Analysis of Online Doctor Reviews and Ranking

Submitted by:
Harshaneel H. Gokhale
Instructed by:
Dr. Yu Cheng, Dr. Huan Liu
Fred Morstatter

Abstract

With the development of computing and the Internet, a lot of things are done online. People are using online systems to express their opinions. These opinions are targeted towards various subjects such as products, places or people. This gives us an opportunity to analyze these reviews and extract important information out of them. In this system we have analyzed reviews about the doctors posted by their patients using natural language processing techniques. We have also used Random Walk with Restarts algorithm to rank the doctors. The fundamental idea behind using Random Walk algorithm is that the Random Walk with Restarts favors the neighbors of the query node. The experiments in this project are mainly conducted on the dataset from IBM Watson, New York which consists around 1000 online reviews for around 200 doctors.

Keywords: Review Ranking, Sentiment analysis, Random Walk with Restarts.

1 Introduction

With the humongous amount of information available on the Internet it is now emerging as an important task is to give users what they want. As stated by Chris Anderson, "The secret to creating a thriving Long Tail business can be summarized in two imperatives: (1) Make everything available. (2) Help me find it."[1]. Any online service should be such that it shows the required information to users in the order of importance. This gives rise to the problem of ranking of the information available online. Search engines such as Google are very efficient in ranking web pages using PageRank algorithm and show it to the users; but it cannot rank the results for very specific problems such as ranking list of product reviews and show that list of product to the user. In this project we aim to solve this kind of specific problem of ranking doctors on the basis of the reviews posted by users.

With the advent of the Internet technology, we have a lot of users telling their experience about different things over the Internet. This makes data available to us for crunching and generating some important insights. The data that we have used for the project is from IBM Watson's RateMDs[2] web system which has provision for patients to express the experience about their doctors. While evaluating doctors, patients have different parameters to base their opinion upon. Those parameters may contain punctuality, knowledge, staff and overall performance. We have readily available scores for these meta-parameters that are given by users on the scale of 0 to 5. With these parameters we also have the textual review written by patients about the doctor. We can take help of natural language processing techniques to understand the polarity and the subjectivity of the opinions of patients and can use this as another feature.

In this report first, we will discuss about the fundamental technologies we have used. Then we will discuss about the structure of ranking system and methodology. At last, we will discuss about the results of the experiments we have performed.

2 VADER (Valence Aware Dictionary for sEntiment Reasoning)

VADER is a sentiment analysis system developed by Hutto, C.J. and Gilbert, E.E.[3] that helps extracting polarity and subjectivity of natural language text. While assigning polarity scores, VADER takes into consideration different slang terms and colloquial phrases. This makes it a useful concept to use. Reviews are not strictly speaking formal. While writing reviews we use a lot of colloquial terms to express our feelings. In such cases it is important to consider such terms into account while assigning polarity scores. VADER maintains a dictionary of commonly used slang terms with their impact.

There is another important feature of reviews. We generally enhance the polarity of the review by using words like very, very much, extremely and etc. For example, I like Dr. Smith! represents positive sentiment; but I like Dr. Smith very much! represents extremely positive sentiment. VADER maintains a dictionary of such enhancing words and increases or decreases the polarity of the sentence if such words are encountered. Figure 1 represents the entire system architecture of VADER system.

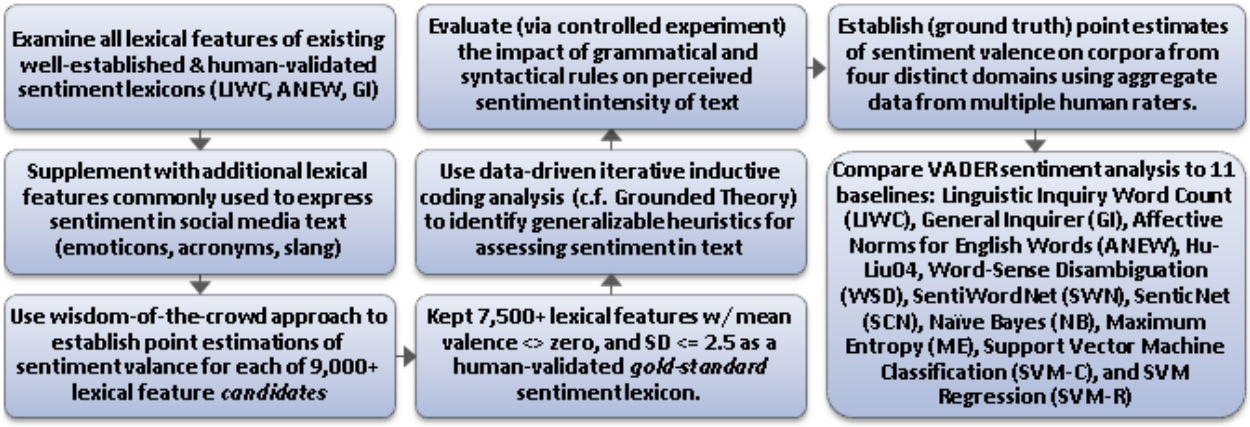

Figure 1: VADER System

3 Random Walk With Restarts

Random Walk with Restarts(RWR) is a very elegant algorithm that is used to rank the nodes of the graph with respect to the query node. Random walk, a series of random variables[4], was first proposed by Karl Pearson[5], has many useful use-cases. In fact, Google's very famous PageRank[6] algorithm is based on RWR.

Imagine random walk as a series of random variables T_i where, T_i is a connected vertex selection for each node of each step ([7], [8], [9]). At every step there is some probability associated with restarting from the original node. General form of RWR is given as,

$$r = c\bar{A}r + (1 - c)q$$

Here, $\bar{A}$ is a column normalized adjacency matrix of the graph, c is the restart probability, q is a query vector and r is the ranking list generated by this method. In closed form, RWR is given as,

$$r = (1 - c)(I - c\bar{A})^{-1}q$$

Generally, computing inverse of a matrix is a costly process for very large matrices. That is why recursive method is used for RWR. In recursive form RWR is given as,

$$\text{Until } r_{t+1} \text{ and } r_t \text{ converge, } r_{t+1} = c\bar{A}r_t + (1 - c)q;$$

Here, r_t is initialized to q and then final ranking is recursively computed. This is very efficient method and saves a lot of time even if matrices are huge.

4 Methodology

Entire process of generating ranking list of the doctors consists of two major steps. Figure 2 gives very high level information about how we can achieve the required goal.

Figure 2: System Overview

4.1 Pre-processing

In this data we have meta features such as doctors' knowledge, staff, punctuality and etc. Users have assigned their scores for these parameters on the scale of 0 to 5. We can use these features as they are. Additionally we have textual reviews from which we can extract sentiment of the user. We have created sentiment polarity using VADER[3] as a feature.

Meta-features have values on the scale of 0 to 5, whereas, sentiment polarity generated by VADER is between -1 to 1. The motivation is to use these feature values as weights of graph edges. So all the meta features are normalized to range 0 to 1 using- $(1.0 - (feature\_value/max\_value))$. In case of meta-features, maximum value is 5. We have also converted sentiment score to the range of 0 to 1. We have divided polarity scores by -0.5 which shrinks the range to -0.5 to 0.5 and it also flips the polarity. So now -0.5 represents extremely positive sentiment and 0.5 represents extremely negative sentiment. Then we add 0.5 to polarity so finally, we have polarity scores between 0 to 1 where 0 represents extremely positive sentiment and 1 represents extremely negative sentiment.

There are several reviews posted for one doctor. Our goal is to rank doctors but not just reviews. That is why, in pre-processing step, we have created groups of reviews posted for the same doctor. Later on, we will merge the scores of reviews for same doctor to identify its position in final ranking.

Since, RWR algorithm is for graphs, we have to convert our data to a graph structure. Simplest and the most intuitive way of doing this is by creating complete bipartite structure between features and the reviews and using normalized feature values as weights. Figure 3 shows the structure of the graph that we have created.

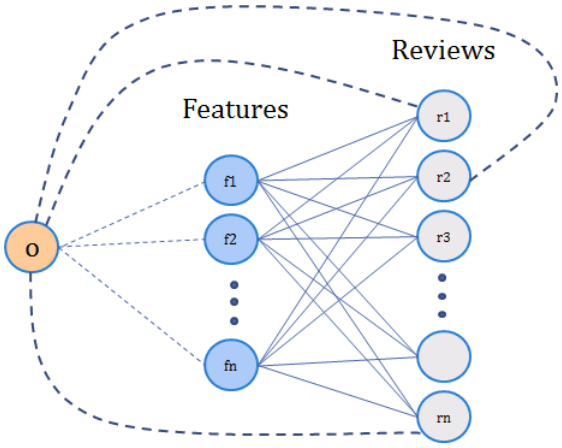

Figure 3: Generated graph for data

In this graph we have also added additional node labeled as 'O'. There are two reasons for adding this node in the graph. (1) RWR ranks neighboring nodes higher in the final ranking. We want to query this graph from all the features such that final ranking will have best reviews with respect to all the features simultaneously. So instead of creating query vector with non-zero values for all the features, we can connect all the features together to one node and query with respect to that node. This also allows us to add feature weights by having different weights for O to feature edges. (2) By adding weak links of small weights, having equal values, between reviews and node 'O', we avoid loss of information in the values of the features without changing rank of the matrix. For example, if Review_1 has feature vector as $[5, 5, 5, 5, 5]$ and Review_2 has feature vactor of $[1, 1, 1, 1, 1]$ then, after column normalization step of RWR, both the vectors will be normalized to $[0.5, 0.5, 0.5, 0.5, 0.5]$. At this point there is no way to distinguish between Review_1 and Review_2 to see which one is better. Addition of weak links between O and reviews with same weight will help avoiding this scenario.

4.2 Feature Importance

As mentioned in the previous section, adding node 'O' in the graph allows us to incorporate feature importance. In this case we will have to study how a particular feature account for the final ranking. To study that we can make use of entropy. Entropy of the data is given by- Entropy $= \sum p_i \log(p_i)$ where p_i is probability of occurrence of every unit in the data. Entropy essentially measures the randomness in the data. If data has more randomness then we can rank all the data points more reliably. For example, out of $A_1 = [1, 1, 2, 2]$ and $A_2 = [1, 2, 3, 4]$, we can distinguish and rank between elements using A_2 better than A_1. So we can say that, higher the entropy, more important is the feature from ranking perspective. We have measured the entropy of the features and assigned them as weights to the edges between 'O' and features.

4.3 Ranking

We have used RWR for ranking using the graph we have created as mentioned in previous sub-section. For RWR, query vector is generated by having entry 1.0 for node 'O' and zero for every other node. Using recursive method of RWR, we can achieve convergence fairly quickly as compared to closed form solution with matrix inverse computation.

4.4 Aggregation

Since we have many reviews for every doctors, we will have to aggregate results of RWR to prepare final ranking list. RWR algorithm assigns scores for every node. We have added scores of all the reviews for a particular doctor to assign final score to that doctor. Aggregation step also consists of normalization and we have normalized doctors' scores by dividing it by number of reviews of that particular doctor. If we do not perform this step then doctors with higher number of reviews is likely to be preferred over others in spite of being lower in the ranking list.

5 Testing and Results

The issue with testing metric is that we do not have ground truth; but we have meta-features which can give us the fair idea about what final ranking list might look like. We have performed same aggregation operation on the reviews using meta-features and then we have sorted the list with respect to the normalized features. Then we have tested results of RWR against synthesized ranking list using meta-features. We have used Spearman's ranking correlation coefficient to see how good the RWR algorithm results are. Spearman's ranking correlation coefficient is given by,

$$\rho = 1 - \frac{6 \sum {d_i}^2}{n(n^2 - 1)}$$

Here, d_i is absolute distance between ranks of the same element in two ranking lists and n is the number of element in the ranking list. We also have to perform the significance test to prove the significance of the results by RWR. It can be done using Fisher transformation and Z-score. Fisher transformation and Z-score is given by,

$$F(r) = \frac{1}{2} \ln \frac{1+r}{1-r} = arctanh(r)$$

$$Z = \sqrt{\frac{n-3}{1.06}} F(r)$$

Here, r is Spearman's ranking correlation coefficient, $F(r)$ is Fisher transformation and Z is Z-score. We can calculate two sided p-value using Z-score. If the p-value that we get is less than 0.01 then we can say that our ranking is significant. Figure 4 demonstrates the results of the experiments we have performed. It shows the performance of RWR for different restart probabilities against correlation coefficient. We can see that for restart probability = 0, ranking is very poor which is quite understandable because, without any restart, new

paths from different neighbors will never be discovered. As soon as we start having non zero restart probabilities, we get considerably good results. We can also see that when restart probability is 1 results are poor. That is mainly because, at every step, restart is performed and other important nodes are neglected.

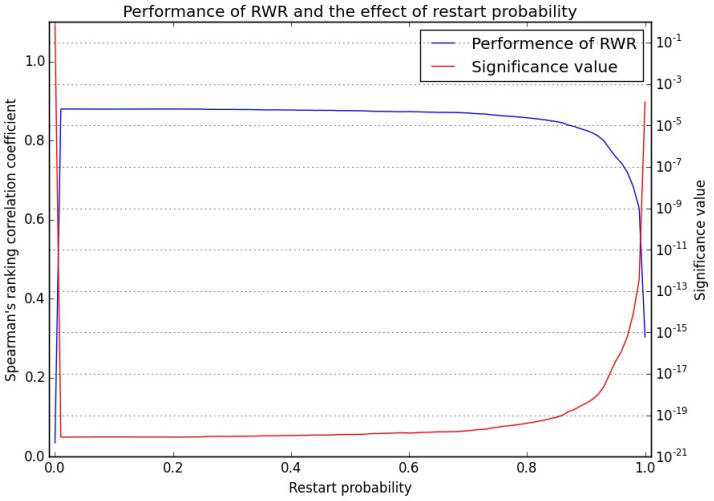

Figure 4: Results of RWR algorithm

Using this method the highest correlation coefficient we have got is 0.87 with significance value(p-value) of 1.97×10^{-21}. Since the p-value is less than 0.01 we can say that results generated by our method are significant.

6 Future Work

RWR has proved itself to be useful in various areas such as recommendation[10]. In this project we can make a provision such that ranking for a particular kind of doctors can be achieved and shown to the user. That will help user to find best doctors for a particular kind of disease or disorder. The core idea of random walk can be used for some other kind of problem like Twitter hashtag prediction where RWR can be used over a tripartite graph of users tweets and hashtags.

7 Conclusion

In the era of information overload, we are constantly looking for the methods that can selectively find out what we are looking for. In this project we have considered problem of ranking of doctors according to their performance. Our method using RWR helps us to achieve that using feature values. It generates reliable results which can make this approach as a baseline for the future work.

The use of this method is not just limited to ranking of doctors. It can be also used to rank products by using problem specific features. With the minor tweak in graph generation, the core concept can also be used for hashtag prediction tweets.

References

[1] C. Anderson, The Long Tail: Why the Future of Business Is Selling Less of More, ser. Hyperion Books. Hyperion, 2006.

[2] RateMDs: `https://www.ratemds.com`

[3] Hutto, C.J. & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014.

[4] C. M. Grinstead and J. L. Snell, Introduction to probability. American Mathematical Soc., 1998.

[5] K. Pearson, The problem of the random walk, Nature, vol. 72, no. 1865, pp. 294, 1905.

[6] L. Page, S. Brin, R. Motwani, and T. Winograd, The pagerank citation ranking: Bringing order to the web. Stanford InfoLab, Technical Report 1999-66, November 1999, previous number = SIDL WP-1999-0120.

[7] D. Aldous and J. fill, Reversible markov chains and random walks on graphs, 2002.

[8] L. Lovasz, Random walks on graphs: A survey, Combinatorics, Paul erdos is eighty, vol. 2, no. 1, pp. 1 46, 1993.

[9] L. Lovasz, Random walks on graphs: A survey, in Combinatorics, Paul Erdos is Eighty, D. Miklos, V. T. Sos, and T. Szonyi, Eds. Budapest: Janos Bolyai Mathematical Society, vol. 2, pp. 353398, 1996.

[10] S. Pongnumkul, K. Motohashi, Random Walk-based Recommendation with Restart using Social Information and Bayesian Transition Matrices in International Journal of Computer Applications (0975 8887) Volume 114 No. 9, March 2015.