Answer Consolidation: Formulation and Benchmarking

Wenxuan Zhou1, Qiang Ning2, Heba Elfardy2, Kevin Small2, Muhao Chen1
1University of Southern California, 2Amazon
{zhouwenx,muhaoche}@usc.edu {qning,helfardy,smakevin}@amazon.com

Abstract

Current question answering (QA) systems primarily consider the single-answer scenario, where each question is assumed to be paired with one correct answer. However, in many real-world QA applications, multiple answer scenarios arise where consolidating answers into a comprehensive and non-redundant set of answers is a more efficient user interface. In this paper, we formulate the problem of answer consolidation, where answers are partitioned into multiple groups, each representing different aspects of the answer set. Then, given this partitioning, a comprehensive and non-redundant set of answers can be constructed by picking one answer from each group. To initiate research on answer consolidation, we construct a dataset consisting of 4,699 questions and 24,006 sentences and evaluate multiple models. Despite a promising performance achieved by the best-performing supervised models, we still believe this task has room for further improvements.

1 Introduction

Open-domain question answering (QA) systems (Voorhees, 1999) aim to answer natural language questions using large collections of reference documents, contributing to real-world applications such as intelligent virtual assistants and search engines. Current QA systems (Zhu et al., 2021) usually adopt a three-stage pipeline consisting of: (1) a passage retriever (Yang et al., 2019; Karpukhin et al., 2020) that selects a small set of passages relevant to the question, (2) a machine reader that examines the retrieved passages and extracts (Wang et al., 2017, 2019) or abstracts (Lewis et al., 2020) the candidate answers, and (3) an answer reranker (Wang et al., 2018; Kratzwald et al., 2019) that fuses features from previous stages to either select one final answer or return the top-ranked answer from the previous stage.

Current QA research (Joshi et al., 2017; Kwiatkowski et al., 2019) primarily examines the case where each question is assumed to have a single correct answer. However, in practice, many questions can have multiple correct answers. For example, the question “Is coffee good for your health?” can be answered with respect to different aspects (e.g., “coffee can help you with weight loss”, “coffee can cause insomnia and restlessness”). To correctly identify different aspects of answers to the same question while mitigating aspect-level redundancy, it is important to consolidate the answers. Answer consolidation is particularly desirable for applications such as intelligent assistants, where responses are desired to be both comprehensive and concise. Additionally, in scenarios where QA is used for knowledge extraction (Bhutani et al., 2019; Du and Cardie, 2020) or claim verification (Yin and Roth, 2018; Zhang et al., 2020), consolidation is also an essential step to identify salient knowledge or evidence while mitigating duplication.

To effectively recognize multiple aspects of answers in QA systems, our first contribution is to introduce and formalize the answer consolidation problem. Specifically, given a question paired with multiple answer snippets, answer consolidation first partitions the snippets into groups where each group represents a single aspect within the answer space. Once partitioned, the final answer set is produced by returning a representative snippet from each group. In this formulation, the answer consolidation task is a post-processing stage that takes predicted answer-mentioning snippets (in this work, sentences) from previous QA stages and produces an answer set that maximizes answer aspect coverage while minimizing answer duplication.

To foster research on the answer consolidation problem, our second contribution is the collec-
tion of a new dataset, namely, QUASI (Question-Answer consolidation). QUASI consists of 4,699 questions such that each question is paired with multiple answer-mentioning sentences grouped according to different aspects. Starting with a Quora-based question source (Chen et al., 2018), noting the potential for multi-aspect answers, we first retrieve 10 relevant answer sentences from the web for each question. These sentences are then examined by three crowd-sourced workers, who exclude sentences that do not contain an answer and group the remaining ones. Finally, individual sentence groupings from different workers are aggregated into a single partitioning. QUASI consists of 24,006 sentences and 19,676 groups, corresponding to an average number of 4.18 aspects per question and 1.22 sentences per group.

Our third contribution is a comprehensive benchmarking for the answer consolidation problem based on QUASI. Specifically, we consider two evaluation settings: (1) classification, where the model predicts whether two sentences are in the same group. (2) sentence grouping, where the model groups the answer sentences. We evaluate a wide selection of zero-shot and supervised methods, including various SoTA sentence embedding models (Reimers and Gurevych, 2019; Gao et al., 2021), cross-encoders (Devlin et al., 2019; Liu et al., 2019), and a newly proposed answer-aware cross-encoder model. In the supervised setting, the answer-aware cross-encoder achieves the best results based on the Matthew correlation coefficient (MCC) score of 87.8% (classification setting) and an adjusted mutual information (AMI) score of 68.9% (sentence grouping setting). As this performance is notably far from perfect, our findings indicate the need for future investigation on this meaningful, but challenging, task.

2 Related Work

QA with multiple answers. Many QA datasets have assumed that a question has a single correct answer (Joshi et al., 2017; Kwiatkowski et al., 2019), while in real scenarios, many questions can have multiple correct answers. Fewer datasets for QA or machine reading comprehension (MRC) have been proposed with the consideration of multiple answers. In extractive MRC, MASH-QA (Zhu et al., 2020) allows one question to be answered by multiple non-consecutive text spans. In abstractive MRC/QA, MS MARCO (Campos et al., 2016) simply treats different workers’ answers as different answers. DuReader (He et al., 2018) merges similar answers during data construction. QReCC (Ananth et al., 2021) allows one worker to provide multiple different answers. Beyond these, WebQuestions (Berant et al., 2013) and GooAQ (Khashabi et al., 2021) include lists of diverse answers, and TREC-QA (Bauďiš and Šedivý, 2015) uses regular expressions to capture multiple answers. However, none of the aforementioned efforts have investigated effective consolidation of multiple answers. In this work, we formally define and collect a dataset for the answer consolidation problem as a complement to previous work. From another perspective, AmbigQA (Min et al., 2020) focuses on the case where a question can be interpreted in different ways, leading to the question disambiguation task. This is fundamentally different from our work that partitions answers to the same question into different coherent subsets. Stance detection (Liu et al., 2021) is concerned with the focused problem of collecting approving/disapproving opinions for a yes-no question, unlike our studied problem where all multi-answer questions are considered and the answers are not limited to binary opinions.

Answer Summarization. Questions with multiple answers are common in online communities. For example, Liu et al. (2008) observe that no more than 48% of best answers on Yahoo! Answers are unique. Many efforts (Song et al., 2017; Chowdhury and Chakraborty, 2019; Fabbri et al., 2021) have been devoted to summarizing reusable answers in community QA. Particularly, AnswerSumm (Fabbri et al., 2021) proposes a dataset where different answers are rewritten to bullet points by humans. While training on summarization data may enable the model to return salient and non-redundant answers, such training only works for abstractive machine readers. A more related work is BERT-DDP (Fujita et al., 2020), which considers the problem of getting a diverse and non-redundant answer set. They construct a dataset based on Yahoo! Chiebukuro where workers are asked to provide an answer set given a question. However, the correct answer set is not unique when answers are equivalent. As they treat all but the annotated answer set as wrong, both training and inference are prone to false negatives. In this paper, we group the answers with respect to their aspects and provide a discriminant rule, such that the correct group assignment is unique.
Diverse passage retrieval. Many information retrieval efforts address the problem of retrieving diverse documents for a query (Clarke et al., 2008; Fan et al., 2018; Abdool et al., 2020). In QA, Min et al. (2021) examine answer diversity in passage retrieval and propose a self-supervised dynamic oracle training objective. However, as passages may contain irrelevant information to the question, the retriever faces the challenge of identifying and integrating answers in passages when assessing answer diversity. In this work, we consider a dedicated task of answer consolidation and leave the problem of identifying answers to previous QA stages.

3 Answer Consolidation Task

Motivation. Many questions can have multiple correct answers, including questions explicitly asking for a multi-answer list (e.g., What are the symptoms of flu?) or questions where different people have different opinions (e.g., debate questions), amongst others. To provide users with a comprehensive view of the answers, the QA system needs to actively identify different answers as opposed to only returning the most popular or top-ranked answer. Additionally, as the same answer may be repeated or paraphrased many times in the reference corpus (e.g., web), the QA system may also need to eliminate the redundant answers. We address these requirements within answer consolidation.

Basic concepts. When answering a specific question, different answers may be given regarding different perspectives, opinions, angles, or parts of the overall answer. We regard such answers as those pertaining to different aspects. Furthermore, we refer to two sentences as equivalent if they contain the same answer aspect(s) and distinct if they express different answer aspect(s). To better identify equivalent/distinct sentences, we propose the following operational discriminant rule: Given two answer-mentioning sentences $s_1, s_2$ for the same question $q$, we can rewrite the answers contained in $s_1$ and $s_2$ into yes-no questions $q'_1$ and $q'_2$, which can be answered by yes/no/irrelevant. Then, if $s_1$ and $s_2$ give coherent answers of yes/no to $q'_2$ and $q'_1$, respectively, then $s_1$ and $s_2$ are considered to represent equivalent aspects. Otherwise, they are considered distinct from each other.

We take the following example:

**Q**: Is coffee good for your health?

**S1**: Coffee can make you slim down.

**S2**: Coffee can relieve headache.

**S3**: Coffee can help with weight loss.

Then we rewrite the answers contained in sentences as the following questions:

**Q’1**: Can coffee make you slim down?

**Q’2**: Can coffee relieve headache?

**Q’3**: Can coffee help you with weight loss?

We can tell that S1 and S3 are equivalent, as they both give coherent answers (yes) to each others’ yes-no questions. We can also tell that S2 is distinct from S1 and S3, as it gives irrelevant answers to Q’1 and Q’3.

Task definition. A formal definition for answer consolidation is that given a question and a set of answer-mentioning sentences, answer consolidation aims at putting sentences into groups such that: (1) each sentence belongs to exactly one group, and (2) sentences from the same/different groups are equivalent/distinct. In this way, each sentence group corresponds to the same answer aspect(s).

We show in §4.3 that although this definition may fail for a pair of sentences if they are partially relevant, it only occurs for 2.6% of sentences, which shows that our operational task definition works well for the majority of sentences in practice.

In this work, we treat answer consolidation as a stand-alone process applied after QA retrieval such that we only take the question and answer-mentioning sentences as input. In this way, the answer consolidation model is independent of the retriever and the reader architectures, and can flexibly adapt to different QA systems. We show in §6.3 that taking sentences instead of answer spans as input leads to better performance.

4 Question-Answer Consolidation Dataset (QUASI)

In this section, we describe the creation of our Question-Answer consolidation dataset (QUASI) including corpus collection (§4.1) and dataset annotation (§4.2). We then provide statistical and quantitative analysis of QUASI (§4.3).

4.1 Corpus

We created QUASI based on the Quora question pairs (QQP) corpus (Chen et al., 2018), which consists of 364k questions pairs, originally designed for predicting whether pairs of questions have the
same meaning. We start with QQP since Quora questions have a high expected propensity of having multiple correct answers. In preprocessing, we removed questions containing spelling errors or non-English words using the Enchant library,\(^4\) and questions containing personal pronouns including \("I", "you", "we", "my", and "your"\) as we found that such questions frequently ask about very specific personal experiences for which the answers may not necessarily contain any noteworthy claims, opinions, or facts in the answer.

Next, we retrieved sentences that were likely to contain multiple answers to the questions. Given a question, we retrieved relevant sentences from the web using a SoTA industry QA retriever, where each sentence was associated with a relevance score and a URL. To ensure the questions have multiple answers, we first removed sentences retrieved from quora.com and kept the questions if the relevance scores of the top three retrieved sentences were larger than specified low-confidence threshold. Finally, we kept the top 10 sentences for each remaining question and pass to crowd workers for sentence group annotation.

4.2 Annotation

We used Amazon Mechanical Turk (AMT) to annotate QUAS1. Each AMT HIT consisted of a question and 11 sentences (including the top-10 relevance scores and one additional attention-check sentence) where the crowd workers were required to: (1) identify sentences that actually contain answers to the question and (2) put answer-mentioning sentences into sentence groups with respect to their aspects. The workers were allowed to skip an answer-mentioning sentence if it was hard to put it into any groups (e.g., sentences containing more than one aspect).

The AMT interface is shown in Figure 1. Annotation was performed by dragging the sentences between blue boxes. The sentence groups could be added or removed using the two buttons. To submit the HIT, workers needed to put all sentences into boxes corresponding to either a specific sentence group, \(\text{not an answer, or hard to put into groups}\).

Cost. Each HIT was assigned to three annotators with pay of $0.50/HIT, leading to target an hourly pay rate of $15. We randomly sampled 5k questions from §4.1 for HIT submission to AMT.

Figure 1: The interface used to collect the dataset. The second sentence is an attention-check sentence.

Quality Control. We used three strategies to ensure the annotation quality:

1. Workers selection. We only allowed crowd workers with acceptance rate \(\geq 98\%\) and had completed at least 5k hits to work on the task. We provided annotation guidelines and examples of 3 annotated hits to instruct the workers.

2. Qualification test. We manually annotated three hits as the qualification test. Workers were required to practice on these three hits and get the correct sentence groups on at least 2 hits to continue working on the task. We pay $0.05 for each submitted hit in the qualification test. Although we had provided detailed instructions, only 24% of workers passed the qualification test.

3. Attention checker. For all hits, we added an attention-check sentence that was randomly sampled from other questions, so that it was unlikely to answer the question. As a part of the task, the worker needed to identify that this attention-check sentence did not involve an answer, otherwise she/he would be blocked from continuing to work on the task.

Label aggregation. To ensure data quality, we aggregated worker annotations. To derive the sentence set for answer consolidation, we begin by only considering sentences put into any sentence group(s) by all crowd workers as eligible, keeping 37,588 out of 50k sentences. Next, we derived the aggregated sentence groups from AMT annotations. As we are not aware of existing methods for this process, we proposed the following algorithm for constructing new sentence groups. First, we sort the sentences by their relevance scores and create a sentence group with the most relevant sentence

\(^4\)https://abiword.github.io/enchant/
We provide statistical and qualitative analyses regarding QUAS1 in this section.

**Annotation quality.** We first analyze the quality of data annotation before label aggregation. In the first annotation task of identifying whether a sentence contains an answer, AMT workers achieved an inter-annotator Fleiss’ kappa of 0.62, an average agreement rate of 90.2%, and a worker agreement rate of 75.9%, respectively. These results show that the overall annotation quality is usable, but with room for improvement. Accordingly, to further improve the data quality, we only keep group annotations on which all workers agree (as stated in §4.2).

**Dataset statistics.** Our final dataset consists of 4,699 questions, 24,006 sentences, and 19,676 groups. On average, there are 4.18 groups per question, and 1.22 sentences per group. Specifically, 97.7% of questions have multiple aspects (sentence groups), and 45.4% of questions have at least one pair of equivalent sentences. In terms of sentence groups, 86.6% of groups have only one sentence, 8.8% of groups have two sentences, and the remaining 4.6% of groups have 3 or more sentences. Overall, this analysis shows that our dataset contains both multi-aspect and redundant answers that align with the challenges of answer consolidation.

**Types of equivalent sentences.** To get a better understanding of the required knowledge to identify equivalent sentences, we randomly sampled 100 sentence pairs in the same group and manually labeled the pairs with the types shown in Table 1. We observed that if the machine reader has the correct answer spans, 56% (formatting and ex-
act match) of the equivalent sentences could be directly identified by string comparison. Another 11% of the equivalent sentences only differed at the lexical level, which may be identified using lemmatization, removal of stop words, or a dictionary of synonym words. 30% of equivalent sentences are semantic variations such that identifying equivalence requires understanding of their meanings and potentially even commonsense reasoning. E.g., for the example given in Table 1, the answer consolidation model needs to understand that oxygen is good air and carbon dioxide corresponds to bad air. For the remaining 3% of pairs, we do not agree with the annotation. Either the sentences do not answer the question, or they do not contain the same aspect(s).

**Limits of the task definition.** During data annotation, 2.6% of answer-mentioning sentences are denoted as “hard to put into groups”. After inspection, we find that these sentences contain more than one aspect of answers. For example, given the question *What are the best places to visit and things to do in San Diego, CA?*, one sentence may be *The San Diego Zoo, Balboa Park, and SeaWorld are the top tourist attractions in San Diego.*, which contains 3 different answers. This sentence overlaps with multiple groups and thus cannot be placed in a single group. Given the low prevalence, we leave consideration of such sentences to future work.

## 5 Approach

In this section, we first tackle the classification setting of answer consolidation (§5.1). Given a question and answer-mentioning sentences, the task is to predict for a pair of sentences whether they are in the same group. We consider different types of models, including sentence embedding models, cross-encoders, and answer-aware cross-encoders.

Then we consider the sentence grouping setting, presenting the method of transforming pairwise predictions to sentence groups (§5.2). For all methods, we use RoBERTaLarge (Liu et al., 2019) as the encoder, noting that other pretrained language models (PLMs) can easily be incorporated as part of these methods.

### 5.1 Sentence Pair Classification

**Sentence embedding models.** Sentence embedding models (Reimers and Gurevych, 2019; Gao et al., 2021) produce for each sentence an embedding vector, with which we can use metrics such as cosine to calculate their similarity. Specifically, given a question $q$ and a sentence $s$, we first tokenize them to $X_q$ and $X_s$ using the RoBERTa tokenizer, and then concatenate them as inputs:

$$<s>X_q X_s</s>$$

Following Gao et al. (2021), we take the $<s>$ embedding in the last layer of PLM as the sentence embedding. Then for a pair of sentences, whether they are in the same group is decided by the cosine similarity of the sentence embedding. The similarity can be converted to binary predictions using the best threshold that is selected on the validation set.

The sentence embedding models can work in both zero-shot and supervised settings. In the zero-shot setting, we directly use the pretrained sentence embedding model to make predictions without fine-tuning. In the supervised setting, given a pair of sentence embedding $h_1$, $h_2$, and label $y \in \{0, 1\}$, where 0 and 1 mean not in/in the same group respectively, we fine-tune the PLM based on the following regression objective of sentence-transformers (Reimers and Gurevych, 2019):

$$L_{reg} = (\cos(h_1, h_2) - y)^2.$$  

**Cross-encoders.** Cross-encoders (Devlin et al., 2019; Liu et al., 2019) take a pair of sentences as the input and predict whether they are in the same group or not. Given a question $q$ and two answer-mentioning sentences $s_1$ and $s_2$, we first tokenize them as $X_q$, $X_{s_1}$, and $X_{s_2}$ using the RoBERTa tokenizer, and then take $X_q X_{s_1}$ and $X_q X_{s_2}$ as two segments of inputs, following the input formats of sentence pair classification tasks (Liu et al., 2019):

$$<s>X_q X_{s_1}</s><s>X_q X_{s_2}</s>$$

Prediction is independently performed on sentence pairs using a binary classifier on the first special (classification) token $<s>$ embedding in the last layer of the PLM. The cross-encoders work in both zero-shot and supervised settings. In the zero-shot setting, we fine-tune the model on the MNLI (Williams et al., 2018) dataset and take entailment as in the same group. In the supervised setting, given the sentence pair embedding (obtained from $<s>$) $h$ and the label $y$, we fine-tune

---

3We find that using 0 for not in the same group achieves better results than using -1.
the model using the binary cross-entropy loss:

\[ p = \sigma (\mathbf{w}^\top \mathbf{h}) , \]
\[ \mathcal{L}_{\text{bce}} = -(y \log p + (1 - y) \log (1 - p)) , \]

where \( p \) is the probability that the sentences are in the same group, \( \sigma \) is the sigmoid function, \( \mathbf{w} \) is a parameter of the classifier. In inference, we convert \( p \) to binary predictions using the best threshold selected on the validation set. The cross-encoders require predicting on all sentence pairs and have higher computational costs than sentence embedding models. However, we observe in experiments that cross-encoders consistently outperform sentence embedding models in the supervised setting.

**Answer-aware (A²) cross-encoders.** §4.3 shows that 56% of equivalent sentences can be directly identified if the model knows the answer spans. Therefore, we provide the answer consolidation models with answers generated from the UnifiedQA\textsubscript{LARGE} model (Khashabi et al., 2020). As the UnifiedQA is trained on both extractive and abstractive datasets, the answers may not be text spans of sentences. Specifically, given a question \( q \), two sentences \( s_1 \) and \( s_2 \), and the generated answers \( a_1 \) and \( a_2 \), we first tokenize them as \( X_p, X_{s_1}, X_{s_2}, X_{a_1}, \) and \( X_{a_2} \), respectively, then construct the input to cross-encoders as:

\[
\langle \text{a} \rangle X_q X_{s_1} X_{a_1} \langle / \text{a} \rangle \langle / \text{a} \rangle X_q X_{s_2} X_{a_2} \langle / \text{a} \rangle
\]

The training process and inference process are the same as the cross-encoders.

**5.2 Sentence Grouping**

In answer consolidation, our ultimate goal is to obtain the consolidated sentence groups. This is done in a two-step approach. The first step is to get the matrix of distances \( D \) between pairs of sentences for a question. We perform this step using models trained in the classification setting. For sentence embeddings, \( D \) is adopted as the pairwise cosine distance matrix. For cross-encoders, each entry of \( D \) equals to 1 minus the predicted probability for a sentence pair. As \( D \) derived in this way may not be always symmetric, we use \( \frac{1}{2} (D + D^\top) \) as the distance matrix instead.

The next step is to transform the distance matrix into sentence groups. Here we apply agglomerate clustering (Han et al., 2011). It uses a bottom-up strategy, starting from letting each sentence form its own cluster, and then recursively merging the clusters if their distance is smaller than a threshold. We use the average distance of sentence pairs as the inter-cluster distance measure and select the best threshold on the validation set. Agglomerate clustering stops when the distances between all clusters are larger than the threshold.

**6 Experiments**

In this section, we present the experimental setup (§6.1), show the main results (§6.2), conduct an ablation study (§6.3), and provide error analysis (§6.4).

**6.1 Experimental Setup**

**Dataset.** We randomly split the 4,699 questions into an 80/10/10 split, which serves as the training, validation, and test set, respectively.

**Evaluation metrics.** We use different evaluation metrics in the two evaluation settings. For the classification setting, we first use the micro \( F_1 \) measure. Considering that classes in the dataset are highly imbalanced (only 11% of sentence pairs are in the same group), we additionally use the Matthews correlation coefficient (MCC; Matthews 1975), which is considered a more class-balanced metric. For the sentence grouping setting, we use clustering metrics including adjusted rand index (ARI; Rand 1971) and adjusted mutual information (AMI; Nguyen et al. 2009). These two metrics take the predicted grouping and the ground-truth grouping, and measure the similarity between them. For all metrics, larger values indicate better performance, and a value of 100% indicates perfect classification/grouping.

**Configuration.** We implement the models using Huggingface’s Transformers (Wolf et al., 2020). The models are optimized with Adam (Kingma and Ba, 2015) using a learning rate of \( 1e^{-5} \), with a linear decay to 0. We fine-tune all models for 10 epochs with a batch size of 32 questions (including all associated sentence pairs). The best model checkpoint and thresholds are selected based on the validation set. We report the average results on 5 runs of training using different random seeds.

**Models.** We use RoBERTa\textsubscript{LARGE} (Liu et al., 2019) as the encoder for all models. For sentence embedding models, we try RoBERTa fine-tuned on two intermediate tasks: 1) SROBERTa (Reimers and Gurevych, 2019) is fine-tuned on natural language inference (NLI) datasets, achieving better
Table 2: Main results on the test set of QUAISI. The best results in the zero-shot and supervised settings are highlighted in **bold**.

6.2 Main Results

The experimental results on both the pairwise classification and sentence grouping settings are reported in Table 2. We observe that in the zero-shot setting, intermediate-task training improves answer consolidation, while the performance remains far behind supervised models. In the supervised setting, cross-encoders consistently outperform the sentence embedding models. Overall, the answer-aware cross-encoder intermediate tuned on MNLI (A²RoBERTa-MNLI) achieves the best results on all metrics, showing that intermediate-task training on MNLI improves performance. Besides, we find that answer-aware cross-encoders outperform regular cross-encoders, showing that answers generated by the machine reader provide additional information that helps consolidation.

6.3 Ablation Study

In this section, we study the model performance based on different input information given to supervised models. We denote the questions, sentences, and answers generated by UnifiedQA as Q, S, and A, respectively. The results are shown in Table 3. Overall, models trained on all inputs (Q+S+A) achieve better results than those that have observed only a subset of the available inputs on most metrics. Removing the sentences leads to the largest drops in performance (e.g., 20.9% in F₁ for SimCSE-RoBERTa and 22.3% in F₁ for RoBERTa-MNLI), which shows that sentences provide useful information for answer consolidation. Using sentences only leads to the second-largest drop in performance, showing that without grounding to questions and answers, consolidation is not simply addressed only with the sentences. Besides, removing questions also leads to more significant drops in performance than removing answers (e.g., 4.9% in F₁ for SimCSE-RoBERTa and 2.3% in F₁ for RoBERTa-MNLI). This shows that it is necessary to understand the answer equivalence within the question context in order to consolidate answers.

6.4 Error Analysis

To get a sense of what knowledge is needed to further improve model performance, we examined sentence pairs incorrectly classified as *not in the same group* by A²RoBERTa in the validation and test sets, where 318 out of 2,614 pairs (12.2%) are wrongly classified. We randomly sample 50 such error cases and categorize them by the answer-equivalence type as defined in Table 1. Of the 50
pains, 1 (2%) is from exact match, 8 (16%) are ambiguous, and the remaining 41 (82%) are from the semantic variation category, showing that it is the most challenging type to tackle.

We further study the specific causes of errors on the 42 unambiguous pairs. Examples of distinct error causes are described in Table 4. We find that 16.7% of the falsely classified sentence pairs contain one answer that entails the other instead of expressing the exact same answers, which should however be considered redundant answers by our definition. 80.9% of pairs are equivalent but require understanding the semantic meanings or entity-specific/commonsense knowledge. The rest 2.4% contain spelling errors that negatively affect model inference.

### 7 Conclusion

In this paper, we formulate and propose the answer consolidation task that seeks to group answers into different aspects. This process can be used to construct a final set of answers that is both comprehensive and non-redundant. We contribute the Question-Answer consolidation dataset (QUASI) for this task and evaluate various models, including sentence embedding models, cross-encoders, and answer-aware cross-encoders. While the best-performing supervised models have achieved promising performance, without that abundant annotation, unsupervised methods still remain far from perfect. This suggests room for further studies on more robust and generalizable solutions for answer consolidation that would largely benefit real-world open-domain QA systems.

### Acknowledgement

We appreciate the reviewers for their insightful comments and suggestions. Wenxuan Zhou and Muhao Chen are supported by the National Science Foundation of United States Grant IIS 2105329.

### References


Xuan Vinh Nguyen, Julien Epps, and James Bailey. 2009. Information theoretic measures for clustering comparison: is a correction for chance necessary? In ICML.


Ellen M. Voorhees. 1999. The trec-8 question answering track report. In TREC.


