yang, and Zhi-Hong Deng. 2017. Inter-weighted alignment network for sentence pair modeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1179–1189, Copenhagen, Denmark. Association for Computational Linguistics.

Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2018. Multi-cast attention networks. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

Harish Tayyar Madabushi, Mark Lee, and John Barnden. 2018. Integrating question classification and deep learning for improved answer selection. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3283–3294, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Quan Hung Tran, Tuan Lai, Gholamreza Haffari, Ingrid Zukerman, Trung Bui, and Hung Bui. 2018. The context-dependent additive recurrent neural net. 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 1274–1283, New Orleans, Louisiana. Association for Computational Linguistics.

Ellen M. Voorhees. 1999. The trec-8 question answering track report. In *In Proceedings of TREC-8*, pages 77–82.

Kexin Wang, Nandan Thakur, Nils Reimers, and Iryna Gurevych. 2021. GPL: generative pseudo labeling for unsupervised domain adaptation of dense retrieval. *CoRR*, abs/2112.07577.

Mengqiu Wang, Noah A Smith, and Teruko Mitamura. 2007. What is the jeopardy model? a quasi-synchronous grammar for qa. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 22–32.

Shuohang Wang and Jing Jiang. 2017. A compare-aggregate model for matching text sequences. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Peng Xu, Davis Liang, Zhiheng Huang, and Bing Xiang. 2021. Attention-guided generative models for extractive question answering. *CoRR*, abs/2110.06393.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2013–2018.

Seunghyun Yoon, Franck Dernoncourt, Doo Soon Kim, Trung Bui, and Kyomin Jung. 2019. A compare-aggregate model with latent clustering for answer selection. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Zeyu Zhang, Thuy Vu, and Alessandro Moschitti. 2021. Joint models for answer verification in question answering systems. 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 3252–3262, Online. Association for Computational Linguistics.

Zeyu Zhang, Thuy Vu, and Alessandro Moschitti. 2022. Double retrieval and ranking for accurate question answering. *CoRR*, abs/2201.05981.

Appendix

A Experimental setup details

A.1 Hyperparameter Selection

For our experiments, we consider different combinations of hyper-parameters. In particular, for training we combined the following parameters in multiple experiments: $lr \in \{1e^{-6}, 5e^{-5}, 1e^{-4}\}$, $k \in \{5, max\}$, batch size $\in \{64, 128, 256\}$, precision $\in \{16, 32\}$ along with using both Adam (Kingma and Ba, 2015) and Adafactor (Shazeer and Stern, 2018) optimizers. We selected the best combination of hyper-parameters (described in Section 5) by manually evaluating the output of different models on a reduced version of the MS-MARCO NLG dataset. Additionally, we also experimented with different hyper-parameters for the decoder. Specifically, we performed a qualitative evaluation of the answers generated using different parameters to select the best configuration: beam search of 5, and forcing the generated answer to have a number of tokens in the interval $[6, 100]$.

A.2 Dataset Statistics

Below we provide the statistics for the different datasets that we use in our experiments.

Dataset	Split	# of Q	QA Pairs
MS-MARCO QA	Train	655006	19543964
MS-MARCO QA	Dev	1000	29309
MS-MARCO NLG	Train	153725	4694170
MS-MARCO NLG	Dev	1000	28353
MS-MARCO NLG	Test	1000	28529
WikiQA	Train (clean)	857	8651
WikiQA	Dev (clean)	121	1126
WikiQA	Test	633	6165
TrecQA	Train	804	32964
TrecQA	Dev	216	9590
TrecQA	Test	340	13416

Table 6: Datasets statistics.

A.3 Computational Setup

We perform each experiment on 8 NVIDIA A100 GPUS (40GB RAM) using DDP[7] as distributed training strategy. To complete the training on MS-MARCO every model takes approximately 6 days, while the experiments on TREC-QA and WikiQA takes 4 hours.

[7] https://pytorch.org/docs/master/generated/torch.\nn.parallel.DistributedDataParallel.html

B Checkpoint Selection using AS2 Model

We observed that minimizing the loss on the development split does not strictly correlate with better answer generation from a human annotation perspective. Thus we experimented with an alternate performance measure for model checkpoint selection. Specifically, we use each checkpoint to generate outputs for a number of validation examples and score them with an AS2 model. We use the average scores produced by the AS2 model as the metric for deciding the best model checkpoint. The differences between using the loss on the development split and the average AS2 score can be noticed in Figure 4. We conducted a manual evaluation of outputs for different checkpoints and determined that using AS2 scores correlates better with our manual judgements than development set loss. We plan to explore this technique further in future work.

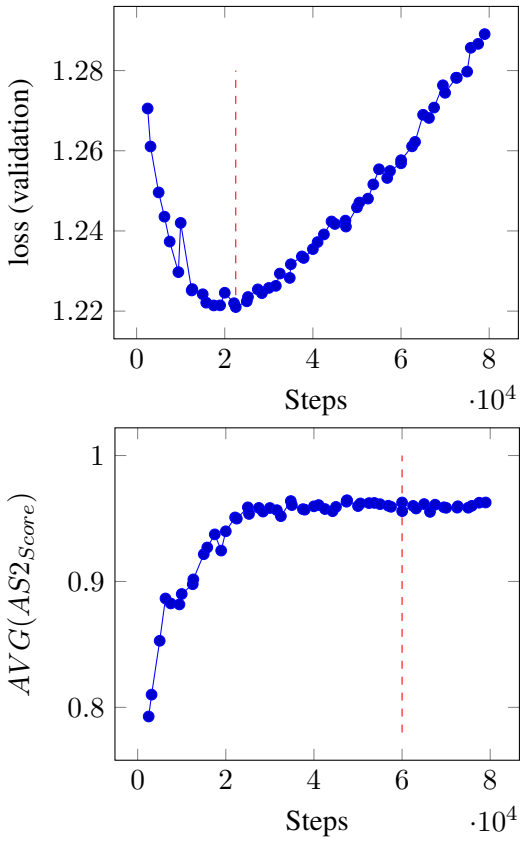

Figure 4: These plots show the differences between using the dev. loss and the average AS2 model score as the measure for model checkpoint selection. Notice that the vertical dashed red line is used to identify the best checkpoint (which is the one with the lowest loss in the first plot, and the one with the highest average AS2 score in the second).

Model	Transfer	Supervised	BLEU NLG answer	BLEU AS2 answer
RoBERTa	-	-	34.76	100.0
T5	**WS**	-	46.67	32.69
T5	**WS+LW**	-	38.47	65.55
T5	**WS+LW+SC**	-	38.85	63.76
T5	-	(Hsu et al., 2021)	36.94	61.22
T5	**WS+LW+SC**	(Hsu et al., 2021)	49.33	37.62

Table 7: Results using BLEU as the metric for measuring performance of answer generation. We train the GenQA models on MS-MARCO NLG for the supervised setting, and MS-MARCO QA for the knowledge transfer setting.

C Details of Human Evaluation

To evaluate our models we used the Amazon Mechanical Turk[8] framework. We designed an annotation task in which we showed to a pool of several high quality annotators (turkers) a question, a target answer generated from our models and, where possible (e.g., MS MARCO-NLG), a well-formed reference answer asking if the target answer was correct or not. For each QA pair (hit) we paid $0.1\$$ and we assigned 5 turkers. Specifically, we selected each Turker considering only masters with an approval rate greater than 95% and with at least 500 hits approved.

D Human Annotations v/s BLEU

Comparing human evaluation to BLEU scores for GenQA model outputs, we find that BLEU is not a reliable performance metric for this task and setting. For BLEU, we use two references for the generated target answer: (i) the manually written answer from MS-MARCO NLG, and (ii) the top ranked answer by the AS2 model that is being used as the teacher for knowledge transfer. The results are shown in Table 7 and appear to be quite random. Neither of the rankings induced by the BLEU metric correspond to the ranking induced by human evaluation (see Table 1).

E Qualitative Evaluation

In this section, we present some qualitative examples of answers generated by our models. In particular, we show (i) the differences between the generated answer with the answer candidates used as input for GenQA, and (ii) how we can manipulate the generated answers by forcing the decoding to start from different **SCO** bucket label tokens.

[8] https://www.mturk.com/

E.1 Generated Answer v/s Input Candidates

From qualitative analysis we observe that the answers generated by our knowledge transfer techniques are generally longer than the answers generated by a GenQA model trained in a fully supervised manner. We present three examples in Table 9. In the first example, both the GenQA answers are correct, and we observe that the AS2 selected top answer contains the correct answer string in it. In the second example, both the GenQA answers are incorrect, however, similar to the previous example, both the generated answers are shorter and syntactically improved versions of the AS2 selected answer. In the third example, the answer generated by our weakly supervised GenQA model is correct (the AS2 selected top answer is also correct) while that generated by the supervised GenQA model is incorrect. Overall, we notice that our weakly supervised models tend to copy and summarize the answer from the input candidates having the highest AS2 scores, while the supervised GenQA model generates more concise and shorter answers.

E.2 Forced Decoding using SCO

In this section, we aim to analyze how the **SCO** bucket tokens can be used to modify the quality of the generated answers from GenQA models. Specifically, we tested our GenQA models trained with **SCO** by forcing the decoder to generate answers starting from each of the different **SCO** bucket tokens: ({**'[_YES_]'**, **'[_PROBABLY_]'**, **'[_MAYBE_]'**, **'[_DOUBT_]'**, **'[_NO_]'**}). We present an anecdotal example in Table 8. We observe that the syntactical quality of the generated answers correlates with the **SCO** bucket token selected as the first token of the generated answer. Furthermore, higher confidence **SCO** bucket tokens (e.g., **'[_YES_]'**, **'[_PROBABLY_]'**) tend to generate shorter and more concise answers, while lower confidence **SCO** bucket tokens like **'[_DOUBT_]'** and **'[_NO_]'** can be used to generate longer sequences that are syntactically inferior.

"what city had a world fair in 1900"	
[_YES_]	"The 1900 world's fair was held in Paris, France."
[_PROBABLY_]	"The 1900 world fair was held in Paris, France."
[_MAYBE_]	"the 1900 world's fair, in paris which opened on march 19, 1900 is still one of the ... most important and influential exhibitions in modern history."
[_DOUBT_]	"in 1902 the exhibition denoting the new century at paris had been reopened to the public ... and there were only 12 sculptures by Louis britton and eight paintings by marcus thomas."
[_NO_]	"after a year of years' recovery but still having trouble finding an adequate buyer, the glass ... was shipped to france in 1900 for the world fair in paris and will be forever lost on this museum trail."

Table 8: An anecdotal example of GenQA using **SCO** by forcing the decoder to start the answer generation from different **SCO** bucket tokens. We observe that the high confidence **SCO** bucket answer is concise and correctly answers the question without the need of any further reasoning.

Inputs	
Question:	when was tom sawyer written
C1: (0.99)	Among his novels are The Adventures of Tom Sawyer (1876) and its sequel, Adventures... of Huckleberry Finn (1885), the latter often called The Great American Novel.
C2: (0.71)	When it first came out in 1876, however, it was comparatively a failure.
C3: (0.34)	Tom Sawyer (1973) G | 1h 43min | Adventure, Musical, Family | 15 March 1973 (USA) Tom Sawyer and... his pal Huckleberry Finn have great adventures on the Mississippi River, pretending to be pirates,... attending their own funeral, and witnessing a murder.
C4: (0.06)	When he was four, Twain's family moved to Hannibal, Missouri, a port town on the Mississippi River... that inspired the fictional town of St. Petersburg in The Adventures of Tom Sawyer and Adventures... of Huckleberry Finn.
Gold answer:	Tom sawyer written is on 1876.
Generated & Selected	
AS2 Selection:	Among his novels are The Adventures of Tom Sawyer (1876) and its sequel, Adventures... of Huckleberry Finn (1885), the latter often called The Great American Novel.
GenQA (Hsu et al., 2021):	Tom Sawyer was written in 1876.
(Ours) WS+LW+SC:	The Adventures of Tom Sawyer is a novel by Mark Twain, first published in 1876.

Inputs	
Question:	which planet is colder saturn or neptune
C1: (0.98)	Saturn: Due to its distance from the Sun, Saturn is a rather cold gas giant ... planet, with an average temperature of -178 °Celsius
C2: (0.90)	Neptune: With temperatures dropping to -218°C in Neptune's upper atmosphere, ... the planet is one of the coldest in our Solar System.
C3: (0.79)	The next four planets we will talk about–Jupiter, Saturn, Uranus, and Neptune–are ... very different to the terrestrial planets in that they have no real solid surfaces,... being dominated by thick, deep atmospheres.
C4: (0.74)	Jupiter is the fifth planet from the Sun, and is the largest planet in the solar system.
Gold answer:	The planet Saturn is colder.
Generated & Selected	
AS2 Selection:	Saturn: Due to its distance from the Sun, Saturn is a rather cold gas giant ... planet, with an average temperature of -178 °Celsius
GenQA (Hsu et al., 2021):	Neptune is colder than Saturn.
(Ours) WS+LW+SC:	Uranus is the fifth planet from the Sun, and is the largest planet in the solar system.

Inputs	
Question:	average of rn
C1: (0.99)	According to the Bureau of Labor Statistics lastest data, the average salary of a registered ... nurse in the United States is $69,790.
C2: (0.98)	The starting salary of a Registered Nurse can range from around $28,000-$50,000 per year on average.
C3: (0.90)	The starting hourly wage of an RN can range from $16.50-26.00 per hour.
C4: (0.81)	Although the average annual salary of a registered nurse in 2011 was $69,110, according to the ... BLS, salaries are affected by a number of factors.
Gold answer:	The average salary of a registered nurse is $69,790.
Generated & Selected	
AS2 Selection:	According to the Bureau of Labor Statistics lastest data, the average salary of a registered ... nurse in the United States is $69,790.
GenQA (Hsu et al., 2021):	The average salary of a registered nurse is $28,000 to $50,000 per year.
(Ours) WS+LW+SC:	The average salary of a Registered Nurse is $69,790 per year.

Table 9: Anecdotal examples from MS-MARGO NLG test set with answers generated from (i) our weakly supervised GenQA model (trained using WS+LW+SC), (ii) a fully supervised GenQA model (Hsu et al., 2021), (iii) the AS2 selected top answer and (iv) the manually written gold answer in the data set. We indicate the AS2 model scores for the input answer candidates to better reason about these examples.