SimsterQ: A Similarity based Clustering Approach to Opinion Question Answering

Aishwarya Ashok *
Dept of Comp Science & Engg
University of Texas at Arlington
Arlington, TX, USA
aishwarya.ashok@mavs.uta.edu

Ganapathy S. Natarajan *
Dept of MEIE
University of Wisconsin - Platteville
Platteville, WI, USA
natarajang@uwplatt.edu

Ramez Elmasri
Dept of Comp Science & Engg
University of Texas at Arlington
Arlington, TX, USA
elmasri@cse.uta.edu

Laurel Smith-Stvan
Dept of Linguistics and TESOL
University of Texas at Arlington
Arlington, TX, USA
stvan@uta.edu

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 between sentence vectors of reviews 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. At the sentence level our system performed slightly better than a more complicated deep learning method. Our system returned answers similar to the review snippets from the Amazon QA Dataset as measured by the cosine similarity. Analysis was also performed on the quality of the clusters generated by our system.

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 are:

1. 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. Provide evidence of an unsupervised simple system having a performance akin or exceeding deep learning systems.

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

* These authors contributed equally to this project and paper.

Proceedings of the 3rd Workshop on e-Commerce and NLP (ECNLP 3), pages 69–76

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:

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

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

3. *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.

4. *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 and some clusters may just be empty. In essence, the algorithm self selects the appropriate number of clusters.

5. *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.

Our system is not limited to separate n observations into k clusters, like the k-means algorithm. N observations are naturally partitioned into up to k clusters. The algorithm naturally selects the appropriate number of clusters by grouping highly similar sentences into each cluster. We present only the sentences from the top 10 clusters; the k may be varied depending on the task at hand. In this research k was selected to be 10, so that we can compare our results with Gupta et al. (2019).

The order of the sentences in the review does not matter. We find the cosine similarity between each sentence and the question and order it from highest to lowest cosine similarity. So, the order in which the sentences occur in the review does not affect the results from our system. We use cosine similarity as it is a commonly used measure to find closeness of sentences using their angles in a vector space.

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 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

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 five different methods of selecting candidate sentences.

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

```
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])
                      // Sim Variant
        else
            answers.add(cluster[c].median)
                  // Median Variant
        end
        Remove sentences added to cluster c
         from sentences
    end
    return answers
```

Algorithm 1: SimsterQ Algorithm

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, sim_ns, sim_med, sim_med_ns, and 3g all use the SimsterQ system described in Algorithm 1. In all methods we returned the top k, where k = 10 or the maximum number of sentences available, whichever is smaller.

4.2 Dataset

The AmazonQA dataset was used in this study (Gupta et al., 2019). 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 different variants of the SimsterQ system.

5 Evaluation

Evaluations were performed at both the sentence level and at the whole answer level.

5.1 Cluster Quality

Our algorithm performs clustering of sentences to find the answers. As previously mentioned, the algorithm self selects the appropriate number of

clusters. However, we need to measure the quality and the number of clusters returned. Two commonly used measures to evaluate cluster quality are Silhouette score and Calinski-Harabasz score. These metrics were calculated for each question separately.

Each answer cluster was decided based on the cosine similarity with the question and the cosine similarity with the top sentence within each cluster. So, in calculating the cluster quality metrics, cosine similarity with question and cosine similarity with first sentence in the cluster were used as the features and the cluster number was used as the labels.

Silhouette score works based on distances and Calinski-Harabasz score works based on dispersion measured as squared distances (sum of squares). So we are reporting both the scores in our analysis.

5.1.1 Silhouette Score

Silhouette score measures cohesion over dispersion of each data point and provides an average measure as a normalized score between -1 and +1. Cohesion is a measure of intra-cluster distance and dispersion is a measure of inter-cluster distance. Values closer to +1 mean separated well defined clusters and values closer to -1 mean highly overlapping clusters - defeating the general purpose of clustering. If 'a' is the mean distance between a point and every other point in the same cluster, and if 'b' is the mean distance between a point and every other point in the nearest cluster, then the silhouette score for that point is defined as:

$$s = \frac{b - a}{max(a, b)} \tag{1}$$

The average s for all points is the Silhouette score for the clustering output.

5.1.2 Calinski - Harabasz Score

Calinski-Harabasz (CH) score is also called the Variance Ratio Criterion. This index provides a score calculated based on the co-variance. CH score is calculated as:

$$CH = \frac{tr(B_k)}{tr(W_k)} \frac{n - k}{k - 1} \tag{2}$$

where, B_k - co-variance matrix between clusters, W_k - co-variance matrix within clusters, n - sample size, k - number of clusters, and tr - trace of the matrix.

A higher CH score is better. The lowest possible CH score is 0 which indicates no dispersion among the clusters.

5.2 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. Okapi BM25 is still widely used as a benchmark in similar tasks (Fan et al., 2019). An advantage of using the Okapi BM25 is that it provides us with a tf-idf based benchmark (Sixto et al., 2016). Word vectors aim to reduce problem complexity by moving away from tf-idf methods which requires us to one-hot-encode the entire vocabulary.

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. This may seem biased, but in the absence of a gold standard we chose the proven and widely used Okapi BM25.

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.

To establish the cosine similarity threshold value 0.7, we used 75 questions from the Musical Instruments category (used only for bench marking purposes) and used top 5 answers that our model returns for each question. We then calculated cosine similarity between the sentences our model returned and the answer provided in the Amazon QA dataset. We took the 75th percentile value, which was 0.7, as the threshold.

5.2.1 Accuracy

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.

5.2.2 Correct Answer

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.

5.2.3 At least 50%

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.

The correct answer and at least 50% were inspired by the accuracy @ x% approach used by different authors working with the Amazon dataset and performing similar tasks (Fan et al., 2019; McAuley and Yang, 2016; Yu and Lam, 2018). In accuracy @ x% the commonly used measure is accuracy @ 50%. This approach helps in identifying the top answers crossing a threshold and has better relationship in real world applications (Fan et al., 2019).

5.3 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 five methods.

6 Results

6.1 Cluster Quality

Cluster quality was measured using the Silhouette score and the Calinski-Harabasz (CH) score. For each question, both these scores were calculated. Silhouette score cannot be calculated when there are less than two clusters. This situation arises for questions where the number of review sentences are limited. These occurrences were removed for analyzing cluster quality. All results presented on cluster quality uses a n = 647.

Figure 1a and Figure 1b show the Silhouette score and CH score for every single question. The algorithm naturally selects between 2 and 6 clusters for most of the questions and both the scores are high in this range. Benchmarks for Silhouette scores vary by task and the hockey-stick or elbow curve is looked at to make decisions about optimal cluster sizes.

Figure 1c and Figure 1d show the mean scores plotted as a function of the number of clusters. Our algorithm naturally limits the clusters to the optimal in most cases. The optimal number of clusters is between 2 and 6, with the CH score indicating 10 clusters having a better mean. Figure 2 shows that of the 647 questions 80% of the questions have the appropriate number of clusters. Using the Pareto (80-20) rule, our algorithm's clustering quality is good, as it chooses the appropriate number of clusters 80% of the time.

6.2 Sentence Level

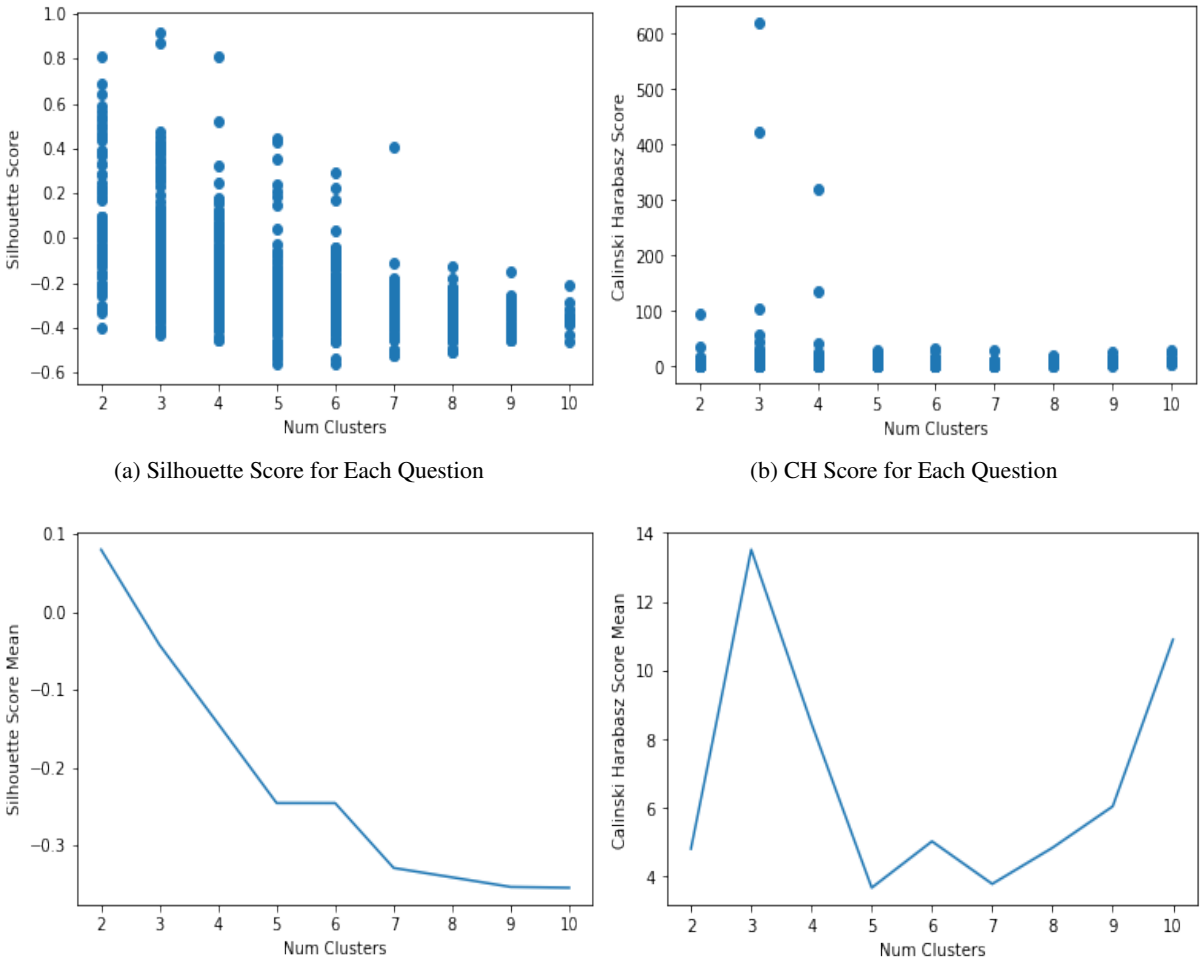
The sentence level evaluation was performed using the Okapi BM25 top sentence as the gold standard. Of the methods based on our system, the sim method consistently performs better than the other methods, as shown in Table 1. Except for the Correct Answer metric, sim method has the highest values in all other cases.

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 a 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 baseline 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.

6.3 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)



(a) Silhouette Score for Each Question

(b) CH Score for Each Question

(c) Mean Silhouette Score for Different Number of Clusters

(d) Mean CH Score for Different Number of Clusters

Figure 1: Clustering Quality Results

Table 1: Sentence Level Results

Score	Metric	Methods				
		sim	sim_ns	sim_med	sim_med_ns	3g
ROUGE-1	F	**45.86**	42.41	42.64	38.98	37.23
	P	45.94	43.17	43.04	39.88	38.72
	R	49.97	45.43	46.01	42.45	39.51
ROUGE-L	F	**42.26**	38.66	38.85	35.21	33.56
	P	44.46	41.63	41.22	38.18	36.90
	R	48.36	43.91	43.96	40.63	37.65
R-Net* ROUGE-L	F	40.22				
Similarity	Accuracy	**91.50**	82.60	91.30	82.80	87.10
	Correct Answer	83.60	72.40	**83.70**	72.90	75.50
	At least 50%	**79.77**	72.05	79.47	79.24	72.66

*This score is based on the work 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. We say this because, ROUGE-L looks for the longest common sub se-

Table 2: Answer Level Results

Score	Metric	Methods				
		sim	sim_ns	sim_med	sim_med_ns	3g
ROUGE-1	F	38.58	34.31	**38.63**	34.24	34.89
	P	63.00	65.99	62.33	65.04	61.96
	R	28.46	24.20	28.58	24.26	25.20
ROUGE-L	F	**29.66**	25.15	29.78	25.16	26.09
	P	59.72	63.28	58.99	62.18	58.74
	R	27.00	23.09	27.08	23.07	23.89
Similarity	Accuracy	**95.94**	91.02	96.36	91.19	93.88

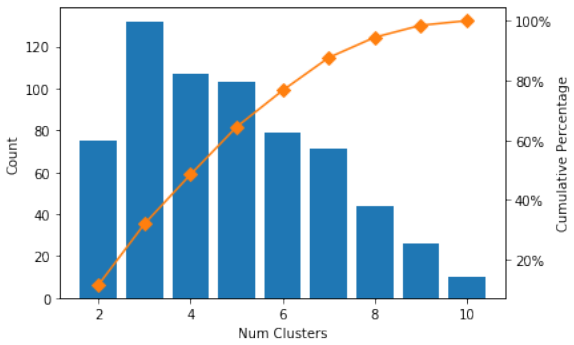


Figure 2: Pareto Chart for Number of Clusters

quence and penalizes shorter sentences. The sim method performs better with thh ROUGE-L and the accuracy metrics. Sim_med is better only with respect to the ROUGE-1 score.

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.

References

Miao Fan, Chao Feng, Mingming Sun, Ping Li, and Haifeng Wang. 2019. Reading customer reviews to answer product-related questions. In *Proceedings of the 2019 SIAM International Conference on Data Mining*, pages 567–575. SIAM.

Mansi Gupta, Nitish Kulkarni, Raghuveer Chanda, Anirudha Rayasam, and Zachary C. Lipton. 2019. Amazonqa: A review-based question answering task. *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*.

Soo-Min Kim and Eduard Hovy. 2005. Identifying opinion holders for question answering in opinion texts. In *Proceedings of AAAI-05 Workshop on Question Answering in Restricted Domains*, pages 1367–1373.

Fangtao Li, Yang Tang, Minlie Huang, and Xiaoyan Zhu. 2009. Answering opinion questions with random walks on graphs. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 737–745. Association for Computational Linguistics.

Jingjing Liu, Yunbo Cao, Chin-Yew Lin, Yalou Huang, and Ming Zhou. 2007. Low-quality product review detection in opinion summarization. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

Julian McAuley and Alex Yang. 2016. Addressing complex and subjective product-related queries with customer reviews. In *Proceedings of the 25th International Conference on World Wide Web*, pages 625–635.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Samaneh Moghaddam and Martin Ester. 2011. Aqa: aspect-based opinion question answering. In *2011 IEEE 11th International Conference on Data Mining Workshops*, pages 89–96. IEEE.

Juan Sixto, Aitor Almeida, and Diego López-de Ipiña. 2016. Improving the sentiment analysis process of spanish tweets with bm25. In *International Conference on Applications of Natural Language to Information Systems*, pages 285–291. Springer.

Swapna Somasundaran, Theresa Wilson, Janyce Wiebe, and Veselin Stoyanov. 2007. Qa with attitude: Exploiting opinion type analysis for improving question answering in on-line discussions and the news. In *ICWSM*.

Veselin Stoyanov, Claire Cardie, and Janyce Wiebe. 2005. Multi-perspective question answering using the opqa corpus. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 923–930. Association for Computational Linguistics.

Mengting Wan and Julian McAuley. 2016. Modeling ambiguity, subjectivity, and diverging viewpoints in opinion question answering systems. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 489–498. IEEE.

Hong Yu and Vasileios Hatzivassiloglou. 2003. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 129–136. Association for Computational Linguistics.

Jianxing Yu, Zheng-Jun Zha, and Tat-Seng Chua. 2012. Answering opinion questions on products by exploiting hierarchical organization of consumer reviews. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 391–401. Association for Computational Linguistics.

Qian Yu and Wai Lam. 2018. Aware answer prediction for product-related questions incorporating aspects. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 691–699.