An Opinion-aware Approach to Contextual Suggestion

Peilin Yang and Hui Fang

Department of Electrical and Computer Engineering
University of Delaware, USA
{franklyn,hfang}@udel.edu

Abstract. In this paper we describe our efforts for TREC Contextual Suggestion task. Our goal of this year is to evaluate the effectiveness of (1) an opinion-based method to model user profiles and rank candidate suggestions; and (2) a template-based summarization method that leverages the information from multiple resources to generate the description of a candidate suggestion. Existing approaches often uses the description or category information of the suggestions to estimate the user profile. However, one limitation of such approaches is that it may not be generalized well to other suggestions. To overcome this limitation, we propose to estimate the user profile based on the reviews of the candidate suggestions. Our preliminary results show that the proposed new method can improve the performance. Moreover, we explore a new way of generating summaries for the candidate suggestions. The main idea is that we want to leverage the information from multiple sources to generate a more comprehensive summary for a candidate suggestion. In particular, the summary includes the category information of the suggestion, meta-description of the website, reviews of the suggestion and the similar example suggestions that the user liked.

1 Introduction

In this year's Contextual Suggestion (CS) Track, we main aims are two folds: (1) to effectively model user profile [5] and (2) to explorer a new description generation method which combines multiple aspects of information.

For user profile modeling, we know from the example suggestions and the task user profile that which suggestions a user likes and dislikes, a straightforward solution is to estimate a positive user profile to model what the user likes and a negative one to model what the user dislikes [4]. Once we estimate such profiles for a user, we can then rank suggestions based on both their similarity to the positive user profile and their dissimilarity to the negative user profile. We propose a new method that models user profile based on reviews about the suggestions. It is obvious that the reviews given by the task users are not available for both example suggestions and candidate suggestions. Our method is designed to take advantage of the wisdom of the crowd. In particular, we leverage the on-line opinions posted by other real world users and use them to approximate the task user's opinion about the suggestion. If a user likes a suggestion, the positive profile of this user would be estimated based on the opinions about this suggestion from similar users, i.e., those who also like the place. The negative profile could be estimated in a similar way. We then explore two ways of estimating the profile based on the opinions. These two methods are mainly different in what kind of information from the opinions is used for profile modeling.

The description generation utilizes several sources of information to generate sentences template which cover multiple aspects of the suggestions. Category is firstly used to introduce the suggestion. The identified contents from suggestion's website are then used to show an "official" introduction. The highlighted reviews from real world users of the suggestion are picked by our review extraction algorithm and included in the description. Moreover, we add a last sentence to the description showing the example suggestions that the user liked and with the same category of the candidate suggestion.

2 Our Method

2.1 System Framework

Figure 1 shows our system framework. It mainly consists of three parts: (1) Useful information gathering, (2) User profile modeling, (3) Candidate suggestion ranking, (4) Description generation.

Useful information gathering component mainly crawls everything that we need to rank the candidate suggestion.

User profile modeling component basically models the user profile using reviews of example suggestions. The user profile is split into positive and negative. It also models the candidate suggestion profile in the similar way.

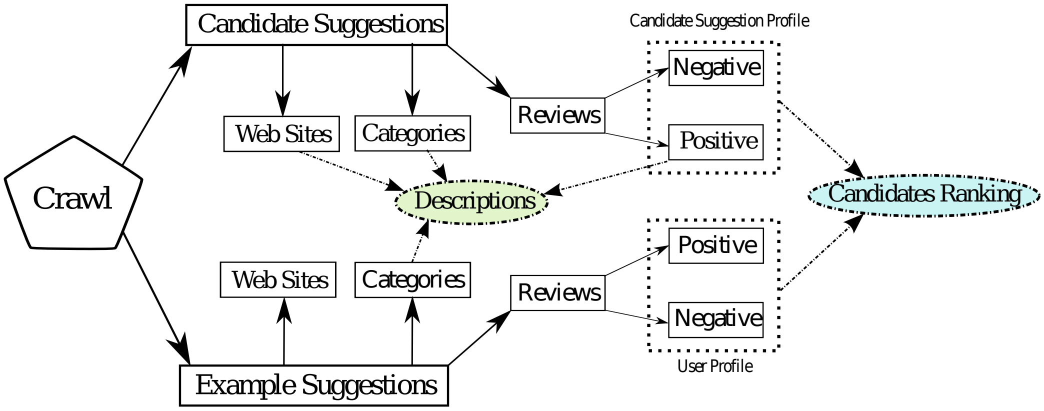

Fig. 1. System Framework

Candidate suggestions ranking is based on both user profile model and candidate suggestion profile model. We will introduce this component in more details later.

The description generation component mainly utilizes the positive candidate suggestion profile, categories and website of candidate suggestion, categories of some example suggestion to summarize a template-based description. We also describe this part in details later this paper.

2.2 Useful Information Gathering

The first step of solving the contextual suggestion problem is to gather useful information. Useful information contains not only the candidate suggestions, but also their web site, reviews associated with them and etc. For candidate suggestions we use open web as they contain comprehensive information than ClueWeb12 does. We crawl the information such as candidate suggestions, main page, categories and reviews of candidates exclusively from Yelp for all contexts. The crawling is done per category. We roughly choose 16 high-level categories that cover most appropriate categories used in the CS task. Approximately 105,871 candidate suggestions are crawled for all contexts, resulting in average 2117 candidate suggestions per context. The median number of candidate suggestions for contexts is 1276. The maximum and minimum numbers are 302 and 8410 respectively. Moreover, we also crawl the web site for each candidate suggestion with maximum pages of 100. The counterpart of example suggestions in Philadelphia is also crawled. The crawled data is stored mainly in json format and is used in both candidate suggestion ranking and description generation.

2.3 User Profile Modeling

Method We first describe a high level framework of our method: Given a user U and a candidate suggestion CS, our method operates as follows:

1. Build positive and negative user profile, i.e., $\mathcal{U}_{pos}$ and $\mathcal{U}_{neg}$, based on the information about example suggestions and the ratings that user gives, i.e, $ES(U)$;
2. Build a positive and negative profile for the candidate suggestion, i.e., $\mathcal{CS}_{pos}$ and $\mathcal{CS}_{neg}$;

A more detailed process of above method is described as below: We first use positive reviews of the example suggestions that the user likes to build positive user profile, and use negative reviews of example suggestions that the user dislikes to build negative user profile. The basic assumption here is that the opinion of a user about a place can be inferred by the opinions of the users who give the same rating (more accurately the similar rating since we merge some ratings into category of positive or negative) as the target user to the same place. For this year's TREC, the ratings of example suggestions ranges from 0 to 4 for both website and description. We define the example suggestion that user likes as the ratings of 3 or 4 on website. Ratings of 0 or 1 are viewed as those user dislikes. Ratings of 2, i.e. the very middle one, is simply ignored as we treat it as neutral rating. For candidate suggestions, there is a rating associated with each review. The rating ranges from 1 to 5 as scalar ratings. The reviews with rating of 4 or 5 are viewed as positive reviews and rating of 1 or 2 are viewed as negative reviews. Reviews with rating 3 are ignored for the same reason as mentioned before.

Formally, the user profiles are estimated as:

$$\mathcal{U}_{pos} = \bigcup_{es_i \in ES(U) \bigcap R_U(es_i)=POS} REP(O_{pos}(es_i))$$
$$\mathcal{U}_{neg} = \bigcup_{es_i \in ES(U) \bigcap R_U(es_i)=NEG} REP(O_{neg}(es_i))$$

where $O_{pos}(es_i)$ represents all positive reviews about es_i, $O_{neg}(es_i)$ represents all negative reviews, and $REP(O(es_i))$ denotes how to represent opinion $O(es_i)$ in the profile. We tried two different $REP(O(es_i))$ in our experiments.

- *Use unique terms from full reviews (UFR)*: Use all unique terms occurred in the positive reviews about a suggestion s to build its positive profile. Similarly, use all unique terms occurred in the negative reviews of the suggestion to build its negative profile.
- *Use review summaries (RS)*: Review summaries are the terms extracted from the reviews using Opinosis Algorithm[2].

We then follow the similar strategy and build the positive and negative profile for a candidate suggestion CS as follows:

$$\mathcal{CS}_{pos} = REP(O_{pos}(CS))$$
$$\mathcal{CS}_{neg} = REP(O_{neg}(CS)).$$

2.4 Candidate Suggestion Ranking

We estimate the similarity between a user and a candidate suggestion as follows:

$$\begin{aligned} S(U, CS) = \alpha \times SIM(\mathcal{U}_{pos}, \mathcal{CS}_{pos}) - \beta \times SIM(\mathcal{U}_{pos}, \mathcal{CS}_{neg}) \\ - \gamma \times SIM(\mathcal{U}_{neg}, \mathcal{CS}_{pos}) + \eta \times SIM(\mathcal{U}_{neg}, \mathcal{CS}_{neg}) \end{aligned} \tag{1}$$

where $SIM(a, b)$ is the text similarity measurement between a and b. In our experiments, we use F2-EXP of axiomatic retrieval model [1] as $SIM(a, b)$. α, β, γ and η are parameters that regulate the impact of the four components to the final similarity score. Their values are chosen in interval [0,1] with the pace of 0.1. Here the intuition is that the similarity score should be positively correlated with $\mathcal{U}_{pos}, \mathcal{CS}_{pos}$ and $\mathcal{U}_{neg}, \mathcal{CS}_{neg}$ and should be negatively correlated with $\mathcal{U}_{pos}, \mathcal{CS}_{neg}$ and $\mathcal{U}_{neg}, \mathcal{CS}_{pos}$. The basic assumption is that people like or dislike a suggestion for similar reasons and such reasons are included in the reviews.

Apparently, higher similarity score between a user and a candidate suggestion means higher ranking of that candidate suggestion. The similarity score are computed for each user/context pair and then we can get all the desired results.

2.5 Description Generation

We use a template to render the descriptions of candidate suggestions. The template basically consists of the following components:

1. An opening sentence.
2. The "official" introduction.
3. The highlighted reviews.
4. The concluding sentence.

The opening sentence is of the form: suggestion's name followed by the fine category of that suggestion. For example, "The big fish grocery is a *shopping store*." If the category of candidate suggestion is not available, we just show that "The xyz is a good place to go." We believe that the straightforward opening sentence makes the description more attractive and more readable.

Both "official" introduction and highlighted reviews depend on the extraction of aspects (features) of that candidate suggestion. This is done by first doing the part-of-speech tagging using Stanford Log-linear Part-Of-Speech Tagger [3] to the positive reviews of the candidate suggestion and then pick a set of nouns that appears

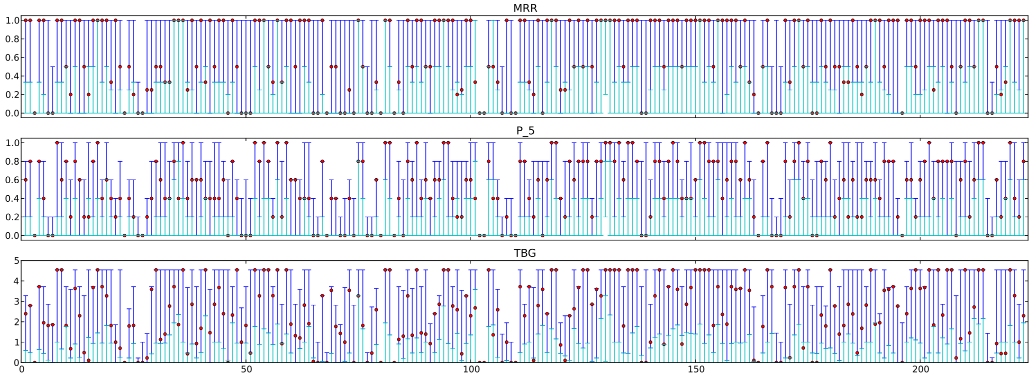

Fig. 2. Detailed Results of UDInfoCS1

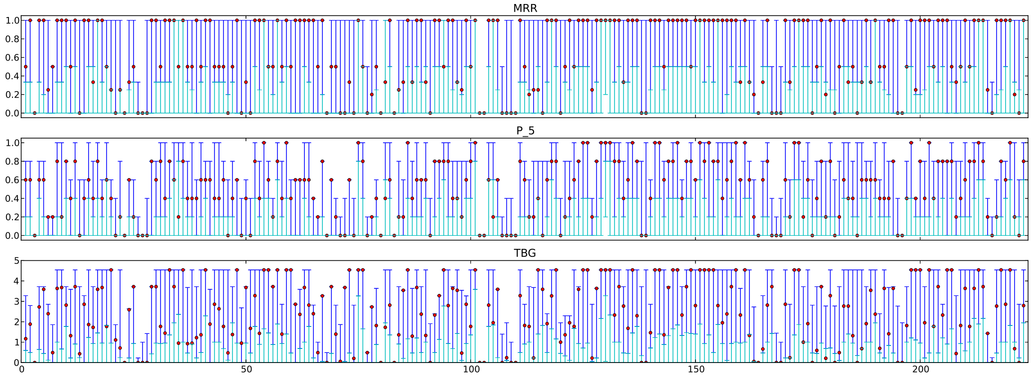

Fig. 3. Detailed Results of UDInfoCS2

most frequently. Apparently, terms in positive reviews have higher chance to be positive. We pick nouns since they are more prone to be real aspects.

After we extracted the most frequent nouns, we pick up the sentences from the web site of the candidate suggestion. We first manually generate a list of 391 positive adjectives such as "fabulous" and "wonderful". Sentences that contain larger number of positive adjectives are chosen as candidates. Sentences from web site's meta description field are also included as candidates. Finally we pick up at most 5 sentences as candidates due to the number of characters limits. We pick up highlighted reviews in the similar way as we pick up sentences from candidate suggestion's web site.

The concluding sentence is of the form: "We recommend this suggestion to you because you liked *abc* and *xyz* in example suggestions." *abc* and *xyz* are example suggestions that have the same fine category with the candidate suggestion.

The number of sentences of each component mentioned above are balanced in order to meet the requirement (less than 512 characters).

3 Submitted Runs and Experiment Results

We submitted two runs: UDInfoCS1 and UDInfoCS2. UDInfoCS1 models user profiles using RS. For description, it combines the opening sentence, meta-description and the picked sentences from the web site, highlighted reviews, and the concluding sentence. UDInfoCS2 models user profiles using UFR. For description, it combines the opening sentence, meta-description (no picked sentences here) of the web site, highlighted reviews, and the concluding sentence. For parameters in the similarity function, we tune them based on the cross-validation on example suggestions. The best parameter sets for both runs are $\alpha = 0.7, \beta = 0.0, \gamma = 0.2, \eta = 0.1$ (they are the same).

Table 1. Overall Mean Performances

	UDInfoCS1	UDInfoCS2
P@5	0.5094	0.4969
MRR	0.632	0.63
TBG	2.4474	2.431

As for judgments, 223 profile/context pairs were sampled and judged. There are four criteria: geographic appropriateness, interestingness of website/description, and time data (how long it took to arrive at the judgment in seconds). Geographic appropriateness has 3 scales: 0,1,2, with 0 not geographically appropriate, 1 marginally geographically appropriate, and 2 geographically appropriate. Interestingness of website/description has 5 scales: 0,1,2,3,4, with 0 strongly uninterested, 1 uninterested, 2 neutral, 3 interested, 4 strongly interested. Moreover, geographic appropriateness were judged partially by NIST accessors and partially by users. Scores from NIST accessors were used if a profile/context pair were judged by both of them.

Three evaluation measurements were reported: P@5 (precision at 5), MRR (mean reciprocal rank) and TBG (time-biased gain). For P@5 and MRR, a suggestion is labeled as relevant only if it has geographic score of 1 or 2, and description score of 3 or 4, and web site score of 3 or 4.

Figure 2 and Figure 3 show the performances of our two runs in terms of all evaluation measurements. The X axis consists of all profile/context pairs, ordered alphabetically in the format of “profile-context”. One red point represents our result for that profile/context. Three bars corresponding to the best, median and worst results of that profile/context. We can see that our runs achieve some of the best results. Most of our results are equal or better than the median results, indicating the effectiveness of our proposed system. Table 1 shows the overall mean performances of our runs in terms of all evaluation measurements.

4 Conclusion

In TREC 2013 Contextual Suggestion Track, we submitted two runs. For ranking the candidate suggestion of each profile/context, we apply the method that measures the similarity between user profile and candidate suggestions' profile. User profile is based on the positive and negative reviews of the example suggestion. Individual candidate suggestion profile is modeled in the similar way. The similarity is computed so that it is positively correlated with positive/positive negative/negative similarity between user profile and candidate suggestion profile and it is also negatively correlated with positive/negative negative/positive similarity between user profile and candidate suggestion profile. For description generation, we apply a template based summarization method. The template mainly contains an opening sentence, “official” introductions, highlighted reviews and a concluding sentence.

For our two runs, UDInfoCS1 uses review summary terms as the representation of the profiles while UDInfoCS2 uses unique terms as the representation of the profiles. For description generation, the only difference of UDInfoCS1 and UDInfoCS2 is that UDInfoCS1 contains the meta description and picked sentences from the website as “official” introduction while UDInfoCS2 only contains the meta description.

The performance of our two submitted runs are in general better than the median performance. Some of the results are even best results, indicating the effectiveness of our proposed method.

References

1. H. Fang and C. Zhai. An exploration of axiomatic approaches to information retrieval. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, pages 480–487, New York, NY, USA, 2005. ACM.
2. K. Ganesan, C. Zhai, and J. Han. Opinosis: a graph-based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 340–348, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.
3. K. Toutanova, D. Klein, C. D. Manning, and Y. Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 173–180, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.
4. P. Yang and H. Fang. An exploration of ranking-based strategy for contextual suggestion. In *Proceedings of 21st Text REtrieval Conference (TREC 2012)*. NIST, November 2012.
5. P. Yang and H. Fang. Opinion-based user profile modeling for contextual suggestions. In *Proceedings of the 2013 Conference on the Theory of Information Retrieval*, ICTIR '13, pages 18:80–18:83, New York, NY, USA, 2013. ACM.