Highlights: A Review Analysis Tool

Urvashi Khandelwal
University of Illinois at Urbana-Champaign, Urbana, IL
khndlwl2@illinois.edu

ABSTRACT

Automatic Information Extraction has been a widely studied field for a very long time, and most subtasks such as Named Entity Recognition, Part of Speech Tagging and Text Summarization are very challenging problems. As the amount of information on the web grows, the need for tools to parse this information and generate human readable versions becomes more of an immediate requirement. Customer reviews for products and services are a focused instance of the general problem and extracting relevant information from user opinions entails similar challenges. In this paper, we propose Highlights: a tool to render information about products and services to users in a more visually helpful manner by highlighting important sentences that represent a certain sentiment, are related to a particular topic relevant to most reviews for that business, and serve as good summaries for the overall quality assessment of the business.

1. INTRODUCTION

Crowdsourced information has become an important part of achieving success as a business. People heavily rely on other users' input on whether to use a product or service. The hope is that a large number of reviews will be able to eliminate most forms of bias, and the opinionss would overall indicate the pros and cons of the product/service in question. For this reason, the amount of crowdsourced information on the web has been growing very quickly. Several websites like Yelp and Amazon have been aggregating user reviews on millions of businesses and products. However, as the amount of information grows, it becomes increasingly difficult to parse the data, let alone find relevant information.

The notion of relevance varies with users. Consider a restaurant on Yelp that has thousands of low ratings and most of those negative reviews focus on the bad chicken. On the other hand, it has only a handful of reviews complimenting the salads. When a vegetarian looks at a summary of the restaurant, they might get deterred by the massive amount of poor reviews, even though the information they are basing this off of is irrelevant to them. Given such scenarios, it is desirable to design a system that allows users to find the information they are looking for faster, more easily, and hopefully in a more visually helpful manner.

This led to the design of Highlights, a review analysis tool that takes the reviews of a business or product and allows the user to "highlight" specific information based on sentiment and topics. For instance, Fig.2(a) shows a review (truncated) for Ghirardelli, returned as the first result for

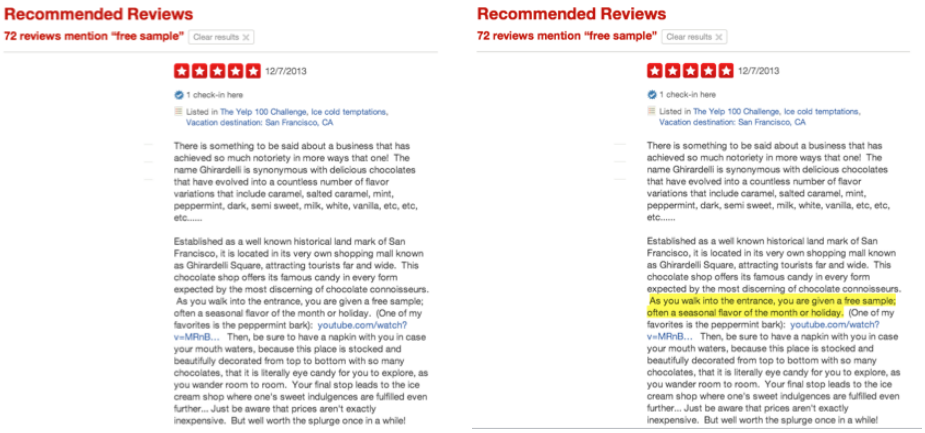
Figure 1: Highlights: A Prototype

the topic "free samples". This review is extremely long, and tedious to parse quickly. The context for the topic is even harder to grasp without reading most of the review to find the keywords. However, in Fig.2(b) the summary sentence related to the topic has been highlighted and the user can quickly read the surrounding text to gauge whether the review would be useful for them. Within Highlights, the user is given the option of selecting the features based on which they would like to see reviews, and the system displays a ranked list of the reviews with highlighted summary sentences.

We organize the paper as follows: in Section 2 we cover some related work in the general area of analyzing reviews in this way, in Section 3 we provide some background, in Section 4 we provide a concrete task definition. We look at techniques in Section 5 and experimental results are presented in Section 6. Finally, we provide some concluding thoughts and ideas for future work in Section 7.

2. RELATED WORK

Review summarization based on various aspects has most certainly been studied widely. Highlights is a system that combines various techniques in the realm of information extraction to process and present review data in a way that is helpful to users. Review Spotlight[1] is a similar system that extracts adjective-noun word pairs and runs sentiment analysis on them, in order to generate word clouds that represent summarized information. Highlights is different from them in the sense that it summarizes reviews and then ranks them based on which reviews might be most useful to the users, who can then scroll through them whilst reading the highlighted summary sentences as well as contextually rel-



(a) Current View (b) View after Highlights

Figure 2: A Yelp review listing for Ghirardelli Ice Cream and Chocolate Shop for the topic "free sample" before and after the use of Highlights.

evant sections to decide which reviews seem useful. This removes the extra step of the user having to deal with word pairs that are placed outside of their original context.

In [2], Wang et. al. define the Latent Aspect Rating Analysis (LARA) problem of mining individual reviewer's opinions on various topical aspects in online reviews. In the followup work, they defined the Latent Aspect Rating Analysis Model (LARAM)[3] which can simulataneously identify latent topical aspects, ratings on those topics and weights placed by reviewers on each of those topics. Although, Highlights suffers from poor topic modeling results, the objective of the system is different from solving the issue of sparsity in opinion mining in online review data. Hence, a potential version of the system could incorporate the work of LARAM to mine better topics and enhance the review summarization task even further. But in this paper, we focus on a simplistic system that serves as an initial prototype, in order to motivate the need for the system itself.

Websites like Yelp and Amazon provide users with review snippets, and Yelp specifically generates snippets containing some key words with a high frequency within the business' reviews. However, neither works with the intuition of dividing the task between positive and negative sentiments in order to present a summarized view of the reviews. It is our belief that these companies which aggregate opinion based data for products and services, could potentially benefit from incorporating this system into their larger framework.

3. BACKGROUND

Highlights is a review analysis framework that combines a number of text mining techniques, including sentiment analysis, topic modeling and text summarization. We briefly introduce each of these components in this section.

3.1 Sentiment Analysis

In [4], Mukherjee defines sentiment analysis as an Information Extraction subtask that looks to extract subjective opinions (e.g. positive/negative sentiment) from a person's questions, statements and comments by analyzing a large number of documents. He identifies some key challenges associated with this task. A rather important challenge that most review data is riddled with is the existence of implicit sentiment as induced by language features such as sarcasm. In order to classify text based on sentiment, it is possible to use traditional classification techniques such as Naive Bayes, Support Vector Machines, MaxEnt etc.

3.2 Topic Modeling

Blei et. al. [5] define the goal of topic modeling as finding short descriptions of members of a collection that enable efficient processing of large collections while preserving the underlying statistical relationships that are critical for tasks such as classification, summarization, similarity and relevance judgements. They proposed Latent Dirichlet Allocation (LDA): a generative probabilistic model of a corpus in which each document is represented as a mixture over latent topics, each of which is a distribution over words in the corppus vocabulary. Hence, extracting these topics is a good way to be able to identify the most important aspects of the content of the document and is critical in summarizing documents and corpora.

3.3 Text Summarization

Radev et. al. [6] define summarization as taking either a single text or multiple texts and condensing the information in a way which still represents the important content of the original text, and is short in length. There are several techniques for both single and multiple document summarizations, however, in this paper we are concerned with topic-driven summaries. The most widely known technique for this is Maximal Marginal Relevance[12] which tries to rank sentences based on some metric such as term based weights, and then selects sentences as summaries based on two ideas: a high rank, and least similarity to already selected sentences. This helps with diversification of the retrieved results.

4. TASK DEFINITION

Consider the problem setting where we have businesses or products, identified as items, and customers for these items, identified as users. Users that have used the items provide text based reviews expressing their opinions about them. Now, the task is to help new users find reviews that provide them with the most relevant information about the items. This is done by selecting reviews based on a particular sen-

Figure 3: Pipeline for Highlights

timent, extracting topics from them and then highlighting pieces of the reviews considered most representative of the opinion. The retrieved reviews are then rendered to the users sequentially, allowing them to scroll through whilst reading the highlighted snippets and surrounding sentences for context. Fig.1, which illustrates a prototype for the system, shows an example of how the user would be able to navigate the options within the tool by toggling buttons to turn features on/off. So if they select "Negative", the summary sentences that are highlighted in the reviews reflect negative sentiment whilst also being associated with the most relevant topics for the business.

5. METHODOLOGY

Now we look at the details of the system implementation. We give a high level overview for the design before looking at each part of the process separately.

Fig.3 illustrates how Highlights moves from reviews to the final visualization rendered to the user. It starts with reviews for an individual item and works on classifying them into positive/negative sentiment. Then, for every sentiment class, it extracts topics to discover the top keywords associated with each sentiment. Finally, using this information, it summarizes based on just topics, or both sentiment and topics, depending on the options the user has selected, and returns the reviews containing the highest ranked summaries highlighted based on the colors of the features selected. In Fig.2(b), for instance, the highlighting is based on just topics, whereas in Fig.1, the highlighting is based on the positive sentiment as well.

5.1 Sentiment Classification and Topic Extraction

For the sentiment classification task, we defined two class labels: positive and negative sentiment. Features used to carry out the classification task were Bag-of-Word features based on token frequencies computed across the corpus - in this case, all the reviews for a particular item. The classification was carried out by a Support Vector Machine from the MeTA toolkit[7]. SVMs classify data points based on decision boundaries learned from the features provided. In this case (as seen from the experiments) it did particularly well at learning the sentiment for short reviews, where each review was treated as a document. Despite the fact that ratings data is available and the sentiment in reviews can be identified without the additional step of classification, it is desirable to run this since it would work to normalize reviews across users who are too generous given their opinions, vs. those that are too strict despite mostly appreciating the item.

For topic extraction, we also used MeTA's topic modeling tool[7] which uses LDA to extract k topics from the text. It uses Collapsed Variational Bayes for statistical inference. The topic modeling is carried out over all reviews, as well as over reviews from each sentiment class. This helps to identify topics that are in general appreciated and contribute to higher ratings, distinguishing them from the top aspects drawing criticisms.

Table 1: Performance Metrics for sentiment classification using MeTA's SVM classifier on the Yelp and IMDb datasets

Dataset	Reviews	Precision	Recall	F1-score
Yelp	113,545	0.897	0.88	0.916
IMDb	50,000	0.891	0.891	0.891

By isolating topics in this way, the system is more robust in dealing with scenarios such as those in the example from the introduction where vegetarians might be deterred from restaurants that have thousands of negative reviews for the chicken, simply because keywords in the few positive reviews were unable to amass the document frequency for higher retrieval scores.

5.2 Summarizer

The text summarizer implementation for this system is independent of any existing summarization toolkits. It was written in Python. The main idea was to compute TF-IDF scores for every token in the documents (reviews) across the corpus (all reviews for an item), and select sentences containing tokens with the highest scores.

$$\text{TF-IDF score} = Token\ Freq\ *\ 1+\log \frac{\#Reviews}{\#Reviews\ with\ Token}$$

The NLTK toolkit[11] was used to tokenize the reviews into words when computing TF-IDF scores and into sentences when carrying out sentence selection.

In addition, the sentence selection was augmented using the topics extracted from the topic modeling using LDA. Two sets of topics were generated where the output included a list of words, ranked by their probabilities, illustrating relevance to the topic. The top N words were selected to incorporate into the summarizer, where N is a parameter. The TF-IDF scores and the occurence of topic related keywords in sentences were equally weighted features for the text summarization because the TF-IDF weighting helped to identify the aspects that made the reviews unique adding diversity to the results retrieved, while the topics helped to eliminate noise in terms of words that were far too rare to be relevant, yet occured frequently in the given review.

6. EXPERIMENTS

6.1 Datasets

The prototype for Highlights was tested by running experiments on the Yelp academic dataset. All businesses with fewer than fifty reviews were pruned in order to reduce sparsity (data per business was still sparse given the typically short length of business reviews). After pruning, 113545 reviews for 735 businesses remained. In addition, some experiments for sentiment analysis and topic modeling were also run on the IMDb Movie Review dataset[9], simply to widen the scope of interpretting the results. This dataset had 50,000 movies, with two reviews per movie. Due to the sparsity in reviews per movie, the system prototype could not be run on this dataset.

6.2 The System: Sentiment, Topics and Summaries

First, we ran some sentiment analysis experiments to gauge the complexity of the task. All reviews were used and the SVM classifier required the libsvm format[8]. Surprisingly,

Table 2: Word intrusion for topic models: Underline - intruder, Bold - guess

Datset	T1	T2	T3	T4	T5
Yelp (5)	food	**menu**	sauc	chicken	flavor
Yelp (1)	call	sale	car	store	**told**
IMDb (Pos)	work	film	**time**	stori	charact
IMDb (Neg)	stori	movi	bad	**it**	film

with just token frequencies, the classifier achieved a 93.7% overall accuracy on the yelp dataset and a 89.1% overall accuracy on the IMDb dataset. Table 1 shows the precision, recall and F1-scores for the two datasets. The ground truth for binary sentiment was set up using ratings data. On a scale of 1 to 5 for yelp, anything below a 3 was negative and anything above was positive, while for IMDb, the pivots were 5 and 6 on a scale from 1 to 10. A high accuracy despite the sparsity of the IMDb dataset demonstrates the robustness of the sentiment extraction of Highlights.

Next, we ran LDA on the reviews. Before working on per business analysis, we ran it on a subset of the reviews with $k = 3$ to see whether topic extraction would give a sense of the major good quality points for items. For Yelp, it was run on the 5 star and 1 star reviews to see what people tend to like and dislike about businesses. For IMDb, we ran it on the positive and negative reviews separately, to see whether it could generate the topics that make most users give positive reviews vs. those that make them dislike the movies. The evaluation for this part of the system was intended to be subjective with the use of word intrusion[10]. The top five words from within a topic were chosen and an intruder word with lower probability from within the topic was also chosen. These words were shuffled and the task was to identify the intruder.

Even with a much larger set, the topic models were quite weak as can be seen from the word intrusion assessment, shown in Table 2. Overall, for all of the 12 topics extracted, there was only 41.7% success in identifying the intruder. This is due to the sparsity of data. With per business analyses, the topics were far worse. However, a beneficial aspect of the topic modeling was the high probability words it generated for each topic. Hence, Highlights uses the LDA with $k = 2$ to generate two sets of topic words, and uses the top 6 words in each to enhance the summarizer, where 6 was a heuristically chosen parameter.

For the summarizer, we ran our implementation over individual businesses and generated the top-N summary sentences, ranked based on the TF-IDF score and topic inclusion. Fig.1 shows two highlighted sentences for the Einstein Bagels business. After running LDA on the reviews, "menu" and "food" were among the high probability words, while "special" and "great" had the highest TF-IDF scores. This demonstrates that the 50% weighting scheme for the two aspects does produce some good results. In this example, the positive sentiment and topics were being covered. From reading the rest of these two reviews, subjectively the two highlighted sentences served as good snippets or summaries to present to the user. It is clear that the lack of an easily implementable evaluation metric for text summarization makes evaluating this part of the system consdierably difficult. Moving forward, a user study to test the system would certainly help to not only evaluate performance, but also gather feedback on what can be improved.

7. CONCLUSION AND FUTURE WORK

We have presented Highlights, a system that serves as a start to exploring the realm of making reviews more useful to users, in terms of the sentiment, topics and in-context summaries. The notion of highlighting, different from [1]'s tag cloud approach, seems to make it easier for users to parse reviews by simply scrolling through retrieved results instead of adding more clicks to the process of groking the reviews. We have presented a three-step approach: sentiment analysis to topic extraction to summary selection, which maps out an intuitive pipeline as to how the users might want to process the information. A future direction for this system would be to enhance each piece of the pipeline with more features. For instance, sentiment analysis on a per sentence basis could help to improve the summaries extracted, since sometimes users give an overall good review but might mention a couple of key things they did not like. For topic models, although it seems that more reviews are the only way to deal with the data sparsity, it could be beneficial to take similar businesses, in terms of products and users that have reviewed them, in order to combine a few businesses before extracting topics from them. It could be interesting to see the impact of this on the system. Finally, for the text summarizer, making the summary sentence selection more selective, following the Maximal Marginal Relevance technique[12], the system might see an improvement in the retrieval. Overall, this system fills a necessary requirement in today's large scale information extraction tasks by making it easier for people to use review data on the web. Preliminary results show that it is a good start to creating a strong and robust tool, and conducting user studies would help to back this claim, as well as gather feedback for what aspects of the system can be further improved.

8. REFERENCES

[1] K. Yatani, M. Novati, A. Trusty, K.N. Truong. Review Spotlight: A User Interface for Summarizing User-generated Reviews Using Adjective-Noun Word Pairs. In *SIGCHI*, 2011.

[2] H. Wang, Y. Lue, C. Zhai. Latent Aspect Rating Analysis on Review Text Data: A Rating Regression Approach. In *SIGKDD*, 2010.

[3] H. Wang, Y. Lue, C. Zhai. Latent Aspect Rating Analysis without Aspect Keyword Supervision. In *SIGKDD*, 2011.

[4] S. Mukherjee. Sentiment Analysis: A Literature Survey. In *CoRR*, 2013.

[5] D. Blei, A. Ng, M. Jordan. Latent Dirichlet Allocation. In *Journal of Machine Learning Research, 3:993–1022*, 2003.

[6] D. Radev, E. Hovy, K. McKeown. Introduction to the special issue on summarization. In *Computational Linguistics., 28(4):399{408}*, 2002.

[7] S. Massung, C. Geigle. MeTA. *http://meta-toolkit.github.io/meta/*

[8] C. Chang, C. Lin. LIBSVM: A Library for Support Vector Machines. In ACM Transactions on Intelligent Systems and Technology, 2011.

[9] A. Maas, R. Daly, P. Pham, D. Huang, A. Ng, C. Potts. Learning Word Vectors for Sentiment Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human*

Language Technologies, 2011.

[10] J. Chang, J. Boyd-Graber, C. Wang, S. Gerrish, D. Blei. Reading Tea Leaves: How Humans Interpret Topic Models In *Neural Information Processing Systems*, 2009.

[11] S. Bird, E. Loper, E. Klein. Natural Language Processing with Python. In *O'Reilly Media Inc.*, 2009.

[12] J. Carbonell, J. Goldstein. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. In *SIGIR*, 1998.