Automatic generation of short informative sentiment summaries

Andrea Glaser and Hinrich Schütze
Institute for Natural Language Processing
University of Stuttgart, Germany
glaseraa@ims.uni-stuttgart.de

Abstract

In this paper, we define a new type of summary for sentiment analysis: a single-sentence summary that consists of a *supporting sentence* that conveys the overall sentiment of a review as well as a convincing reason for this sentiment. We present a system for extracting supporting sentences from online product reviews, based on a simple and unsupervised method. We design a novel comparative evaluation method for summarization, using a crowdsourcing service. The evaluation shows that our sentence extraction method performs better than a baseline of taking the sentence with the strongest sentiment.

1 Introduction

Given the success of work on sentiment analysis in NLP, increasing attention is being focused on how to present the results of sentiment analysis to the user. In this paper, we address an important use case that has so far been neglected: quick scanning of short summaries of a body of reviews with the purpose of finding a subset of reviews that can be studied in more detail. This use case occurs in companies that want to quickly assess, perhaps on a daily basis, what consumers think about a particular product. One-sentence summaries can be quickly scanned – similar to the summaries that search engines give for search results – and the reviews that contain interesting and new information can then be easily identified. Consumers who want to quickly scan review summaries to pick out a few reviews that are helpful for a purchasing decision are a similar use case.

For a one-sentence summary to be useful in this context, it must satisfy two different "information needs": it must convey the sentiment of the review, but it must also provide a specific reason for that sentiment, so that the user can make an informed decision as to whether reading the entire review is likely to be worth the user's time – again similar to the purpose of the summary of a web page in search engine results.

We call a sentence that satisfies these two criteria a *supporting sentence*. A supporting sentence contains information on the sentiment as well as a specific reason for why the author arrived at this sentiment. Examples for supporting sentences are *"The picture quality is very good"* or *"The battery life is 2 hours"*. Non-supporting sentences contain opinions without such reasons such as *"I like the camera"* or *"This camera is not worth the money"*.

To address use cases of sentiment analysis that involve quick scanning and selective reading of large numbers of reviews, we present a simple unsupervised system in this paper that extracts one supporting sentence per document and show that it is superior to a baseline of selecting the sentence with the strongest sentiment.

One problem we faced in our experiments was that standard evaluations of summarization would have been expensive to conduct for this study. We therefore used crowdsourcing to perform a new type of comparative evaluation method that is different from training set and gold standard creation, the dominant way crowdsourcing has been used in NLP so far.

In summary, our contributions in this paper are as follows. We define supporting sentences, a new type of sentiment summary that is appropriate in situations where both the sentiment of a review and a good reason for that sentiment need to be

Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, pages 276–285,
Avignon, France, April 23 - 27 2012.

conveyed succinctly. We present a simple unsupervised method for extracting supporting sentences and show that it is superior to a baseline in a novel crowdsourcing-based evaluation.

In the next section, we describe related work that is relevant to our new approach. In Section 3 we present the approach we use to identify supporting sentences. Section 4 describes the feature representation of sentences and the classification method. In Section 5 we give an overview of the crowdsourcing evaluation. Section 6 discusses our experimental results. In Sections 7 and 8, we present our conclusions and plans for future work.

2 Related Work

Both sentiment analysis (Pang and Lee, 2008; Liu, 2010) and summarization (Nenkova and McKeown, 2011) are important subfields of NLP. The work most relevant to this paper is work on summarization methods that addresses the specific requirements of summarization in sentiment analysis. There are two lines of work in this vein with goals similar to ours: (i) aspect-based and pro/con-summarization and (ii) approaches that extract summary sentences from reviews.

An aspect is a component or attribute of a product such as *"battery"*, *"lens cap"*, *"battery life"*, and *"picture quality"* for cameras. Aspect-oriented summarization (Hu and Liu, 2004; Zhuang et al., 2006; Kim and Hovy, 2006) collects sentiment assessments for a given set of aspects and returns a list of pros and cons about every aspect for a review or, in some cases, on a per-sentence basis.

Aspect-oriented summarization and pro/con-summarization differ in a number of ways from supporting sentence summarization. First, aspects and pros&cons are taken from a fixed inventory. The inventory is typically small and does not cover the full spectrum of relevant information. Second, in its most useful form, aspect-oriented summarization requires classification of phrases and sentences according to the aspect they belong to; e.g., *"The camera is very light"* has to be recognized as being relevant to the aspect *"weight"*. Developing a component that assigns phrases and sentences to their corresponding categories is time-consuming and has to be redone for each domain. Any such component will make mistakes and undetected or incorrectly classified aspects can result in bad summaries.

Our approach enables us to find strong supporting sentences even if the reason given in that sentence does not fit well into the fixed inventory. No manual work like the creation of an aspect inventory is necessary and there are no requirements on the format of the reviews such as author-provided pros and cons.

Aspect-oriented summarization also differs in that it does not differentiate along the dimension of quality of the reason given for a sentiment. For example, *"I don't like the zoom"* and *"The zoom range is too limited"* both give reasons for why a camera gets a negative evaluation, but only the latter reason is informative. In our work, we evaluate the quality of the reason given for a sentiment.

The use case we address in this paper requires a short, easy-to-read summary. A well-formed sentence is usually easier to understand than a pro/con table. It also has the advantage that the information conveyed is accurately representing what the user wanted to say – this is not the case for a presentation that involves several complex processing steps and takes linguistic material out of the context that may be needed to understand it correctly.

Berend (2011) performs a form of pro/con summarization that does not rely on aspects. However, most of the problems of aspect-based pro/con summarization also apply to this paper: no differentiation between good and bad reasons, the need for human labels to train a classifier, and inferior readability compared to a well-formed sentence.

Two previous approaches that have attempted to extract sentences from reviews in the context of summarization are (Beineke et al., 2004) and (Arora et al., 2009). Beineke et al. (2004) train a classifier on rottentomatoes.com summary sentences provided by review authors. These sentences sometimes contain a specific reason for the overall sentiment of the review, but sometimes they are just catchy lines whose purpose is to draw moviegoers in to read the entire review; e.g., *"El Bulli barely registers a pulse stronger than a book's"* (which does not give a specific reason for why the movie does not register a strong pulse).

Arora et al. (2009) define two classes of sentences: qualified claims and bald claims. A qualified claim gives the reader more details (e.g., *"This camera is small enough to fit easily in a*

coat pocket") while a bald claim is open to interpretation (e.g., *"This camera is small"*). Qualified/bald is a dimension of classification of sentiment statements that is to some extent orthogonal to quality of reason. Qualified claims do not have to contain a reason and bald claims can contain an informative reason. For example, *"I didn't like the camera, but I suspect it will be a great camera for first timers"* is classified as a qualified claim, but the sentence does not give a good reason for the sentiment of the document. Both dimensions (qualified/bald, high-quality/low-quality reason) are important and can be valuable components of a complete sentiment analysis system.

Apart from the definition of the concept of supporting sentence, which we believe to be more appropriate for the application we have in mind than rottentomatoes.com summary sentences and qualified claims, there are two other important differences of our approach to these two papers. First, we directly evaluate the quality of the reasons in a crowdsourcing experiment. Second, our approach is unsupervised and does not require manual annotation of a training set of supporting sentences.

As we will discuss in Section 5, we propose a novel evaluation measure for summarization based on crowdsourcing in this paper. The most common use of crowdsourcing in NLP is to have workers label a training set and then train a supervised classifier on this training set. In contrast, we use crowdsourcing to directly evaluate the relative quality of the automatic summaries generated by the unsupervised method we propose.

3 Approach

Our approach is based on the following three premises.

(i) *A good supporting sentence conveys both the review's sentiment and a supporting fact.* We make this assumption because we want the sentence to be self-contained. If it only describes a fact about a product without evaluation, then it does not on its own explain which sentiment is conveyed by the article and why.

(ii) *Supporting facts are most often expressed by noun phrases.* We call a noun phrase that expresses a supporting fact a *keyphrase*. We are not assuming that *all* important words in the supporting sentence are nominal; the verb will be needed in many cases to accurately convey the reason for the sentiment expressed. However, it is a fairly safe assumption that part of the information is conveyed using noun phrases since it is difficult to convey specific information without using specific noun phrases. Adjectives are often important when expressing a reason, but frequently a noun is also mentioned or one would need to resolve a pronoun to make the sentence a self-contained supporting sentence. In a sentence like *"It's easy to use"* it is not clear what the adjective is referring to.

(iii) *Noun phrases that express supporting facts tend to be domain-specific; they can be automatically identified by selecting noun phrases that are frequent in the domain – either in relative terms (compared to a generic corpus) or in absolute terms.* By making this assumption we may fail to detect supporting sentences that are worded in an original way using ordinary words. However, in a specific domain there is usually a lot of redundancy and most good reasons occur many times and are expressed by similar words.

Based on these assumptions, we select the supporting sentence in two steps. In the first step, we determine the n sentences with the strongest sentiment within every review by classifying the polarity of the sentences (where n is a parameter). In the second step, we select one of the n sentences as the best supporting sentence by means of a weighting function.

Step 1: Sentiment Classification

In this step, we apply a sentiment classifier to all sentences of the review to classify sentences as positive or negative. We then select the n sentences with the highest probability of conforming with the overall sentiment of the document. For example, if the document's polarity is negative, we select the n sentences that are most likely to be negative according to the sentiment classifier. We restrict the set of n sentences to sentences with the "right" sentiment because even an excellent supporting sentence is not a good characterization of

the content of the review if it contradicts the overall assessment given by the review. Only in cases where there are fewer than n sentences with the correct sentiment, we also select sentences with the "wrong" sentence to obtain a minimum of n sentences for each review.

Step 2: Weighting Function

Based on premises (ii) and (iii) above, we score a sentence based on the number of noun phrases that *occur with high absolute and relative frequency* in the domain. We only consider simple nouns and compound nouns consisting of two nouns in this paper. In general, compound nouns are more informative and specific. A compound noun may refer to a specific reason even if the head noun does not (e.g., *"life"* vs. *"battery life"*). This means that we need to compute scores in a way that allows us to give higher weight to compound nouns than to simple nouns.

In addition, we also include counts of nouns and compounds in the scoring that do not have high absolute/relative frequency because frequency heuristics identify keyphrases with only moderate accuracy. However, theses nouns and compounds are given a lower weight.

This motivates a scoring function that is a weighted sum of four variables: number of simple nouns with high frequency, number of infrequent simple nouns, number of compound nouns with high frequency, and number of infrequent compound nouns. High frequency is defined as follows. Let $f_{dom}(p)$ be the domain-specific absolute frequency of phrase p, i.e., the frequency in the review corpus, and $f_{wiki}(p)$ the frequency of p in the English Wikipedia. We view the distribution of terms in Wikipedia as domain-independent and define the relative frequency as in Equation 1.

$$f_{rel}(p) = \frac{f_{dom}(p)}{f_{wiki}(p)} \tag{1}$$

We do not consider nouns and compound nouns that do not occur in Wikipedia for computing the relative frequency. A noun (resp. compound noun) is deemed to be of high frequency if it is one of the $k\%$ nouns (resp. compound nouns) with the highest $f_{dom}(p)$ and at the same time is one of the $k\%$ nouns (resp. compound nouns) with the highest $f_{rel}(p)$ where k is a parameter.

Based on these definitions, we define four different sets: F_1 (the set of nouns with high frequency), I_1 (the set of infrequent nouns), F_2 (the set of compounds with high frequency), and I_2 (the set of infrequent compounds). An infrequent noun (resp. compound) is simply defined as a noun (resp. compound) that does not meet the frequency criterion.

We define the score s of a sentence with n tokens $t_1 \dots t_n$ (where the last token t_n is a punctuation mark) as follows:

$$\begin{aligned} s = \sum_{i=1}^{n-1} & \; w_{f_2} \cdot [\![(t_i, t_{i+1}) \in F_2]\!] \\ & + w_{i_2} \cdot [\![(t_i, t_{i+1}) \in I_2]\!] \\ & + w_{f_1} \cdot [\![t_i \in F_1]\!] \\ & + w_{i_1} \cdot [\![t_i \in I_1]\!] \end{aligned} \tag{2}$$

where $[\![\phi]\!] = 1$ if ϕ is true and $[\![\phi]\!] = 0$ otherwise. Note that a noun in a compound will contribute to the overall score in two different summands.

The weights w_{f_2}, w_{i_2}, w_{f_1}, and w_{i_1} are determined using logistic regression. The training set for the regression is created in an unsupervised fashion as follows. From each set of n sentences (one per review), we select the two highest scoring, i.e., the two sentences that were classified with the highest confidence. The two classes in the regression problem are then the top ranked sentences vs. the sentences at rank 2. Since taking all sentences turned out to be too noisy, we eliminate sentence pairs where the top sentence is better than the second sentence on almost all of the set counts (i.e., count of members of F_1, I_1, F_2, and I_2). Our hypothesis in setting up this regression was that the sentence with the strongest sentiment often does not give a good reason. Our experiments confirm that this hypothesis is true.

The weights w_{f_2}, w_{i_2}, w_{f_1}, and w_{i_1} estimated by the regression are then used to score sentences according to Equation 2.

We give the same weight to all keyphrase compounds (and the same weight to all keyphrase nouns) – in future work one could attempt to give higher weights to keyphrases with higher absolute or relative frequency. In this paper, our goal is to establish a simple baseline for the task of extraction of supporting sentences.

After computing the overall weight for each sentence in a review, the sentence with the highest weight is chosen as the supporting sentence – the sentence that is most informative for explaining the overall sentiment of the review.

4 Experiments

4.1 Data

We use part of the Amazon dataset from Jindal and Liu (2008). The dataset consists of more than 5.8 million consumer-written reviews of several products, taken from the Amazon website. For our experiment we used the digital camera domain and extracted 15,340 reviews covering a total of 740 products. See table 1 for key statistics of the data set.

Type	Number
Brands	17
Products	740
Documents (all)	15,340
Documents (cleaned)	11,624
Documents (train)	9,880
Documents (test)	1,744
Short test documents	147
Long test documents	1,597
Average number of sents	13.36
Median number of sents	10

Table 1: Key statistics of our dataset

In addition to the review text, authors can give an overall rating (a number of stars) to the product. Possible ratings are 5 (very positive), 4 (positive), 3 (neutral), 2 (negative), and 1 (very negative). We unify ratings of 4 and 5 to "positive" and ratings of 1 and 2 to "negative" to obtain polarity labels for binary classification. Reviews with a rating of 3 are discarded.

4.2 Preprocessing

We tokenized and part-of-speech (POS) tagged the corpus using TreeTagger (Schmid, 1994). We split each review into individual sentences by using the sentence boundaries given by TreeTagger. One problem with user-written reviews is that they are often not written in coherent English, which results in wrong POS tags. To address some of these problems, we cleaned the corpus after the tokenization step. We separated word-punctuation clusters (e.g., *word...word*) and removed emoticons, html tags, and all sentences with three or fewer tokens, many of which were a result of wrong tokenization. We excluded all reviews with fewer than five sentences. Short reviews are often low-quality and do not give good reasons. The cleaned corpus consists of 11,624 documents. Finally, we split the corpus into training set (85%) and test set (15%) as shown in Table 1. The average number of sentences of a review is 13.36 sentences, the median number of sentences is 10.

4.3 Sentiment Classification

We first build a sentence sentiment classifier by training the Stanford maximum entropy classifier (Manning and Klein, 2003) on the sentences in the training set. Sentences occurring in positive (resp. negative) reviews are labeled positive (resp. negative). We use a simple bag-of-words representation (without punctuation characters and frequent stop words). Propagating labels from documents to sentences creates a noisy training set because some sentences have sentiment different from the sentiment in their documents; however, there is no alternative because we need per-sentence classification decisions, but do not have per-sentence human labels.

The accuracy of the classifier is 88.4% on "propagated" sentence labels.

We use the sentence classifier in two ways. First, it defines our *baseline* BL for extracting supporting sentences: the baseline simply proposes the sentence with the highest sentiment score that is compatible with the sentiment of the document as the supporting sentence.

Second, the sentence classifier selects a subset of candidate sentences that is then further processed using the scoring function in Equation 2. This subset consists of the $n = 5$ sentences with the highest sentiment scores of the "right" polarity – that is, if the document is positive (resp. negative), then the $n = 5$ sentences with the highest positive (resp. negative) scores are selected.

4.4 Determining Frequencies and Weights

The absolute frequency of nouns and compound nouns simply is computed as their token frequency in the training set. For computing the relative frequency (as described in Section 3, Equation 1), we use the 20110405 dump of the English Wikipedia.

In the product review corpora we studied, the percentage of high-frequency keyphrase compound nouns was higher than that of simple nouns. We therefore use two different thresholds for absolute and relative frequency. We de-

fine F_1 as the set of nouns that are in the top $k_n = 2.5\%$ for both absolute and relative frequencies; and F_2 as the set of compounds that are in the top $k_p = 5\%$ for both absolute and relative frequencies. These thresholds are set to obtain a high density of good keyphrases with few false positives. Below the threshold there are still other good keyphrases, but they cannot be separated easily from non-keyphrases.

Sentences are scored according to Equation 2. Recall that the parameters w_{f_2}, w_{i_2}, w_{f_1}, and w_{i_1} are determined using logistic regression. The obtained parameter values (see table 2) indicate the relative importance of the four different types of terms. Compounds are the most important term and even those with a frequency below the threshold k_p still provide more detailed information than simple nouns above the threshold k_n; the value of w_{i_2} is approximately twice the value w_{f_1} for this reason. Non-keyphrase nouns are least important and are weighted with only a very small value of $w_{i_1} = 0.01$.

Phrase	Par	Value
keyphrase compounds	w_{f_2}	1.07
non-keyphrase compounds	w_{i_2}	0.89
keyphrase nouns	w_{f_1}	0.46
non-keyphrase nouns	w_{i_1}	0.01

Table 2: Weight settings

The scoring function with these parameter values is applied to the $n = 5$ selected sentences of the review. The highest scoring sentence is then selected as the supporting sentence proposed by our system.

For 1380 of the 1744 reviews, the sentence selected by our system is different from the baseline sentence; however, there are 364 cases (20.9%) where the two are the same. Only the 1380 cases where the two methods differ are included in the crowdsourcing evaluation to be described in the next section. As we will show below, our system selects better supporting sentences than the baseline in most cases. So if baseline and our system agree, then it is even more likely that the sentence selected by both is a good supporting sentence. However, there could also be cases where the $n = 5$ sentences selected by the sentiment classifier are all bad supporting sentences or cases where the document does not contain any good supporting sentences.

5 Comparative Evaluation with Amazon Mechanical Turk

One standard way to evaluate summarization systems is to create hand-edited summaries and to compute some measure of similarity (e.g., word or n-gram overlap) between automatic and human summaries. An alternative for extractive summaries is to classify all sentences in the document with respect to their appropriateness as summary sentences. An automatic summary can then be scored based on its ability to correctly identify good summary sentences. Both of these methods require a large annotation effort and are most likely too complex to be outsourced to a crowdsourcing service because the creation of manual summaries requires skilled writers. For the second type of evaluation, ranking sentences according to a criterion is a lot more time consuming than making a binary decision – so ranking the 13 or 14 sentences that a review contains on average for the entire test set would be a significant annotation effort. It would also be difficult to obtain consistent and repeatable annotation in crowdsourcing on this task due to its subtlety.

We therefore designed a novel evaluation methodology in this paper that has a much smaller startup cost. It is well known that relative judgments are easier to make on difficult tasks than absolute judgments. For example, much recent work on relevance ranking in information retrieval relies on relative relevance judgments (one document is more relevant than another) rather than absolute relevance judgments. We adopt this general idea and only request such relative judgments on supporting sentences from annotators. Unlike a complete ranking of the sentences (which would require $m(m-1)/2$ judgments where m is the length of the review), we choose a setup where we need to only elicit a single relative judgment per review, one relative judgment on a sentence pair (consisting of the baseline sentence and the system sentence) for each of the 1380 reviews selected in the previous section. This is a manageable annotation task that can be run on a crowdsourcing service in a short time and at little cost.

We use Amazon Mechanical Turk (AMT) for this annotation task. The main advantage of AMT is that cost per annotation task is very low, so that we can obtain large annotated datasets for an af-

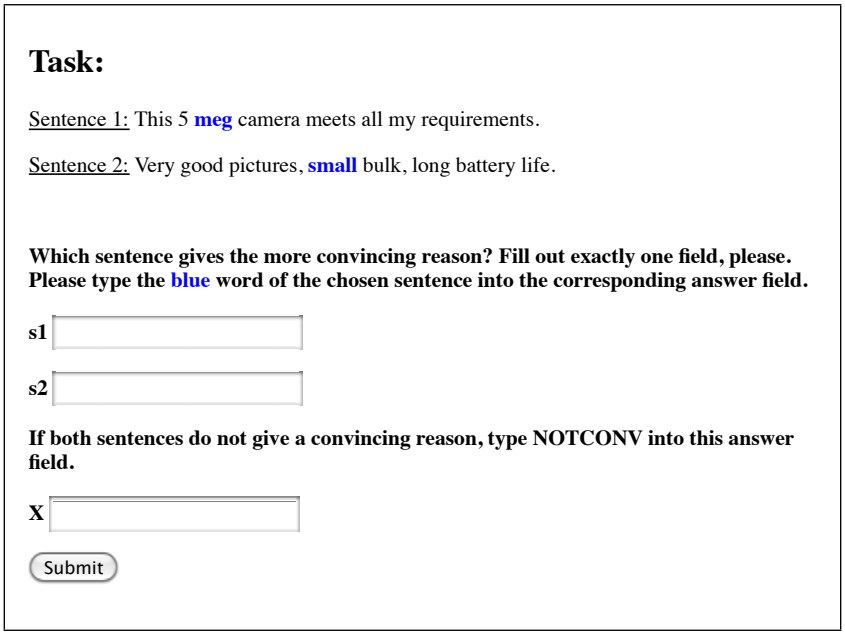

Figure 1: AMT interface for annotators

fordable price. The disadvantage is the level of quality of the annotation which will be discussed at the end of this section.

5.1 Task Design

We created a HIT (Human Intelligence Task) template including detailed annotation guidelines. Every HIT consists of a pair of sentences. One sentence is the baseline sentence; the other sentence is the system sentence, i.e., the sentence selected by the scoring function. The two sentences are presented in random order to avoid bias.

The workers are then asked to evaluate the relative quality of the sentences by selecting one of the following three options:

1. Sentence 1 has the more convincing reason
2. Sentence 2 has the more convincing reason
3. Neither sentence has a convincing reason

If both sentences contain reasons, the worker has to compare the two reasons and choose the sentence with the more convincing reason.

Each HIT was posted to three different workers to make it possible to assess annotator agreement. Every worker can process each HIT only once so that the three assignments are always done by three different people.

Based on the worker annotations, we compute a gold standard score for each sentence. This score is simply the number of times it was rated better than its competitor. The score can be 0, 1, 2 or 3. HITs for which the worker chooses the option "Neither sentence has a convincing reason" are ignored when computing sentence scores.

The sentence with the higher score is then selected as the best supporting sentence for the corresponding review.

In cases of ties, we posted the sentence pair one more time for one worker. If one of the two sentences has a higher score after this reposting, we choose it as the winner. Otherwise we label this sentence pair "no decision" or "N-D".

5.2 Quality of AMT Annotations

Since our crowdsourcing based evaluation is novel, it is important to investigate if human annotators perform the annotation consistently and reproducibly.

The Fleiss' κ agreement score for the final experiment is 0.17. AMT workers only have the instructions given by the requesters. If they are not clear enough or too complicated, workers can misunderstand the task, which decreases the quality of the answers. There are also AMT workers who spam and give random answers to tasks. Moreover, ranking sentences according to the quality of the given reason is a subjective task. Even if the sentence contains a reason, it might not be convincing for the worker.

To ensure a high level of quality for our dataset,

	Experiment	# Docs	BL	SY	N-D	B=S
1	AMT, first pass	1380	27.4	57.9	14.7	-
2	AMT, second pass	203	46.8	45.8	7.4	-
3	AMT final	1380	34.3	**64.6**	1.1	-
4	AMT+[B=S]	1744	27.1	**51.1**	0.9	20.9

Table 3: AMT evaluation results. Numbers are percentages or counts. BL = baseline, SY = system, N-D = no decision, B=S = same sentence selected by baseline and system

we took some precautions. To force workers to actually read the sentences and not just click a few boxes, we randomly marked one word of each sentence blue. The worker had to type the word of their preferred sentence into the corresponding answer field or `NOTCONV` into the special field if neither sentence was convincing. Figure 1 shows our AMT interface design.

For each answer field we have a gold standard (the words we marked blue and the word `NOTCONV`) which enables us to look for spam. The analysis showed that some workers mistyped some words, which however only indicates that the worker actually typed the word instead of copying it from the task. Some workers submitted inconsistent answers, for instance, they typed a random word or filled out all three answer fields. In such cases we reposted this HIT again to receive a correct answer.

After the task, we counted how often a worker said that neither sentence is convincing since a high number indicates that the worker might have only copied the word for several sentence pairs without checking the content of the sentences. We also analyzed the time a worker needed for every HIT. Since no task was done in less than 10 seconds, the possibility of just copying the word was rather low.

6 Results and discussion

The results of the AMT experiment are shown in table 3. As described above, each of the 1380 sentence pairs was evaluated by three workers. Workers rated the system sentence as better for 57.9% of the reviews, and the baselines sentence as better for 27.4% of the reviews; for 14.7% of reviews, the scores of the two sentences were tied (line 1 of Table 3). The 203 reviews in this category were reposted one more time (as described in Section 5). The responses were almost perfectly evenly split: about 47% of workers preferred the baseline system, 46% the system sentence; 7.4% of the responses were undecided (line 2). Line 3 presents the consolidated results where the 14.7% ties on line 1 are replaced by the ratings obtained on line 2 in the second pass.

The consolidated results (line 3) show that our system is clearly superior to the baseline of selecting the sentence with the strongest sentiment. Our system selected a better supporting sentence for 64.6% of the reviews; the baseline selected a better sentence for 34.3% of the reviews. These results exclude the reviews where baseline and system selected the same sentence. If we assume that these sentences are also acceptable sentences (since they score well on the traditional sentiment metrics as well as on our new content keyword metric), then our system finds a good supporting sentence for 72.0% of reviews (51.1+20.9) whereas the baseline does so for only 48.0% (27.1+20.9).

6.1 Error Analysis

Our error analysis revealed that a significant proportion of system sentences that were worse than baseline sentences did contain a reason. However, the baseline sentence also contained a reason and was rated better by AMT annotators. Examples (1) and (2) show two such cases. The first sentence is the baseline sentence (BL) which was rated better. The system sentence (SY) contains a similar or different reason. Since rating reasons is a very subjective task, it is impossible to define which of these two sentences contains the better reason and depends on how the workers think about it.

(1) BL: *The best thing is that everything is just so easily displayed and one doesn't need a manual to start getting the work done.*

SY: *The zoom is incredible, the video was so clear that I actually thought of making a 15 min movie.*

(2) BL: *The colors are horrible, indoor shots are horrible, and too much noise.*

SY: *Who cares about 8 mega pixels and 1600 iso when it takes such bad quality pictures.*

In example (3) the system sentence is an incomplete sentence consisting of only two noun phrases. These cut-off sentences are mainly caused by incorrect usage of grammar and punctuation by the reviewers which results in wrongly determined sentence boundaries in the preprocessing step.

(3) BL: *Gives peace of mind to have it fit perfectly.*

SY: *battery and SD card.*

In some cases, the two sentences that were presented to the worker in the evaluation had a different polarity. This can have two reasons: (i) due to noisy training input, the classifier misclassified some of the sentences, and (ii) for short reviews we also used sentences with the non-conforming polarity. Sentences with different polarity often confused the workers and they tended to prefer the positive sentence even if the negative one contained a more convincing reason as can be seen in example (4).

(4) BL: *It shares same basic commands and setup, so the learning curve was minimal.*

SY: *I was not blown away by the image quality, and as others have mentioned, the flash really is weak.*

A general problem with our approach is that the weighting function favors sentences with many noun phrases. The system sentence in example (5) contains many noun phrases, including some highly frequent nouns (e.g., *"lens", "battery"*), but there is no convincing reason and the baseline sentence has been selected by the workers.

(5) BL: *I have owned my cd300 for about 3 weeks and have already taken 700 plus pictures.*

SY: *It has something to do with the lens because the manual says it only happens to the 300 and when I called Sony tech support the guy tried to tell me the battery was faulty and it wasn't.*

Finally, there are a number of cases where our assumption that good supporting sentences contain keyphrases is incorrect. For example, sentence (6) does not contain any keyphrases indicative of good reasons. The information that makes it a good supporting sentence is mainly expressed using verbs and particles.

(6) *I have had an occasional problem with the camera not booting up and telling me to turn it off and then on again.*

7 Conclusion

In this work, we presented a system that extracts supporting sentences, single-sentence summaries of a document that contain a convincing reason for the author's opinion about a product. We used an unsupervised approach that extracts keyphrases of the given domain and then weights these keyphrases to identify supporting sentences. We used a novel comparative evaluation methodology with the crowdsourcing framework Amazon Mechanical Turk to evaluate this novel task since no gold standard is available. We showed that our keyphrase-based system performs better than a baseline of extracting the sentence with the highest sentiment score.

8 Future work

Our method failed for some of the about 35% of reviews where it did not find a convincing reason because of the noisiness of reviews. Reviews are user-generated content and contain grammatically incorrect sentences and are full of typographical errors. This problem makes it hard to perform preprocessing steps like part-of-speech tagging and sentence boundary detection correctly and reliably. We plan to address these problems in future work by developing a more robust processing pipeline.

Acknowledgments

This work was supported by Deutsche Forschungsgemeinschaft (Sonderforschungsbereich 732, Project D7) and in part by the IST Programme of the European Community, under the PASCAL2 Network of Excellence, IST-2007-216886. This publication only reflects the authors' views.

References

Shilpa Arora, Mahesh Joshi, and Carolyn P. Rosé. 2009. Identifying types of claims in online customer reviews. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, NAACL-Short '09, pages 37–40, Stroudsburg, PA, USA. Association for Computational Linguistics.

Philip Beineke, Trevor Hastie, Christopher Manning, and Shivakumar Vaithyanathan. 2004. Exploring sentiment summarization. In *Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*. AAAI Press. AAAI technical report SS-04-07.

Gábor Berend. 2011. Opinion expression mining by exploiting keyphrase extraction. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1162–1170, Chiang Mai, Thailand, November. Asian Federation of Natural Language Processing.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pages 168–177, New York, NY, USA. ACM.

Nitin Jindal and Bing Liu. 2008. Opinion spam and analysis. In *WSDM '08: Proceedings of the international conference on Web search and web data mining*, pages 219–230, New York, NY, USA. ACM.

Soo-Min Kim and Eduard Hovy. 2006. Automatic identification of pro and con reasons in online reviews. In *Proceedings of the COLING/ACL on Main conference poster sessions*, COLING-ACL '06, pages 483–490, Stroudsburg, PA, USA. Association for Computational Linguistics.

Bing Liu. 2010. Sentiment analysis and subjectivity. *Handbook of Natural Language Processing*, 2nd ed.

Christopher Manning and Dan Klein. 2003. Optimization, maxent models, and conditional estimation without magic. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: Tutorials - Volume 5*, NAACL-Tutorials '03, pages 8–8, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ani Nenkova and Kathleen McKeown. 2011. Automatic summarization. *Foundations and Trends in Information Retrieval*, 5(2-3):103–233.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*, Manchester, UK.

Li Zhuang, Feng Jing, and Xiao-Yan Zhu. 2006. Movie review mining and summarization. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, CIKM '06, pages 43–50, New York, NY, USA. ACM.