SimsterQ: A Similarity based Clustering Approach to Opinion Question Answering

Anonymous authors

Abstract

In recent years, there has been an increase in online shopping resulting in an increased number of online reviews. Customers cannot delve into the huge amount of data when they are looking for specific aspects of a product. Some of these aspects can be extracted from the product reviews. In this paper we introduced SimsterQ - a clustering based system for answering questions that makes use of word vectors. Clustering was performed using cosine similarity scores of sentence vectors and questions. Two variants (Sim and Median) with and without stopwords were evaluated against traditional methods that use term frequency. We also used an n-gram approach to study the effect of noise. We used the reviews in the Amazon Reviews dataset to pick the answers. Evaluation was performed both at the individual sentence level using the top sentence from Okapi BM25 as the gold standard and at the whole answer level using review snippets as the gold standard. Our system returned answers similar to the review snippets from the Amazon QA Dataset as measured by the cosine similarity.

1. Introduction

In the recent years, the volume of online shopping has increased rapidly. This has resulted in the increase in availability of online reviews and question-answers related to a product. Traditional Question Answering (QA) systems are factual in nature. For example, “Which year did World War I end?” 1918. In opinion QA, answers to questions are based on the customers’ opinions. The customers’ opinions help other users to decide whether to purchase a product. This process is time consuming for the users to look at thousands of reviews to find the required information. Our paper aims at answering questions, users have, using customer reviews. We used the product reviews to extract the relevant sentences, with minimal to no overlap in meaning, and present it to the user. We make use of the AmazonQA dataset to answer binary (yes/no) questions.

The main focused contribution of this paper is: using an unsupervised clustering based system (SimsterQ) with five different variants to answer binary questions using information
in the product reviews. To the best of our knowledge, we do not know of other systems that have used clustering to answer opinion based questions using product reviews.

2. Related Work

Early work in opinion question answering addressed separating facts from opinions [Yu and Hatzivassiloglou, 2003], and the authors used a Naïve Bayes classifier to identify polarity of the opinions. Kim and Hovy [2005] aimed at identifying the opinion holder of the opinions. Stoyanov et al. [2005] explained the differences between fact based and opinionated answers and how traditional QA systems will not be able to handle multiple perspectives for answers. Some works aimed at using community based question-answers to provide unique answers to questions [Liu et al., 2007, Somasundaran et al., 2007]. Moghaddam and Ester [2011] made use of online reviews to answer questions on aspects of a product. Li et al. [2009] and Yu et al. [2012] used graphs and trees to answer opinion questions. Wan and McAuley [2016] modeled ambiguity and subjectivity in opinion QA using statistical models.

Gupta et al. [2019] give baselines for answer generation systems given the question and reviews. We use their results as the baseline for our evaluation. We also discuss the dataset from this paper in 4.2. While most systems used in the works described above are supervised learning models, our system used unsupervised learning to answer binary (yes/no) questions.

3. System Description

The answer selection process to get the top k sentences has the following steps:

  Relevant reviews selection: We group all reviews by the asin/product id. We pick those reviews with the same product id as the questions.

  Sentence level similarity We process the reviews by removing punctuations and html tags. We split the reviews by sentences and find the cosine similarity between each sentence and the question.

  Filtering sentences below threshold We filter the above set by removing sentences below a threshold. The threshold is set to 0.5 so that sentences that have minimal to no similarity with the question are removed from consideration as candidate sentences.

  Grouping sentences with similar meaning/information We order the sentences by the similarity score in descending order. We then form clusters by picking the top sentence and grouping it with sentences that have high similarity (threshold value = 0.9). We repeat this until all sentences are clustered. Note that some clusters will have only one sentence at this point.

  Selecting top k-sentences We then pick our top k = 10 answers from our top 10 clusters. These 10 clusters in essence have the highest similarity scored sentences with the question. We either pick the first sentence in each cluster or we pick the sentence with median length from each cluster.

For the cosine similarity calculation, we use word2vec to calculate the sentence vector as sum of the word vectors of the words in the sentence. The calculation of sentence vector was
Function Similarity (question,reviews):
    sentences ← split(reviews)
    sentences ← list(ordered by cosine sim)
    return sentences, cosine sim

Function Cluster (sentences, cosine sim, threshold, median):
    answers ← empty
    c = 0
    while sentences not empty do
        c += 1
        cluster[c].append(sentences[0])
        for i ← 1 to num(sentences) do
            if sim (sentences[0],sentences[i]) > threshold then
                cluster[c].append(sentences[i])
            end
        end
        if median == False then
            answers.add(cluster[c][0])
        else
            answers.add(cluster[c].median)
        end
        Remove sentences added to cluster c from sentences
    end
    return answers

Algorithm 1: SimsterQ Algorithm

to take advantage of the compositionality property using word2vec [Mikolov et al., 2013].
We used word vectors of dimension 100 trained on the 2015 wikidump.

4. Experimental Setup

4.1 Methods Used

The term frequency method (tf) is one of the earliest and popular method used for information retrieval. Tf (term frequency) is usually combined with idf (inverse document frequency); but in this paper, only tf was performed as our work is at the sentence level. In our paper given a question about a product, we collected all the reviews available for that product. We then split the reviews into sentences(we will refer to these as candidate sentences) and performed 7 methods.

Term Frequency (tf) used the frequency of words in the question and compared it with the frequency in candidate sentences. Term Frequency no stopwords (tfns) used the tf method but without stopwords. These two methods, tf and tfns, still used the clustering
function of Algorithm 1. Clustering was used to reduce the repetition of information in the top 10 answers.

Similarity (sim) made use of cosine similarity between the question and candidate sentences. The other methods were variants of this method. Similarity no stopwords (simns) used the similarity method but without the stopwords. Similarity median (med) uses of the sentence with median length in a cluster versus the first sentence in the cluster as in sim. Similarity Median no stopwords (medns) used the similarity median but without stopwords.

The last method was the 3-gram method (3g). In this we split the question into 3-grams and we used the same method as sim. We used 3-gram since the shortest question in the dataset is three words long. From the clusters, we picked only sentences that have been returned by at least half the n-gram phrases. The 3-gram model was done with the idea that splitting longer questions into smaller parts will help grasp the meaning, i.e. we expected shorter phrases to incorporate more information than the whole sentence. Sim, simns, med, medns, and 3g all use the SimsterQ system described in Algorithm 1. In all methods we returned the top k, where k = 10 sentences.

4.2 Dataset

The AmazonQA dataset was used in this study. The dataset has both yes/no (binary) and open-ended questions. The fields we used are question id, question Type, question Text, answers, review snippets, asin/ product id, and category. The dataset was built based on previous parallel datasets provided by Wan and McAuley [2016].

The first dataset consists of question on Amazon for products and the answers provided by users who bought those products. The second dataset was the Amazon Reviews Dataset. Amazon Reviews dataset contains 142.8 million reviews for different products in 24 product categories.

The problem with using the parallel datasets was that the evaluation was a difficult task. The answer generation by our model was using the product reviews but the gold standard is from answers written by Amazon users. For the same reason we do not use the answers as the gold standard.

The AmazonQA bridges this gap by providing relevant review snippets for each question. In addition, the dataset has a variable to identify if the question can be answered satisfactorily using the reviews alone. We found this more appropriate for our task since our intention is to provide top k sentences from the reviews that will answer a question.

We used five categories of products in our research. The five categories were Automotive, Baby, Beauty, Pet Supplies, and Tools and Home Improvement. We chose these categories as they are likely to have products that are not similar and likely to have questions that do not overlap.

We randomly picked 200 questions from each category for a total of 1000 questions. We took the reviews from the Amazon Reviews dataset since we already worked on this dataset for our previous research. The reviews were used to provide answers using the tf methods and the different variants of the SimsterQ system.

5. Evaluation

Evaluations were performed at both the sentence level and at the whole answer level.
5.1 Sentence Level Evaluation

At the sentence level, we pick the top 1 sentence, using Okapi BM25, as the gold standard. To retrieve the top 1 sentence using Okapi BM25, we used the question as the query and the product reviews as the documents.

For each sentence in the answers returned by our system, we use the top sentence as the gold standard to calculate ROUGE-1 and ROUGE-L scores. The average of the ROUGE scores with the max ROUGE-L F-score for each instance is reported. In addition to providing the F1 scores, Precision and Recall scores are also reported. In QA tasks, the relevance of the answers may be more important than how well the answers capture the essence of the question (a common benchmark for question answering and summarization tasks). So, P and R scores are reported to better interpret the results.

ROUGE is usually used to evaluate summarization task and may not be the best metric to measure our system performance which does a opinion based QA task which are different from the traditional QA systems. So cosine similarity was used as a metric to evaluate our system generated answer sentences against the gold standard. Three different metrics were calculated based on how well our system was able to exceed a cosine similarity threshold of 0.7 when compared against the gold standard.

5.1.1 Accuracy

was calculated based on the total number of all answer sentences. In our case, accuracy for each method was the fraction of the sentences that had a cosine similarity, with the gold standard, of more than 0.7.

5.1.2 Correct Answer

was found as the fraction of questions for which our methods returned at least one answer that had a cosine similarity, with the gold standard, of more than 0.7. This was a measure of how reliable the methods were in returning at least one relevant answer based on the reviews.

5.1.3 At least 50%

correct answers for each question was the third evaluation metric. This was calculated as the fraction of questions for which our methods returned more than 50% of answer sentences that had a cosine similarity, with the gold standard, of more than 0.7.

5.2 Answer Level Evaluation

At the answer level, we use the review snippets returned by the AmazonQA authors as the gold standard. We calculate the ROUGE scores and cosine similarity between the gold standard and each of the seven methods.
6. Results

6.1 Sentence Level

The sentence level evaluation was performed using the Okapi BM25 top sentence as the gold standard. Term frequency based methods always seem to perform better. Of the methods based on our system, the sim method consistently performs better than the other methods. Our system outperforms the R-Net baseline (Rouge-L: 40.22) used by Gupta et al. [2019]. Our system is supposed to be applied at the sentence level and the results indicate that an unsupervised system such as ours could outperform more complicated deep learning models. If there is a trade-off sought between computing time and accuracy, our system performs similar to or better than the baselines used by Gupta et al. [2019].

Rouge score is not the best metric for tasks such as opinion question answering. We believe the cosine similarity is a better metric to measure how close the retrieved answer is to the gold standard. Overall the sim method is able to provide an answer more than 70% similar to the gold standard answer 91.5% of the time. From the sentences returned by our system as candidate answers, 72% of the time at least half the candidate sentences are good answers. This shows that our system is consistent and accurate at providing good answers.

6.2 Answer Level

At the answer level the top candidate sentences (up to 10) returned by our system were compared against the review snippets as the gold standard. The review snippets were top review sentences returned by the system used by Gupta et al. [2019].

Average ROUGE scores are reported in Table 2. Both systems aim at providing the best candidate sentences. Looking at the precision scores, it is clear that our system performance is good in terms of returning relevant sentences, similar in content to the gold standard. The sim method still is the best performing method.

<table>
<thead>
<tr>
<th>Score</th>
<th>Metric</th>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>tf</td>
</tr>
<tr>
<td>ROUGE-1</td>
<td>F</td>
<td>51.50</td>
</tr>
<tr>
<td></td>
<td>P</td>
<td>50.71</td>
</tr>
<tr>
<td></td>
<td>R</td>
<td>58.35</td>
</tr>
<tr>
<td>ROUGE-L</td>
<td>F</td>
<td>47.49</td>
</tr>
<tr>
<td></td>
<td>P</td>
<td>49.50</td>
</tr>
<tr>
<td></td>
<td>R</td>
<td>56.76</td>
</tr>
</tbody>
</table>

Table 2: Answer Level Results
Looking at the similarity scores it is clear that the candidate sentences returned by our system is almost exactly similar to the sentences returned by Gupta et al. [2019]. Once again our system is able to perform on par with a more complicated system.

### 7. Conclusions and Future Work

This paper introduced SimsterQ - a unsupervised clustering based system to answer questions about products by accessing the reviews of the products. Five different variants of this system were evaluated using 1000 yes/no questions. At the sentence level sim performed better with the highest ROUGE and Similarity scores. Sim method returns the top sentence from each of the 10 clusters created.

When evaluating the entire answer, our system performed better than the baseline ROUGE score from the R-Net method.

In future SimsterQ will be used with open-ended questions. The challenge with open-ended questions will be the evaluation. Perspectives expressed in the reviews need not necessarily match the perspectives in the gold standard answer. We want to evaluate the performance of SimsterQ on other datasets.

In the Amazon question/answer data set not every question has a good relevant answer. The answers are sometimes a single user’s opinion. SimsterQ will be used to provide a new gold standard answer to the binary questions. The code and usage will be provided on GitHub upon publication/acceptance of this paper.

### Acknowledgments

Omitted for blind review.

### References

Mansi Gupta, Nitish Kulkarni, Raghuveer Chanda, Anirudha Rayasam, and Zachary C. Lipton. Amazonqa: A review-based question answering task. Proceedings of the Twenty-

---

Table 2: Answer Level Results

<table>
<thead>
<tr>
<th>Score</th>
<th>Metric</th>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>tf</td>
</tr>
<tr>
<td>ROUGE-1</td>
<td>F</td>
<td>45.60</td>
</tr>
<tr>
<td></td>
<td>P</td>
<td>62.64</td>
</tr>
<tr>
<td></td>
<td>R</td>
<td>36.70</td>
</tr>
<tr>
<td>ROUGE-L</td>
<td>F</td>
<td>37.72</td>
</tr>
<tr>
<td></td>
<td>P</td>
<td>59.00</td>
</tr>
<tr>
<td></td>
<td>R</td>
<td>34.38</td>
</tr>
<tr>
<td>Similarity</td>
<td>Accuracy</td>
<td>98.32</td>
</tr>
</tbody>
</table>


