TL;DR? Tips Generation for Yelp Reviews

Rui Meng (rui.meng@pitt.edu)
Jun Fu (juf29@pitt.edu)
Mengdi Wang (mew133@pitt.edu)

1 Introduction

TL;DR is short for "Too long; didn't read", indicating an article is too long for reader to digest. In the meantime this Internet slang indicates a common phenomenon that readers tend to read concise content. Online review provides customers a particular way to rate and comment on certain topics, which performs as an important information channel of sharing and communicating ideas among customers. Many popular websites or apps, such as Yelp, Amazon, and Netflix, allow customers to leave reviews for services or goods. For example, many customers read reviews on Yelp in order to get an instructive opinion on a certain business. Oftentimes it is the case that hundreds or thousands of long reviews about a single business, making it difficult for people make evaluation for that business. Though Yelp offers the function that allowing people to write short reviews (tips), most people prefer leaving long messages.

Automatic summarization is a key task in natural language processing and it has a variety of applications in the real world. One such application is with regard to product or restaurant reviews, which aiming to enable customers to grab the most important information without reading over all reviews, and thus save their time and effort. Our goal in this project is to derive short tips automatically, on either single review article or all the reviews about a business. It would be doubtless valuable for improving users' experience by highlighting the essence of reviews.

There are two typical methods for summarizing a document. The first one is extractive summarization, in which the summary is made up of words, phrases, and/or sentences that are extracted from the original document. Another method is in a generative manner, by applying language model to generate novel text snippet that summarizes the information in the original document. Based on the fact that a large amount of tips are extracted from reviews, we believe that the Yelp tips generation task can be realized by conducting the extractive summarization method, generating tips for Yelp reviews by extracting sentences from original text. More specifically, given a review document, we can summarize the user generated review into one or few sentences, and subsequently aggregate all the summarized sentences as the summary of the given business.

Meanwhile, the large amount of tips as well as reviews data offers us the opportunity to explore the possibility of training a generative summarization model. So in the last part of our project, we present an experiment result of a state-of-the-art generative summarization model based on neural network [1].

2 Related Work

In [2], a summary is defined as "a text that is produced from one or more texts, that conveys important information in the original text(s), and that is no longer than half of the original text(s) and usually significantly less than that". We could capture three important aspects that characterize research on automatic summarization as in [3]: Firstly, summaries could be produced from a single document or multiple documents; the second is summaries should preserve important information; the last characteristic is, summaries should be short.

The problem of summarization has been studied in many research work, most of which relies on verbatim *extraction* of sentences to address the problem of single-document summarization. Most early work on single-document summarization focused on *technical documents* [4]. The key idea is to derive a *significance factor* that reflects the number of occurrences of significant words within a sentence. All the sentences are ranked in order of their significance factor and the top ranking sentences are selected to form the auto summarization. Afterwards, with the advent of machine learning techniques in NLP, a series of publications appeared in the 1990s employing statistical techniques to realize the document summarization. Most initial systems relied on naive-Bayes methods[5] based on the features independence assumptions. However, in the real world the assumptions may not hold. Other method [6] focused on the choice of appropriate features and on learning algorithms. While the previous approched are mostly feature-based and non-sequential, [7] modeled the problem of extracting a sentence from a document using a *hidden Markov Model* (HMM), only using three features: sentence position, number of terms in the sentence and the likeliness of the sentence terms given the document terms. [8] used log-linear models to show empirically that it works better than a naive-Bayes model.

A few works dealt with more complicated problem which generating novel sentences for summarization. A recent work [1] combined a neural language model with a contextual input encoder and proposed a fully data-driven approach for generating abstractive summaries.

3 Methodology

In our project, we mainly focus on two tasks: 1) generating a summary (one sentence) on single review article; 2) generating a summary (consisting of a few sentences) of a given business. Our approach to those two tasks can be broken down into two main steps: text representation and summary sentence extraction.

3.1 Text Representation

Text representation denotes how to model and represent a part of text, basically the part can be a word, phrase, sentence and document. Text representation plays a fundamental role in almost all the natural language processing tasks as we have to convert the text into a form which computer can "understand" before conducing any further processing. In this task, we mainly consider the sentence-level modeling as we are aiming to extract key sentences from review documents.

3.1.1 Bag-of-Words Model

Bag-of-words model is a simplifying representation which commonly used in natural language processing and information retrieval. In this model, a text (such as a sentence or a document) is represented as the bag (multiset) of its words, disregarding grammar and even word order but keeping multiplicity.

TF-IDF, short for term frequency-inverse document frequency, is one of the most commonly-used weighting method for bag-of-word model. It is a numerical statistic that is intended to reflect how important a word is to a document in the collection or corpus. Each document could be represented by a vector in which each element corresponds the importance of one word in the corpus. The importance of a word in one document is determined by the product of the frequency (TF) and inverse-document frequency (IDF).

$$\text{TF-IDF} = f_{ij} \times \log \frac{n}{d_j}$$

where f_{ij} is (relative) frequency of word j in document i and $\log \frac{n}{d_j}$ is the inverse document frequency.

3.1.2 Topic Model

Topic model is a type of statistical model, assuming that a document concerns multiple topics in different proportions. LDA (Latent Dirichlet Allocation) [9] is a widely used topic modeling method in language processing. In LDA, each document may be viewed as a mixture of various topics. This is similar to probabilistic latent semantic analysis (pLSA), except that in LDA the topic distribution is assumed to have a Dirichlet prior. We could perform LDA on the documents and use the distribution of each document on N topics as presentation of this document.

3.1.3 Distributed Representation

Another important text representation is distributed representation. Compared to representing text with discrete and independent words, distributed representation projects each word into vectors of real numbers in a low-dimensional space by neural network, which has been shown to be very effective at language modeling and compositionality.

Word2vec[10] is a state-of-the-art neural language modeling implementation. In this project, we apply word2vec to obtain the vector representation of each word, and sentence vector is generated by taking the average of the vectors of each word contained in the sentence.

3.2 Summary Sentence Extraction

In this project, we mainly explored the extractive summarization methods. The advantage of extractive summarization is we can obtain a fairly good summary based on some statistical methods, which can meet both readability and saliency. Here we apply two extractive methods, one is density-based method and the other one is graph-based.

The major drawback of extractive summarization is apparent: it's expression ability is limited to the original text, as extractive model cannot generate a sentence which doesn't appear in the text at all. However we found that a generative method, which usually requires a large amount of training data, is not applicable in our case, due to the fact that only few usable data in Yelp (though we extracted about 20,000 review-tip pairs, the quality of tips is not as good as summaries generated by linguists).

Actually we ran a state-of-the-art generative summarization model[1] on our data, but the result is fairly bad, largely due to the fact that neural network model needs training on a considerably large and high-quality dataset. The original work trained this generative model on a dataset with around 9.5 million document-summary pairs. We will talk about the result of this experiment in Section 4.5.

3.2.1 Density-based Method

As we have two tasks (summary on single review or on all reviews of a business), we will describe how we use density-based method to deal with these two tasks separately.

a. Summarize on Single Review

In this task we aim to extract a summarization for a single review document. As one review does not have too many sentences and in many cases one review talks about one main topic. We want to choose a sentence containing information that is commonly repeated among all sentences in this review as the summary. This is very intuitive, because a good summarization should contain the information that is common between all of the reviews. We choose the sentence by minimizing a score criterion that is approximately inversely proportional to the density of sentence vectors around a the given sentence vector. That is to say, we want to find the density center of the sentence vectors. The score could be written like this:

$$\text{score} = \sum_{i=1}^{k} dist(s, s_i)$$

where dist is the euclidean distance and k is the k nearest neighbors.

b. Summarize on Reviews of a Business

One business may have many reviews, thus it is extremely difficult for users to obtain useful information of a business, even if we obtain one summary sentence for every review. Therefore, we extract summary for each business as our second task. In this task, we use the same score method mentioned above and extract M sentences as the summary of a particular business. However we do not want those M sentence to be similar, one intuitive improvement is to add a penalty term to the

score in order to generate diverse summaries:

$$\text{score} = \sum_{i=1}^{k} dist(s, s_i) + penalty(s)$$

$$penalty(s) = \sum_{s_e \in ExtractedSentences} (\text{cosine}(s, s_e))^2$$

$$\text{cosine}(s, s_e) = \frac{s \cdot s_e}{||s||||s_e||}$$

where cosine(s, s_e) is the cosine similarity between sentences s and s_e.

3.2.2 Graph-based Method

Graph-based methods for sentence ranking productively exploit repetition in the input, both on the word and sentence level. In this way, graph-based methods combine the advantages from word frequency and sentence clustering methods. TextRank[11] is a representative of Graph-based methods, which has been proved to be very effective on keyphrase and summary sentence extraction.

Graph-based ranking algorithm PageRank has been successfully used in social network, citation analysis and many applications. The importance of elements can be revealed by analyzing the linking structure of a given network. Similarly, the importance of words and sentences can be measured in the network analyzing way too.

Formally, following the definition in PageRank, let $G = V, E$ be a weighted undirected graph with the set of vertices V and set of edges E. The score of a vertex V_i is defined as follows:

$$Score(V_i) = (1 - d) + d * \sum_{V_j \in IN(V_i)} \frac{\omega_{ji}}{\sum_{V_k \in Out(V_j)} \omega_{jk}} * WS(V_j)$$

where d is a damping factor that can be set between 0 and 1. $IN(V_i)$ and $OUT(V_i)$ donate the adjacent nodes of V_i which linked with input and output edge separately. Here as it's a undirected graph, so $IN(V_i)$ and $OUT(V_i)$ are the same set. ω_{ij} represents for the weight of edge from V_i to V_j.

Based on the above standard graph definition, if we set each sentence in text as a vertex in graph,and set the similarity of each pair of sentences as edge weight, then we can get a sentence graph and learn the importance of each sentence by PageRank. Therefore we can extract the most important sentence in sentence graph and treat it as the summary sentence in text. Also different sentence representation can be used here to explore it's influence on computing sentence similarity.

4 Experiment

4.1 Dataset

Our dataset is provided by the Yelp Dataset Challenge: `https://www.yelp.com/dataset_challenge`. This dataset includes information about local businesses in 10 cities across 4 countries. It contains around 2,225,213 reviews and 591,864 tips written by around 552,000 users for 77,000 businesses.

We want to find the reviews from which can also extract users' tips. These tips could be used as the reference summaries (ground truth) when evaluating on summaries extracted by our model. By using the business_id and user_id information, we can easily identify the reviews and tips which talking about the same business and written by the same user. We collected more than 19,000 reviews which have tips inside. However not all the reviews are in good quality, some of which only contain very little content. So we decide to filter out the low-quality reviews whose length is less 20 words. Finally we end up with a dataset of 17,483 review-tip pairs.

A process of sentence segmentation is required, which is accomplished by a sentence tokenization package in NLTK. It's worth noting that the NLTK package cannot perfectly tokenize the reviews into sentences. This is because the online reviews use many ungrammatical punctuations which make it difficult to detect the sentence boundary. Thus it may have some bad effects on the final summarization results.

4.2 Experimental Setting

As the word usage in Yelp is really casual and diverse, there are more than 1 million unique words. This is really disadvantageous for bag-of-word model, as the number of dimension is the number of unique word in the corpus, making the vectors very sparse. We ranked all the words by frequency and kept the top 100,000 words as vocabulary. In addition, we set up both the number of topics in LDA and the number of dimension in word2vec to 200.

4.3 Evaluation Metric

As for the evaluation, we reviewed most of mainstream evaluation methods from [12]. Though there are tens of different metrics trying to capture the quality of machine-generated summary from different aspects, few of them are well recognized by researchers. Document Understanding Conference (DUC) evaluation turns out to be the most objective and accurate evaluation method for summarization systems. However as DUC evaluation is conducted by human, it's inapplicable for us to conduct this method. After thoughtful consideration and comparison, we decide to apply two well-established evaluation metrics in our experiment, which are BLEU[13] and ROUGE[14].

4.3.1 BLEU

BLEU, or Bilingual Evaluation Understudy, is a precision based metric used primarily for machine translation NLP tasks. The metric is as follows; for a candidate or test summary, and a reference summary, let us define the following variables:

- $m_w(i) =$ number of occurrences of ith n-gram in the test summary.
- $m_{ref}(j) =$ number of occurrences of jth n-gram in the reference summary.
- $m_(max) = min(m_w(i), m_{ref}(j))$. If the ith n-gram does not appear in the reference summary, this value is 0
- $w_t =$ total of all n-grams in test summary.

Then the BLEU metric is given by the following formula:

$$\text{BLEU} = \frac{\sum_{i \in \text{n-grams}} m_{max}(i)}{w_t} \tag{1}$$

In [13], it was noted that unigrams satisfied adequacy and had the advantage that they were simpler. Longer n-grams were better for measuring fluency. Since we are only evaluating a summarization task that uses extraction, i.e. we are no interested in measuring fluency, we chose to use unigrams in the evaluation.

4.3.2 ROUGE

[14] introduced a set of metrics called Recall-Oriented Understudy for Gisting Evaluation (ROUGE) that have become standards of automatic evaluation of summaries.

In what follow, let $R = \{r_1, ..., r_m\}$ be a set of reference summaries, and let s be a summary generated automatically by some systems. Let $\Phi_n(d)$ be a binary vector representing the n-grams contained in a document d; the i-th component $\phi^i(d)$ is 1 if the i-th n-gram is contained in d and 0 otherwise. The metric ROUGE-N is an n-gram recall based statistic that can be computed as follows:

$$\text{ROUGE} = \frac{\sum_{r \in R} < \Phi_n(r), \Phi_n(s) >}{\sum_{r \in R} < \Phi_n(r), \Phi_n(r) >} \tag{2}$$

where $< ., . >$ denotes the usual inner product of vectors. This measure is closely related to BLEU which is described above.

4.4 Result and Discussion

Table 1 shows the results of different summarization method on review-level (single document) task, which aiming to extract one sentence as summary for each review article. From this table, we can see that the graph-based method works better than density-based method in general. We can also see that among all the representations the bag-of-word model obtains the best performance in both density-based and graph-based method, indicating that the word frequency may be more helpful with revealing important sentences on review-level task. As one review document only contains less than 20 sentences, the semantic representations (Word2Vec and LDA) usually provide too condensed information, which may not be suitable for this task.

Table 1: Results of review-level task

Extracting method	Density-based			Graph-based		
Representation	Word2vec	LDA	BOW	Word2vec	LDA	BOW
Rouge	0.46	0.42	**0.51**	0.38	0.53	**0.57**
BLEU	0.54	0.54	**0.72**	0.55	0.77	**0.81**

Table 2 shows the summarization results on business-level task, which aiming to extract top K sentences as tips for each business. This table shows that graph-based method performs better than density-based method in general. And similar to the result in review-level task, the bag-of-word representation has better performance than the other two representation methods overall.

Table 2: Results of business-level task

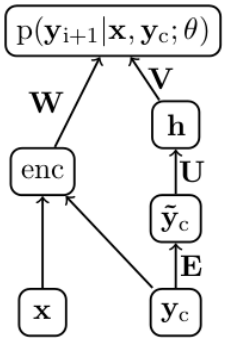
Extracting method	Density-based			Graph-based		
Representation	Word2vec	LDA	BOW	Word2vec	LDA	BOW
Rouge	**0.58**	0.22	0.39	0.47	0.36	**0.50**
BLEU	0.20	**0.41**	0.30	0.31	**0.45**	0.41

4.5 Exploration of State-of-the-Art Generative Model

In the final part of our project, we want to explore the possibility that whether generative models are applicable on solving this Yelp tips generation task. Recently neural network is widely applied on many different NLP tasks and performs its strength on extracting semantic features. For our task, we applied a generative summarization method which based on Neural Attention Model[1]. Basically, this method utilizes a attention-based neural language model that generates each word of the summary conditioned on the input sentence.

Given an input sentence, the goal is to produce a condensed summary. Let the input consist of a sequence of M words $x_1, ..., x_M$ coming from a fixed vocabulary V of size $|V| = V$. We will represent each word as an indicator vector $x_i \in \{0,1\}^V$ for $i \in \{1, ..., M\}$, sentences as a sequence of indicators, and X as the set of possible inputs. Furthermore define the notation $x_{[i,j,k]}$ to indicate the sub-sequence of elements i, j, k.

Figure 1: A network diagram for the NNLM decoder with additional encoder element.



Considering the problem of generating summaries, we define the set $\mathcal{Y} \subset (\{0,1\}^V, ..., 0, 1^V)$ as all possible sentences of length N , i.e. for all i and $y\mathcal{Y}$, y_i is an indicator. We say a system is abstractive if it tries to find the optimal sequence from this set $\mathcal{Y}$,

$$argmax_{y \in \mathcal{Y}} s(x, y)$$

under a scoring function $s : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$. The scoring function can be defined as:

$$s(x, y) \approx \sum_{i=0}^{N-1} log\ p(y_{i+1}|x, y_c; \theta)$$

Then we trained this model on our Yelp review data, including 17,483 review-tip pairs. As our dataset is relatively small, the training process is really quick, finished within 10 minutes. Then

we tested it on ten Yelp reviews. The testing data and generated tips are shown in Figure 2 and Figure 3 respectively. From the testing result we could see that the tips generated by the generative model is fairly bad, largely because the neural network model failed to learn a accurate linguistic expression from the small size training set. Actually, the original work trained this generative model on a dataset with around 9.5 million document-summary pairs.

Figure 2: Screenshot of testing reviews and corresponding tips

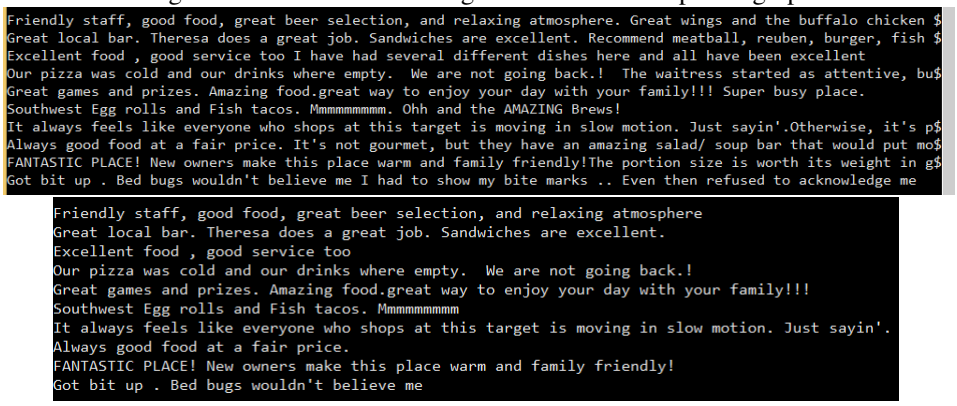
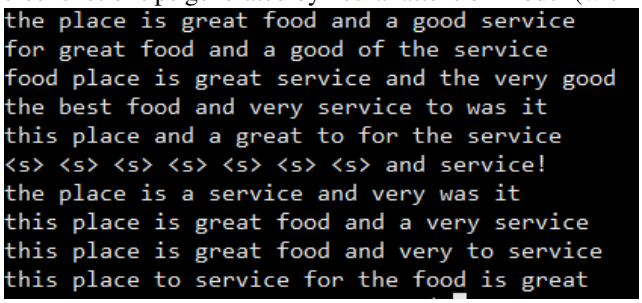


Figure 3: Screenshot of tips generated by neural attention model (with #length=10)



5 Conclusion

Review summarization is an important application of natural language processing, which is very useful for providing customs brief feedback generated on large amount of online reviews. In this project, we apply multiple text representation and summarization methods to derive short summaries on business reviews.

As for the text representation part, our experimental result indicates the robustness of traditional bag-of-word model, it achieved the best performance on both density-based and graph-based method. The other two semantic representations failed to make competitive results. As our task mainly concerns on summarizing on short text, the semantic advantage of these models may be not helpful on extractive methods.

As for the comparison between the two extractive methods, we find that graph-based method achieves a significant improvement over density-based method. The advantage of density-based method on capturing global information may explain this performance difference.

A generative summarization method based on neural attention model is also explored here. Though this method does not work very well, the thought of employing neural network representation and translation model enlightens us about possible improvement on our project.

In the future, we plan to try other state-of-the-art representative methods, for example doc2vec. We may also apply other sentence importance scoring methods, based on various semantic features and multiple semantic similarities.

References

[1] Alexander M Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*, 2015.

[2] Dragomir R Radev, Eduard Hovy, and Kathleen McKeown. Introduction to the special issue on summarization. *Computational linguistics*, 28(4):399–408, 2002.

[3] Dipanjan Das and André FT Martins. A survey on automatic text summarization. *Literature Survey for the Language and Statistics II course at CMU*, 4:192–195, 2007.

[4] Hans Peter Luhn. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165, 1958.

[5] Julian Kupiec, Jan Pedersen, and Francine Chen. A trainable document summarizer. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 68–73. ACM, 1995.

[6] Eduard Hovy and Chin-Yew Lin. Automated text summarization and the summarist system. In *Proceedings of a workshop on held at Baltimore, Maryland: October 13-15, 1998*, pages 197–214. Association for Computational Linguistics, 1998.

[7] Shun-ichi Amari and Hiroshi Nagaoka. *Methods of information geometry*, volume 191. American Mathematical Soc., 2007.

[8] Miles Osborne. Using maximum entropy for sentence extraction. In *Proceedings of the ACL-02 Workshop on Automatic Summarization-Volume 4*, pages 1–8. Association for Computational Linguistics, 2002.

[9] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[10] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[11] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into texts. Association for Computational Linguistics, 2004.

[12] Ani Nenkova. Summarization evaluation for text and speech: issues and approaches. In *INTERSPEECH*, 2006.

[13] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

[14] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8, 2004.