Accessible Abstract: Is Tim Cook the same person as Timothy Donald Cook? You might think so, but the way we train computers to answer questions would say they aren’t. We show that keeping track of multiple names (and it’s really simple) can create better question answering systems. Simply by adding alternate answers mined from knowledge bases, we can improve accuracy 1-2 points on major QA datasets.

Links:
- Research Talk [https://youtu.be/YdOmmhFKkf0](https://youtu.be/YdOmmhFKkf0)

Downloaded from [http://umiacs.umd.edu/~jbg/docs/2021_emnlp_answer_equiv.pdf](http://umiacs.umd.edu/~jbg/docs/2021_emnlp_answer_equiv.pdf)

Contact Jordan Boyd-Graber (jbg@boydgraber.org) for questions about this paper.
What’s in a Name?
Answer Equivalence For Open-Domain Question Answering
Chenglei Si
Computer Science
University of Maryland
clsi@terpmail.umd.edu

Chen Zhao
Computer Science
University of Maryland
chenz@cs.umd.edu

Jordan Boyd-Graber
CS, LSC, UMIACS, and iSchool
University of Maryland
jbg@umiacs.umd.edu

Abstract
A flaw in QA evaluation is that annotations often only provide one gold answer. Thus, model predictions semantically equivalent to the answer but superficially different are considered incorrect. This work explores mining alias entities from knowledge bases and using them as additional gold answers (i.e., equivalent answers). We incorporate answers for two settings: evaluation with additional answers and model training with equivalent answers. We analyse three QA benchmarks: Natural Questions, TriviaQA and SQuAD. Answer expansion increases the exact match score on all datasets for evaluation, while incorporating it helps model training over real-world datasets. We ensure the additional answers are valid through a human post hoc evaluation.1

1 Introduction: A Name that is the Enemy of Accuracy
In question answering (QA), computers—given a question—provide the correct answer to the question. However, the modern formulation of QA usually assumes that each question has only one answer, e.g., SQuAD (Rajpurkar et al., 2016), HotpotQA (Yang et al., 2018), DROP (Dua et al., 2019). This is often a byproduct of the prevailing framework for modern QA (Chen et al., 2017; Karpukhin et al., 2020): a retriever finds passages that may contain the answer, and then a machine reader identifies the (as in only) answer span.

In a recent position paper, Boyd-Graber and Börschinger (2020) argue that this is at odds with the best practices for human QA. This is also a problem for computer QA. A BERT model (Devlin et al., 2019) trained on Natural Questions (Kwiatkowski et al., 2019, NQ) answers Tim Cook to the question “Who is the Chief Executive Officer of Apple?” (Figure 1), while the gold answer is only Timothy Donald Cook, rendering Tim Cook as wrong as Tim Apple. In the 2020 NeurIPS Efficient QA competition (Min et al., 2021), human annotators rate nearly a third of the predictions that do not match the gold annotation as “definitely correct” or “possibly correct”.

Despite the near-universal acknowledgement of this problem, there is neither a clear measurement of its magnitude nor a consistent best practice solution. While some datasets provide comprehensive answer sets (e.g., Joshi et al., 2017), subsequent datasets such as NQ have not... and we do not know whether this is a problem. We fill that lacuna.

Section 2 mines knowledge bases for alternative answers to named entities. Even this straightforward approach finds high-precision answers not included in official answer sets. We then incorporate this in both training and evaluation of QA models to accept alternate answers. We focus on three popular open-domain QA datasets: NQ, TriviaQA and SQuAD. Evaluating models with a more permissive evaluation improves exact match (EM) by 4.8 points on TriviaQA, 1.5 points on NQ, and 0.7 points on SQuAD (Section 3). By augmenting training data with answer sets, state-of-the-art models improve on NQ and TriviaQA, but not on SQuAD (Section 4), which was created with a single evidence passage in mind. In contrast, augmenting the answer allows diverse evidence sources to provide an answer.
After reviewing other approaches for incorporating ambiguity in answers (Section 5), we discuss how to further make QA more robust.

2 Method: An Entity by any Other Name

This section reviews the open-domain QA (ODQA) pipeline and introduces how we expand gold answer sets for both training and evaluation.

2.1 ODQA with Single Gold Answer

We follow the state-of-the-art retriever–reader pipeline for ODQA, where a retriever finds a handful of passages from a large corpus (usually Wikipedia), then a reader, often multi-tasked with passage reranking, selects a span as the prediction.

We adopt a dense passage retriever (Karpukhin et al., 2020, DPR) to find passages. DPR encodes questions and passages into dense vectors. DPR searches for passages in this dense space: given an encoded query, it finds the nearest passage vectors in the dense space. We do not train a new retriever but instead use the released DPR checkpoint to query the top-k (in this paper, k = 100) most relevant passages.

Given a question and retrieved passages, a neural reader reranks the top-k passages and extracts an answer span. Specifically, BERT encodes each passage p, concatenated with the question q as

\[ P^i = \text{BERT}(p_i; q) \]

where \( P^i \) is the maximum sequence length and \( h \) is the hidden size of BERT. Three probabilities use this representation to reveal where we can find the answer. The first probability \( P_r(p_i) \) encodes whether passage \( i \) contains the answer. Because the answer is a subset of the longer span, we must provide the index where the answer starts \( j \) and where it ends \( k \). Given the encoding of passage \( i \), there are three parameter matrices \( w \) that produce these probabilities:

\[
P_r(p_i) = \text{softmax}(w_r(P^i_{[0,:]})); \tag{1}
\]

\[
P_s(t_j) = \text{softmax}(w_s(P^i_{[j,:]})); \quad \text{and} \tag{2}
\]

\[
P_e(t_k) = \text{softmax}(w_e(P^i_{[k,:]})). \tag{3}
\]

where \( P^i_{[0,:]} \) represents the \([\text{CLS}]\) token, and \( w_r, w_s \) and \( w_e \) are learnable weights for passage selection, start span and end span. Training updates weights with one positive and \( m - 1 \) negative passages among the top-100 retrieved passages for each question (we use \( m = 24 \)) with log-likelihood of the positive passage for passage selection (Equation 1) and maximum marginal likelihood over all spans in the positive passage for span extraction (Equations 2–3).

To study the effect of equivalent answers in reader training, we focus on the distant supervision setting where we know what the answer is but not where it is (in contrast to full supervision where we know both). To use the answer to discover positive passages, we use string matching: any of the top-k retrieved passages that contains an answer is considered correct. We discard questions without any positive passages. This framework is consistent with modern state-of-the-art ODQA pipelines (Alberti et al., 2019; Karpukhin et al., 2020; Zhao et al., 2020; Asai et al., 2020; Zhao et al., 2021, inter alia).

2.2 Extracting Alias Entities

We expand the original gold answer set by extracting aliases from Freebase (Bollacker et al., 2008), a large-scale knowledge base (KB). Specifically, for each answer in the original dataset (e.g., Sun Life Stadium), if we can find this entry in the KB, we then use the “common.topic.alias” relation to extract all aliases of the entity (e.g., [Joe Robbie Stadium, Pro Player Park, Pro Player Stadium, Dolphins Stadium, Land Shark Stadium]). We expand the answer set by adding all aliases. We next describe how this changes evaluation and training.

2.3 Augmented Evaluation

For evaluation, we report the exact match (EM) score, where a predicted span is correct only if the (normalized) span text matches with a gold answer exactly. This is the adopted metric for span extraction datasets in most QA papers (Karpukhin et al., 2020; Lee et al., 2019; Min et al., 2019, inter alia). When we incorporate the alias entities in evaluation, we get an expanded answer set \( \mathcal{A} \equiv \{a_1, ..., a_n\} \). For a given span \( s \) predicted by the model, we compute EM score of \( s \) if the span matches any correct answer \( a \) in the set \( \mathcal{A} \):

\[
\text{EM}(s, \mathcal{A}) = \max_{a \in \mathcal{A}} \{\text{EM}(s, a)\}. \tag{4}
\]

2.4 Augmented Training

When we incorporate the alias entities in training, we treat each retrieved passage as positive if it contains either the original answer or the extracted alias entities. As a result, some originally negative passages become positive since they may contain the aliases, and we augment the original training
Table 1: **Avg. Original Answers** denotes the average number of answers per question in the official test sets. **Matched Ans.** denotes the percentage of original answers that have aliases in the KB. **Avg. Augmented Answers** denotes the average number of answers in our augmented answer sets. **Last two rows:** number of positive questions (questions with matched positive passages) in the original / augmented training set for each dataset. NQ and TriviaQA have more augmented answers than SQuAD.

<table>
<thead>
<tr>
<th>Data</th>
<th>Model</th>
<th>Single Ans</th>
<th>Ans Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>NQ</td>
<td>Baseline</td>
<td>34.9</td>
<td>36.4</td>
</tr>
<tr>
<td></td>
<td>+ Augment Train</td>
<td>35.8</td>
<td>37.2</td>
</tr>
<tr>
<td>TRIVIAQA</td>
<td>Baseline</td>
<td>49.9</td>
<td>54.7</td>
</tr>
<tr>
<td></td>
<td>+ Augment Train</td>
<td>50.0</td>
<td>55.9</td>
</tr>
<tr>
<td>SQuAD</td>
<td>Baseline</td>
<td>18.9</td>
<td>19.6</td>
</tr>
<tr>
<td></td>
<td>+ Augment Train</td>
<td>18.3</td>
<td>18.9</td>
</tr>
</tbody>
</table>

Table 2: Evaluation results on QA datasets compared to the original “Single Ans” evaluation under the original answer set, using the augmented answer sets (“Ans Set”) improves evaluation. Retraining the reader with augmented answer sets (“Augment Train”) is even better for most datasets, even when evaluated on the datasets’ original answer sets. Results are the average of three random seeds.

Then, we train on this augmented training set in the same way as in Equations 1–3.

### 3 Experiment: Just as Sweet

We present results on three QA datasets—NQ, TriviaQA and SQuAD—on how including aliases as alternative answers impacts *evaluation* and *training*. Since the official test sets are not released, we use the original dev sets as the test sets, and randomly split 10% training data as the held-out dev sets. All of these datasets are extractive QA datasets where answers are spans in Wikipedia articles.

**Statistics of Augmentation.** Both SQuAD and TriviaQA have one single answer in the test set (Table 1). While NQ also has answer sets, these represent annotator ambiguity given a passage, not the full range of possible answers. For example, different annotators might variously highlight Lenin or Chairman Lenin, but there is no expectation to exhaustively enumerate all of his names (e.g., Vladimir Ilyich Ulyanov or Vladimir Lenin). Although the default test set of TriviaQA uses one single gold answer, the authors released answer aliases minded from Wikipedia. Thus, we directly use those aliases for our experiments in Table 2. Overall, a systematic approach to expand gold answers significantly increases gold answer numbers.

TriviaQA has the most answers that have equivalent answers, while SQuAD has the least. Augmenting the gold answer set increases the positive passages and thus increases the training examples, since questions with no positive passages are discarded (Table 1), particularly for TriviaQA’s entity-centric questions.

**Implementation Details.** For all experiments, we use the *multiset.bert-base-encoder* checkpoint of DPR as the retriever and use *bert-base-uncased* for our reader model. During training, we sample one positive passage and 23 negative passages for each question. During evaluation, we consider the top-10 retrieved passages for answer span extraction. We use batch size of 16 and learning rate of 3e-5 for training on all datasets.

**Augmented Evaluation.** We train models with the original gold answer set and evaluate under two settings: 1) on the original gold answer test set; 2) on the answer test set augmented with alias entities. On all three datasets, EM score improves (Table 2). TriviaQA shows the largest improvement, as most answers in TriviaQA are entities (93%).

**Augmented Training.** We incorporate the alias answers in training and compare the results with single-answer training (Table 2). One check that this is encouraging the models to be more robust and not a more permissive evaluation is that augmented training improves EM by about a point even...
Table 4: Annotation of fifty sampled augmented training examples from each dataset. Most training examples are still correct except for SQuAD, where additional answers are incorrect a third of the time. How the new answers are wrong is broken down in the bottom half of the table.

<table>
<thead>
<tr>
<th>Error Type</th>
<th>NQ</th>
<th>SQuAD</th>
<th>TriviaQA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct</td>
<td>48</td>
<td>31</td>
<td>41</td>
</tr>
<tr>
<td>Debatable</td>
<td>0</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Wrong</td>
<td>1</td>
<td>16</td>
<td>6</td>
</tr>
<tr>
<td>Invalid</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Non-equivalent</td>
<td>1</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>Wrong context</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Wrong alias</td>
<td>0</td>
<td>10</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 4: Annotation of fifty sampled augmented training examples from each dataset. Most training examples are still correct except for SQuAD, where additional answers are incorrect a third of the time. How the new answers are wrong is broken down in the bottom half of the table.

on the original single answer test set evaluation. However, TriviaQA improves less, and EM decreases on SQuAD with augmented training. The next section inspects examples to understand why augmented training accuracy differs on these datasets.

Freebase vs Wikipedia Aliases. We present the comparison of using Wikipedia entities and Freebase entities for augmented evaluation and training on TriviaQA. We show the augmented evaluation and training results in Table 3. Using Wikipedia entities increases in EM score under augmented evaluation (e.g., the baseline model scores 54.13 under Wiki-expanded augmented evaluation, as compared to 51.75 under Freebase-expanded augmented evaluation). This is mainly because TriviaQA answers have more matches in Wikipedia titles than in Freebase entities. On the other hand, the difference between the two alias sources is rather small for augmented training. For example, using Wikipedia for answer expansion improves the baseline from 49.31 to 49.42 under single-answer evaluation, while using Freebase improves it to 49.53.

4 Analysis: Does QA Retain that Dear Perfection with another Name?

A sceptical reader would rightly suspect that accuracy is only going up because we have added more correct answers. Clearly this can go too far... if we enumerate all finite length strings we could get perfect accuracy. This section addresses this criticism by examining whether the new answers found with augmented training and evaluation would still satisfy user information-seeking needs (Voorhees, 2019) for both the training and test sets.

Accuracy of Augmented Training Set. We annotate fifty passages that originally lack an answer but do have an answer from the augmented answer set (Table 4). We classify them into four categories: correct, debatable, and wrong answers, as well as invalid questions that are ill-formed or unanswerable due to annotation error. The augmented examples are mostly correct for NQ, consistent with its EM jump with augmented training. However, augmentation often surfaces wrong augmented answers for SQuAD, which explains why the EM score drops with augmented training.

We further categorize why the augmentation is wrong into three categories (Table 6): (1) Non-equivalent entities, where the underlying knowledge base has a mistake, which is rare in high quality KBs; (2) Wrong context, where the corresponding context is not answering the question; (3) Wrong alias, where the question asks about specific alternate forms of an entity but the prediction is another alias of the entity. This is relatively common
in SQuAD. We speculate this is a side-effect of its creation: users write questions given a Wikipedia paragraph, and the first paragraph often contains an entity’s aliases (e.g., “Vladimir Ilyich Ulyanov, better known by his alias Lenin, was a Russian revolutionary, politician, and political theorist”), which are easy questions to write.

Accuracy of Expanded Answer Set. Next, we sample fifty test examples that models get wrong under the original evaluation but that are correct under augmented evaluation. We classify them into four categories: correct, debatable, wrong answers, and the rare cases of invalid questions. Almost all of the examples are indeed correct (Table 5), demonstrating the high precision of our answer expansion for augmented evaluation. In rare cases, for example, for the question “Who sang the song Tell Me Something Good?”, the model prediction Rufus is an alias entity, but the reference answer is Rufus and Chaka Khan. The authors disagree whether that would meet a user’s information-seeking need because Chaka Khan, the vocalist, was part of the band Rufus. Hence, it was labeled as debatable.

5 Related Work: Refuse thy Name

Answer Annotation in QA Datasets. Some QA datasets such as NQ and TyDi (Clark et al., 2020) n-way annotate dev and test sets where they ask different annotators to annotate the dev and test set. However, such annotation is costly and the coverage is still largely lacking (e.g., our alias expansion obtains many more answers than NQ’s original multi-way annotation). AmbiqQA (Min et al., 2020) aims to address the problem of ambiguous questions, where there are multiple interpretations of the same question and therefore multiple correct answer classes (which could in turn have many valid aliases for each class). We provide an orthogonal view as we are trying to expand equivalent answers to any given gold answer while AmbiqQA aims to cover semantically different but valid answers.

Query Expansion Techniques. Automatic query expansion has been used to improve information retrieval (Carpineto and Romano, 2012). Recently, query expansion has been used in NLP applications such as document re-ranking (Zheng et al., 2020) and passage retrieval in ODQA (Qi et al., 2019; Mao et al., 2021), with the goal of increasing accuracy or recall. Unlike this work, our answer expansion aims to improve evaluation of QA models.

Evaluation of QA Models. There are other attempts to improve QA evaluation. Chen et al. (2019) find that current automatic metrics do not correlate well with human judgements, which motivated Chen et al. (2020) to construct a dataset with human annotated scores of candidate answers and use it to train a BERT-based regression model as the scorer. Feng and Boyd-Graber (2019) argue for instead of evaluating QA systems directly, we should instead evaluate downstream human accuracy when using QA output. Alternatively, Risch et al. (2021) use a cross-encoder to measure the semantic similarity between predictions and gold answers. For the visual QA task, Luo et al. (2021) incorporate alias answers in visual QA evaluation. In this work, instead of proposing new evaluation metrics, we improve the evaluation of ODQA models by augmenting gold answers with aliases from knowledge bases.

6 Conclusion: Wherefore art thou Single Answer?

Our approach for matching entities in a KB is a simple approach to improve QA accuracy. We expect future improvements—e.g., entity linking source passages would likely improve precision at the cost of recall. Future work should also investigate the role of context in deciding the correctness of predicted answers. Beyond entities, future work should also consider other types of answers such as non-entity phrases and free-form expressions.

As the QA community moves to ODQA and multilingual QA, robust approaches will need to holistically account for unexpected but valid answers. This will better help users, use training data more efficiently, and fairly compare models.

Acknowledgements

We thank members of the UMD CLIP lab, the anonymous reviewers and meta-reviewer for their suggestions and comments. Zhao is supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via the BETTER Program contract 2019-19051600005. Boyd-Graber is supported by NSF Grant IIS-1822494. Any opinions, findings, conclusions, or recommendations expressed here are those of the authors and do not necessarily reflect the view of the sponsors.
References


Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2021. Generation-Augmented Retrieval for Open-domain Question Answering. arXiv, abs/2009.08553.


