Sentiment Summarization: Evaluating and Learning User Preferences

Kevin Lerman
Columbia University
New York, NY
klerman@cs.columbia.edu

Sasha Blair-Goldensohn
Google, Inc.
New York, NY
sasha@google.com

Ryan McDonald
Google, Inc.
New York, NY
ryanmcd@google.com

Abstract

We present the results of a large-scale, end-to-end human evaluation of various sentiment summarization models. The evaluation shows that users have a strong preference for summarizers that model sentiment over non-sentiment baselines, but have no broad overall preference between any of the sentiment-based models. However, an analysis of the human judgments suggests that there are identifiable situations where one summarizer is generally preferred over the others. We exploit this fact to build a new summarizer by training a ranking SVM model over the set of human preference judgments that were collected during the evaluation, which results in a 30% relative reduction in error over the previous best summarizer.

1 Introduction

The growth of the Internet as a commerce medium, and particularly the Web 2.0 phenomenon of user-generated content, have resulted in the proliferation of massive numbers of product, service and merchant reviews. While this means that users have plenty of information on which to base their purchasing decisions, in practice this is often too much information for a user to absorb. To alleviate this information overload, research on systems that automatically aggregate and summarize opinions have been gaining interest (Hu and Liu, 2004a; Hu and Liu, 2004b; Gamon et al., 2005; Popescu and Etzioni, 2005; Carenini et al., 2005; Carenini et al., 2006; Zhuang et al., 2006; Blair-Goldensohn et al., 2008).

Evaluating these systems has been a challenge, however, due to the number of human judgments required to draw meaningful conclusions. Often systems are evaluated piecemeal, selecting pieces that can be evaluated easily and automatically (Blair-Goldensohn et al., 2008). While this technique produces meaningful evaluations of the selected components, other components remain untested, and the overall effectiveness of the entire system as a whole remains unknown. When systems are evaluated end-to-end by human judges, the studies are often small, consisting of only a handful of judges and data points (Carenini et al., 2006). Furthermore, automated summarization metrics like ROUGE (Lin and Hovy, 2003) are non-trivial to adapt to this domain as they require human curated outputs.

We present the results of a large-scale, end-to-end human evaluation of three sentiment summarization models applied to user reviews of consumer products. The evaluation shows that there is no significant difference in rater preference between any of the sentiment summarizers, but that raters do prefer sentiment summarizers over non-sentiment baselines. This indicates that even simple sentiment summarizers provide users utility. An analysis of the rater judgments also indicates that there are identifiable situations where one sentiment summarizer is generally preferred over the others. We attempt to learn these preferences by training a ranking SVM that exploits the set of preference judgments collected during the evaluation. Experiments show that the ranking SVM summarizer's cross-validation error decreases by as much as 30% over the previous best model.

Human evaluations of text summarization have been undertaken in the past. McKeown et al. (2005) presented a task-driven evaluation in the news domain in order to understand the utility of different systems. Also in the news domain, the Document Understanding Conference[1] has run a number of multi-document and query-driven summarization shared-tasks that have used a wide

[1] http://duc.nist.gov/

Proceedings of the 12th Conference of the European Chapter of the ACL, pages 514–522,
Athens, Greece, 30 March – 3 April 2009.

iPod Shuffle: 4/5 stars

"In final analysis the iPod Shuffle is a decent player that offers a sleek compact form factor an excessively simple user interface and a low price" ... "It's not good for carrying a lot of music but for a little bit of music you can quickly grab and go with this nice little toy" ... "Mine came in a nice bright orange color that makes it easy to locate."

Figure 1: An example summary.

range of automatic and human-based evaluation criteria. This year, the new Text Analysis Conference[2] is running a shared-task that contains an opinion component. The goal of that evaluation is to summarize answers to opinion questions about entities mentioned in blogs.

Our work most closely resembles the evaluations in Carenini et al. (2006, 2008). Carenini et al. (2006) had raters evaluate extractive and abstractive summarization systems. Mirroring our results, they show that both extractive and abstractive summarization outperform a baseline, but that overall, humans have no preference between the two. Again mirroring our results, their analysis indicates that even though there is no overall difference, there are situations where one system generally outperforms the other. In particular, Carenini and Cheung (2008) show that an entity's *controversiality*, e.g., mid-range star rating, is correlated with which summary has highest value.

The study presented here differs from Carenini et al. in many respects: First, our evaluation is over different extractive summarization systems in an attempt to understand what model properties are correlated with human preference irrespective of presentation; Secondly, our evaluation is on a larger scale including hundreds of judgments by hundreds of raters; Finally, we take a major next step and show that it is possible to automatically learn significantly improved models by leveraging data collected in a large-scale evaluation.

2 Sentiment Summarization

A standard setting for sentiment summarization assumes a set of documents $D = \{d_1, \ldots, d_m\}$ that contain opinions about some entity of interest. The goal of the system is to generate a summary S of that entity that is representative of the average opinion and speaks to its important aspects. An example summary is given in figure 1. For simplicity we assume that all opinions in D are about the entity being summarized. When this assumption fails, one can parse opinions at a finer-level (Jindal and Liu, 2006; Stoyanov and Cardie, 2008)

In this study, we look at an extractive summarization setting where S is built by extracting representative bits of text from the set D, subject to pre-specified length constraints. Specifically, assume each document d_i is segmented into candidate text excerpts. For ease of discussion we will assume all excerpts are sentences, but in practice they can be phrases or multi-sentence groups. Viewed this way, D is a set of candidate sentences for our summary, $D = \{s_1, \ldots, s_n\}$, and summarization becomes the following optimization:

$$\arg\max_{S \subseteq D} \mathcal{L}(S) \quad \text{s.t.: } \textsc{length}(S) \leq K \qquad (1)$$

where $\mathcal{L}$ is some score over possible summaries, $\textsc{length}(S)$ is the length of the summary and K is the pre-specified length constraint. The definition of $\mathcal{L}$ will be the subject of much of this section and it is precisely different forms of $\mathcal{L}$ that will be compared in our evaluation. The nature of LENGTH is specific to the particular use case.

Solving equation 1 is typically NP-hard, even under relatively strong independence assumptions between the sentences selected for the summary (McDonald, 2007). In cases where solving $\mathcal{L}$ is non-trivial we use an approximate hill climbing technique. First we randomly initialize the summary S to length $\sim K$. Then we greedily insert/delete/swap sentences in and out of the summary to maximize $\mathcal{L}(S)$ while maintaining the bound on length. We run this procedure until no operation leads to a higher scoring summary. In all our experiments convergence was quick, even when employing random restarts.

Alternate formulations of sentiment summarization are possible, including aspect-based summarization (Hu and Liu, 2004a), abstractive summarization (Carenini et al., 2006) or related tasks such as opinion attribution (Choi et al., 2005). We choose a purely extractive formulation as it makes it easier to develop baselines and allows raters to compare summaries with a simple, consistent presentation format.

2.1 Definitions

Before delving into the details of the summarization models we must first define some useful functions. The first is the sentiment polarity function that maps a lexical item t, e.g., word or short phrase, to a real-valued score,

$$\textsc{lex-sent}(t) \in [-1, 1]$$

[2] http://www.nist.gov/tac/

The LEX-SENT function maps items with positive polarity to higher values and items with negative polarity to lower values. To build this function we constructed large sentiment lexicons by seeding a semantic word graph induced from WordNet with positive and negative examples and then propagating this score out across the graph with a decaying confidence. This method is common among sentiment analysis systems (Hu and Liu, 2004a; Kim and Hovy, 2004; Blair-Goldensohn et al., 2008). In particular, we use the lexicons that were created and evaluated by Blair-Goldensohn et al. (2008).

Next we define sentiment intensity,

$$\text{INTENSITY}(s) = \sum_{t \in s} |\text{LEX-SENT}(t)|$$

which simply measures the magnitude of sentiment in a sentence. INTENSITY can be viewed as a measure of subjectiveness irrespective of polarity.

A central function in all our systems is a sentences normalized sentiment,

$$\text{SENT}(s) = \frac{\sum_{t \in s} \text{LEX-SENT}(t)}{\alpha + \text{INTENSITY}(s)}$$

This function measures the (signed) ratio of lexical sentiment to intensity in a sentence. Sentences that only contain lexical items of the same polarity will have high absolute normalized sentiment, whereas sentences with mixed polarity items or no polarity items will have a normalized sentiment near zero. We include the constant α in the denominator so that SENT gives higher absolute scores to sentences containing many strong sentiment items of the same polarity over sentences with a small number of weak items of the same polarity.

Most sentiment summarizers assume that as input, a system is given an overall rating of the entity it is attempting to summarize, $R \in [-1, 1]$, where a higher rating indicates a more favorable opinion. This rating may be obtained directly from user provided information (e.g., star ratings) or automatically derived by averaging the SENT function over all sentences in D. Using R, we can define a mismatch function between the sentiment of a summary and the known sentiment of the entity,

$$\text{MISMATCH}(S) = (R - \frac{1}{|S|} \sum_{s_i \in S} \text{SENT}(s_i))^2$$

Summaries with a higher mismatch are those whose sentiment disagrees most with R.

Another key input many sentiment summarizers assume is a list of salient entity aspects, which are specific properties of an entity that people tend to rate when expressing their opinion. For example, aspects of a digital camera could include picture quality, battery life, size, color, value, etc. Finding such aspects is a challenging research problem that has been addressed in a number of ways (Hu and Liu, 2004b; Gamon et al., 2005; Carenini et al., 2005; Zhuang et al., 2006; Branavan et al., 2008; Blair-Goldensohn et al., 2008; Titov and McDonald, 2008b; Titov and McDonald, 2008a). We denote the set of aspects for an entity as A and each aspect as $a \in A$. Furthermore, we assume that given A it is possible to determine whether some sentence $s \in D$ mentions an aspect in A. For our experiments we use a hybrid supervised-unsupervised method for finding aspects as described and evaluated in Blair-Goldensohn et al. (2008).

Having defined what an aspect is, we next define a summary diversity function over aspects,

$$\text{DIVERSITY}(S) = \sum_{a \in A} \text{COVERAGE}(a)$$

where $\text{COVERAGE}(a) \in \mathbb{R}$ is a function that weights how well the aspect is covered in the summary and is proportional to the importance of the aspect as some aspects are more important to cover than others, e.g., "picture quality" versus "strap" for digital cameras. The diversity function rewards summaries that cover many important aspects and plays the redundancy reducing role that is common in most extractive summarization frameworks (Goldstein et al., 2000).

2.2 Systems

For our evaluation we developed three extractive sentiment summarization systems. Each system models increasingly complex objectives.

2.2.1 Sentiment Match (SM)

The first system that we look at attempts to extract sentences so that the average sentiment of the summary is as close as possible to the entity level sentiment R, which was previously defined in section 2.1. In this case $\mathcal{L}$ can be simply defined as,

$$\mathcal{L}(S) = -\text{MISMATCH}(S)$$

Thus, the model prefers summaries with average sentiment as close as possible to the average sentiment across all the reviews.

There is an obvious problem with this model. For entities that have a mediocre rating, i.e., $R \approx 0$, the model could prefer a summary that only contains sentences with no opinion whatsoever. There are two ways to alleviate this problem. The first is to include the INTENSITY function into $\mathcal{L}$,

$$\mathcal{L}(S) = \alpha \cdot \text{INTENSITY}(S) - \beta \cdot \text{MISMATCH}(S)$$

Where the coefficients allow one to trade-off sentiment intensity versus sentiment mismatch.

The second method, and the one we chose based on initial experiments, was to address the problem at inference time. This is done by prohibiting the algorithm from including a given positive or negative sentence in the summary if another more positive/negative sentence is not included. Thus the summary is forced to consist of only the most positive and most negative sentences, the exact mix being dependent upon the overall star rating.

2.2.2 Sentiment Match + Aspect Coverage (SMAC)

The SM model extracts sentences for the summary without regard to the content of each sentence relative to the others in the summary. This is in contrast to standard summarization models that look to promote sentence diversity in order to cover as many important topics as possible (Goldstein et al., 2000). The sentiment match + aspect coverage system (SMAC) attempts to model diversity by building a summary that trades-off maximally covering important aspects with matching the overall sentiment of the entity. The model does this through the following linear score,

$$\mathcal{L}(S) = \alpha \cdot \text{INTENSITY}(S) - \beta \cdot \text{MISMATCH}(S) + \gamma \cdot \text{DIVERSITY}(S)$$

This score function rewards summaries for being highly subjective (INTENSITY), reflecting the overall product rating (MISMATCH), and covering a variety of product aspects (DIVERSITY). The coefficients were set by inspection.

This system has its roots in *event-based summarization* (Filatova and Hatzivassiloglou, 2004) for the news domain. In that work an optimization problem was developed that attempted to maximize summary informativeness while covering as many (weighted) sub-events as possible.

2.2.3 Sentiment-Aspect Match (SAM)

Because the SMAC model only utilizes an entity's overall sentiment when calculating MISMATCH, it is susceptible to degenerate solutions. Consider a product with aspects A and B, where reviewers overwhelmingly like A and dislike B, resulting in an overall SENT close to zero. If the SMAC model finds a very negative sentence describing A and a very positive sentence describing B, it will assign that summary a high score, as the summary has high intensity, has little overall mismatch, and covers both aspects. However, in actuality, the summary is entirely misleading.

To address this issue, we constructed the sentiment-aspect match model (SAM), which not only attempts to cover important aspects, but cover them with appropriate sentiment. There are many ways one might design a model to do this, including linear combinations of functions similar to the SMAC model. However, we decided to employ a probabilistic approach as it provided performance benefits based on development data experiments. Under the SAM model, each sentence is treated as a bag of aspects and their corresponding mentions' sentiments. For a given sentence s, we define A_s as the set of aspects mentioned within it. For a given aspect $a \in A_s$, we denote $\text{SENT}(a_s)$ as the sentiment associated with the textual mention of a in s. The probability of a sentence is defined as,

$$p(s) = p(a^1, \ldots, a^n, \text{SENT}(a_s^1), \ldots, \text{SENT}(a_s^n))$$

which can be re-written as,

$$\prod_{a \in A_s} p(a, \text{SENT}(a_s)) = \prod_{a \in A_s} p(a) p(\text{SENT}(a_s)|a)$$

if we assume aspect mentions are generated independently of one another. Thus we need to estimate both $p(a)$ and $p(\text{SENT}(a_s)|a)$. The probability of seeing an aspect, $p(a)$, is simply set to the maximum likelihood estimates over the data set D. Furthermore, we assume that $p(\text{SENT}(a_s)|a)$ is normal about the mean sentiment for the aspect μ_a with a constant standard deviation, σ_a. The mean and standard deviation are estimated straight-forwardly using the data set D. Note that the number of parameters our system must estimate is very small. For every possible aspect $a \in A$ we need three values: $p(a)$, μ_a, and σ_a. Since $|A|$ is typically small – on the order of 5-10 – it is not difficult to estimate these models even from small sets of data.

Having constructed this model, one logical approach to summarization would be to select sentences for the summary that have highest probability under the model trained on D. We found,

however, that this produced very redundant summaries – if one aspect is particularly prevalent in a product's reviews, this approach will select all sentences about that aspect, and discuss nothing else. To combat this we developed a technique that scores the summary as a whole, rather than by individual components. First, denote $\text{SAM}(D)$ as the previously described model learned over the set of entity documents D. Next, denote $\text{SAM}(S)$ as an identical model, but learned over a candidate summary S, i.e., given a summary S, compute $p(a)$, m_a, and σ_a for all $a \in A$ using only the sentences from S. We can then measure the difference between these models using KL-divergence:

$$\mathcal{L}(S) = -\text{KL}(\text{SAM}(D), \text{SAM}(S))$$

In our case we have $1 + |A|$ distributions – $p(a)$, and $p(\cdot|a)$ for all $a \in A$ – so we just sum the KL-divergence of each. The key property of the SAM system is that it naturally builds summaries where important aspects are discussed with appropriate sentiment, since it is precisely these aspects that will contribute the most to the KL-divergence. It is important to note that the short length of a candidate summary S can make estimates in $\text{SAM}(S)$ rather crude. But we only care about finding the "best" of a set of crude models, not about finding one that is "good" in absolute terms. Between the few parameters we must learn and the specific way we use these models, we generally get models useful for our purposes.

Alternatively we could have simply incorporated the DIVERSITY measure into the objective function or used an inference algorithm that specifically accounts for redundancy, e.g., maximal marginal relevance (Goldstein et al., 2000). However, we found that this solution was well grounded and required no tuning of coefficients.

Initial experiments indicated that the SAM system, as described above, frequently returned sentences with low intensity when important aspects had luke-warm sentiment. To combat this we removed low intensity sentences from consideration, which had the effect of encouraging important luke-warm aspects to mentioned multiple times in order to balance the overall sentiment.

Though the particulars of this model are unique, fundamentally it is closest to the work of Hu and Liu (2004a) and Carenini et al. (2006).

3 Experiments

We evaluated summary performance for reviews of consumer electronics. In this setting an entity to be summarized is one particular product, D is a set of user reviews about that product, and R is the normalized aggregate star ratings left by users. We gathered reviews for 165 electronics products from several online review aggregators. The products covered a variety of electronics, such as MP3 players, digital cameras, printers, wireless routers, and video game systems. Each product had a minimum of four reviews and up to a maximum of nearly 3000. The mean number of reviews per product was 148, and the median was 70. We ran each of our algorithms over the review corpus and generated summaries for each product with $K = 650$. All summaries were roughly equal length to avoid length-based rater bias[3]. In total we ran four experiments for a combined number of 1980 rater judgments (plus additional judgments during the development phase of this study).

Our initial set of experiments were over the three opinion-based summarization systems: SM, SMAC, and SAM. We ran three experiments comparing SMAC to SM, SAM to SM, and SAM to SMAC. In each experiment two summaries of the same product were placed side-by-side in a random order. Raters were also shown an overall rating, R, for each product (these ratings are often provided in a form such as "3.5 of 5 stars"). The two summaries on either side were shown below this information with links to the full text of the reviews for the raters to explore.

Raters were asked to express their preference for one summary over the other. For two summaries S_A and S_B they could answer,

1. No preference
2. Strongly preferred S_A (or S_B)
3. Preferred S_A (or S_B)
4. Slightly preferred S_A (or S_B)

Raters were free to choose any rating, but were specifically instructed that their rating should account for a summaries representativeness of the overall set of reviews. Raters were also asked to provide a brief comment justifying their rating. Over 100 raters participated in each study, and each comparison was evaluated by three raters with no rater making more than five judgments.

[3] In particular our systems each extracted four text excerpts of roughly 160-165 characters.

Comparison (A v B)	Agreement (%)	No Preference (%)	Preferred A (%)	Preferred B (%)	Mean Numeric
SM v SMAC	65.4	6.0	52.0	42.0	0.01
SAM v SM	69.3	16.8	46.0	37.2	0.01
SAM v SMAC†	73.9	11.5	51.6	36.9	0.08
SMAC v LT†	64.1	4.1	70.4	25.5	0.24

Table 1: Results of side-by-side experiments. *Agreement* is the percentage of items for which all raters agreed on a positive/negative/no-preference rating. *No Preference* is the percentage of agreement items in which the raters had no preference. *Preferred A/B* is the percentage of agreement items in which the raters preferred either A or B respectively. *Mean Numeric* is the average of the numeric ratings (converted from discreet preference decisions) indicating on average the raters preferred system A over B on a scale of -1 to 1. Positive scores indicate a preference for system A. † significant at a 95% confidence interval for the mean numeric score.

We chose to have raters leave pairwise preferences, rather than evaluate each candidate summary in isolation, because raters can make a preference decisions more quickly than a valuation judgment, which allowed for collection of more data points. Furthermore, there is evidence that rater agreement is much higher in preference decisions than in value judgments (Ariely et al., 2008).

Results are shown in the first three rows of table 1. The first column of the table indicates the experiment that was run. The second column indicates the percentage of judgments for which the raters were in agreement. Agreement here is a *weak agreement*, where three raters are defined to be in agreement if they all gave a no preference rating, or if there was a preference rating, but no two preferences conflicted. The next three columns indicate the percentage of judgments for each preference category, grouped here into three coarse assignments. The final column indicates a numeric average for the experiment. This was calculated by converting users ratings to a scale of 1 (strongly preferred S_A) to -1 (strongly preferred S_B) at 0.33 intervals. Table 1 shows only results for items in which the raters had agreement in order to draw reliable conclusions, though the results change little when all items are taken into account.

Ultimately, the results indicate that none of the sentiment summarizers are strongly preferred over any other. Only the SAM v SMAC model has a difference that can be considered statistically significant. In terms of order we might conclude that SAM is the most preferred, followed by SM, followed by SMAC. However, the slight differences make any such conclusions tenuous at best. This leads one to wonder whether raters even require any complex modeling when summarizing opinions. To test this we took the lowest scoring model overall, SMAC, and compared it to a leading text baseline (LT) that simply selects the first sentence from a ranked list of reviews until the length constraint is violated. The results are given in the last row of 1. Here there is a clear distinction as raters preferred SMAC to LT, indicating that they did find usefulness in systems that modeled aspects and sentiment. However, there are still 25.5% of agreement items where the raters did choose a simple leading text baseline.

4 Analysis

Looking more closely at the results we observed that, even though raters did not strongly prefer any one sentiment-aware summarizer over another overall, they mostly did express preferences between systems on individual pairs of comparisons. For example, in the SAM vs SM experiment, only 16.8% of the comparisons yielded a "no preference" judgment from all three raters – by far the highest percentage of any experiment. This left 83.2% "slight preference" or higher judgments.

With this in mind we began examining the comments left by raters throughout all our experiments, including a set of additional experiments used during development of the systems. We observed several trends: 1) Raters tended to prefer summaries with lists, e.g., pros-cons lists; 2) Raters often did not like text without sentiment, hence the dislike of the leading text system where there is no guarantee that the first sentence will have any sentiment; 3) Raters disliked overly general comments, e.g., "The product was good". These statements carry no additional information over a product's overall star rating; 4) Raters did recognize (and strongly disliked) when the overall sentiment of the summary was inconsistent with the star rating; 5) Raters tended to prefer different

systems depending on what the star rating was. In particular, the SMAC system was generally preferred for products with neutral overall ratings, whereas the SAM system is preferred for products with ratings at the extremes. We hypothesize that SAM's low performance on neutral rated products is because the system suffers from the dual imperatives of selecting high intensity snippets and of selecting snippets that individually reflect particular sentiment polarities. When the desired sentiment polarity is neutral, it is difficult to find a snippet with lots of sentiment, whose overall polarity is still neutral, thus SAM may either ignore that aspect or include multiple mentions of that aspect at the expense of others; 6) Raters also preferred summaries with grammatically fluent text, which benefitted the leading text baseline.

These observations suggest that we could build a new system that takes into account all these factors (weighted accordingly) or we could build a rule-based meta-classifier that selects a single summary from the four systems described in this paper based on the global characteristics of each. The problem with the former is that it will require hand-tuning of coefficients for many different signals that are all, for the most part, weakly correlated to summary quality. The problem with the latter is inefficiency, i.e., it will require the maintenance and output of all four systems. In the next section we explore an alternate method that leverages the data gathered in the evaluation to automatically learn a new model. This approach is beneficial as it will allow any coefficients to be automatically tuned and will result in a single model that can be used to build new summaries.

5 Summarization with Ranking SVMs

Besides allowing us to assess the relative performance of our summarizers, our evaluation produced several hundred points of empirical data indicating which among two summaries raters prefer. In this section we explore how to build improved summarizers with this data by learning preference ranking SVMs, which are designed to learn relative to a set of preference judgments (Joachims, 2002).

A ranking SVM typically assumes as input a set of queries and associated partial ordering on the items returned by the query. The training data is defined as pairs of points, $\mathcal{T} = \{(x_i^k, x_j^k)_t\}_{t=1}^{|\mathcal{T}|}$, where each pair indicates that the i^{th} item is preferred over the j^{th} item for the k^{th} query. Each input point $x_i^k \in \mathbb{R}^m$ is a feature vector representing the properties of that particular item relative to the query. The goal is to learn a scoring function $s(x_i^k) \in \mathbb{R}$ such that $s(x_i^k) > s(x_j^k)$ if $(x_i^k, x_j^k) \in \mathcal{T}$. In other words, a ranking SVM learns a scoring function whose induced ranking over data points respects all preferences in the training data. The most straight-forward scoring function, and the one used here, is a linear classifier, $s(x_i^k) = w \cdot x_i^k$, making the goal of learning to find an appropriate weight vector $w \in \mathbb{R}^m$.

In its simplest form, the ranking SVM optimization problem can be written as the following quadratic programming problem,

$$\min \frac{1}{2}||w||^2 \quad \text{s.t.: } \forall (x_i^k, x_j^k) \in \mathcal{T}, \\ s(x_i^k) - s(x_j^k) \geq \text{PREF}(x_i^k, x_j^k)$$

where $\text{PREF}(x_i^k, x_j^k) \in \mathbb{R}$ is a function indicating to what degree item x_i^k is preferred over x_j^k (and serves as the margin of the classifier). This optimization is well studied and can be solved with a wide variety of techniques. In our experiments we used the SVM-light software package[4].

Our summarization evaluation provides us with precisely a large collection of preference points over different summaries for different product queries. Thus, we naturally have a training set $\mathcal{T}$ where each query is analogous to a specific product of interest and training points are two possible summarizations produced by two different systems with corresponding rater preferences. Assuming an appropriate choice of feature representation it is straight-forward to then train the model on our data using standard techniques for SVMs.

To train and test the model we compiled 1906 pairs of summary comparisons, each judged by three different raters. These pairs were extracted from the four experiments described in section 3 as well as the additional experiments we ran during development. For each pair of summaries (S_i^k, S_j^k) (for some product query indexed by k), we recorded how many raters preferred each of the items as v_i^k and v_j^k respectively, i.e., v_i^k is the number of the three raters who preferred summary S_i over S_j for product k. Note that $v_i^k + v_j^k$ does not necessarily equal 3 since some raters expressed no preference between them. We set the loss function $\text{PREF}(S_i^k, S_j^k) = v_i^k - v_j^k$, which in some cases

[4] http://svmlight.joachims.org/

could be zero, but never negative since the pairs are ordered. Note that this training set includes all data points, even those in which raters disagreed. This is important as the model can still learn from these points as the margin function PREF encodes the fact that these judgments are less certain.

We used a variety of features for a candidate summary: how much capitalization, punctuation, pros-cons, and (unique) aspects a summary had; the overall intensity, sentiment, min sentence sentiment, and max sentence sentiment in the summary; the overall rating R of the product; and conjunctions of these. Note that none of these features encode which system produced the summary or which experiment it was drawn from. This is important, as it allows the model to be used as standalone scoring function, i.e., we can set $\mathcal{L}$ to the learned linear classifier $s(S)$. Alternatively we could have included features like *what system was the summary produced from*. This would have helped the model learn things like the SMAC system is typically preferred for products with mid-range overall ratings. Such a model could only be used to rank the outputs of other summarizers and cannot be used standalone.

We evaluated the trained model by measuring its accuracy on predicting a single preference prediction, i.e., given pairs of summaries (S_i^k, S_j^k), how accurate is the model at predicting that S_i is preferred to S_j for product query k? We measured 10-fold cross-validation accuracy on the subset of the data for which the raters were in agreement. We measure accuracy for both weak agreement cases (at least one rater indicated a preference and the other two raters were in agreement or had no preference) and strong agreement cases (all three raters indicated the same preference). We ignored pairs in which all three raters made a no preference judgment as both summaries can can be considered equally valid. Furthermore, we ignored pairs in which two raters indicated conflicting preferences as there is no gold standard for such cases.

Results are given in table 2. We compare the ranking SVM summarizer to a baseline system that always selects the overall-better-performing summarization system from the experiment that the given datapoint was drawn from, e.g., for all the data points drawn from the SAM versus SMAC experiment, the baseline always chooses the SAM summary as its preference. Note that in most experiments the two systems emerged in a statistical tie, so this baseline performs only slightly better than chance. Table 2 clearly shows that the ranking SVM can predict preference accuracy much better than chance, and much better than that obtained by using only one summarizer (a reduction in error of 30% for strong agreement cases).

	Preference Prediction Accuracy	
	Weak Agr.	Strong Agr.
Baseline	54.3%	56.9%
Ranking SVM	61.8%	69.9%

Table 2: Accuracies for learned summarizers.

We can thus conclude that the data gathered in human preference evaluation experiments, such as the one presented here, have a beneficial secondary use as training data for constructing a new and more accurate summarizer. This raises an interesting line of future research: can we iterate this process to build even better summarizers? That is, can we use this trained summarizer (and variants of it) to generate more examples for raters to judge, and then use that data to learn even more powerful summarizers, which in turn could be used to generate even more training judgments, etc. This could be accomplished using Mechanical Turk[5] or another framework for gathering large quantities of cheap annotations.

6 Conclusions

We have presented the results of a large-scale evaluation of different sentiment summarization algorithms. In doing so, we explored different ways of using sentiment and aspect information. Our results indicated that humans prefer sentiment informed summaries over a simple baseline. This shows the usefulness of modeling sentiment and aspects when summarizing opinions. However, the evaluations also show no strong preference between different sentiment summarizers. A detailed analysis of the results led us to take the next step in this line of research – leveraging preference data gathered in human evaluations to automatically learn new summarization models. These new learned models show large improvements in preference prediction accuracy over the previous single best model.

Acknowledgements: The authors would like to thank Kerry Hannan, Raj Krishnan, Kristen Parton and Leo Velikovich for insightful discussions.

[5]http://www.mturk.com

References

D. Ariely, G. Loewenstein, and D. Prelec. 2008. Coherent arbitrariness: Stable demand curves without stable preferences. *The Quarterly Journal of Economics*, 118:73105.

S. Blair-Goldensohn, K. Hannan, R. McDonald, T. Neylon, G.A. Reis, and J. Reynar. 2008. Building a sentiment summarizer for local service reviews. In *WWW Workshop on NLP in the Information Explosion Era*.

S.R.K. Branavan, H. Chen, J. Eisenstein, and R. Barzilay. 2008. Learning document-level semantic properties from free-text annotations. In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL)*.

G. Carenini and J. Cheung. 2008. Extractive vs. nlg-based abstractive summarization of evaluative text: The effect of corpus controversiality. In *International Conference on Natural Language Generation (INLG)*.

G. Carenini, R.T. Ng, and E. Zwart. 2005. Extracting knowledge from evaluative text. In *Proceedings of the International Conference on Knowledge Capture*.

G. Carenini, R. Ng, and A. Pauls. 2006. Multi-document summarization of evaluative text. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Y. Choi, C. Cardie, E. Riloff, and S. Patwardhan. 2005. Identifying sources of opinions with conditional random fields and extraction patterns. In *Proceedings the Joint Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT-EMNLP)*.

E. Filatova and V. Hatzivassiloglou. 2004. A formal model for information selection in multi-sentence text extraction. In *Proceedings of the International Conference on Computational Linguistics (COLING)*.

M. Gamon, A. Aue, S. Corston-Oliver, and E. Ringger. 2005. Pulse: Mining customer opinions from free text. In *Proceedings of the 6th International Symposium on Intelligent Data Analysis (IDA)*.

J. Goldstein, V. Mittal, J. Carbonell, and M. Kantrowitz. 2000. Multi-document summarization by sentence extraction. *In Proceedings of the ANLP/NAACL Workshop on Automatic Summarization*.

M. Hu and B. Liu. 2004a. Mining and summarizing customer reviews. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*.

M. Hu and B. Liu. 2004b. Mining opinion features in customer reviews. In *Proceedings of National Conference on Artificial Intelligence (AAAI)*.

N. Jindal and B. Liu. 2006. Mining comprative sentences and relations. In *Proceedings of 21st National Conference on Artificial Intelligence (AAAI)*.

T. Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD)*.

S.M. Kim and E. Hovy. 2004. Determining the sentiment of opinions. In *Proceedings of Conference on Computational Linguistics (COLING)*.

C.Y. Lin and E. Hovy. 2003. Automatic evaluation of summaries using n-gram cooccurrence statistics. In *Proceedings of the Conference on Human Language Technologies and the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*.

R. McDonald. 2007. A Study of Global Inference Algorithms in Multi-document Summarization. In *Proceedings of the European Conference on Information Retrieval (ECIR)*.

K. McKeown, R.J. Passonneau, D.K. Elson, A. Nenkova, and J. Hirschberg. 2005. Do Summaries Help? A Task-Based Evaluation of Multi-Document Summarization. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*.

A.M. Popescu and O. Etzioni. 2005. Extracting product features and opinions from reviews. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

V. Stoyanov and C. Cardie. 2008. Topic identification for fine-grained opinion analysis. In *Proceedings of the Conference on Computational Linguistics (COLING)*.

I. Titov and R. McDonald. 2008a. A joint model of text and aspect ratings. In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL)*.

I. Titov and R. McDonald. 2008b. Modeling online reviews with multi-grain topic models. In *Proceedings of the Annual World Wide Web Conference (WWW)*.

L. Zhuang, F. Jing, and X.Y. Zhu. 2006. Movie review mining and summarization. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*.