Learning Answer Generation using Supervision from Automatic Question Answering Evaluators

Matteo Gabburo[1*], **Siddhant Garg**[2], **Rik Koncel Kedziorski**[3†], **Alessandro Moschitti**[2]
[1]University of Trento , [2]Amazon Alexa AI, [3]Kensho Technologies, Inc.
matteo.gabburo@unitn.it
{sidgarg,amosch}@amazon.com
rikka@kensho.com

Abstract

Recent studies show that sentence-level extractive QA, i.e., based on Answer Sentence Selection (AS2), is outperformed by Generation-based QA (GenQA) models, which generate answers using the top-k answer sentences ranked by AS2 models (a la retrieval-augmented generation style). In this paper, we propose a novel training paradigm for GenQA using supervision from automatic QA evaluation models (GAVA). Specifically, we propose three strategies to transfer knowledge from these QA evaluation models to a GenQA model: (i) augmenting training data with answers generated by the GenQA model and labelled by GAVA (either statically, before training, or (ii) dynamically, at every training epoch); and (iii) using the GAVA score for weighting the generator loss during the learning of the GenQA model. We evaluate our proposed methods on two academic and one industrial dataset, obtaining a significant improvement in answering accuracy over the previous state of the art.

1 Introduction

Recent research on retrieval-based Question Answering (QA) systems has been focused on two main tasks: (i) Answer Sentence Selection (AS2) e.g., (Garg et al., 2020), which, given a question and a list of answer candidates, chooses the most relevant answer that correctly answers the question; and (ii) Machine Reading (MR) e.g., (Chen et al., 2017), which, given a question and a reference text, involves finding a text span that directly answers the question. While effective, both the strategies (AS2 and MR) have some limitations: (i) the text might not include all the information necessary to answer a question, (ii) the text might include unnecessary, distracting information, or (iii) the text expresses the answer in a convoluted (indirect) format. Additionally, the text style and sentiment may be inappropriate for answering, or might be structurally too dependent on longer discourse context to enable usage as a stand-alone answer.

These drawbacks have motivated researchers to explore text generation systems for writing 'better' answers in the open-domain abstractive QA setting. For example, in the MR domain, RAG (Lewis et al., 2020b) generates an answer from a set of documents which are selected by dense passage retrieval models. For the domain of AS2, previous research has focused on summarizing answers from relevant paragraphs/evidences (Lewis et al., 2020a), or synthesizing information from the top ranked answer candidates of an AS2 system (Hsu et al., 2021; Muller et al., 2022; Gabburo et al., 2022).

The latter, termed as GenQA, has shown improvements in answer generation from the perspective of both answering accuracy and style suitability. The main distinguishing feature of GenQA from a generation-based MR approach is the length of the answer: the former uses an entire sentence as the target answer, while the latter in practice uses a short text (primarily targeting entity names). Therefore GenQA offers a more general and challenging research setting for answer generation.

Training effective GenQA models is made challenging by the cost and difficulty of obtaining large-scale, high quality training data. This typically requires human annotators to read the questions and relevant top k retrieved paragraphs/sentences, and then re-write them into a self-contained, and concise natural answer (sentence/paragraph).

Recent research (Vu and Moschitti, 2021; Bulian et al., 2022) has proposed effective automatic QA evaluation models based on transformer encoders for sentence-form answers. Training these QA evaluators only requires access to question answer pairs with annotations of correctness/incorrectness of the answers. This style of data annotation is much cheaper to perform than writing high-quality answers for training for GenQA models. In this work

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

we explore the novel idea of using automatic QA evaluators for training GenQA models, which enables a faster and cheaper design implementation.

In this paper, we reduce the amount of data needed for training a GenQA model using supervision from Automatic QA Evaluators. Our first contribution is to propose GAVA: an automatic QA evaluation approach that extends AVA (Vu and Moschitti, 2021) by (i) exploiting multiple reference answers and (ii) evaluating LM-generated answers instead of extracted answers. This way, we obtain a more robust and accurate QA evaluator that can effectively supervise the training of GenQA models. We propose three novel methods to use GAVA for refining the training of GenQA.

The first consists of (i) generating multiple possible answers using a baseline GenQA model for questions belonging to the GenQA training dataset, and (ii) then refining the set of generated answers by only retaining those with the highest GAVA scores (corresponding to "correct" or "high quality" answers). These generated answers are used as alternate gold standard answers (in addition to the annotators' written answers) to create additional training examples for GenQA. We term this approach GAVA-SDA (Static Data Augmentation).

The second approach extends GAVA-SDA, performing data augmentation dynamically at every epoch instead of off-line before training. This intuitively is more effective as the GenQA model continuously improves during the training. Specifically, at every epoch, we use GAVA to score the list of generated answers along with the k input answer candidates. We then use the top scoring answer as the GenQA target and the next top-k scoring answers as inputs for GenQA. We term this approach GAVA-DDA (Dynamic Data Augmentation).

The third approach uses GAVA as a scoring function for loss weighting during the training of GenQA. Specifically, we generate an answer using a GenQA model for a training sample, and weight the GenQA model loss of this instance using the GAVA score corresponding to the generated answer. Intuitively, this makes the GenQA model learn more from instances associated with higher GAVA-scoring answers (which corresponds to "correct" or "high quality" answers). We term this approach GAVA-LW (Loss Weighting).

We perform empirical evaluation on two academic and one industrial QA dataset (de-identified customer questions from Alexa personal assistant), and show that our three proposed techniques using GAVA for training a GenQA model produce significant improvements in answering accuracy over a baseline GenQA approach. We also show that the answers generated by these improved GenQA models consistently achieve higher GAVA scores on average than the baseline. We will release the code along with the trained GenQA and GAVA models at `https://github.com/amazon-science/wqa-genqa-gava` to enable easy replication of our experimental results.

2 Related Work

Answer Generation: Several research works (Izacard and Grave, 2021; Lewis et al., 2020b) have studied the problem of generating short answer spans (typically entity level) for MR systems. The most relevant of these works for GenQA is the work of Asai et al. (2022), that trains an answer generation model using the evidentiality of retrieved passages. Xu et al. (2021) uses decoder cross-attention patterns to generate extractive answer spans. Fajcik et al. (2021) generate answer spans by using a combination of a generative and extractive reader (aggregating information over multiple passages). An independent, but related line of research is question-based summarization, and there have been several research works in this field: (Iida et al., 2019; Deng et al., 2020).

Hsu et al. (2021) proposed the first formulation for generating complete answer sentences using evidences retrieved by an answer sentence selection (AS2) model. This model was termed GenQA, and it uses the top-k most relevant answer sentence candidates for a question as input context to an encoder-decoder model to generate a natural sounding complete answer sentence for this question. Muller et al. (2022) extend GenQA for multiple languages by using answer sentence candidates from multiple languages as input context for the GenQA model. Recently, Gabburo et al. (2022) propose training of GenQA models using unlabeled data by leveraging weak supervision from trained AS2 ranking models. This approach was shown to combine well with the supervised GenQA approach (Hsu et al., 2021) to improve the answering accuracy. Note that all of these approaches are different from the ones described in the previous paragraph as they aim to generate complete answer sentences, and not just short answer spans.

Evaluation of QA Systems: Token level simi-

larity metrics like BLEU (Papineni et al., 2001), ROUGE (Lin, 2004), METEOR (Banerjee and Lavie, 2005), etc have been shown (Reiter, 2018) to not extend to sentence-form QA evaluation. For MR tasks, Yang et al. (2018) adapt BLEU and ROUGE metrics, but limit their evaluation to only yes-no and entity questions. Si et al. (2021) uses multiple gold reference answers (extracted from Knowledge Bases) to be used as references for evaluating answer span extraction.

There have been several learnable automatic metrics: BERTScore (Zhang et al., 2020), COMET (Rei et al., 2020), BLEURT (Sellam et al., 2020), etc. that have been proposed for some tasks in NLP such as MT and Summarization. These are based on transformer encoder models. Chen et al. (2019) proposed extending BERTScore for MR tasks using the question and paragraph context in addition to the answer. In similar line of work, Vu and Moschitti (2021) propose AVA which is an automatic QA evaluation metric that learns a transformer to encode the question, a reference gold answer and the target answer to be evaluated. Very recently, Bulian et al. (2022) also present similar findings as AVA, by proposing BEM which can be used for evaluating sentence-level extractive QA (AS2). AVA and BEM have not been evaluated for GenQA systems previously. Hsu et al. (2021) and (Gabburo et al., 2022) show that automatic metrics like BLEU, BLEURT, BERTScore do not correlate well with human judgements for evaluating accuracy of GenQA systems. We extend AVA for our experiments as the automatic QA evaluation system.

3 Automatic QA Evaluation using Multiple References (GAVA)

Vu and Moschitti (2021) propose AVA: an automatic evaluation models for QA based on a transformer encoder. It is applied to a question and a complete answer sentence to determine the correctness or incorrectness of the answer. Formally, we denote the AVA model with $\mathcal{E}$, which takes as input a question q, a target answer a, and a reference r, i.e., gold standard (GS) answer, and outputs a correctness probability score, $s \in [0, 1]$. AVA is trained on the same labeled data of AS2, i.e., question answer pairs, where each question has multiple annotated answer candidates available.

Though the AVA approach was empirically shown to be accurate for evaluating AS2 systems Vu and Moschitti (2021), there are some limitations associated with it: (i) several questions may have diverse correct answers, e.g., "Tell me a winner of the US Open?", and (ii) the same answer may be expressed in very different formats, e.g., "How old is Joe Biden?", "Biden is 80 years old" v/s. "The president has just entered his life's eighth decade". Furthermore, AVA does not use negative references when evaluating correctness of answers, while incorrect answers can also help refine the prediction of correctness/incorrectness. Note that most AS2 datasets have multiple annotated answers (combination of correct and incorrect labels), and thus it is straightforward to use them for building data to train AVA with multiple positive and negative references. Intuitively, a GenQA system synthesizes an answer using different pieces of information spread across many relevant candidates (while suppressing any irrelevant information), aligning well with the idea of using multiple references for QA evaluation.

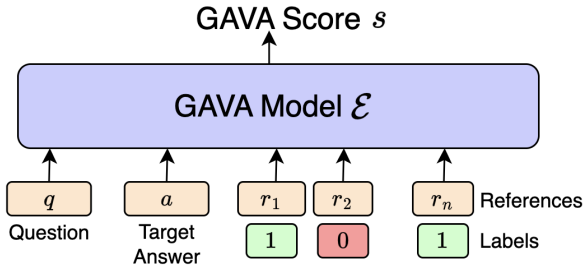

Figure 1: Multi-reference GAVA that uses multiple positive and negative reference answers to evaluate the correctness/incorrectness of a target answer for a particular question and produces a score $s \in [0, 1]$.

We term this approach: GAVA (AVA for generation-based models), which uses multiple references (combining positive and negative references) $\{r_1, r_2, \ldots, r_n\}$. Fig 1 shows the GAVA architecture: which uses a transformer encoder, taking as input: a question q, a target answer a, and n references. The information about the nature of the positive/negative references is encoded by prepending each reference with a prompt indicating the type of reference it is.

3.1 Comparison between GAVA and AVA

In subsequent sections of the paper, we will use the QA evaluator as a teacher to transfer knowledge for training GenQA models. We hypothesize that this knowledge transfer improves the answer generation capability of GenQA models by enabling the model to discriminate between good and poor supporting answer candidates. Thus a stronger QA

evaluator teacher will benefit in training more effective GenQA models. Here we perform an empirical comparison between GAVA and the baseline AVA model, to show that the former achieves a higher correlation with human annotations.

We consider two Answer Sentence Selection (AS2) datasets: WikiQA (Yang et al., 2015) and TREC-QA (Wang et al., 2007). We use a DebertaV3-Large (He et al., 2021) pre-trained model for both AVA and GAVA, and set n=5 reference answers per question for the latter. We measure the Pearson correlation between the human annotations and the two QA evaluators for each dataset under two configurations in Table 1: (i) **Extractive QA (AS2)** using the answer candidates available in the datasets, and (ii) **Generative QA (GenQA)** using answers that are written using a T5-Large GenQA model Hsu et al. (2021). The results indicate the empirical superiority of GAVA over AVA as an automatic QA evaluation metric, which stems from the usage of multiple references, including negative ones.

Model	Extractive QA (AS2)		Generative QA (GenQA)	
	WikiQA	TREC-QA	WikiQA	TREC-QA
AVA	0.632	0.797	0.678	0.647
GAVA	**0.676**	**0.842**	**0.690**	**0.671**

Table 1: Comparison between AVA and GAVA on WikiQA and TREC-QA. The models are compared in terms of Pearson correlation between the evaluation system prediction and the human evaluation. The best results for each dataset are highlighted in bold.

4 Generative QA (GenQA)

Answer generation-based QA (GenQA) refers to a text generation model for generating an answer to a question. Specifically, a generation model $\mathcal{M}$ is provided a question q and some context as the input, and generates an answer g. Hsu et al. (2021) proposed GenQA for generating natural sounding complete answer sentences by leveraging labeled datasets having high quality human authored answers as the targets for generation.

Specifically, a dataset, $\mathcal{D}$, for training a GenQA model, $\mathcal{M}$, contains examples of the format: $\Big(q, \{a_1, a_2, \ldots, a_k\}, t\Big)$ where q is the question, $\{a_1, \ldots, a_k\}$ are the k answer candidates used as input context to $\mathcal{M}$, and t is the target output answer (GS human authored answer).

Gabburo et al. (2022) extended this line of work by proposing a novel approach to train GenQA models using unlabeled data by transferring knowledge from an AS2 model (that is used to produce silver labels). Specifically, for each question q, 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.

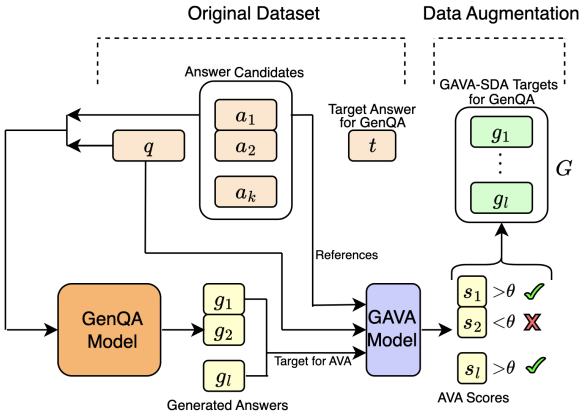

Figure 2: Illustrative representation of the GAVA-Static Data Augmentation (GAVA-SDA) approach.

5 GAVA for Training GenQA

In this section, we propose three approaches for training GenQA models using GAVA.

5.1 Static Data Augmentation (GAVA-SDA)

We create new training examples starting from $\mathcal{D}$, using a GAVA model, $\mathcal{E}$, and a base GenQA model $\mathcal{M}_0$ (trained only on $\mathcal{D}$). The new examples are added to $\mathcal{D}$ to create an improved training corpora for learning an improved GenQA model, $\mathcal{M}$.

For every question, $q \in \mathcal{D}$, along with its answer candidates $\{a_1, \ldots, a_k\}$ as input context, we apply $\mathcal{M}_0$ to generate multiple possible answers $\{g_1, g_2, \ldots, g_l\}$, using a probabilistic decoding approach (Wiher et al., 2022). Then, we apply the GAVA model to each of the generated answers g_i to obtain GAVA scores, s_i, of correctness i.e., $\mathcal{E}(q, g_i, \{a_1, .., a_k\})$. Then using a pre-defined threshold θ, we filter and pick only those answers $G = \{g_i : s_i \geq \theta\}$. We use this set of generated and filtered answers as alternate targets for generation to produce new examples for training GenQA, $\Big(q, \{a_1, \ldots, a_k\}, g\Big)$, where $\Big(q, \{a_1, \ldots, a_k\}, t\Big) \in \mathcal{D}$ and $g \in G$. Fig. 2 illustrates this approach.

It should be noted that: (i) θ is a parameter that can be tuned to increase the probability of correctness of the generated answers g_i. However, a very high θ will lead to filtering out a majority of the generated answers, leading to a very small augmented set (trade-off between size and quality). For our experiments, we used a value of θ that generated a large set of good quality diverse answers, as indicated by the GAVA score. (ii) Training a GenQA model on the augmented data can refine its predictions, biasing the generation towards "good-quality" answers. Overall, this produces improvement in quality and accuracy of the generated answers.

5.2 Dynamic Data Augmentation (GAVA-DDA)

We can improve the GAVA-SDA approach by producing new examples at regular intervals during the training, e.g., at the beginning of every epoch. This makes the data augmentation approach more adaptive, improving the learning of the GenQA model $\mathcal{M}$. As $\mathcal{M}$ improves during training, it will generate improved and higher quality answers, which can then be selected by GAVA to augment for the subsequent iterations. These 'higher-quality' answers can, in turn, improve the GenQA model's generation ability. In other words, instead of using a static base GenQA model $\mathcal{M}_0$, for the generation of the augmented data, we use the latest GenQA model $\mathcal{M}$, trained on the latest augmented data in the training routine.

Additionally, we refine the input context to the GenQA model during training. After obtaining the filtered set of generated and selected answers from GAVA: G, we combine them with the answers from D, i.e., $\mathbb{A} = \{a_1, \ldots, a_k, t\} \cup G$. We then use $\mathcal{E}$ to score $\mathbb{A}$, and pick the topmost ranked answer as the target for GenQA, and the next k top ranked answers as the input context for GenQA (following the same idea as (Gabburo et al., 2022)). Intuitively, this can improve the quality of both the input context that the GenQA model is using for generation, as well as the output target answer.

We combine the above two modifications into a single approach and call this Dynamic Data Augmentation (GAVA-DDA).

5.3 Loss Weighting (GAVA-LW)

GAVA-SDA and GAVA-DDA transfer the knowledge of the GAVA evaluation model for training GenQA by augmenting training data. Both approaches do not modify the GenQA training approach. In contrast, GAVA-LW uses the GAVA score to modify the GenQA training loss.

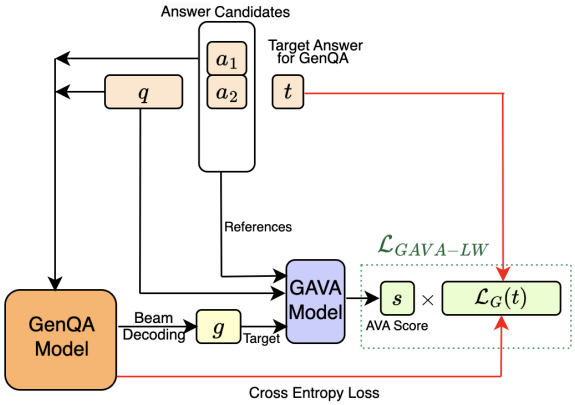

Figure 3: Illustrative representation of the GAVA-Loss Weighting (GAVA-LW) approach.

More formally, for training $\mathcal{M}$ with an example $\Big(q, \{a_1, \ldots, a_k\}, t\Big) \in \mathcal{D}$, we apply three steps: (i) compute the standard cross entropy loss $\mathcal{L}_{\mathcal{G}}(t)$ of GenQA model $\mathcal{M}$ on input $\Big(q, \{a_1, \ldots, a_k\}\Big)$ with target t (ii) run inference procedure on $\Big(q, \{a_1, \ldots, a_k\}\Big)$ to obtain model generation g (iii) compute the GAVA score of g using $\mathcal{E}(q, g, \{a_1, .., a_k\})$. We then use the GAVA score to weight the GenQA training loss as follows:

$$\mathcal{L}_{GAVA-LW} = \Big(1 - \mathcal{E}\big(q, g, \{a_1, .., a_k\}\big)\Big) \times \mathcal{L}_{\mathcal{G}}(t)$$

The GAVA-LW approach is illustrated in Fig.3. Intuitively, we want to make the model learn to improve its predictions for examples where the answer quality given by GAVA is low. Thus, for these examples, we give a weight to the training loss with the GAVA score. The $\mathcal{L}_{GAVA-LW}$ formulation (i) helps the model emphasize *harder* training samples (on which the model is currently not performing well) during learning, and (ii) trains a stronger more generalized system.

6 Experiments

6.1 Datasets

For training and evaluating our models, we consider two academic and one industrial dataset representing real world customer questions.

WQA Web Question Answers (WQA) is a public dataset defined in (Zhang et al., 2021). The dataset contains 149,513 questions, each associated with about 15 answer candidates. Both questions and answers are retrieved from a large-scale web index.

Each QA pair is manually annotated for answer correctness by professional annotators.

MS-MARCO Bajaj et al. (2018) proposed MS-MARCO, originally for MR tasks, comprising ∼800k queries retrieved from the Bing search engine along with ∼10 labeled answer passages. Following Gabburo et al. (2022), we transform MS-MARCO to obtain a large dataset of QA pairs, where the answers are sentences and not passages/paragraphs. Using a SOTA AS2 model (DeBERTav3-xl (He et al., 2021) trained on the ASNQ (Garg et al., 2020) dataset), we pick the top-2 ranked answer sentences from a positively labeled passage as positive answer candidates for the question. Similar to Gabburo et al. (2022), we randomly sub-sample 1k questions from the dev. set for evaluation (we use human annotations for our experiments and using the entire 100k dev. set would be extremely expensive to annotate).

IQAD Industrial QA Dataset (Garg and Moschitti, 2021; Di Liello et al., 2022b) is a large scale internal industrial QA dataset containing non-representative de-identified user questions from Alexa personal assistant. IQAD contains ∼10k questions, each having ∼200 answer candidates retrieved using a large scale web index (over 100M documents). Each question has ∼15 answer candidates with human annotations of correctness/incorrectness. Results on IQAD are presented relative to a baseline due to the data being internal.

6.2 Models and Evaluation

For our experiments we consider two types of models (i) GAVA evaluation models, as described in Section 3, and (ii) GenQA models, using techniques described in Section 5. For GAVA $\mathcal{E}$, we use a DebertaV3-Large (He et al., 2021) pre-trained model using up to n=5 reference answers per question. We train two GAVA models: one on WQA and one on IQAD, using the former for both the public datasets. [1] For GenQA, we consider a baseline model from Gabburo et al. (2022), which is a T5-Large (Raffel et al., 2020) encoder-decoder transformer trained using weak supervision on MS-MARCO. We consider this as the baseline GenQA model $\mathcal{M}_0$, and apply all of our techniques: GAVA-SDA, GAVA-DDA, GAVA-LW starting from this. Unless otherwise stated, we use θ=0.9 for GAVA-SDA and GAVA-DDA. For the GenQA models, we take k=5 answer candidates as inputs, and select the best checkpoint, corresponding to highest AVA-Score on the development set. We present complete experimental details in Appendix.

We perform human evaluation of the generated answers using Amazon MTurk (5 annotations per QA pair, taking average of these scores). We selected a pool of turkers having an approval rate higher than 95% with more than 500 approved hits. We designed our annotation task by providing the annotator with (i) the question, (ii) the generated answer, and (iii) a correct reference answer. For each hit (question + generated answer pair), we pay the turker 0.1$ and obtain 5 independent annotations. Using these annotations, we compute the answering accuracy over the entire evaluation set: number of correct answers divided by the total number of generated answers. We also evaluate models using the automatic GAVA metric.

6.3 Main Results

We evaluate GenQA models trained with our three proposed techniques in Table 2 using human evaluation of accuracy and GAVA-Score (automatic evaluation). We observe that across all datasets, our approaches outperform the baseline and are able to improve GenQA training, as evidenced by both human and automatic evaluation.

Specifically for WQA, we observe that the GAVA-SDA approach achieves the highest answering accuracy (improving an impressive +21.3% points over the baseline). The experiments on WQA indicate the ideal case, where we can have a GAVA model trained on the same dataset (due to availability of some annotations of correctness). We even observe improvement in the GAVA score for our approaches (which is expected, since we are using this model to supervise the training of GenQA). Interestingly, we do not see a perfect correlation between the human-induced and GAVA-induced relative ordering of the four techniques.

On MS-MARCO, we again observe that GAVA-SDA achieves the highest answering accuracy (+10.2% points over the baseline), and here there is a perfect correlation between the human evaluation and GAVA. This evaluation on MS-MARCO demonstrates the transferability of using GAVA for teaching GenQA across data distributions (the GAVA model used here is trained on WQA, as the sentence version of MS-MARCO does not have

[1] WQA contains human annotations of answer correctness which can be used as references for training a strong GAVA model. The answer sentence version of MS-MARCO does not contain human annotations of answer correctness.

Approach	WQA		MS-MARCO		IQAD	
	Accuracy	GAVA-Score	Accuracy	GAVA-Score	Accuracy	GAVA-Score
GenQA-WS (Gabburo et al., 2022)	0.655	0.409	0.775	0.770	Baseline	Baseline
(Ours) GAVA-SDA	**0.868**	0.498	**0.877**	**0.869**	+8.55%	-1.48%
(Ours) GAVA-DDA	0.769	0.439	0.843	0.855	**+9.85%**	**+0.54%**
(Ours) GAVA-LW	0.796	**0.527**	0.794	0.784	+8.81%	+0.51%

Table 2: Answering accuracy (manual evaluation) and AVA-Score on WQA, MS-MARCO and IQAD datasets. Results on IQAD are presented relative to the baseline (due to the data being internal). For WQA and MS-MARCO, we use an AVA model trained on WQA, and for IQAD we use an AVA model trained on IQAD. Best results for each dataset are highlighted in bold.

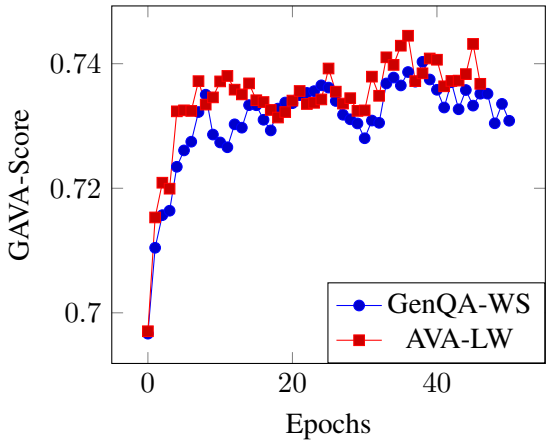

Figure 4: Comparison of baseline GenQA-WS and AVA-LW on WQA in terms of GAVA-Score (on validation split) varying across training epochs (GAVA-LW achieves higher GAVA-Score throughout training).

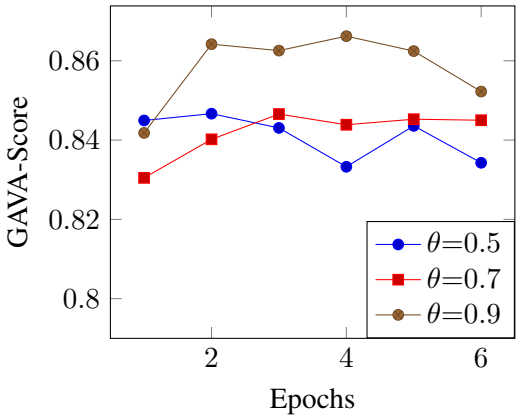

Figure 5: Comparison of three different GAVA-SDA models trained on MS-MARCO using different thresholds θ in terms of GAVA-Score (on validation split) varying across training epochs. The model trained with the largest value of θ achieves the highest GAVA-Score.

human annotations).

On the industrial dataset, IQAD, we observe that the GAVA-LW loss weighting approach achieves the highest accuracy (+9.85% relative improvement over the baseline). The results on IQAD lend support to our approaches extending to a real world scenario with actual customer questions.

6.4 Analysis and Ablations

Variation of GAVA Score over Training To understand how the GAVA score of our proposed techniques improves over the baseline, we plot its variation over the training epochs. We pick the MS-MARCO dataset and GAVA-LW as our approach to compare with, and present results in Fig. 4. From the figure, we observe that the GAVA-LW achieves a higher GAVA score than the GenQA baseline throughout the training regime. This demonstrates the knowledge transfer from the GAVA model for training GenQA, as the GenQA model is able to achieve a higher GAVA score over training epochs.

Variation of Threshold θ for GAVA-SDA: As discussed in Section 5, θ is a tunable parameter that decides the quantity v/s quality trade-off for data augmentation. We aim to study how the choice of θ affects the training of the GenQA model. We consider the GAVA-SDA approach, and the MS-MARCO dataset, and use three different values of θ=$\{0.5, 0.7, 0.9\}$. We follow the same experimental setting described in Section 6.2, and present results in Table 4. The results suggest a trend of achieving a higher final GenQA accuracy using a higher value of θ. This highlights that the quality of the generated answers (for augmenting) is more important for downstream answer generation than the quantity (Higher θ will pick "better-quality" answers, but increase the number of answers getting filtered out). Additionally we plot the GAVA-Score on the development set across training with these different values of θ in Fig. 5. We observe that the GAVA-Score for the model trained with θ=0.9 is always higher than the score of the other models across the entire training.

Correlation with other Automatic Metrics We perform a study to analyze how well can other automatic evaluation metrics perform for the task of evaluating answer generation. Specifically, we consider BLEU, BLEURT, and BERTScore. We

Approach	Manual	GAVA-Score	BLEURT	BERTScore	BLEU
GenQA-WS	0.775	0.770	0.587	0.492	30.8
GAVA-SDA(θ=0.5)	0.846	0.857	0.578	0.496	35.2
GAVA-SDA(θ=0.7)	0.861	0.864	0.576	0.491	34.9
GAVA-SDA(θ=0.9)	0.877	0.869	0.573	0.486	34.1
GAVA-DDA	0.843	0.855	0.627	0.541	38.1
GAVA-LW	0.794	0.784	0.627	0.502	32.2
Correlation (Pearson)		0.979	-0.429	-0.035	0.679
Correlation (Spearman's)		1	-0.812	-0.6	0.429

Table 3: Evaluation of different GenQA models using automatic evaluation metrics: BLEU, BLEURT, BERTScore in addition to GAVA-Score on the MS-MARCO dataset. We present the correlation each metric achieves with human annotation. GAVA achieves the best correlation with human evaluation of answer accuracy.

θ	| Augmented Set |	Accuracy	GAVA-Score
0.5	91,348	0.846	0.857
0.7	84,272	0.861	0.864
0.9	69,557	**0.877**	**0.869**

Table 4: Variation of GenQA accuracy by changing θ for GAVA-SDA approach, on the MS-MARCO dataset. We present human and automatic (GAVA-Score) evaluation. | Augmented Set | indicates the number of data augmentation examples created using a particular value of θ (Lower $\theta \rightarrow$ more augmentation examples).

use the MS-MARCO dataset, and evaluate several different GenQA models trained using our approaches. We evaluate performance using each of these automatic metrics and GAVA; and present the Pearson and Spearman's rank correlation between these metrics and the manual evaluation in Table 3. From the table, we observe that GAVA achieves the strongest correlation with human evaluation, highlighting its efficacy to be used as an automatic QA evaluation metric. Other text similarity matching metrics achieve poor correlation with the human annotation of answer correctness.

6.5 Qualitative results

We perform a qualitative analysis highlighting anecdotal examples to study success and failure cases of our answer-generation approach. Specifically, we pick the MS-MARCO dataset and the GAVA-DDA approach, and present both success and failure cases of answer generation to gain insights into the strengths and limitations of our approach.

Table 5 shows instances where GAVA-DDA successfully generates accurate answers. These examples highlight various sub-tasks that the model implicitly performs. Firstly, the model demonstrates its ability to synthesize information from multiple answer candidates. For example, for the question *"How do I get from DC to Alexandria VA?"*, the model correctly synthesized information from each of the reference answer candidates into the generated answer about the *Metrorail service* connecting the two locations. Second, the model exhibits reasoning capabilities highlighting identification of the correct reference answer candidate, along with improving it's style suitability for answering the input question. This is observed in the second example with the question *"How long should a central air conditioner last?"*, where the model identified the first reference *"10 to 20 years - sometimes longer"* to contain the most relevant information for answering the question. At times, the model acts as an answer sentence selection (AS2) model that simply re-ranks (without any modification) and generates one of the reference answer candidates.

Table 6 presents some examples where the model hallucinates and produces incorrect answers. This is highlighted in the question about Albany Minnesota's Population, where the model hallucinates and introduces an incorrect year in the generated answer, even when it is not present in any of the input reference candidates. Additionally, at times, the model may be unable to synthesize a good answer due to lacking evidence in the retrieved reference candidates. This is highlighted in the question about the earth's magnetic field. This limitation emphasizes the importance of reliable and accurate answer candidates for grounding the answer generation from the model.

7 Conclusion

In this paper we propose a novel training paradigm of learning answer generation systems (GenQA) using supervision from automatic QA evaluation met-

1)	Question	How do I get from DC to Alexandria VA?
	Answer #1	Check out the Metrorail system map and WMATA's official trip planner to plot the best route to your destination.
	Answer #2	To get to downtown DC from Alexandria, take the Yellow Line toward Greenbelt or Fort Totten, or take the Blue Line toward Largo Town Center or Stadium-Armory.Check out the Metrorail system map and
	Answer #3	WMATA's official trip planner to plot the best route to your destination.o get to downtown DC from Alexandria, take the Yellow Line toward Greenbelt or Fort Totten, or take the Blue Line toward Largo Town Center or Stadium-Armory.
	Answer #4	In addition to the King St-Old Town Metrorail station, Alexandria is serviced by three other stations: Braddock Road, Eisenhower Ave, and Van Dorn Street.
	GenQA Answer	The best way to get to Alexandria from Washington, DC is by Metrorail.
2)	Question	How long should a central air conditioner last?
	Answer #1	10 to 20 years - sometimes longer.
	Answer #2	You wouldn't run your car without regular maintenance and tune ups and you shouldn't run your a/c that way either - if you want it to last as long as possible. You should have a service tech come out once a year for a tune up.
	Answer #3	Source(s): 20 years working for a major manufacturer of central heating and air conditioning.
	GenQA Answer	The average life span of a central air conditioner is 10 to 20 years.
3)	Question	How long before a bearded dragon starts to grow
	Answer #1	When grown, males are larger than females, with notably bigger heads.
	Answer #2	Determining the gender of your bearded dragon before maturity isn't easy.
	Answer #3	Bearded dragons reach sexual maturity between the ages of 8 months and 18 months.
	GenQA Answer	Bearded dragons reach sexual maturity between the ages of 8 months and 18 months.

Table 5: Examples of correctly generated answers using GAVA-DDA approach on the MS-MARCO dataset. Example (1) highlights that the model is correctly able to synthesize a correct answer using the reference answer candidates for the question. Example (2) highlights a case where the GenQA model uses information from a single reference answer, but reformulates it's style using the question to present as an answer. Example (3) highlights a case where the GenQA model is effectively functioning as an answer ranker, as it directly copies the best answer candidate among the references to produce the generated answer.

1)	Question	Albany MN population ?
	Answer #1	Albany, Minnesota, as per 2017 US Census estimate, has a community population of 2,662 people.
	Answer #2	Albany is located in Stearns County, 20 miles west of St. Cloud and 80 miles northwest of Minneapolis/St. Paul on Interstate 94 (I-94). Albany has direct access to State Highway 238, which originates in Albany.
	GenQA Answer	The population was 2,662 at the 2010 census.
2)	Question	Which layer is responsible for the earth's magnetic field?
	Answer #1	Best Answer: The Earth's magnetic field is produced by convective currents in the outer core.
	Answer #2	The (presumably) molten iron core.
	Answer #3	The outer core is liquid iron.
	GenQA Answer	The outer core is liquid iron.

Table 6: Examples of incorrectly generated answers using GAVA-DDA approach on the MS-MARCO dataset. Example (1) highlights a case of hallucination during generation where the model introduces an incorrect year in the generated answer, even when it is not present in any of the input reference candidates. Example (2) highlights a failure case of GenQA where the model is unable to synthesize a good answer due to lacking evidence in the retrieved reference candidates.

rics based on transformer encoders. We propose three strategies: augmenting the training corpora with high GAVA-scoring generated answers for training the GenQA model (either statically once before training, or dynamically at every training epoch); and using the GAVA score for weighting the loss during the learning of the GenQA model. We perform empirical evaluation on two academic and one industrial dataset and show that our approaches outperform the baseline with both human annotations and automatic QA evaluation metrics (GAVA score). For future work, we plan to investigate how automatic QA evaluator based preferences align with human-annotated preferences for training larger LMs via reinforcement learning (Lambert et al., 2022). This would involve using GAVA as the RLHF reward model.

Limitations

The main limitation of our methodology is that the training of Generative Question Answering models requires the usage of large GPU resources, which may not be easily available to all researchers. Regarding the performance of the model and quality of the generated answers, our approach can be affected by possible bias induced by the evaluation system we are using. 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. Automatic QA evaluation systems do not achieve perfect correlation with human annotations, which indicates a gap with respect to human evaluation. For safety critical applications, human evaluation of generated answers still remains the best means for evaluation.

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.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Jannis Bulian, Christian Buck, Wojciech Gajewski, Benjamin Börschinger, and Tal Schuster. 2022. Tomayto, tomahto. beyond token-level answer equivalence for question answering evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 291–305, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Anthony Chen, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Evaluating question answering evaluation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 119–124, Hong Kong, China. Association for Computational Linguistics.

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.

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*.

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*, Seattle, Washington. 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. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11806–11816, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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.

Matteo Gabburo, Rik Koncel-Kedziorski, Siddhant Garg, Luca Soldaini, and Alessandro Moschitti. 2022. Knowledge transfer from answer ranking to answer generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9481–9495, Abu Dhabi, United Arab Emirates. 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.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing.

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.

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

Nathan Lambert, Louis Castricato, Leandro von Werra, and Alex Havrilla. 2022. Illustrating reinforcement learning from human feedback (rlhf). *Hugging Face Blog*. Https://huggingface.co/blog/rlhf.

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.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Benjamin Muller, Luca Soldaini, Rik Koncel-Kedziorski, Eric Lind, and Alessandro Moschitti. 2022. Cross-lingual open-domain question answering with answer sentence generation. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 337–353, Online only. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, page 311. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

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.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702. Association for Computational Linguistics.

Ehud Reiter. 2018. A Structured Review of the Validity of BLEU. *Computational Linguistics*, 44(3):393–401.

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. Association for Computational Linguistics.

Chenglei Si, Chen Zhao, and Jordan Boyd-Graber. 2021. What's in a name? answer equivalence for open-domain question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9623–9629, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Thuy Vu and Alessandro Moschitti. 2021. AVA: an automatic eValuation approach for question answering systems. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5223–5233. Association for Computational Linguistics.

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, Prague, Czech Republic. Association for Computational Linguistics.

Gian Wiher, Clara Meister, and Ryan Cotterell. 2022. On Decoding Strategies for Neural Text Generators. *Transactions of the Association for Computational Linguistics*, 10:997–1012.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

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

An Yang, Kai Liu, Jing Liu, Yajuan Lyu, and Sujian Li. 2018. Adaptations of ROUGE and BLEU to better evaluate machine reading comprehension task. In *Proceedings of the Workshop on Machine Reading for Question Answering*, pages 98–104, Melbourne, Australia. Association for Computational Linguistics.

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.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. Number: arXiv:1904.09675.

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.

Appendix

A Implementation Details

A.1 Computational Setting

We train our models on a machine with 8 NVIDIA A100 with 40Gb of VRAM and 1.1Tb of RAM. Our framework is based on Pytorch (Paszke et al., 2019) and hugging face (Lhoest et al., 2021; Wolf et al., 2020).

A.2 GAVA training

The multi-reference GAVA models are structurally closely related with the multi-sentence answer selection models proposed in (Di Liello et al., 2022a). We train two different multi-reference GAVA models starting from a DeBERTaV3-Large model (He et al., 2021) on two different datasets: WQA for the experiments on public dataset, and IQAD for the industrial scenario. For both the settings we use Adam (Kingma and Ba, 2015) with a learning rate of $1e-06$, a batch size of 32 and $fp32$ for 20 epochs. We shuffle the training set at the beginning of each epoch and we evaluate our model 4 times on the development set considering different performance measure. At the end of the training, we select the best checkpoint maximizing the area-under-the-curve (AUROC) on the development set.

A.3 GenQA based models training

We train our approaches on WQA, MS-MARCO and IQAD starting from a T5-Large model pre-trained using WS on MS-MARCO (Gabburo et al., 2022). To train the models we use Adam as optimizer with lr=$5e-06$, $fp32$, and batch size of 32 shuffling the training set at the beginning of each epoch. For MS-MARCO, we train the model for 15 epochs while for WQA and IQAD, we train the model for 30 epochs. We select the best checkpoint in term of GAVA-Score computed on the development set. We adopt an early stopping criterion stopping the training when the model does not improve for 3 epochs.