# Explanations for CommonsenseQA: New Dataset and Models

**Shourya Aggarwal**[1], **Divyanshu Mandowara**[1], **Vishwajeet Agrawal**[1],
**Dinesh Khandelwal**[2], **Parag Singla**[1], **Dinesh Garg**[2]
[1]IIT Delhi, [2]IBM Research AI, India
aggarwal.shourya@gmail.com, divyanshu.mandowara92@gmail.com
vishwa.grawal@gmail.com, dikhand1@in.ibm.com
parags@cse.iitd.ac.in, garg.dinesh@in.ibm.com

## Abstract

CommonsenseQA (CQA) (Talmor et al., 2019) dataset was recently released to advance the research on common-sense question answering (QA) task. Whereas the prior work has mostly focused on proposing QA models for this dataset, our aim is to *retrieve* as well as *generate* explanation for a given *(question, correct answer choice, incorrect answer choices)* tuple from this dataset. Our explanation definition is based on certain desiderata, and translates an explanation into a set of *positive and negative common-sense properties (aka facts)* which not only *explain* the *correct answer choice* but also *refute* the *incorrect ones*. We human-annotate a first-of-its-kind dataset (called ECQA) of positive and negative properties, as well as free-flow explanations, for $11K$ QA pairs taken from the CQA dataset. We propose a latent representation based *property retrieval* model as well as a GPT-2 based *property generation model* with a novel two step fine-tuning procedure. We also propose a *free-flow explanation generation* model. Extensive experiments show that our retrieval model beats BM25 baseline by a relative gain of 100% in $F_1$ score, property generation model achieves a respectable $F_1$ score of 36.4, and free-flow generation model achieves a similarity score of 61.9, where last two scores are based on a human correlated semantic similarity metric.

**Question:**
Where is a frisbee in play likely to be?
**Answer Choices:**
outside park roof tree air

**Our Explanation:**
Positives Properties
1) A frisbee is a concave plastic disc designed for skimming through the air as an outdoor game.
Negative Properties
1) A frisbee can be outside anytime, even while not in play.
2) A frisbee can be in a park anytime, even while not in play.
3) A frisbee can be on a roof after play.
4) A frisbee can be in a tree after play.
Free-Flow (FF) Explanation
A frisbee is a concave plastic disc designed for skimming through the air as an outdoor game, so while in play it is most likely to be in the air. A frisbee can be outside or in a park anytime, and other options are possible only after play.

**CoS Explanation (Rajani et al., 2019):**
A frisbee floats on air.

Table 1: An example from CQA dataset along with our human-annotated explanation, containing positive properties to support correct answer choice (in green), negative properties to refute the incorrect choices (in red), and free-flow natural language explanation (in blue). The CoS explanation shown above from a prior work (Rajani et al., 2019) is less informative than ours.

## 1 Introduction

The field of automated question answering (QA) has witnessed a rapid progress in the past few years, sometimes beating even human performance (Zhang et al., 2020). The reasons behind this trend include (i) emergence of large-sized QA datasets such as SQuAD (Rajpurkar et al., 2016), HotpotQA (Yang et al., 2018), CommonsenseQA (Talmor et al., 2019), NaturalQA (Kwiatkowski et al., 2019), etc., and (ii) emergence of powerful, large scale, pre-trained, neural language models such as Transformer (Vaswani et al., 2017), BERT (Devlin et al., 2019), GPT (Brown et al., 2020), etc.

Much of the prior work in QA has focused on building models for only predicting the correct answer. In this paper, we tackle the problem

*Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics
and the 11th International Joint Conference on Natural Language Processing*, pages 3050–3065
August 1–6, 2021. ©2021 Association for Computational Linguistics

of generating an explanation for the answer of a question. While existing work has looked at explaining the answer predicted by a model (Amini et al., 2019), we take up the task of explaining the given gold (correct) answer in a model oblivious fashion (Jansen et al., 2018). We do this in the context of common-sense QA task and work with `CommonsenseQA` dataset. Explaining the known gold answers for common-sense QA is an important research problem and is far from being solved (Rajani et al., 2019). Two major hurdles in solving this problem include (i) lack of any desiderata for what constitutes an explanation (Horacek, 2017) and (ii) unavailability of QA datasets comprising high quality human-annotated explanations.

In this work, we address the entire stack of automatically generating explanations for the `CommonsenseQA` task. This includes setting up a desiderata for the explanation, curation of a dataset in accordance with the desiderata, proposing baselines models, and careful experimentation. Our overall contributions can be summarized as:

1. We present a set of characteristics (*refutation complete, comprehensive, minimal,* and *coherent*) for what constitutes an explanation. For any given *(question, correct answer choice, incorrect answer choices)* tuple, our explanation constitutes a set of *positive properties* to justify the correct answer choice and a set of *negative properties* to refute the incorrect ones.

2. We human annotate positive and negative properties for $11K$ QA pairs from the recently released `CommonsenseQA` (`CQA`) dataset (Talmor et al., 2019). We also curate a free-flow explanation for each QA pair. An example of our human annotated explanation is shown in Table 1[1]. We call our dataset as *ECQA (Explanations for CommonsenseQA)* and publicly release[2] it for future research.

3. We propose a set of models for the task of retrieval as well as generation of explanations. Our retrieval system, called as `eXplanation Retriever` (`XR`), represents properties in a latent space, and retrieves the facts against a `CQA` example from a given common-sense knowledge corpus. Our generation system, called as `eXplanation Generator` (`XG`), comprises a novel two step fine-tuned property generation model (`XGP`) to generate common-sense properties and a free-flow explanation generation model (`XGF`).

4. We perform extensive experiments to demonstrate the effectiveness of `XR` and `XG` systems. We use an $F_1$ based evaluation, calculated via exact property match when retrieving using *gold corpus* of facts. For property generation, and retrieval using a *silver corpus* in the absence of gold facts, $F_1$ is computed using a *semantic similarity metric* carefully picked to have a high correlation with human judgment. `XR` outperforms BM25 by a relative gain of $100\%$ for the gold corpus, and $70\%$ for the sliver corpus. `XGP` achieves a $F_1$ score of $36.4$, while `XGF` achieves a semantic similarity score of $61.9$. We publicly release our code and trained models [3].

## 2 Related Work

Bulk of the recent literature on automated QA is focused on either (i) proposing a new kind of dataset (Unger et al., 2014; Rajpurkar et al., 2016; Ling et al., 2017; Joshi et al., 2017; Trivedi et al., 2017; Welbl et al., 2017; Yang et al., 2018; Kwiatkowski et al., 2019; Talmor et al., 2019; Miao et al., 2020), or (ii) proposing a model with improved answer accuracy (Amini et al., 2019; Bhargav et al., 2020; Chen et al., 2020). As far as explanation in QA is concerned, we can either (i) explain the model's predicted answer, or (ii) explain the given gold answer without worrying about the model. For certain QA tasks (e.g. *KBQA*, *MathQA*, *VQA*), former explanation task is more meaningful. For other QA tasks (e.g. *Common-sense QA*, *ScienceQA*), the later form of explanation may be more meaningful. In both, one of the key challenge is to ground the definition of explanation.

Knowledge-Base QA task (Berant et al., 2013) requires the QA model to output a logical query (e.g. `SPARQL` or `SQL`) which is then executed over the underlying KB to get the answer. This logical query itself serves as an explanation. The MathQA task (Ling et al., 2017; Amini et al., 2019) requires the model to output a theorem-like proof, program, or algebraic construct which is executed to get the answer. Again, such a theorem serves as an explanation. For ScienceQA task, an expla-

[1] An additional example is given in Appendix A.1.

[2] `https://github.com/dair-iitd/ECQA-Dataset`

[3] `https://github.com/dair-iitd/ECQA`

| Datasets | Reasoning Type | Reasoning Steps | Refutation | Knowledge Base of Facts | Free Flow Explanation |
|---|---|---|---|---|---|
| WorldTree V2 | Scientific | Multi-hop | N | Y | N |
| COS-E | Common-sense | Single-hop | N | N | Y |
| QASC | Scientific | Two-hop | N | Y | N |
| OpenBookQA | Scientific | Multi-hop | N | Y | N |
| **ECQA** | **Common-sense** | Multi-hop | **Y** | **Y** | **Y** |

Table 2: Comparison of various properties of the different multi-choice QA explanation datasets. $4^{th}$, $5^{th}$, and $6^{th}$ columns refer to whether the dataset (i) provides refutation for incorrect choices, (ii) comes with a knowledge corpus of facts, (iii) provides a free-flow natural language explanation, respectively.

nation naturally comprises relevant scientific facts coming from a given corpus. WorldTree (Jansen et al., 2018) and WorldTree V2 (Xie et al., 2020) are corpora of elementary multiple-choice science questions with gold explanations for correct answer choice. OpenBookQA (Mihaylov et al., 2018) is a ScienceQA dataset built over the WorldTree corpus. QASC (Khot et al., 2020) is a middle school level multiple-choice ScienceQA dataset.

For other QA tasks, such as common-sense QA, reading comprehension QA (RCQA), visual QA (VQA), grounding the definition of explanation is not so obvious (Horacek, 2017) and hence, they lack labeled data as well. In the case of RCQA and VQA (Ghosh et al., 2018), there have been attempts to explain the predicted answers. Clark et al. (2020) studied the logical reasoning capacity of transformer based language models on various RCQA tasks. Bhagavatula et al. (2019) have proposed an NLI dataset for abductive reasoning. Wang et al. (2019) introduced the task of sense-making where given a pair of natural language statements, the goal is to pick the more sensible statement in the pair. Kotonya and Toni (2020) have proposed a dataset of explainable fact-checking in the public health domain and defined coherence properties to evaluate explanation quality.

As far as common-sense QA is concerned, we are not aware of much prior work on generating human understandable natural language explanations either for the *predicted answer* or for the given *gold answer*. CQA (Talmor et al., 2019) is a popular, multiple choice, common-sense QA dataset. The goal behind original CQA task is confined only till answering the questions and hence almost all the submissions (Ma et al., 2019; Khashabi et al., 2020; Zhu et al., 2020; Yang et al., 2020) to the leaderboard of the CQA dataset focus just on answering the question and not generating explanations. As far as explaining the gold answers of CQA questions are concerned, except for the works by Rajani et al. (2019), the literature is quite slim – both from the perspective of the explanation annotated datasets and models. Rajani et al. (2019) recently annotated explanations for the CQA dataset and called those explanations as CoS explanation (CoS-E for short). CoS-E are much shorter than our ECQA explanations (refer Table 1) and their aim was to leverage them in training a QA model so as to boost its answering accuracy. Their QA model first predicts CoS-E followed by leveraging the same to answer the question. Also, it is designed to generate only single-hop explanation which justifies only the correct answer choice and does not refute any incorrect answer choice. Table 2 compares our ECQA dataset with other relevant explanation datasets. To the best of our knowledge, both our ECQA annotation and XR, XG systems for explaining the CQA dataset are first-of-a-kind.

## 3 Explanations for CommonsenseQA

The broad idea behind explaining common-sense QA is to capture how humans would justify if a QA pair is presented to them. However, grounding a precise definition for this human justification is still hard due to subjectivity (Horacek, 2017). Furthermore, depending on the type of reasoning involved in the QA task, form and shape of an explanation may vary. Though, it is hard to give a single definition of the explanation for QA pairs coming from the CQA dataset, we believe one can still approach this by means of putting forward desiderata or desired characteristics of a well-formed explanation:

**Comprehensive:** Any information or reasoning, which is necessary to explain the answer should be present. This requires writing common-sense facts that are not present in the question but are essential for explanation.

**Refutation Complete:** While it should explain why an answer choice is correct, it should also explain why rest of the choices are incorrect or not best suited as answer.
**Minimal:** It should not contain any irrelevant or redundant information, especially the ones which are already present in the question.
**Coherent:** All the facts and statements should be written in a coherent and free-flow form to get a meaningful and natural explanation.

### 3.1 Formatting of the Explanation

The next question is how to translate above desiderata into a right format of the explanation for the purpose of machine generation. A naïve approach would be to consider it as a sequence of tokens or words, but it is unclear how to define metrics for deciding whether such a sequence satisfies the desiderata or not. So, we alternatively suggest two different formats for the explanations.

***1. Property Set* Format:** Given a CQA tuple $(q, a, I)$ where, $q$ is the question, $a$ is the correct answer choice, $I$ is the list of incorrect choices, this format suggests compiling a set $S$ of commonsense atomic facts (aka properties) such that each property in $S$ is required to either justify the correct answer choice or refute an incorrect answer choice. Furthermore, this format also requires the set $S$ to be minimal in the sense that dropping any property from $S$ may fail to either justify correct answer choice or refute one or more incorrect answer choices. Also, it's good to ensure that each property statement in $S$ is atomic in the sense that it is confined to a single fact and can't be further broken down into two independent facts. In summary, $S$ contains all those atomic properties that are needed for the explanation and nothing more.

Conceptually, we further partition this set $S$ into $S^+$ and $S^-$ and call the respective properties as *positive* and *negative*, respectively. Positive properties justify the correct answer choice and negative properties refute the incorrect answer choices. Our ECQA dataset has precisely annotated these sets for the QA pairs in CQA dataset. An example of such $S^+$ and $S^-$ sets is given in the Table 1.

**2. *Free Flow (FF)* Format:** This format essentially converts the *question*, the *answer choices*, and the *knowledge fact statements* from the sets $S^+$ and $S^-$ into a well-formed, coherent, free-flow style paragraph. This is important since this is how a human might perceive an explanation to be.

## 4 ECQA Dataset

We partnered with a private firm to crowdsource the annotations in *property set (S)* format for the CQA dataset. The firm utilized their in-house annotation and quality control teams for this purpose. For each question in the CQA dataset, an annotator was shown the question, its target concept (as given in CQA), all five answer choices, and the correct answer choice. As described earlier, the annotators were then asked to write the following: A set $S^+$ of positive properties, another set $S^-$ of negative properties and a free-flowing English explanation using the facts encapsulated in sets $S^+$ and $S^-$.

Each question in the CQA dataset comes with a label called *target concept*. We sorted all the questions according to their *target concepts* and provided questions of the same target concept to a single annotator. This prevented from conflicting statements appearing in positive and negative properties, and also helped speed up the annotation. An outcome of this exercise is shown in Table 1.

While it is difficult to guarantee that annotated property set is *comprehensive*, we tried to ensure it by asking annotators writing at least one property for each answer choice. We also asked them to write *simple* sentences by breaking down the complex sentences into two or more so that it helps in maintaining *minimality*. For the *comprehensiveness* and *minimality* of the final *free-flow* explanation, we explicitly asked them to include everything that appear in properties and avoid introducing anything from question and answer choices. The dataset quality at the ground level was ensured by a separate team of the partner firm, and random checks were performed by the authors as well.

### 4.1 Dataset Analysis

In this section, we highlight various insights regarding our ECQA dataset. There are a total of 10962 questions in the train and validation sets of CQA, and we get annotations for all of them. Top 3 rows of Table 3 gives the average count and the word length of properties per question. We also give the average word length of ECQA free-flow (FF) and CoS-E free-flow explanation for comparison.

In order to measure how much information ECQA free-flow annotations provide, we calculated number of distinct words (nouns, verbs, adjectives, and adverbs based on POS tagging) and report their average numbers in Table 4. The first three rows compare the information content in CQA, CoS-E

| Statistic | Avg. # per ques. | Avg. # words |
|---|---|---|
| $S^+$ | 2.05 | 9.94 |
| $S^-$ | 4.26 | 10.42 |
| $S$ | 6.32 | 10.27 |
| $FF$ | 1 | 49.52 |
| CoS-E | 1 | 6.82 |

Table 3: ECQA dataset statistics

and ECQA, while fourth and fifth rows tell what extra is present in a single annotation of the two explanation datasets w.r.t to CQA. This gives us a rough idea that the annotation introduces new entities and relations required for the explanation. Comparison using word-overlap metrics and additional data insights are presented in the Appendix A.9.

| Dataset | NN* | VB* | JJ* | RB* |
|---|---|---|---|---|
| CQA | 7.92 | 3.75 | 1.39 | 0.60 |
| CoS-E | 3.42 | 2.67 | 1.01 | 0.49 |
| ECQA | 10.22 | 7.83 | 3.12 | 2.20 |
| CoS-E \ CQA | 1.15 | 0.88 | 0.41 | 0.21 |
| ECQA \ CQA | 4.75 | 5.22 | 1.85 | 1.82 |

Table 4: Comparing information content through important words in CQA, CoS-E and ECQA.

### 4.2 Human Validation Experiments

We performed two human validation experiments to assess the absolute (and relative to CoS-E) quality of our ECQA dataset. In the first experiment, we asked three human judges to validate 100 samples each from our ECQA dataset. Out of 100 samples, 50 samples were common across judges (for normalization and correlation analysis) and 50 were different. Both $S^+$ and $S^-$ property sets were judged on a 3-points[4] scale to capture how well (negative)positive properties are justifying (in)correctness of (in)correct answer choice(s). Table 5 lists down the mean ($\mu$), standard deviation ($\sigma$), standard error ($e$), and average Pearson's correlation coefficient ($\rho$) for both positive and negative properties. $83.33\%$ of the samples were rated a perfect 2 score for positive properties and $66.67\%$ were rated perfect 2 for negative properties. We computed Pearson's correlation coefficient as follows. For each of the 50 commonly labeled samples, we first computed the *average score* across all the judges. Then, we computed Pearson's coefficient between scores of an individual judge and the corresponding average scores. Finally, we took the average of these individual coefficients across all judges (Gaona, 2014; Agirre et al., 2012). In the second experiment, we asked a set of three different human judges to compare the ECQA explanations with CoS explanations for the same 100 samples as in previous validation experiment. For each question, both explanations were randomly shuffled and resulting pair of explanations was called as $(E1, E2)$. The judges were asked to compare $E1$ with $E2$ on each of the following aspects: *comprehensiveness, refutation completeness, minimality/non-redundancy*, and *overall quality*. The comparison was logged on a 4-point scale[5]. Column 2 of Table 6 lists down the $\%$ times our explanation stood better than CoS-E. In all the four aspects, ECQA is judged to be outperforming CoS-E by a huge margin. Pearson's coefficient can be computed for each quality measure (column) and property (row) in Table 6, giving a $4 \times 4$ matrix of coefficient values with an average value of $0.774$. The detailed coefficient matrix is given in Appendix A.7.

| Aspect | $\mu$ | $\sigma$ | e | $\rho$ |
|---|---|---|---|---|
| $S^+$ | **1.799** | $0.566$ | $0.057$ | $0.765$ |
| $S^-$ | **1.588** | $0.604$ | $0.060$ | $0.748$ |

Table 5: Absolute Dataset Quality Experiment: Positive and Negative properties as rated by human judges

| Aspect | ECQA better | CoS-E better | Both Good | Both Bad |
|---|---|---|---|---|
| Comprehensive | **79.00** | 1.33 | 12.67 | 7.00 |
| RC | **84.33** | 0.33 | 1.67 | 13.67 |
| M/NR | **76.00** | 5.33 | 9.67 | 9.00 |
| Overall | **92.33** | 0.33 | 0.33 | 7.00 |

Table 6: Human Judgements for Relative Dataset Quality Experiment: ECQA and CoS-E. Numbers are averaged over 3 judges. RC: Refutation Complete and M/NR: Minimality/Non-redundancy

We do not report Cohen's Kappa score since it can have problems when dealing with skewed preferential distributions, i.e., when one choice is overwhelmingly preferred over the other (Feinstein and Cicchetti, 1990). In such scenarios, Kappa

[4] 0: complete garbage, 1: partial but incomplete reasoning, 2: satisfactory reasoning.

[5] 1: $E1$ better than $E2$, 2: $E2$ better than $E1$, 3: Both good, 4: Both bad

score can be low (and misleading) despite very high inter-annotator agreement due to the high chances of random agreement between the annotators. This is true in our case since ECQA explanations are highly preferred over CoS-E ones, by the judges.

## 5 Explanation Retrieval

This section describes our proposed eXplanation Retriever (XR) system to retrieve $S^+$ and $S^-$ property sets from a given property corpus for a given question. XR consists of two modules - (i) *property ranker*, and (ii) *property selector*. The experimentation code and trained models for this and the following section are publicly released. [6]

### 5.1 Property Ranker

Input to *property ranker* is a tuple $(q, a, c)$, where $q$ is a question (in natural language), $a$ is one of the answer choices (natural language) for the question $q$, and $c$ is token *'not'* if the answer choice $a$ is incorrect and empty string otherwise. Property ranker ranks the properties in the given corpus based on the given tuple $(q, a, c)$. The architecture of *property ranker* comprises two parameter shared sub-modules, namely *QA Encoder* ($E_1$) and *Property Encoder* ($E_2$). Module $E_1$ takes a tuple $(q, a, c)$ as input and outputs a vector $\boldsymbol{z}_{qac}$ in a 512-dimensional latent space $\mathcal{Z}$. Design of module $E_1$ is inspired by *sentence transformers (SBERT)* (Reimers and Gurevych, 2019) and comprises a BERT layer followed by single mean-pooling and a fully connected layer. We picked dimensions of the latent space through hyperparameter tuning on validation set. Module $E_2$ takes a property statement $p^*$ (in natural language) as input and returns a vector $\boldsymbol{z}_{p^*}$ in the same latent space $\mathcal{Z}$. $E_2$'s architecture is identical to the $E_1$, with parameter shared at every layer level.

**Training:** For training property ranker, we use SBERT library.[7] We initialize the BERT with *pre-trained bert-base-uncased* (Devlin et al., 2019). Weights of the fully connected layer are initialized randomly. In ECQA dataset, multiple properties from the corresponding sets $S^+$ or $S^-$ could form the relevant properties (each referred as $p^*$) for a given $(q, a, c)$. For the correct answer choice, all properties from the corresponding $S^+$ set are valid $p^*$. In case of incorrect choice, we first match the stemmed answer choice with the annotated properties from the set $S^-$ and pick all the matches as valid properties $p^*$, and remove all those tuples from the dataset where we cannot map to any property. Approximately 2% $(q, a, c)$ tuples get dropped from our experiments in this manner. Additionally, 32 questions in the original CQA dataset were marked as ambiguous by our annotators, and hence, we drop them from all our experiments. So there are multiple training examples for a query $(q, a, c)$ corresponding to each matched relevant property ($p^*$). Input part of each training example comprises a pair of $(q, a, c)$ and a relevant commonsense property $p^*$. Output part of each training example comprises vector representations $\boldsymbol{z}_{qac}$ and $\boldsymbol{z}_{p^*}$. The model is trained using a loss function, which forces $\boldsymbol{z}_{qac}$ and $\boldsymbol{z}_{p^*}$ to come closer in the latent space $\mathcal{Z}$. We use *multiple negatives ranking* (MNR) (Henderson et al., 2017) as the loss, which is negative *log-softmax* over similarity of $\boldsymbol{z}_{qac}$ and $\boldsymbol{z}_{p^*}$.[8]

**Inference:** For inference, we first start with a given property corpus $\mathcal{S}$ and encode all of them in the latent space using property encoder $E_2$. Now, we pass any given tuple $(q, a, c)$ through $E_1$ and obtain its latent vector representation $\boldsymbol{z}_{qac}$. Finally, we output a ranked list of the properties in the set $\mathcal{S}$ w.r.t to their cosine similarity with vector $\boldsymbol{z}_{qac}$.

### 5.2 Property Selector

The candidate properties retrieved by the *property ranker* are passed to this *property selection* module along with the query $(q, a, c)$. This *property selector* module then filters out a smaller size relevant properties set from the given larger size retrieved properties set . We experiment with two variants of this module - (i) Top-$k$, and (ii) Alignment-based Iterative Retriever (AIR) (Yadav et al., 2020).

Top-$k$ module picks top-$k$ properties from the ranked list returned by *property ranker* module. Top-$k$ is a naïve yet effective property selection module. We use ECQA dataset statistics to decide value for $k$. Based on Table 3, we select top-3 properties for the correct answer choice and top-1 property for an incorrect answer choice.

AIR (Yadav et al., 2020) is a state-of-the-art unsupervised explanation retrieval algorithm. It iteratively subselects multi-hop explanations from a given set by measuring the alignment between question, answer, and explanation sentences using

[6] https://github.com/dair-iitd/ECQA

[7] https://www.sbert.net/

[8] Cosine similarity and MSE losses did not perform well.

GloVe embeddings (Pennington et al., 2014). We use AIR to select the relevant set of properties from the top 50 properties given by the property ranker.

### 5.3 Experiments and Results for XR System

**Dataset:** We first randomly split our annotated ECQA dataset into a **70** : **10** : **20** partition to form *train*, *val*, and *test* sets, respectively. For all our experiments, we train the proposed *property ranker* using the ECQA train set and validate it using the ECQA val set. We experiment with both *gold* and *silver* corpus of properties during inference. The *gold* corpus consists of properties in the ECQA dataset (including training, val, and test sets). Similarly, the *silver* corpus is the set of train and val set of ECQA dataset and an additional large size corpus of common-sense facts, called as *Open Mind Common Sense* (OMCS) corpus (SINGH, 2002)[9]. The sizes of *gold* and *silver* corpus are 63975 and 901202, respectively.

**Metrics:** We use $F_1$ score between the sets of gold and retrieved properties to compare the performance for retrieval from the gold corpus. Retrieval from the silver corpus can never fetch us the ground-truth properties for a tuple $(q, a, c)$, since they are not contained in that corpus. One way to overcome this is to align the retrieved properties set to the ground truth properties set. We propose using *a maximum unweighted bipartite matching* based metric to find such an alignment score. For this, we first create a complete bipartite graph between the ground truth and the retrieved set of properties. To each edge in the graph, we assign a score based on the semantic similarity of the corresponding property sentences. For this we use lexical and semantic similarity metrics such as *STS-BERT score*[10], *SPICE* (Anderson et al., 2016), *CIDEr* (Vedantam et al., 2015), *METEOR* (Banerjee and Lavie, 2005), and *ROUGE* (Lin, 2004). We prune the edges in bipartite graph that have semantic similarity score less than some threshold value $(\tau)$. We then apply a maximum unweighted bipartite matching algorithm (Kuhn, 1955) on the pruned graph to obtain a matching of predicted *silver* properties with ground-truth *gold* properties. We then calculate usual $F_1$ score assuming the matched properties as the correctly retrieved ones. In Table 8 we report STS-BERT and SPICE based $F_1$ scores as these two metrics are the most correlated with human judgment. Results on other metrics are reported in Appendix A.8. Details regarding our experiment to discover correlation between the five semantic similarity metrics and the human judgment, and the procedure to obtain metric-specific thresholds $(\tau)$ is given in the Appendix A.6.

**Hyperparameters:** We tune hyperparameters of *property ranker* by maximizing the average cosine similarity over the validation set. Table 7 shows the best hyperparameters for our proposed *property ranker* obtained using grid search over validation set, where the parameters were searched in the given range. We use the model which achieves the best results on validation set in 5 epochs. We set warm-up steps and BERT hidden layer dimension to default values of 100[11] and 768, respectively.

| **Parameter** | **Value** | **Range** |
|---|---|---|
| Learning rate | $2{\times}10^{-5}$ | $[10^{-5}, 10^{-3}]$ |
| Dimension of $\mathcal{Z}$ | 512 | $\{128, 256, 512\}$ |
| Max training epochs | 5 | – |
| BERT sequence length | 30 | $\{20, 30\}$ |
| $k$ for positive properties | 3 | $\{3, 5, 10\}$ |
| $k$ for negative properties | 1 | $\{1, 2, 3\}$ |

Table 7: Best hyperparameters for *property ranker*. $\mathcal{Z}$ denote the latent space.

**Results:** We have also considered the popular information retrieval method BM25 (Robertson and Zaragoza, 2009) as another choice for the *property ranker* module. We have used the publicly available implementation of BM25[12]. Table 8 shows the performance comparison of XR system on *gold* and *silver* corpus for different choices of the *property ranker* and *property selector* modules. Our proposed *property ranker* with top-$k$ as *property selector* outperforms all other combinations with a significant margin. In Appendix A.3, we report some *anecdotal examples* of retrieved properties.

## 6 Explanation Generation

In this section we will describe our proposed GPT-2 (Radford et al., 2019) based explanation generation system called eXplanation Generator (XG). Note that XG does not use any corpus of common-sense properties at the inference time to generate explanations. XG has two variants – (i)

[9] The OMCS corpus has around 800,000 common-sense facts and was used to build ConceptNet.

[10] https://pypi.org/project/semantic-text-similarity/

[11] Default value taken from SBERT documentation

[12] https://pypi.org/project/rank-bm25/

| XR System | F$_1$ Score (%) | | |
|---|---|---|---|
| | Gold Corpus | Silver Corpus | |
| | Exact | STS-BERT | SPICE |
| BM25 + AIR | 22.2 | 15.1 | 18.4 |
| BM25 + top-$k$ | 25.6 | 16.2 | 19.8 |
| Ours + AIR | 33.0 | 25.0 | 25.4 |
| Ours + top-$k$ | **49.7** | **27.6** | **28.5** |

Table 8: Explanation retrieval results over *gold* and *silver* corpus for different choices of *property ranker* and *property selector* modules in the XR system."Ours" stands for our proposed *property ranker*.

XGP to generate common-sense properties, and (ii) XGF to generate the free-flow explanations across all the answer choices. In all our experiments, we use random sampling to generate the output tokens using GPT-2 and report average numbers over 3 different runs.

### 6.1 Property Generation (XGP)

Input to the XGP is a tuple $(q, a, c)$ and it generates a set of properties to justify/refute the given answer choice for the given question. The architecture for XGP is the same as GPT-2 but we fine-tune it in a customized manner as described below.

**Training:** We do a novel two-step fine-tuning of GPT-2 and refer to this model as XGP. In the first step, we fine-tune GPT-2 to ensure that it can generate sentences that resemble common-sense properties. For this, we fine-tune GPT-2 on language modeling task using a corpus of common-sense properties: ECQA train set plus OMCS corpus. We use perplexity to evaluate the quality of language model on the val set and save the model which achieves the lowest perplexity in 5 epochs. The input to our model is: ⟨BOP⟩ property ⟨EOP⟩, where property is word-pieces tokens of property and ⟨BOP⟩ and ⟨EOP⟩ are special tokens to mark the beginning and end of a property.

In the second step, we fine-tune it to learn how to generate a set of properties. Given a query tuple $(q, a, c)$ and a sequence of gold properties, say $(p_1^*, ..., p_k^*)$, we create input to GPT-2 as: ⟨BOS⟩ question: $q$ $a$ is $c$ the answer because ⟨BOP⟩ $p_1^*$ ⟨EOP⟩ ... ⟨BOP⟩ $p_k^*$ ⟨EOP⟩ ⟨EOS⟩

In this input template, the following set of strings are always constant: question:, is, and the answer because. Tokens ⟨BOS⟩ and ⟨EOS⟩ denotes the beginning and end of the sequence. We use *train* set of ECQA, preserving the ordering of properties from the annotation, so as to generate the fine-tuning data in the above template for the second fine-tuning step. We fine-tune for 5 epochs and save the model that achieves the lowest perplexity on the ECQA val set.

In order to establish the novelty of this 2 step fine-tuning, we create another model (XGP-W) by performing only 2nd step fine-tuning on pre-trained GPT-2 and compare it with XGP.

**Inference:** We use test set of ECQA to test XGP. The input to model is: ⟨BOS⟩ question: $q$ $a$ is $c$ the answer because ⟨BOP⟩. The model generates tokens until it generates ⟨EOS⟩ token. We parse output and collect a set of multiple properties between consecutive ⟨BOP⟩ and ⟨EOP⟩ tokens.

**Experiments:** Table 9 shows the comparison of XGP and XGP-W using the bipartite graph based metric discussed in section 5. Note that we have also included the best retrieval model on the *silver* corpus from Table 8 to show that our generation models perform significantly better than it. The maximum output token limit of GPT-2 in both the models is set to 150. We report some *anecdotal examples* of generated properties in Appendix A.4.

| Model | F$_1$ Score (%) | |
|---|---|---|
| | STS-BERT | SPICE |
| Ours + top-$k$ (Silver Corpus) | 27.6 | 28.5 |
| XGP-W | 33.0 | 30.1 |
| XGP | **36.4** | **32.2** |

Table 9: Comparison of XGP, XGP-W, and the best XR model using *silver* corpus.

### 6.2 Free-Flow Explanation Generation (XGF)

We now discuss models to generate the free-flow natural language explanations, given a *question, all answer choices*, and the *correct answer choice*. There are two different variants of XGF with different training strategies and inference prompts.

#### 6.2.1 XGF-I

We use GPT-2 to directly output the free-flow explanation $f$ given an input tuple $(q, o, ca)$, where $q$ is question, $o$ is sequence of all the answer choices for the question $q$, and $ca$ is the correct answer.

**Training:** We fine-tune GPT-2 for 5 epochs on train set of ECQA using standard language modeling objective. The input to GPT-2 during training is: ⟨BOS⟩ question: *q* The options are *o*. The best answer is *ca* because *f* ⟨EOS⟩. Validation is done on val set of ECQA using perplexity measure.

**Inference:** During inference on ECQA test set, the prompt is given till because token and generation is done until ⟨EOS⟩ token.

### 6.2.2 XGF-II

Here we generate the free-flow explanations in a two-step manner. In the first step, we generate the properties for each answer choice of a question using the trained XGP (section 6.1) model. After generating all the properties, we feed them in conjunction with *question, all the choices*, and *correct answer* to our GPT-2 based system XGF-II so as to generate the free-flow explanation.

**Training:** The fine-tuning of pre-trained GPT-2 proceeds in two-steps. First, we fine-tune on gold properties from the ECQA dataset. We take the model that achieves lowest perplexity on val set in 5 epochs. After fine-tuning on gold properties, we now fine-tune XGF-II for 5 epochs on the properties generated by XGP.

**Inference:** At inference time, we first generate the properties for each answer choice using XGP. Using these properties, XGF-II generate the free-flow explanation.

**Experiments:** Table 10 shows STS-BERT and SPICE scores between ground-truth and generated explanations by XGF. Both XGF variants give similar results. Note that we set the maximum output token limit of GPT-2 to 250[13]. We also tried free-flow generation with bare pre-trained GPT-2 but it resulted in complete garbage output. We report an *anecdotal example* of generated free-flow explanations in Appendix A.5.

| Model | STS-BERT | SPICE |
|---|---|---|
| XGF-I | 62.5 | 32.1 |
| XGF-II | 61.9 | 31.3 |

Table 10: Semantic Similarity Scores of XGF models.

## 7 Conclusion and Future Work

We have presented desiderata of what constitutes an explanation in the case of common-sense QA. Based on it, we generated a human-annotated explanation dataset ECQA for CommonsenseQA. We have also proposed models to retrieve and generate common-sense facts required to justify the answer choice. We have publicly released our crowdsourced ECQA dataset and code/models. In future work, we plan to explore directions to design RL-based schemes for joint training of property ranker and property selector components in the XR system and joint training of XGP and XGF-II to generate free-flow explanation. Another direction is to improve the accuracy and interpretability of the existing models for CommonsenseQA using the ECQA dataset.

## Acknowledgments

This work was supported by an IBM AI Horizons Network (AIHN) grant. Parag Singla is being supported by IBM SUR awards and Visvesvaraya young faculty fellowship by Govt. of India. We would like to acknowledge the use of IIT Delhi HPC facility, IBM cloud facility, and IBM Cognitive Computing Cluster (CCC) for carrying out various experiments. We thank Avaljot Singh and Koyal Mukherjee for helpful inputs and discussions during the early stages, and Mausam for his critical comments which helped improve the paper. We would also like to thank Yatin Nandwani and Keshavi Sai Kolluru who gave useful comments on the initial draft. We would like to thank the members of IBM-AIHN team for their support and suggestions. Finally, we thank anonymous reviewers for their insightful comments in improving the final version of the paper.

[13] As free-flow explanations are longer than properties, we set the maximum output token limit of GPT-2 to 250 for XGF models compared to 150 used for XGP models.

## Ethical Impact

This paper is concerned about proposing a brand new dataset on explanations of common-sense question answers. The dataset was crowdsourced through a private firm and all the ethical consideration were taken into account including proper remuneration to the human annotators as well as their consent to use the dataset for our research purposes. We have also ensured that there are no *personally identifiable information* or *offensive content* in our annotations. We also sought permission from authors of the CQA dataset to add our annotation on top of that dataset. As far as external libraries used in our code base is concerned, we have sought appropriate permissions from authors of all those external libraries which are available in public domain but do not have any license specified.

As far as implications of our research contributions is concerned, it can advance the state-of-the-art research on automated question answering requiring common-sense knowledge. This research can also advance technologies in the areas such as *automated dialog*, *machine debate*, etc. In fact, generating an explanation for the correct answer choice of a question help design *fair* and *unbiased* QA and dialog systems. These systems could offer huge value in sectors such as *customer support, e-commerce, online education, home automation, etc.*

## References

Eneko Agirre, Daniel M. Cer, Mona T. Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity. In *Proceedings of the 6th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2012*, pages 385–393.

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. MathQA: Towards Interpretable Math Word Problem Solving with Operation-Based Formalisms. In *Proceedings of NAACL-HLT*, pages 2357–2367.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. SPICE: Semantic propositional image caption evaluation. In *Proceedings of ECCV*, pages 382–398.

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.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of EMNLP*, pages 1533–1544.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen-tau Yih, and Yejin Choi. 2019. Abductive Commonsense Reasoning. In *Proceedings of ICLR*.

G. P. Shrivatsa Bhargav, Michael R. Glass, Dinesh Garg, Shirish K. Shevade, Saswati Dana, Dinesh Khandelwal, L. Venkata Subramaniam, and Alfio Gliozzo. 2020. Translucent Answer Predictions in Multi-Hop Reading Comprehension. In *Proceedings of AAAI*, pages 7700–7707.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal andf Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Proceedings of NeurIPS*, pages 1877–1901.

Xinyun Chen, Chen Liang, Adams Wei Yu, Denny Zhou, Dawn Song, and Quoc V. Le. 2020. Neural Symbolic Reader: Scalable Integration of Distributed and Symbolic Representations for Reading Comprehension. In *Proceedings of ICLR*.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2020. Transformers as Soft Reasoners over Language. In *Proceedings of IJCAI*, pages 3882–3890.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Alvan R Feinstein and Domenic V Cicchetti. 1990. High agreement but low kappa: I. The problems of two paradoxes. *Journal of clinical epidemiology*, 43(6):543–549.

Miguel Angel Ríos Gaona. 2014. *Methods for measuring semantic similarity of texts*. Ph.D. thesis, University of Wolverhampton, UK.

Shalini Ghosh, Giedrius Burachas, Arijit Ray, and Avi Ziskind. 2018. Generating Natural Language Explanations for Visual Question Answering Using Scene Graphs and Visual Attention. In *Proceedings of XAI@ICML*.

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *arXiv preprint arXiv:1705.00652*.

Helmut Horacek. 2017. Requirements for Conceptual Representations of Explanations and How Reasoning Systems Can Serve Them. In *Proceedings of the 1st Workshop on Explainable Computational Intelligence (XCI 2017)*.

Peter Jansen, Elizabeth Wainwright, Steven Marmorstein, and Clayton Morrison. 2018. WorldTree: A Corpus of Explanation Graphs for Elementary Science Questions supporting Multi-hop Inference. In *Proceedings of LREC*.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *Proceedings of ACL*, pages 1601–1611.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UNIFIEDQA: Crossing Format Boundaries with a Single QA System. In *Findings of ACL: EMNLP 2020*, pages 1896–1907.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. QASC: A Dataset for Question Answering via Sentence Composition. In *Proceedings of AAAI*, pages 8082–8090.

Neema Kotonya and Francesca Toni. 2020. Explainable Automated Fact-Checking for Public Health Claims. In *Proceedings of EMNLP*, pages 7740–7754.

Harold W Kuhn. 1955. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97.

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

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

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program Induction by Rationale Generation: Learning to Solve and Explain Algebraic Word Problems. In *Proceedings of ACL*, pages 158–167.

Kaixin Ma, Jonathan Francis, Quanyang Lu, Eric Nyberg, and Alessandro Oltramari. 2019. Towards Generalizable Neuro-Symbolic Systems for Commonsense Question Answering. In *Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing*, pages 22–32.

Shen-Yun Miao, Chao-Chun Liang, and Keh-Yih Su. 2020. A Diverse Corpus for Evaluating and Developing English Math Word Problem Solvers. In *Proceedings of ACL*, pages 975–984.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a Suit of Armor Conduct Electricity? A New Dataset for Open Book Question Answering. In *Proceedings of EMNLP*, pages 2381–2391.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of EMNLP*, pages 1532–1543.

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

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain Yourself! Leveraging Language Models for Commonsense Reasoning. In *Proceedings of ACL*, pages 4932–4942.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of EMNLP*, pages 2383–2392.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of EMNLP*, pages 3973–3983.

Stephen Robertson and Hugo Zaragoza. 2009. *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc.

P SINGH. 2002. The public acquisition of common sense knowledge. In *Proceedings of AAAI Spring Symposium on Acquiring (and Using) Linguistic (and World) Knowledge for Information Access, 2002*. AAAI.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. In *Proceedings of NAACL-HLT*, pages 4149–4158.

Priyansh Trivedi, Gaurav Maheshwari, Mohnish Dubey, and Jens Lehmann. 2017. LC-QuAD: A corpus for complex question answering over knowledge graphs. In *Proceedings of ISWC*, pages 210–218.

Christina Unger, Corina Forascu, Vanessa López, Axel Cyrille Ngonga Ngomo, Elena Cabrio, Philipp Cimiano, and Sebastian Walter. 2014. Question Answering over Linked Data (QALD-4). In *Proceedings of CLEF*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Proceedings of NeurIPS*, pages 5998–6008.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. CIDEr: Consensus-based image description evaluation. In *Proceedings of CVPR*, pages 4566–4575.

Cunxiang Wang, Shuailong Liang, Yue Zhang, Xiaonan Li, and Tian Gao. 2019. Does it Make Sense? And Why? A Pilot Study for Sense Making and Explanation. In *Proceedings of ACL*, pages 4020–4026.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. Crowdsourcing Multiple Choice Science Questions. In *Proceedings of NUT@EMNLP*, pages 94–106.

Zhengnan Xie, Sebastian Thiem, Jaycie Martin, Elizabeth Wainwright, Steven Marmorstein, and Peter Jansen. 2020. WorldTree V2: A Corpus of Science-Domain Structured Explanations and Inference Patterns supporting Multi-Hop Inference. In *Proceedings of LREC*, pages 5456–5473.

Vikas Yadav, Steven Bethard, and Mihai Surdeanu. 2020. Unsupervised Alignment-based Iterative Evidence Retrieval for Multi-hop Question Answering. In *Proceedings of ACL*, pages 4514–4525.

Yiben Yang, Chaitanya Malaviya, Jared Fernandez, Swabha Swayamdipta, Ronan Le Bras, Ji-Ping Wang, Chandra Bhagavatula, Yejin Choi, and Doug Downey. 2020. Generative Data Augmentation for Commonsense Reasoning. In *Findings of ACL: EMNLP 2020*, pages 1008–1025.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *Proceedings of EMNLP*, pages 2369–2380.

Zhuosheng Zhang, Junjie Yang, and Hai Zhao. 2020. Retrospective Reader for Machine Reading Comprehension. *CoRR*, abs/2001.09694.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2020. FreeLB: Enhanced Adversarial Training for Natural Language Understanding. In *Proceedings of ICLR*.

# A Appendices

## A.1 Additional Example of ECQA Annotations

Table 11 shows an additional example of CommonsenseQA, along with our human-annotated explanations, containing positive properties to support the correct answer choice (in green), negative properties to refute the incorrect choices (in red), and free-flow natural language explanation (in blue).

**Question:**
What is something that people do early in the day?
**Answer Choices:**
believe in god   make tools   skydive
smoke pot   eat eggs

**Our Explanation:**
Positives Properties
1) People generally eat breakfast early morning.
2) People most often eat eggs as breakfast.
Negative Properties
1) Believing in god is not restricted to a specific part of a day.
2) People generally do not make tools early in the day.
3) Skydive is an irrelevant answer.
4) People usually do not smoke pot early in the day.
Free-Flow Explanation (FF)
People generally eat breakfast early morning which most often consists eggs. People generally do not make tools or smoke pot early in the day. Skydive is an irrelevant answer.

Table 11: Example of CommonsenseQA with our annotated explanation

## A.2 Experimental Details

**Computing Infrastructure:** We run all our experiments on a machine with a single Tesla P100 GPU (16 GiB) and 8 Intel(R) Xeon(R) E5-2690 v4 @ 2.60GHz CPUs with 59 GiB of physical memory. Training times for all our different models within the proposed XR and XG systems were within 4 hours.

**Implementation Details:** All our models are implemented in PyTorch[14]. We used SBERT[15] to implement our property retriever system XR. For our proposed *property ranker* module, we used a BERT-base-uncased, followed by a mean pooling layer, and then a dense layer of size 512. We use Huggingface transformer package[16] to fine-tune GPT-2 for all our generation models.

## A.3 Anecdotal Examples: Property Retrieval

Table 12 shows some hand-picked examples where our proposed XR system retrieves a set of properties to either support or refute the given option.

**Query** $(q, a, c)$**:** (the person used a candle to navigate up the spiral staircase, where were they likely?, Light house, True)
**Gold set** $(p^*)$**:** {'light house has a spiral staircase', 'light house is a structure', 'a candle can be used inside a light house'}
**Ours+top-**$k$**:** {'light house has a spiral staircase', 'a candle can be used inside a light house', 'light house is a structure'}
**BM25+top-**$k$**:** {'a candle can be used inside a light house', 'light house has a spiral staircase', 'candle is used to counter insufficient lighting'}

**Query** $(q, a, c)$**:** (sally took her medicine and experienced strong side effects. what did doctors say about the side effects?, Distinguished, False)
**Gold set** $(p^*)$**:** {'distinguished means important or respected'}
**Ours+top-**$k$**:** {'distinguished means important or respected'}
**BM25+top-**$k$**:** {'sally belong to the house and her neighbour'}

**Query** $(q, a, c)$**:** (two friends wanted to spend a quiet evening together, what did they go see?, Restaurant, False)
**Gold set** $(p^*)$**:** {'restaurant will not be quiet'}
**Ours+top-**$k$**:** {'restaurant is where people go and eat'}
**BM25+top-**$k$**:** {'restaurant will not be quiet'}

Table 12: Anecdotal examples of retrieved properties by our proposed XR system.

[14] https://pytorch.org/
[15] https://www.sbert.net/
[16] https://huggingface.co/transformers/

### A.4 Anecdotal Examples: Property Generation

Table 13 shows some hand-picked examples of generated properties by `XGP` model and retrieved properties by `XR` system from the silver corpus.

| |
|---|
| **Query** $(q, a, c)$**:** (where do people go to pray?, Church, True)<br>**Gold set** ($p^*$)**:** {'prayer is done at church', 'people go to church for praying'}<br>**Ours+top-**$k$**:** {'people go to church to worship and pray', 'some people go to church to pray to their god', 'if you want to pray then you should go to church'}<br>**`XGP`:** {'people go to a church to pray', 'church is a place of worship'} |
| **Query** $(q, a, c)$**:** (how would someone feel after going for run?, Falling down, False)<br>**Gold set** ($p^*$)**:** {'going for a run generally won't make one fall down'}<br>**Ours+top-**$k$**:** {'something that might happen as a consequence of going for a run is falling down'}<br>**`XGP`:** {'person doesn't fall down after going for run always', 'falling down is not a feeling'} |
| **Query** $(q, a, c)$**:** (what is the best course of action for a person to take if they want to get up early?, Dawn, False)<br>**Gold set** ($p^*$)**:** {'dawn is not an action'}<br>**Ours+top-**$k$**:** {'something that might happen when you get up early is you see the dawn'}<br>**`XGP`:** {'dawn is not the best action for a person to take if he wants to get up early'} |

Table 13: Anecdotal examples of generated properties by `XGP` model and retrieved properties by the best `XR` system using silver corpus.

### A.5 Anecdotal Examples: Free-Flow Explanation Generation

Table 14 gives an example of free-flow explanation generation by the two variants of `XGF` system.

### A.6 Human-Metric Correlation Experiment

The semantic textual similarity (STS) task aims to measure semantic similarity between two sentences quantitatively. $N$-gram matching based metrics (CIDEr, METEOR, and ROUGE)[17] only capture the lexical and syntactic similarity and are not suitable for capturing the semantic similarity between two sentences. SPICE uses a semantic representation known as *scene graphs* to overcome the problem with $n$-gram matching. STS-BERT[18] is an implementation interface for a pre-trained BERT model followed by a linear regression head, fine-tuned over STS-B dataset (Wang et al., 2018) to compute semantic similarity between English sentences. It can also be used to provide a similarity score in our case. We designed an experiment to find which metric correlates better with human judgments. We took 100 random samples of queries $(q, a, c)$, picked a valid gold, and one of the `XGP` generated properties. We human-annotated whether the picked gold and `XGP` generated property are semantically similar or not. We also cal-

| |
|---|
| **Input:** question: sally went to a store and spent money. most of her purchases were fresh but some were frozen. what did she do? The options are made a mistake, control people, get things, buy food, pay bills. The best answer is buy food because<br>**Gold free-flow:** buying food is an activity done at store which requires money. it can be both fresh and frozen. all the other options are incorrect as they are not related to spending money to buy something.<br>**Pre-trained GPT-2:** they have a good idea of where you want to buy. if they don't know, buy food. If you are going to buy food, it should be on the menu. if you are going to buy a food item that is going to get you a lot of attention and attention is not being paid to you, buy it. buy the food. (*"buy the food"* repeated 33 times)<br>**`XGF-I`:** food is sold in store so sally bought fresh food from a store and bought it from a store. all the other options are not fesiable as they are not a place where sally spent money.<br>**`XGF-II`:** buying food is done by sally. she went to a store to buy food. controlling people is not done by buying food. getting things is not done by sally. paying bills is not related to spending money. |

Table 14: Anecdotal example of free-flow explanation generation by 2 variants of `XGF` system and pre-trained GPT-2.

[17] BLEU was least correlated with human judgment, therefore it was not included in further experiments.

[18] https://pypi.org/project/semantic-text-similarity/

culate all the metrics scores between both sets of properties. If the score is greater than a threshold $\tau$, we say the properties are semantic similar, otherwise not. Threshold $\tau$ for each metric is selected by maximizing the $F_1$ score for these 100 selected samples. We also calculated Pearson's Correlation coefficient between metric scores and human annotations. We compared the $F_1$ scores of different metrics and found STS-BERT score and SPICE to be having the highest $F_1$ scores and maximum human correlation.

**Thresholds verification:** We designed another experiment to verify these thresholds. We took 200 random queries $(q, a, c)$ along with one of their gold properties from ECQA dataset. We asked a different annotator to write semantically similar property to gold property for each of the first 100 queries and semantically dissimilar property for the other 100 queries. We used the thresholds calculated in the previous experiment to calculate the $F_1$ scores using different metrics on these two sets of properties. STS-BERT score and SPICE metric have the highest $F_1$ scores in this experiment also. Table 15 shows the thresholds ($\tau$), corresponding $F_1$ scores, and Pearson's correlation coefficients with human annotation for different metrics in Experiment 2. We used the same thresholds ($\tau$) for retrieval using silver corpus and property generation results reported in the Table 8 and 9 of the main paper using our proposed unweighted bipartite matching based metric.

| Measure | ST | SP | C | M | R |
|---|---|---|---|---|---|
| **Threshold** | 0.6 | 0.4 | 0.3 | 0.3 | 0.3 |
| **$F_1$ Score (%)** | **78.1** | **64.7** | 35.3 | 54.3 | 36.5 |
| **PC (%)** | **59.6** | 35.0 | 31.0 | 47.7 | 21.8 |

Table 15: Human correlation with different metrics. PC:Pearson's Coefficient, ST: STS-BERT, SP: SPICE, C:CIDEr, M:METEOR, R:ROUGE

## A.7 Human Validation Experiment

Table 16 lists the Pearson's correlation coefficients for human judgements in Relative Dataset Quality Experiment, for each quality measure (column) and property (row) combination in Table 6. Pearson's coefficient is computed as follows: for each judge, we calculate the correlation coefficient between the scores given by the judge, and the average of the scores across all the judges, for commonly labeled 50 samples. This is followed by computation of the average of this coefficient across all the judges for each entry in the table.

| Aspect | ECQA better | CoS-E better | Both Good | Both Bad |
|---|---|---|---|---|
| Comprehensive | 78.0 | 92.0 | 84.9 | 75.7 |
| RC | 68.7 | - | 92.0 | 79.8 |
| M/NR | 65.1 | 86.9 | 61.7 | 73.5 |
| Overall | 74.0 | - | - | 74.5 |

Table 16: Pearson's correlation coefficient for Relative Dataset Quality Experiment: ECQA and CoS-E. RC: Refutation Complete, M/NR: Minimality/Non-redundancy, - means 1 or more annotators never picked this option.

## A.8 More Retrieval and Generation Results

Table 17 shows $F_1$ scores for retrieval from silver corpus and property generation using different metrics. Table 18 compares the free-flow explanation generated by XGF-I and XGF-II.

| System | $F_1$ Score (%) | | | | |
|---|---|---|---|---|---|
| | ST | SP | C | M | R |
| **Retrieval** | | | | | |
| BM25+AIR | 15.1 | 18.4 | 3.2 | 4.3 | 13.1 |
| BM25+top-$k$ | 16.2 | 19.8 | **4.1** | 4.5 | 10.1 |
| Ours+AIR | 25.0 | 25.4 | 3.3 | 5.4 | **14.7** |
| Ours+top-$k$ | **27.6** | **28.5** | 4.0 | **5.5** | 14.1 |
| **Generation** | | | | | |
| XGP-W | 33.0 | 30.1 | 9.8 | 12.3 | 22.6 |
| XGP | **36.4** | **32.2** | **11.1** | **13.7** | **25.7** |

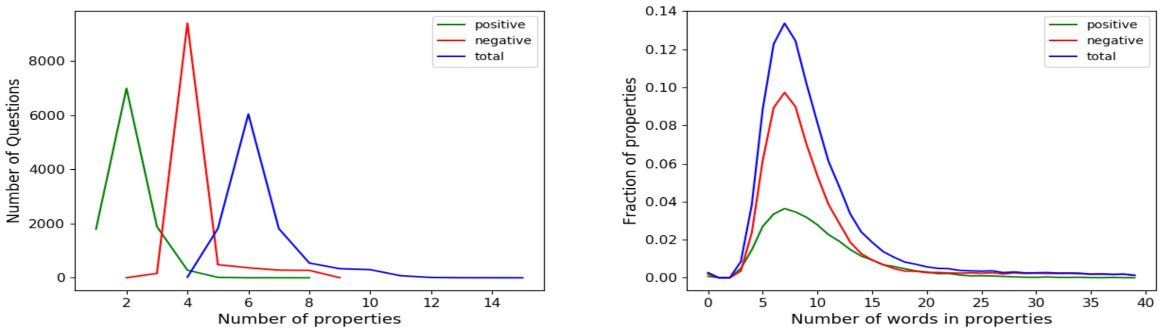
Table 17: Explanation retrieval results over *silver* corpus for different XR systems and property generation results by the XGP models for all 5 metrics. ST: STS-BERT, SP: SPICE, C:CIDEr, M:METEOR, R:ROUGE

| System | ST | SP | C | M | R |
|---|---|---|---|---|---|
| XGF-I | 62.5 | 32.1 | 20.3 | 17.5 | 12.2 |
| XGF-II | 61.9 | 31.3 | 18.7 | 17.2 | 12.5 |

Table 18: Semantic similarity scores for free-flow generation on all 5 metrics. ST: STS-BERT, SP: SPICE, C:CIDEr, M:METEOR, R:ROUGE

## A.9 Data Insights

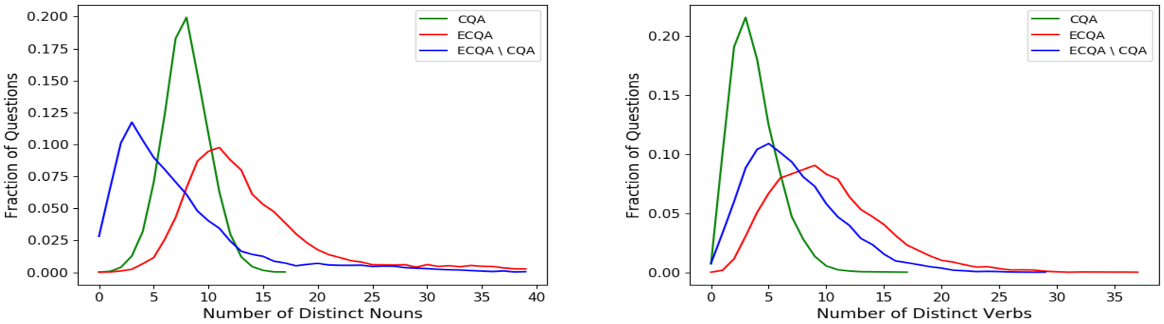
This section provides some more insights about the ECQA dataset. Table 19 gives the word-overlap



(a)

(b)

Figure 1: (a) Distribution of number of properties per question. (b) Distribution of properties length in words.



(a)

(b)

Figure 2: Distribution of number of distinct nouns (a) and verbs (b) in CQA vs ECQA. For each $n$ on x axis, the plot gives the fraction of questions which have $n$ distinct nouns/verb.

metric scores like BLEU-4 and ROUGE between the explanation and the corresponding question text for both CoS−E and ECQA. The scores are low for both CoS−E and ECQA. We note these scores may not be reflective of the true picture about overlap of information content between the explanation and the question text because of two reasons: (a) explanations may paraphrase the content in the original question text artificially resulting in a low score, (b) the score may be low due to difference in the length of the explanation and question. Thus, we have focused on the number of (distinct) important words present only in the explanation, as a metric for information content in the main paper (Table 4).

| Dataset | BLEU-4 | ROUGE |
|---|---|---|
| CoS−E \ CQA | 18.0 | 16.2 |
| ECQA \ CQA | 18.3 | 24.5 |

Table 19: Comparing information content through word-overlap metrics in CQA, CoS−E and ECQA.

We give distribution of number of properties in Figure 1a and length of properties in Figure 1b. The green curve corresponds to positive properties, red curve corresponds to negative properties and the blue curve corresponds to total properties. The distribution of extra number of nouns and verbs in the ECQA dataset are given in Figure 2a and 2b respectively. Here, the green curve corresponds to CQA dataset (number of distinct words in question and answer choices). The red curve corresponds to ECQA dataset (number of distinct words in properties and free-flow explanation). Finally, the blue curve represents the ECQA \ CQA plot corresponding to the number of *novel words* (present in the properties and free-flow explanation but not in the question and answer choices). This, in turn, gives a rough idea of the extra information present in our annotations.

We analyzed the rare novel words present in our annotations and found that on average, every annotation has 0.23 words which do not appear anywhere else in the corpus, 0.7 words which appear less than 10 times and 2.4 words appearing less than 100 times in the whole corpus of about 1.5 million words. This gives an idea about the diversity of extra information in our annotations, indicating the inherent hardness for any machine to generate it without access to external relevant common-sense facts.