Natural Language Summarization of Evaluative Arguments

Giuseppe Carenini, Raymond Ng, and Adam Pauls
Deptartment of Computer Science
University of British Columbia Vancouver, Canada
{carenini,rng,adpauls}@cs.ubc.ca

Abstract

We present and compare two approaches to the task of summarizing evaluative arguments. The first is a domain-independent sentence extraction-based approach, while the second is a weakly domain-dependent language generation-based approach. We evaluate these approaches in a user study and find that they quantitatively perform equally well. Qualitatively, we find that they perform well for different but complementary reasons. We conclude that an effective method for summarizing evaluative arguments must effectively synthesize the two approaches.

1 Introduction

Many organizations are faced with the challenge of summarizing large corpora of text data. One important application is evaluative text, i.e. any document expressing an evaluation of an entity as either positive or negative. For example, many websites collect large quantities of online customer reviews of consumer electronics. Summaries of this literature could be of great strategic value to product designers, planners and manufacturers. Beyond customer reviews, there are other equally important commercial applications, such as the summarization of travel logs, and non-commercial applications, such as the summarization of candidate reviews.

Most previous work on multi-document summarization however has been focusing on factual text (e.g., news () Newsblaster, biographies (**?**)). With the exception of (cite related work) little attention has been paid to multi-document summarization of evaluative text. The general problem we consider in this paper is how to effectively summarize a large corpora of evaluative text about a single entity (e.g., a product). We based our study on the following analysis of the similarities and differences between multi-document summarization of evaluative vs. factual text.

Generally speaking, factual documents tend to be written in the third person and contain a consistent set of facts. The goal of a summarizer is to select the most important facts and present them in a sensible ordering by avoiding repetition. Previous work has shown that this can be effectively achieved by carefully extracting and ordering the most informative sentences from the original documents in a domain-independent way. Notice however that when the source documents are assumed to contain inconsistent information (e.g., conflicting reports of a natural disaster [RIPTIDES]), a different approach is needed. The summarizer needs first to extract the information from the documents, then process such information to identify overlaps and inconsistencies between the different sources and finally produce a summary that point out and explain those inconsistencies. In RIPTIDES, the summary is produced by combining language generated from the extracted information with sentences extracted from the source reports.

A corpus of evaluative text typically contains a large number of possibly inconsistent facts, as opinions on the same entity feature may be uniform or varied. Thus, summarizing a corpus of evaluative text is much more similar to summarizing conflicting reports than a consistent set of factual documents.

Based on this observation, we argue that any strategy to effectively summarize evaluative text about a single entity should rely on a preliminary phase of information extraction from the target corpus as it is the case for summarizing conflicting factual documents. In particular, the summarizer should at least know for each document: what features of the entity were evaluated, the polarity of the evaluations and their strengths.

In this paper, we explore this hypothesis by considering two alternative approaches. First, we developed a sentence-extraction based summarizer that uses the information extracted from the corpus to select and rank sentences from the corpus. We implemented this system, called MEAD*, by adapting MEAD [ref], an open-source framework for multi-document summarization. Second, we developed a summarizer that produces summaries primarily by generating language from the information extracted from the corpus. We implemented this system, called the Summarizer of

Evaluative Arguments), by adapting the Generator of Evaluative Arguments (GEA) [ref] a framework for generating user tailored evaluative arguments.

We have performed an empirical evaluation of MEAD* and SEA in a user study. In this evaluation, we also tested the effectiveness of human generated summaries (HGS) as a topline and of summaries generated by MEAD without access to the extracted information as a baseline. The results indicate that SEA and MEAD* quantitatively perform equally well above MEAD and below HGS. Qualitatively, we find that they perform well for different but complementary reasons. While SEA appears to provide a more coherent overview of the source text, MEAD* seems to provide a more varied language and detail about customer opinions.

In the reminder of the paper, we first summarize our framework for information extraction from evaluative text. Then, we present MEAD* and SEA. After that, we describe our user study and discuss the results. We conclude with a discussion of related work.

2 Extraction of Information from Evaluative Text

2.1 Feature Extraction

Knowledge extraction from evaluative text about a single entity is typically decomposed into three distinct phases: the determination of features of the entity evaluated in the text, the strength of each evaluation, and the polarity of each evaluation. For instance, the information extracted from the sentence *"The menus are very easy to navigate but the user preference dialog is somewhat difficult to locate."* should be that the "menus" and the "user preference dialog" features are evaluated, and that the "menus" receive a very positive evaluation while the "user preference dialog" is evaluated rather negatively.

For these tasks, we adopt the approach described in detail in (**?**). This approach relies on the work of (**?**) for the tasks of strength and polarity determination. For the task of feature extraction, it enhances earlier work (**?**) by mapping the extracted features into a user-defined hierarchy of features which describes the entity of interest. Carenini *et al.* (**?**) show that the resulting mapping reduces redundancy and provides conceptual organization of the crude features. In this paper, we use this organization to generate a textual summary of the extracted knowledge in SEA.

Before, continuing, we shall describe the terminology we use when discussing the extracted knowledge. For a corpus of reviews, there is a set of extracted crude features

$$CF = \{cf_j\} \;\; j = 1...n$$

Figure 1: Partial view of UDF taxonomies for a digital camera.

- Camera
  - Lens
    - Digital Zoom
    - Optical Zoom
    - ...
  - Editing/Viewing
    - Viewfinder
    - ...
  - Flash
  - ...

- Image
  - Image Type
    - TIFF
    - JPEG
    - ...
  - Resolution
    - Effective Pixels
    - Aspect Ratio
  - ...

For example, crude features for a digital camera might include "picture quality", "viewfinder", and "lens". Each sentence s_k in the corpus contains a set of evaluations called $eval(s_k)$. Each evaluation contains both a polarity and a strength represented as an integer in the range $[-3, -2, -1, +1, +2, +3]$ where $+3$ is the most positive possible evaluation and -3 is the most negative possible evaluation.

There is also a hierarchical set of user-defined features

$$UDF = \{udf_i\} \;\; i = 1...m$$

See Figure 1 for a sample UDF. The process of hierarchically organizing the extracted produces a mapping from CF to UDF features. We call the set of crude features mapped to the user-defined feature udf_i $map(udf_i)$. For example, the crude features "unresponsiveness", "delay", and "lag time" would all be mapped to the user-defined feature "delay between shots".

For each cf_j, there is a set of polarity and strength evaluations $ps(cf_j)$ corresponding to each evaluation of cf_j in the corpus. We call the set of polarity/strength evaluations directly associated with udf_i

$$PS_i = \bigcup_{cf_j \epsilon map(udf_i)} ps(cf_j)$$

3 MEAD*: A Sentence Extraction-based Summarization Approach

Most modern summarization systems use sentences extracted from the source text as the basis for summarization (). Some systems perform some manipulation of the extracted sentences, for example, anaphora resolution (()) or sentence compression (()), while others leave the extracted sentences entirely intact ((), ()). Extraction-based approaches have the advantage of avoiding the difficult task of natural language generation, thus maintaining domain-independence because the system need not be aware of specialized vocabulary for its target domain.

An extraction-based summarizer must perform two key tasks: it must (i) choose informative sentences and (ii) create a linguistically coherent summary from those sentences. In practice, most of the effort directed is towards the former. Not only is this task the easier of the two, it is also more important for creating informative summaries. While linguistic coherence is important, it is not necessarily essential in all summarization tasks (XXX is this true? citations?).

Because of the widespread and well-developed use of sentence extractors in the summarization community, we chose to develop our own sentence extractor as a first attempt at summarizing evaluative arguments. Because we wanted to make use of the task-specific information extraction described in Section 2, we did not directly use an existing system. Rather, we adapted MEAD (), an open-source framework for multi-document summarization, to suit our purposes. We refer to our adapted version of MEAD as MEAD*.

The MEAD framework decomposes sentence extraction into three steps

1. *Feature Calculation*: Some numerical feature(s) are calculated for each sentence. Several independent features may be calculated for each sentence, for example, a score based on document position and a score based on the TF*IDF of a sentence.

2. *Classification*: The features calculated during Feature Calculation are combined into a single numerical score for each sentence.

3. *Reranking*: The numerical score for each sentence is adjusted relative to other sentences. This allows the system to avoid redundancy in the final set of sentences by lowering the score of sentences which are similar to already selected sentences.

In addition to providing a framework for accomplishing each task, MEAD also provides several common algorithms for each step. It provides support for one of the most common sentence-level feature calculations in the summarization literature, namely the similarity of each sentence to the 'document centroid' (cf. (), (), ()). By default, MEAD also calculates features for each sentence based on document position and sentence length. For classification, the default algorithm used by MEAD is to use a weighted linear combination of the individual feature scores. The default reranking is to zero out the scores of sentences which exceed a user-defined threshold of similarity with already selected sentences as calculated by the cosine measure of vector similarity (()).

We found from early experimentation that the most informative sentences could be accurately determined by examining the extracted CFs. Thus, we created our own sentence-level feature based on the number, strength, and polarity of CFs extracted for each sentence.

$$CF\_sum(s_k) = \sum_{ps_i \epsilon \ eval(s_k)} |ps_i|$$

During system development, we found this measure to be effective because it was sensitive to the number of CFs mentioned in a given sentence as well as to the strength of the evaluation for each CF. However, many sentences may have the same CF_sum score (especially sentences which contain an evaluation for only one CF). In such cases, we used the centroid feature as a 'tie-breaker'. We accomplished this by calculating both the centroid score as implemented by MEAD 3.07[1] and our own CF_sum score at the feature calculation stage. We then weighted the CF_sum score three times that of the centroid score at the classification stage. Since the centroid scores computed by MEAD are normalized between 0 and 1, any change in CF_sum would outweigh changes in the centroid score.

At the reranking stage, we adopted a different algorithm than the default in MEAD*. We placed each sentence which contained an evaluation of a given CF into a 'bucket' for that CF. Because a sentence could contain more than one CF, a sentence could be placed in multiple buckets. We then selected the top-ranked sentence from each bucket, starting with the bucket containing the most sentences (largest $|ps(cf_j)|$), never selecting the same sentence twice. Once one sentence had been selected from each bucket, the process was repeated[2]. This selection algorithm accomplishes two important tasks: firstly, it avoids redundancy by only selecting one sentence to represent each CF (unless all other CFs have already been represented), and secondly, it gives priority to CFs which are mentioned more frequently in the text.

An alert reader might wonder why we did not use the UDF during sentence selection. We did this because we wanted MEAD* to be entirely domain independent. Using the UDF would inject user-defined domain specific knowledge into the summarizer.

4 SEA: Natural Language Generated-based Summarization Approach

The extraction-based approach described in the previous section has several disadvantages. We already discussed problems with the linguistic coherence of the summary, but more specific problems arise in our particular task of summarizing customer evaluations. Firstly, sentence extraction does not give the reader any explicit information about of the distribution of evaluations, for example, how many users mentioned a given feature and whether user opinions were uniform or varied. It also does not give an aggregate view of user evaluations because it only presents one evaluation for

[1] The centroid calculation requires an IDF database. We constructed an IDF database from several corpora of reviews. We used used a stop word list provided by ().

[2] In practice the process would only be repeated in summaries long enough to contain sentences for each CF, which is very rare.

each CF. It may be that a very positive evaluation for one CF was selected for extraction, even though most evaluations were only somewhat positive (or very negative).

We thus also developed a system presents such information in generated natural language. This system calculates several important characteristics of the source text by aggregating the information extracted from Section 2. These characteristics are described in Section 4.1. The presentation of these characteristics in natural language is described in Section **??**.

4.1 Aggregation of Extracted Information

A good summary of evaluative text should communicate three key aspects of the source text to the user: (i) which features of the evaluated entity were most 'important' to the users (ii) some aggregate of the user opinions for important features (iii) the reasons behind each user opinion. We discuss the determination of (i) in Section 4.1.1 and (ii) in Section 4.1.2. Our generation-based system currently relies on links to the source text for communicating (iii); this is further discussed in Section 4.3.

4.1.1 Feature Selection

Without any prior knowledge of the reader of the summary or the entity evaluated by the evaluative text, we must rely on the statistics of the extracted features to determine which features are the most important. We thus approach the task of selecting the most 'important' features by defining a 'measure of importance' for each feature of the evaluated entity. Unlike our extraction-based system, which operated on a sentence level and thus utilized the CFs in each sentence, our generation-based system can exploit the users knowledge of the entity in the form of the UDF to operate on a conceptual level. We define the 'direct importance' of a feature in the UDF as

$$dir\_moi(udf_i) \;=\; \sum_{ps_k \epsilon PS_i} |ps_k|^2$$

where by 'direct' we mean the importance derived only from that feature and not from its children. The basic premise of this metric is that a feature's importance is proportional to the number of evaluations of that feature in the corpus. However, it seems reasonable that stronger evaluations should be given more weight in the measure of importance than weaker ones. That is, a single evaluation of a feature with a polarity/strength of ± 3 should contribute more to the importance of a feature than an evaluation of ± 1 or ± 2. The sum of squares used for $dir\_moi(udf_i)$ accomplishes both of these goals because it is increased by the number of evaluations, but weighted heavily towards stronger evaluations.

This 'direct' measure of importance, however, is incomplete, as each non-leaf node in the UDF effectively serves a dual purpose. It is both a feature upon which a user might comment and a category for grouping its sub-features. Thus, a non-leaf node should be important if either its children are important or the node itself is important (or both). To this end, we have defined the total measure of importance $moi(udf_i)$ as

$$\begin{cases} dir\_moi(udf_i) & ch(udf_i) \;=\; \emptyset \\ [\alpha \; dir\_moi(udf_i) \; + & \\ (1-\alpha) \; \sum_{udf_k \epsilon ch(udf_i)} moi(udf_k)] & \text{otherwise} \end{cases}$$

where $ch(udf_i)$ refers to the children of udf_i in the hierarchy and α is some real parameter in the range $[0.5, 1]$. In this measure, the importance of a node is a combination of its direct importance and of the importance of its children. The parameter α may be adjusted to vary the relative weight of the parent and children. Setting $\alpha = 0.5$ would weight a node equally with its children, while $\alpha = 1.0$ would ignore the importance of the children. We used $\alpha = 0.9$ for our experiments. This setting resulted in more informative summaries during system development.

In order to perform feature selection using this metric, we must also define a selection procedure. The most obvious is a simple greedy selection – sort the nodes in the UDF by the measure of importance and select the most important node until a desired number of features is included. However, because a node derives part of its 'importance' from its children, it is possible for a node's importance to be dominated by one or more of its children. Including both the child and parent node would be redundant because most of the information is contained in the child. We thus choose a dynamic greedy selection algorithm in which we recalculate the importance of each node after each round of selection, with all previously selected nodes removed from the tree. In this way, if a node that dominates its parent's importance is selected, its parent's importance will be reduced during later rounds of selection. This approach mimics the behaviour of several sentence extraction-based summarizers (e.g. (**?**; **?**)) which define a metric for sentence importance and then greedily select the sentence which minimizes similarity with already selected sentences and maximizes informativeness.

4.1.2 Opinion Aggregation

We approach the task of aggregating opinions from the source text in a similar fashion to our measure of importance. We calculate an 'orientation' for each UDF by aggregating the polarity/strength evaluations of all related CFs into a single value. We define the 'direct orientation' of a UDF as the average of the strength/polarity evaluations of all related CFs

$$dir\_ort(udf_i) \;=\; \underset{ps_k \epsilon PS_i}{avg} \; ps_k$$

As with our measure of importance, we must also include the orientation of a feature's children in its orientation. Because a feature in the UDF conceptually groups its children, the orientation of a feature should include some information about the orientation of its children. We thus define the total orientation $ort(udf_i)$ as

$$\begin{cases} dir\_ort(udf_i) & ch(udf_i) \ = \ \emptyset \\ [\alpha \; dir\_ort(udf_i) + & \\ (1-\alpha) \ \ avg_{udf_k \epsilon ch(udf_i)} \, ort(udf_k)] & \text{otherwise} \end{cases}$$

This metric produces a real number between -3 and $+3$ which serves as an aggregate of user opinions for a feature. The overall orientation of user opinions is deemed to be positive (+) if this number is positive and negative (-) if this number is negative. We use the same value of α as in $moi(udf_i)$.

4.1.3 Opinion Distribution

Communicating user opinions to the reader is not simply a matter of classifying each feature as being evaluated negatively or positively – the reader may also want to know if all users evaluated a feature in similar way or if evaluations were varied. We thus also need a method of determining the modality of the distribution of user opinions. For this purpose, we aggregate the polarity/strength evaluations for each UDF in a slightly different way. We tally all positive polarity/strength evaluations (or negative if $ort(udf_i)$ is negative) for a node and its children. Each evaluation is weighted according to its strength (i.e. a +2 evaluation counts as two 'votes'). We then determine the total 'vote' count as the sum of the absolute values of all evaluations for a node and its children. If the fraction of positive votes for a feature is lies within a certain threshold of 0.5 (i.e. a perfect split between negative and positive evaluations), then we classify the feature as 'bimodal', that is, we claim that the distribution of evaluations is varied enough that the variance should be reported to the reader. In this case, overall orientation of user opinions on that feature is deemed to be divided (+/-). Otherwise, the feature is classified as 'unimodal', i.e. we need only to communicate one aggregate opinion to the reader.

4.2 Adapting the Generator of Evaluative Arguments (GEA)

Having defined metrics for importance, orientation, and modality of user opinions, we may now proceed to the task of communicating these characteristics to reader. The automatic generation of a natural language summary involves the following additional tasks (**?**): (i) structuring the content by ordering and grouping the selected content elements as well as by specifying discourse relations (e.g., supporting vs. opposing evidence) between the resulting groups; (ii) microplanning, which involves lexical selection and sentence planning; and (iii) sentence realization, which produces English text from the output of the microplanner. For all these tasks, we have adapted the Generator of Evaluative Arguments (GEA) (**?**), a framework for generating user tailored evaluative arguments.

4.2.1 Content Structuring

GEA tailors evaluative arguments about a given entity to a quantitative model of the user preferences that is very similar to the UDF, as it is also describes the entity as a hierarchy of features. Our adaptation relies on this key similarity. In essence, GEA organizes and realizes the selected content as text by applying a strategy based on argumentation theory (**?**) that considers the strength and polarity of the user evaluation of each feature represented in the user model. Our summarizer applies the same strategy to organize and realize the selected content. However, instead of using the strength and polarity of a user evaluation of each feature, it uses the number of evaluations and an aggregate of the customers opinions about each feature respectively. This aggregate is a function similar in form to the measure of importance used for content selection.

4.2.2 Microplanning and Realization

Giuseppe: I'll do this in a few paragraphs.

4.3 Sample Sentences

Because the information extraction portion of our system identifies features at the sentence level, we can maintain a mapping for each feature in the UDF back to all sentences which evaluated it. This enables us to link evaluations of features in the summary to relevant data from the source text. Because we want our summary to convey the reasons for user evaluations to the reader, this mapping is important. We thus decided to provide 'sample sentences' for each evaluation in the summary. These sentences serve the dual purpose of confirming the evaluations presented in the summary and presenting the reader with additional information about user evaluations. Sample sentences are selected using techniques similar to the ones developed for MEAD*.

5 Evaluation

We evaluated our two summarizers by performing a user study in which users evaluated both systems. Some participants in this study also evaluated human-written summaries as a topline and summaries generated by MEAD as a baseline. Baseline MEAD summaries were generated with all options set to default.

5.1 The Experiment

Twenty-one undergraduate students recruited via an online user experiment system participated in our experiment. Each participant was given a set of 20 customer reviews randomly selected from a corpus of reviews. Half of the participants received reviews from a corpus of 46 reviews of the Canon G3 digital camera and half received them from a corpus of 101 reviews of the Apex 2600 Progressive Scan DVD player, both obtained from Hu and Liu (**?**). The reviews from these corpora which serve as input to our system have been manually annotated with crude features, strength, and polarity. We used a 'gold standard' for crude feature, strength, and polarity extraction because we wanted our experiments to focus on our summary and not be confounded by errors in the knowledge extraction phase.

The participant was told to pretend that they work for the manufacturer of the product (either Canon or Apex). They were told that they would have to provide a 100 word summary of the reviews to the quality assurance department. The purpose of these instructions was to prime the user to the task of looking for information worthy of summarization. They were then given 20 minutes to explore the set of reviews. The participant could access the reviews through a hypertext interface in which any review could be accessed by clicking on its title. During this time, the participant was allowed to take notes on paper or in a text editor on the computer.

After 20 minutes, the participant was asked to stop. The participant was then given a set of instructions which explained that the company was testing a computer-based system for automatically generating a summary of the reviews s/he has been reading. S/he was then shown the summary of the 20 reviews generated either by MEAD, MEAD*, SEA, or written by a human. The summaries were displayed in a web browser. The upper portion of the browser contained the text of the summary with 'footnotes' linking to sample sentences for each evaluation. Clicking on one of the footnotes caused a sample sentence to be shown in the bottom of the screen. Sample sentences were shown along with the entire review from which they were extracted. The sample sentence itself was highlighted. Once finished, the participant was asked to fill out a questionnaire assessing the summary along several dimensions related to its effectiveness. The participant could still access the summary while s/he worked on the questionnaire.

Our questionnaire consisted of nine questions, the exact working of which may be found in Appendix A. The first five questions were the SEE lingustic well-formedness questions used at the 2005 Document Understanding Conference (**?**). The next three questions were designed to assess the content of the summary. We based our questions on the Responsive evaluation at DUC 2005; however, we were interested in a more specific evaluation of the content that one overall rank. As such, we split the content into three separate questions. The final question in the questionnaire asked the participant to rank the overall quality of the summary holistically.

5.2 Quantitative Results

The quantitative results are summarized in Table 5.2. One participant responded to Questions 1 and 2 with free text comments which did not match his numerical answer. In each case, "Agree" was swapped for "Disagree" and vice versa.

At first glance, it appears from the results that MEAD* and SEA performed at a roughly an equal level, while the baseline summaries performed significantly lower and the human summaries performed significantly higher. Indeed, this is about the only statistically significant effect we can extract from our study (ANOVA analysis with Bonferroni adjusted t-test all p¡ ????). XXX Raymond do better stats analysis.

This calculation was performed on the macro-average of all numerical scores for each summary type. We also wanted to investigate any possible differences in linguistic well-formedness (questions 1-5) and summary content (questions 6-8). We performed a two-way ANOVA test with summary type as rows and the question sets as columns. We found no significant main effects or interaction effects

Since we used the same linguistic quality questions in our experiment as those used at the DUC 2005 () automatic summarization conference, we can also roughly compare our summarizers to the state of the art in multi-document summarization. In Table **??**, we see that in terms of overall linguistic quality, our summarizers perform on par with the median linguistic quality of summarizers at DUC.

Another popular metric for summary evaluation is the ROUGE metric (). This is an automatic evaluation system which compares test summaries to reference summaries. We also evaluated both MEAD*- and SEA-based summaries with ROUGE, using our human summaries as reference summaries. However, the results are inconclusive: MEAD* outscores SEA on the ROUGE-1 metric (unigram matching), while SEA outsocres MEAD* on ROUGE-2 (bigram matching) and, all scores for MEAD* are well within the 95% confidence interval for the corresponding SEA scores.

5.3 Qualitative Results

*MEAD**: The most interesting aspect of the comments made by participants who evaluated MEAD*-based summaries was that they rarely criticized the summary for being nothing more than a set of extracted sentences. In fact, some users seemed to see structure in the summaries which where not intentionally present. For example, one user claimed that the summary had a "*simple sentence first, then ideas are fleshed out, and ends with a fun impact statement*" and also liked the "*fun quotes like 'two thumbs up!'* ". Other users, while noticing that the summary was solely quotation, still felt the summary was adequate ("*Shouldn't just copy consumers . . . However, it summarized various aspects of the cuonsmer's opinions . . .* ").

Otherwise, most comments about the summarizer centered either on its structure or on its content. Regarding the structure, participants complained that the summary had no logical structure, jumped awkwardly

	SEA		**MEAD***		MEAD		Human		DUC		
Question	Avg.	Dev.	Avg.	Dev.	Avg.	Dev.	Avg.	Dev.	Med.	Min.	Max.
Grammaticality	3.43	1.13	2.71	0.76	3.14	0.90	4.29	0.76	3.86	2.60	4.34
Non-redundancy	3.14	1.57	3.86	0.90	3.57	0.98	4.43	1.13	4.44	3.96	4.74
Referential clarity	3.86	0.69	4.00	1.15	3.00	1.15	4.71	0.49	2.98	2.16	4.14
Focus	4.14	0.69	3.71	1.60	2.29	1.60	4.14	0.69	3.16	2.38	3.94
Structure and Coherence	2.29	0.95	3.00	1.41	1.86	0.90	4.43	0.53	2.10	1.60	3.24
Linguistic Average	*3.37*	*1.19*	*3.46*	*1.24*	*2.77*	*1.24*	*4.4*	*0.74*	*3.31*	*2.54*	*4.08*
Recall	2.33	1.03	2.57	0.98	1.57	0.53	3.57	1.27	–	–	–
Precision	4.17	1.17	3.50	1.38	2.17	1.17	3.86	1.07	–	–	–
Accuracy	4.00	0.82	3.57	1.13	2.57	1.4	4.29	0.76	–	–	–
Content Average	*3.5*	*1.26*	*3.21*	*1.2*	*2.1*	*1.12*	*3.9*	*1.04*	–	–	–
Overall	3.14	0.69	3.14	1.21	2.14	1.21	4.43	0.79	–	–	–
Macro Average	*3.39*	*0.73*	*3.34*	*0.51*	*2.48*	*0.65*	*4.24*	*0.34*	–	–	–

Table 1: Quantative results of user responses to our questionnaire on a scale from 1 (Strongly Disagree) to 5 (Strongly Agree). See Appendix A for the exact wording of the questions.

between positive and negative evaluations, and that the sentences did not “flow together”. There is little that can be done about the structure of the MEAD*-based summaries. The problems raised by participants are largely problems intrinsic to extraction-based summaries. While systems exist which attempt to create more coherent summaries through [sentence compression, anaphora resoultion, etc.] (), we do not think these methods will work for our application. Generally, multidocument summarizers are targeted for factual information (e.g. news report () Newsblaster). Factual information tends to be written in the third person and contain a (hopefully) consistent set of facts (maybe mention RIPTIDES, which takes into account inconsistent facts). Evaluative text, on the other hand, may or may not be written in the first person and generally contains varying perspectives on the same issue. Creating a logically coherent summary by extracting sentences with inconsistent facts and even deictic [right word?] point of view is a very non-trivial task.

With regard to the content, participants claimed that (i) there were conflicting evaluations of the same feature, (ii) features like the physical appearance should be left out, while problems with customer service (in the case of the DVD player) and praise of the excellent price (in the case of the digital camera) should have been included, (iii) the summary did not reflect overall opinions (it included positive evaluations of the DVD player even though most evaluatiosn were negative), and (iv) evaluations of some features were repeated. Since comments along the lines of (ii) were common for both MEAD* and SEA summaries, we discuss them below along with SEA. Of the remaining comments, many an be addressed by modifications to our system. In particular, one aspect with our sentence selection procedure that led to many of the above problems is that we do not check which CF evaluations are ‘tagging along’ with a given sentence. For example, a sentence like “*Great colors , pictures and white balance.*” may be chosen because it evaluates the image quality, but it also contains evaluations of the white balance and colours in the image. With this in mind, we could address (i) by ensuring that the each evaluation of a CF in a candidate sentence matches the polarity of all evaluations of the same CF already present in the summary. We could address (iii) by only including sentences whose CF evaluations having polarities matching the majority polarity for each CF. Finally, (iv) could be avoided by not selecting sentences which contain evaluations of CFs already in the summary. However, these solution could only go so far: it may be that a ‘bucket’ for a given CF simply only contains sentences whose CF evaluations are conflicting, repetitive, or non-representative. In other words, it may be that no subset of sentences from the corpus satisfy all these constraints while still evaluating the most important features.

SEA: Comments about the structure of the summaries generated by SEA mentioned the “coherent but robotic” feel of the summaries, the repetition of “users/customers” and lack of pronoun use, the lack of flow between sentences, and the repeated use of generic terms such as “good”. These are largely issues of microplanning, or rather, lack thereof. The original GEA system included not only pronomialization, but also used the FUF/SURGE () system along with a set of adjectives tailored to specific features (e.g. “The location is convenient” vs. “The location is good”) to enrich the linguistic content of the summary. Such complex microplanning is a time-consuming process which we did not carry out. Furthermore, providing feature-specific lexicalizations creates a high degree of domain-dependence in the summarizer, something which we wish to avoid.

In terms of content, there were two main sets of complaints. Firstly, participants wanted more “details” in the summary, for instance, they wanted examples of the “manual features” mentioned by SEA. Note that this is one complaint absent from the MEAD* summaries. That is, where the MEAD*-based summaries lack structure but contain detail, SEA summaries provide a general, structured overview while lacking in specifics. As further proof of this conclusion, in a separate user study (cite ??? what about blind review), we evaluated an interactive multimedia summarizer which combined textual summaries generated by SEA with a visual method for exploring the data in the original corpus. In this study, the textual summaries (evaluated with a question similar to our Overall question) scored higher (average of 3.71 versus 3.14) than in the present study. Additionally one participant explicitly stated the generic nature of the SEA summary: "*I can get a general idea from the text and detailed information from the graphics.*" We thus believe that a better summarizer can be created by combining the strengths of sentence extraction and our generated-based approach. Not only would this provide more detail, it would also provide more domain-specific vocabulary in the summary, thus addression the microplanning problems mentioned in the previous paragraph. We discuss this further along with our future work.

The other set of complaints related to the problem already mentioned in the discussion of MEAD*, namely, that customers disagreed with the choice of features in the summary. We note first that this was the one area in which even *human* summaries do not score very well – human summaries averaged only 3.57 on Recall and 3.86 on Precision, i.e. below the level of 'Agree'. As such, we cannot expect the summary to perform flawlessly in this regard. The difficulty of content selection is best highlighted with the example of the "physical appearance" of the digital camera in some of the MEAD* and SEA summaries. One reason participants may have disagreed with the summarizer's decision to include the physical appearance in the summary may have been that some evaluations of the physical appearance were quite subtle. For example, the sentence "*This camera has a design flaw*" was annotated in our corpus as evaluating the physical appearance, although not all readers would agree with that annotation. A human may disregard this evaluation as unimportant, but our summarizer has no way of distinguishing it from other evaluations of the physical appearance (except by strength). It is also worth noting that in one case, a participant judged the physical appearance to be unimportant based on an incorrect assertion about the set of reviews. He stated that "*I wouldn't have mentioned the physical appearance because that came up in only one review*", although the appearance was in fact mentioned in two separate reviews. Interestingly, the same participant also wanted information about the size of the memory cards of the digital camera included in the summary, even though such comments were present in only three reviews. It is probable that the preference for not including evaluations of the physical appearance is not solely objective: participants may intrinsically assign less importance to physical appearance than to other features, something which our summarizer does not yet attempt to model. We also discuss this further in our future work.

However, both SEA and MEAD* summaries still scored substantially lower than human summaries, and so there is clearly room for improvement. For example, one common complaint about summaries of the DVD player reviews was that discussion of Apex's terrible customer service was not discussed. Interestingly, the customer service feature was omitted from both SEA and MEAD* summaries, but for entirely different reasons. In the case of SEA, the omission was due to a faulty UDF. The $UDFs$ were not designed for the purposes of summarization [XXX Giuseppe: note about where they're from], and as such, features such as "price" and "customer service", though present in the set of CFs, were not present in the UDF. In our final system, we envision a process in which the user creates and refines the UDF and associated mapping to his/her own needs. Thus, the above errors could be fixed by users of the system.

In the case of MEAD*, no sentence about the customer service was extracted simply because other CF 'buckets' were larger. Interestingly, in the case of one particular set of 20 reviews, we determined that MEAD* would have included a sentence about the customer service as the 6th sentence, but the summary was cut off at 5 due to length constrictions. The 5 sentences that were included pertained to the CFs "player", "play", "DVD", "format", "DVD player". These CFs are clearly quite redundant. Had we applied the same domain knowledge from the UDF present in SEA, this redundancy could have been eliminated, and a sentence about the customer service would have been included.

6 Related work

Opinion extraction from evaluative text has received a good deal of attention recently (cite OPINE, others). However, we have only been able to find two systems in the literature which purport to summarize the extracted opinions. Hu and Liu (2005) summarize their extracted features by displaying each feature along with a two counts: the number of sentences evaluating the feature positively, and the number of sentences evaluating the feature negatively. In each case, the user may also view all sentences which evaluative a feature positively or negatively. They display the features in order of longest (in terms of word count) first; sentences are displayed in no particular order. This summarization approach has several weaknesses with respect to our system:. Firstly, it displays all the extracted information without selecting (or ordering) features by any measure of importance. Secondly, it does not group the evaluated features in any way. For example, it may be that evaluations for "lens cap" and "lens" are displayed very far from each other in the summary. This makes it difficult for the user to discern groupings in the evaluations. Because SEA groups features using the UDF, our generated summaries do not have this problem. Finally, this approach does not attempt to communicate any relationship among the features. The user must ascertain for herself which evaluations are general (e.g. “image”) and which are more specific (e.g. “resolution”),and furthermore, how general evaluations related to specific ones (e.g. “users liked the images even though they disliked the resolution”).

A second system which attempts to summarize evaluative text (cite Manning) attempts to extract a single ‘sentiment sentence’ for a single movie review. They treat the problem as a sentence classification problem. To solve the problem, they use naive Bayesian classifier trained on a set of movie reviews which associated sentiment sentences taken from Rottentomatoes.com This task differs from ours in that (i) our system summarizes multiple documents as opposed to a single review; (ii) our system selects more than one sentence for each review. In terms of implementation, their system is a supervised machine-learning system. In contrast, our system is entirely unsupervised and only mildly domain-dependent because of the entity-specific lexicalizations XXX mention earlier and UDF.

Additionally, we have found no other work comparing sentence extraction- and generation-based approaches. In fact, very few summarization systems perform natural language generation. One notable exception is RIPTIDES (cite) – a system designed for a task which is somewhat similar to ours. RIPTIDES is designed for summarizing multiple (possibly conflicting) reports on disasters such earthquakes and terrorist attacks. It extracts information via semi-supervised pattern matching. The information is then fitted into ‘slots’ of a domain-specific schema for each event. The sytem produces a summary consisting of sentences generated by filling in sentence templates with information from the event schema, as well as sentences extracted from the source reports. The event schema performs a similar task to the UDF in our system, and RIPTIDES’s approach to reporting conflicting reports on factual information may also be applicable to reporting on conflicting opinions. However, this system is not directly applicable to our task of summarizaing evaluative text because it does not possess the ability to extract opinions from the text, only factual information. Furthermore, the information extraction process is supervised, making it more labour-intensive to adapt this system to a given domain.

7 Conclusions and Future Work

We have presented and compared a sentence extraction- and language generation based approach to summarizing evaluative text. A formative user study of our summarizers found that, quantitatively, they performed equally well relative to each other, while significantly outperforming a baseline standard approach to multidocument summarization. Qualitatively, comments from participants in the user study indicate that the summarizers have different strengths and weaknesses. On the one hand, though providing varied language and detail about customer opinions, our extraction-based summaries lack a coherent structure and fail to give and overview of the opinions expressed in the evaluative text. On the other, our language generation-based summarizies provide a general, coherent overview of the source text, while sounding ‘robotic’, repetitive, and vague.

These differences are, fortunately, quite complimentary. We plan in the future to utilize the overall structure present in the generation-based summaries to provide structure to language extracted directly from the source text. This approach would reduce the tendency of the the summaries to sound vague and robotic. In order to accomplish this, we will need to extend beyond whole sentence extraction and more intelligently select isolated phrases from the source text. One recent approach to feature extraction from evaluative text (**?**) does exactly that. Furthermore, several multidocument summarizers perform sentence compression on extracted sentences (). Such methods could also be helpful in extracting detail succinctly directly from the source text.

There are two other areas on which we plan to work in the future. We believe that summaries could be made even more useful by tailoring the data presented in the summary to a preference model of the user, as was done in GEA. Additionally, we hope to combine the evaluative information present in customer reviews with factual information (e.g. product specifications) about the entity of interest.

The other area in which we hope to work is automatic induction of the $UDFs$ from the CFs. There has been significant work recently on automatically inducing ontologies (e.g. cite ()). This would permit the entire summarization process, from feature extraction to summary output, to be entirely operated. While user revision would still be essential to create useful summaries, no human input would be required to ‘bootstrap’ the system.

Appendix A: Questionnaire used in the the Formative User Study

The questionnaire presented to the participants included the following statements. Participants indicated their degree of agreement on the standard 5 point Likert scale (one of ”Strongly Disagree”, ”Disagree”, ”Neutral”, ”Agree”, ”Strongly Agree” or ”No Opinion”). The participant was provided with space for free text comments after each question.

1. *The summary has no datelines, system-internal formatting, capitalization errors or obviously ungrammatical sentences (e.g., fragments, missing components) that make the text difficult to read.* (Formatting)

2. *There is no unnecessary repetition in the summary. Unnecessary repetition might take the form of whole sentences that are repeated, or repeated facts, or the repeated use of a noun or noun phrase (e.g., ”Bill Clin-*

ton") when a pronoun ("he") would suffice. (Non-repetition)

3. *It is easy to identify who or what the pronouns and noun phrases in the summary are referring to. If a person or other entity is mentioned, it is clear what their role in the story is. So, a reference would be unclear if an entity is referenced but its identity or relation to the story remains unclear.* (Referential clarity)

4. *The summary has a focus; sentences only contain information that is related to the rest of the summary.* (Focus)

5. *The summary is well-structured and well-organized. The summary is not just a heap of related information, but builds from sentence to sentence to a coherent body of information about the product reviews.* (Structure and Coherence)

6. *The summary contains all of the information you would have included from the source text.* (Recall)

7. *The summary contains no information you would NOT have included from the source text.* (Precision)

8. *All information expressed in the summary accurately reflects the information contained in the source text.* (Accuracy)

9. *Overall, the summary was an ideal summary of the source text.* (Overall)