

University of Amsterdam at TREC 2015: Contextual Suggestion Track

Seyyed Hadi Hashemi

Mostafa Dehghani

Jaap Kamps

University of Amsterdam, Amsterdam, The Netherlands

{hashemi|dehghani|kamps}@uva.nl

ABSTRACT

This paper presents the University of Amsterdam’s participation in the TREC 2015 Contextual Suggestion Track. Creating effective profiles for both users and contexts is the main key to build an effective contextual suggestion system. To address these issues, we investigate building models of users’ profiles for effective contextual personalization, and building language models of contextual features for effective contextual customization. Our main aim is to answer the questions: How to build a user-specific profiles that penalizes terms having high probability in negative language models? Can parsimonious language models improve user and context profile’s effectiveness? How to combine both models and benefit from both a contextual customization using the context language model and a contextual personalization using the users profile? Our main findings are the following: First, although using parsimonious language model leads to a more compact language model as users’ profile, the personalization performance is as good as using standard language models for building users’ profiles. Second, we extensively analyze effectiveness of three different approaches in taking the negative profiles in to account, which improves performance of contextual suggestion models that just uses positive profiles. Third, we learn an effective model for contextual customization and analyze the importance of different contexts in contextual suggestion task. Finally, we propose a linear combination of contextual customization and personalization, which improves the performance of contextual suggestion using either contextual customization or personalization based on all the common used IR metrics.

1. INTRODUCTION

In this paper, we present the University of Amsterdam participation in the TREC 2015 Contextual Suggestion Track. The main goal of this track is to investigate search techniques for complex information needs that are highly dependent on context and user interests. In each run, participants have to produce a ranked list of suggestions for each pair of profile and context.

Each *profile* corresponds to a user who has judged suggestions’ given in an specific context. The user profiles contain a five-point scale rating for each pair of profile and example suggestion. The *context* provided in TREC 2015 is richer than the context being used in previous years [1–3]. In TREC 2012, Contextual Suggestion organizers provided contexts having geographical and temporal aspects. However, judging temporal aspect of the context was difficult for

the NIST assessors, so it has not been used for the TREC 2013 and TREC 2014.

In TREC 2015, the contextual suggestion organizers provides a city the user is located in, a trip type, a trip duration, a type of group the person is travelling with, and the season the trip will occur in as contexts of the venue recommendation. Hopefully, almost all of the given contextual suggestion requests have information about all types of the mentioned contexts, which makes it a very interesting data to test contextual suggestion systems.

TREC 2015 Contextual Suggestion track’s setup is different from the previous tracks. In TREC 2013 and 2014, participants used to submit two different kinds of runs in the Contextual Suggestion Track: 1) open web, or 2) ClueWeb12. However, in TREC 2015, contextual suggestion organizers distributed a data collection, by running contextual suggestion pre-TREC task, among the participants.

In addition, in TREC 2015, participants were allowed to participate in contextual suggestion live or batch experiments. In the live experiment, participants return a list of attraction IDs in the TREC 2015 contextual suggestion collection, but in the batch experiment, participant rank attraction IDs that have been suggested during the live experiment. In this paper, we discuss our participation in the batch experiment that helps us to test our contextual suggestion approach.

In this paper, our main aim is to study the question: *How to build effective language models of both users’ and contexts’ profiles for improving contextual suggestion?* Specifically, we answer the following research questions:

1. *How to build an effective user profile for suggestion candidates personalization?*
 - (a) *Is parsimonious language model based personalization able to improve baseline personalization model in Contextual Suggestion?*
 - (b) *Does considering negative profiles improve baseline contextual suggestion using positive profiles?*
2. *What is the effect of using parsimonious language model based contextual customization in overall performance of contextual suggestion?*
 - (a) *How effective is doing contextual customization for contextual suggestion?*
 - (b) *What is the most important context among the contexts being used in TREC Contextual Suggestion Track?*

- (c) *What is the effect of using a linear combination of contextual customization and contextual personalization for the contextual suggestion?*

The rest of this paper is organized as follows. In Section 2 we review some related work on Contextual Suggestion track. In Section 3, we detail our models of Contextual Suggestion, and Section 4 is devoted to the experimental results. Finally, we present the conclusions and future work in Section 5.

2. RELATED WORK

In the TREC 2012 Contextual Suggestion Track, participants were allowed to use the open web to retrieve suggestion candidates. All of them used the webpages of the aggregator websites such as Yelp, Google Places, Foursquare and Trip Advisor. A considerable fraction of the participants used category of suggestion candidates that is available in the Yelp website. In that track, the given context had geographical and temporal aspects.

In the TREC 2013 and TREC 2014, the participants could use either the open web or the ClueWeb12 dataset, but there were only seven submitted runs out of 34 in 2013 and 6 out of 31 in 2014 that were ClueWeb12 runs [2]. The common approach of the open web runs were retrieving a bag of relevant venues to the given context based on the aggregators’ API such as Yelp API, and then re-rank the suggestion candidates based on the user profiles and/or the suggestion categories.

As the most related work, in TREC 2014, University of Amsterdam experimented with the use of anchor text representations in the language modeling framework, and they analyzed effects of using positive, neutral, and negative profiles in personalization of the suggestion candidates [4]. However, the TREC Contextual Suggestion test collection was not reusable [5], and they could not test different types of ratings of example suggestions. In this paper, we analyze our proposed approach in using both positive and negative profiles in personalization and customization of suggestion candidates.

3. APPROACH

Our contribution in Contextual Suggestion Track has two main parts. One part is personalization of suggestions candidates using users’ profiles, which includes just users’ preferences in rating example suggestions. The other part is customization of suggestion candidates based on the given contextual information like trip duration and type, and users’ profile including their age and gender, but not their preferences in rating example suggestions.

3.1 Parsimonious language models

In our proposed models in contextual personalization and customization, we have used parsimonious language models. Therefore, in this section, we first detail parsimonious language models.

Parsimonious language model is introduced by [6] in the context of information retrieval which aims to make a compact and precise estimation. In parsimonious language model, given a raw probabilistic estimation, i.e. standard language model, the goal is to reestimate the model so that non-essential terms of the raw estimation are eliminated with regards to a background language model.

Generally, in parsimonious language model, probability of terms that are common in the background estimation are pushed to zero. In other words, parsimonious language model represent a document with its terms which make it distinguishable from other documents in the collection by penalizing raw inference of terms that are better explained by the background estimation.

In order to achieve this, an Expectation-Maximization algorithm is employed to reestimate the language model. Consider t determines a possible term and $\tilde{\theta}_d$ is a parsimonious language model of document d . The probability of each term t given parsimonious language model $\tilde{\theta}_d$, $P(t|\tilde{\theta}_d)$ is calculated by iterating through the following steps:

$$E - step : e_t = tf_{t,d} \cdot \frac{\alpha P(t|\tilde{\theta}_d)}{\alpha P(t|\tilde{\theta}_d) + (1 - \alpha)P(t|\theta_C)} \quad (1)$$

$$M - step : p(t|\tilde{\theta}_d) = \frac{e_t}{\sum_{t' \in V} e_{t'}} \quad (2)$$

where, V is the set of all terms with non-zero probability in the initial estimation and θ_C shows the collection language model as background estimation. In the initial E-step, $P(t|\theta_C)$ is estimated using maximum likelihood estimation.

In the E-step, terms with high probability in the document language model having relatively high probability in the background language model are penalized, and continuing iteration, their probability adjusted toward zero. This way, terms which do not explain the document model in a specific way are discarded. After each M-step, terms with probability below a predefined threshold are eliminated. α in Equation 2 controls level of parsimonization so that the lower values of α result in a more parsimonious language models. The iteration is repeated a fixed number of iterations or until the estimates do not change significantly anymore.

3.2 Personalization using PLM

In this section, we propose how to apply parsimonious language models (PLM) in contextual suggestion as a personalization problem. To this aim, we have used parsimonious language model for building a user’s profile in order to estimate relevance of suggestion candidates to the given user. Then, we estimate relevance of each suggestion candidate to the given user by KL-divergence of the user’s PLM and standard language model of the suggestion candidate.

PLM1.

We first detail our first submission in TREC 2015 (i.e., PLM1), in which we have used parsimonious language model in order to build more specific and smaller positive profiles for users.

To this aim, example suggestions being rated higher than 2 are considered as evidences to build positive profiles as mixture language model of their positive preferences using rates as the mixture weight. At the final step of this model, parsimonious language model of the positive mixture LM is built by doing 100 iterations.

The same thing is done on tags assigned to each example suggestion by users, instead of terms in the corresponding web pages. Therefore, we have two users’ positive profiles, which are based on either rated web pages’ contents or tags given by users, for estimation of the final relevance of a sug-

Table 1: Discrimination of users’ age to 6 age range groups

range group	age range
<i>range0</i>	[0,18)
<i>range1</i>	[18,26)
<i>range2</i>	[26,34)
<i>range3</i>	[34,42)
<i>range4</i>	[42,50)
<i>range5</i>	[50,∞)

gestion candidate to a user. We simply calculate the final score as the linear combination of the relevance of the suggestion candidate to the profile based on web pages’ contents (i.e., $score_{webpages}$) and the profile based on users’ given tags (i.e., $score_{tags}$):

$$score_{final} = \alpha score_{webpages} + \beta score_{tags}.$$

In our experiments, we have heuristically considered $\alpha = 1$ and $\beta = 2$.

PLM2.

In PLM2, we tried to build even more specific language models for the positive profiles of users in comparison to the PLM1. To this aim, we have built parsimonious language model of the PLM1 using negative language model as a background language model. Therefore, we penalized terms with the high probability in both the collection language model and the negative profile language model.

The same thing is done on tags assigned to each example suggestion by users, instead of terms in the rated web pages. Same as PLM1, the final score of PLM2 is calculated by the linear combination of the relevance of the suggestion candidate to the profile based on web pages’ contents and the profile based on users’ given tags.

3.3 Customization using PLM

In this section, we detail our proposed contextual customization approach using parsimonious language models of different contexts. TREC 2015 contextual suggestion track provides a reach set of contextual information for almost all of the users’ requests, which makes it possible to test our approach in this test collection.

In order to customize suggestion candidates, parsimonious language models of each class of given contexts are built prior to doing customization for each request. Specifically, PLM of different group of people (e.g., Family), season (e.g., Spring), trip type (e.g., Holiday), trip duration (e.g., Weekend), person age (i.e., discrimination of age to 6 different ranges, which is shown in Table 1), and person gender (e.g., Female) are built. Then, in each request, we estimate relevance of the given suggestion candidates to the different given contexts by calculating the KL-divergence of the standard language models of suggestion candidates and the PLM of different contexts that was built in advance.

In order to have the most effective contextual customization using the best linear combination of contextual relevance of different contexts to the given suggestion candidate, the pairwise SVM rank learning to rank model is used [7]. According to users’ preferences in rating example suggestions, it is straight forward to assume rank preferences of example suggestions for the given request.

In order to do a pairwise learning to rank for contextual customization, we represent each suggestion s_i for the given contextual request r_j by a feature vector v_{ij} . Learning in this pairwise approach is finding a vector \vec{w} such that the maximum number of rank preference constraints are satisfied on the given training set. Specifically, for each satisfied constraint:

$$\{(s_1, s_2) | s_1 \in S(r_j), s_2 \in S(r_j), P_{r_j}(s_1) > P_{r_j}(s_2)\} \iff \vec{w}v_{1j} > \vec{w}v_{2j},$$

in which, $S(r_j)$ is a set of example suggestion rated for the given request r_j , and P_{r_j} is a set of preference ratings for the given request r_j . We have used SVM rank algorithm to estimate the optimal value of \vec{w} . By having an optimal \vec{w} , we use $\vec{w}v_{ij}$ as estimation of the relevance of a suggestion candidate s_i to the given request r_j .

3.4 Linear Combination of Suggestion Personalization and Customization

In order to have a contextual suggestion model, which benefit from both contextual customization and personalization, a linear combination of them is used to fuse the contextual personalization with the contextual customization suggestion ranking. To this aim, the following equation is used to combine the two rankings:

$$p(s|r, \beta) = \beta p_{Personalization}(s|r) + (1 - \beta) p_{Customization}(s|r),$$

where $p(s|r, \beta)$ is the relevance probability of a suggestion s to a request r based on the weight β and $1 - \beta$ given to *Personalization* and *Customization* rankings. In this paper, we have used the reciprocal rank of suggestions for estimating their relevance probability to the given request. The performance of this model and the optimal β parameter is discussed in Section 4.

4. EXPERIMENTS

In this section, we mainly answer our research questions. In fact, we answer the research question: *How to build effective language models of both users’ and contexts’ profiles for improving contextual suggestion?*

4.1 Personalization

This section studies our first research question: *How to build an effective user profile for suggestion candidates personalization?* Building effective profiles for the given users is one of the key steps of doing content-based recommendation.

4.1.1 PLM based Personalization vs. SLM based Personalization

We first look at the question: *Is parsimonious language model based personalization able to improve baseline personalization model in Contextual Suggestion?* As it is mentioned in Section 3, we proposed using parsimonious language model to built users’ specific language models to have a more compact and as effective users profiles as using the standard language model for giving fast effective suggestions for the given request.

Table 2 shows that using parsimonious language model considering the collection language model as a background language model (PLM1) is performing almost same as the baseline contextual suggestion using standard language model of users’ positive preferences (i.e., LM_P). However, the

Table 2: Effects of using parsimonious language model in modeling users’ profiles

Method	p@5	MRR	MAP
LM_P	0.5194	0.6839	0.5498
$PLM1$	0.5204	0.6765	0.5523
PLM_N	0.5270	0.7174	0.5619

model using parsimonious language model of a user, considering the user negative profile as a background language model, performs better than the other two models, and improves the baseline. Specifically, it improves the baseline % 2 in terms of $P@5$ and % 5 in terms of MRR . This experiment also shows that negative users’ preferences are an important source of information that should be used in contextual suggestion problem.

4.1.2 Impact of Using Negative Profiles

This section answers our research question: *Does considering negative profiles improve baseline contextual suggestion using positive profiles?*

To this aim, we have used 3 different approaches to take the negative users preferences in to account in doing contextual suggestion. The first one is the $PLM2$, in which we have built parsimonious language model of $PLM1$ by considering negative profiles’ language models as background language models. This approach makes very specific and compact language models of users positive preferences.

The second approach for considering negative preferences in the personalization step is building a parsimonious language model of users’ positive preferences by considering negative profiles’ language models as background language models. This time, we do not build parsimonious language model by considering both collection language model and negative profiles’ language model as background language models, but by just considering negative profiles’ language model as background language models. This model is less user specific in comparison to the first model and we have tried to just penalize terms in the negative users’ profiles, but not the terms in the neutral users’ profiles.

The third model (LM_N) is based on using standard language models for both positive and negative users’ profiles, and simply using the following linear combination for computing the final suggestions’ relevance to the given request:

$$score_{final}(s|r) = score_{positive}(s|r) - score_{negative}(s|r),$$

in which, $score_{positive}(s|r)$ and $score_{negative}(s|r)$ are the relevance score of a suggestion candidate s to the given positive and negative profile of a request r .

According to Table 3, the last model (i.e., LM_N), which uses larger language models to compute the relevance of suggestion candidates to the given request and does not use parsimonious language models for building users’ profiles, is performing better than the other two approaches in terms of $P@5$ and MAP . However, in terms of MRR , the PLM_N , which does not penalize terms in the neutral profiles but penalize terms in the negative profiles, is performing better than the other two approaches.

4.2 Customization

This section studies the impact of contextual suggestion customization on the overall performance of contextual sug-

Table 3: Effects of using negative profiles in building users’ profiles

Method	p@5	MRR	MAP
LM_P	0.5194	0.6839	0.5498
$PLM2$	0.5024	0.4929	0.5483
PLM_N	0.5270	0.7174	0.5619
LM_N	0.5526	0.7031	0.5684

Table 4: Contextual customization vs. personalization

Method	p@5	MRR	MAP
LM_P	0.5298	0.6866	0.5575
PLM_C	0.5277	0.7004	0.5642

gestion, trying to answer our second research question: *What is the effect of using parsimonious language model based contextual customization in overall performance of contextual suggestion?*

4.2.1 PLM based Customization

We first look at the question: *How effective is doing contextual customization for contextual suggestion?* To this aim, we have filtered those requests, which do not have all the contextual information defined in the TREC contextual suggestion track. Therefore, experiments of this section is done based on 191 out of 211 requests provided by the track organizers. In order to build precise profiles for different classes of different contexts, we have used parsimonious language model. In fact, unlike the users positive profile, which include just less than or equal to 30 rated web pages, contexts’ profiles are made of much more web pages, which makes it more reasonable to use parsimonious language model for making a more compact language models having less terms from other contexts.

As it is shown in Table 4, although we do not use user specific ratings, the PLM based customization (PLM_C) is performing better than the baseline personalization based on standard language models of positive profiles (i.e., LM_P) in ranking suggestion candidates.

This experiment indicates that the contextual information is a very important source of information that can be used to customize suggestions for different group of users.

4.2.2 Context Importance

We second look at the question: *What is the most important context among the contexts being used in TREC Contextual Suggestion Track?* As we mentioned in Section 3, we have learnt a \vec{w} weight vector for an optimal contextual customization of the suggestion candidates. Based on the learnt weight vector, this is very interesting to know that the trip duration is the most important context in the contextual customization of the suggestions, and the least important one among the provided contexts is the trip type. Figure 2 demonstrates the relative importance of different contexts being examined in this track.

4.2.3 Linear Combination of Customization and Personalization

We now look at the question: *What is the effect of using a linear combination of contextual customization and contex-*

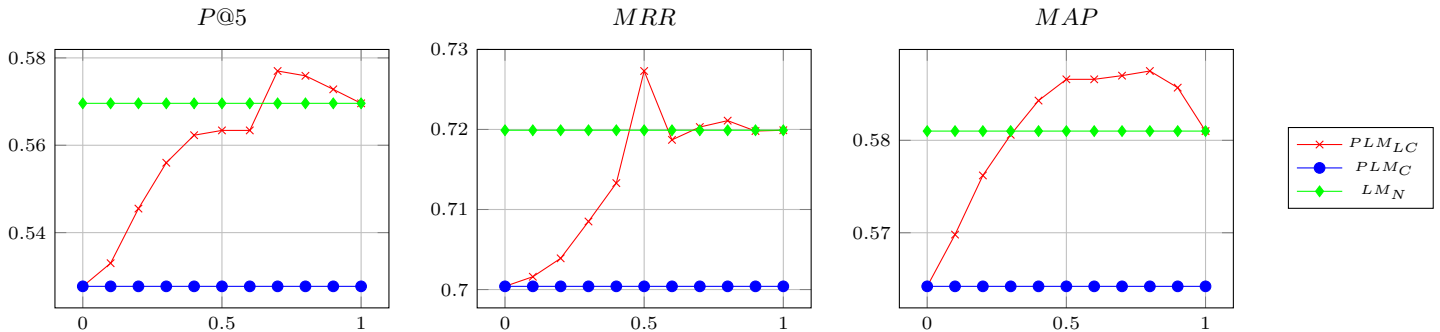


Figure 1: Impact of relative fusion weight (β) on $P@5$, MRR , and MAP .

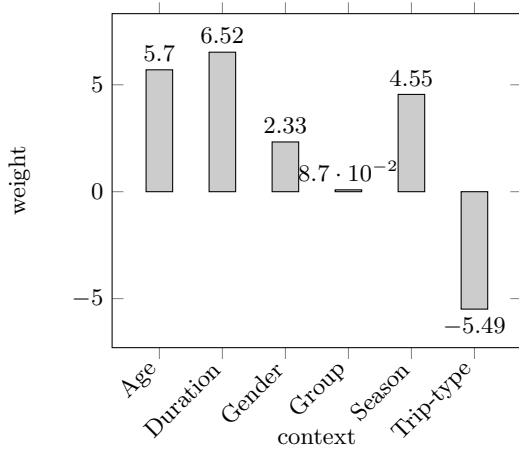


Figure 2: Contexts importance in the TREC 2015 contextual suggestion based on the relative contexts weights being learnt for contextual customization.

tual personalization for the contextual suggestion? As the contextual customization is using a different layer of suggestion candidates personalization based on the example suggestions rated by similar requests, and on the other hand, the proposed personalization approaches just use users' preferences but not other provided contexts of the requests, combination of these two suggestions rank list can improve the overall performance of the contextual suggestion.

To this aim, as we mentioned in Section 3, we have used a linear combination of the customization and personalization approach, whose optimal parameter is being analyzed in Figure 1. In this experiment, we have used our best personalized contextual suggestion ranker (i.e. LM_N) as a personalized ranked list, and fuse it by our proposed contextual customization ranked list (i.e., PLM_C). Figure 1 shows that a linear combination of the customized ranking and the personalized one is performing better than our best contextual personalization approach in all the common IR metrics being used in this paper. Specifically, considering $\beta = 0.7$, PLM_{LC} , whose $P@5$ is 0.5770, is our best contextual suggestion run in term of $P@5$.

5. CONCLUSION AND FUTURE WORK

In this paper, we studied Contextual Suggestion problem through building effective language model based profiles for both users and contexts. According to our experiments in doing contextual personalization, although parsimonious language models does not help much in improving baseline contextual suggestion ranking, it provides a more compact profiles, faster personalization approach with an acceptable effectiveness in personalizing search result. Moreover, we discussed three different contextual suggestion approaches taking negative users' profiles into account, and conclude that the one using standard language models and simply subtracting suggestion candidates' relevance to the negative profiles from the positive profiles, performs better than the ones using parsimonious language models to penalize terms in negative profiles. In addition, we proposed a contextual customization approach using parsimonious language model to build contexts profiles. We learnt an optimal weights of suggestion candidates relevance to the contexts by using pairwise SVM learning to rank model. We observed that the trip duration is the most important context in estimating the relevance of suggestion candidates to the given requests. The contextual customization performs almost as effective as the contextual personalization, which shows that the contextual customization is quite promising in solving cold-start problem in the contextual suggestion task. Finally, we have analyzed that the linear combination of the contextual customization and personalization performs better than the others by having $P@5 = 0.5770$ for the $\beta = 0.7$. As a future work, we continue to work on building a more effective contextual customization approach by clustering contextual requests.

Acknowledgments

This research is funded in part by the European Community's FP7 (project meSch, grant # 600851) and the Netherlands Organization for Scientific Research (ExPoSe project, NWO CI # 314.99.108; DiLiPaD project, NWO Digging into Data # 600.006.014).

REFERENCES

- [1] A. Dean-Hall, C. L. A. Clarke, J. Kamps, P. Thomas, and E. Voorhees. Overview of the TREC 2012 contextual suggestion track. In *Proceedings of TREC 2012*, 2012.

- [2] A. Dean-Hall, C. L. A. Clarke, J. Kamps, P. Thomas, N. Simone, and E. Voorhees. Overview of the TREC 2013 contextual suggestion track. In *Proceedings of TREC 2013*, 2013.
- [3] A. Dean-Hall, C. L. A. Clarke, J. Kamps, P. Thomas, and E. Voorhees. Overview of the TREC 2014 contextual suggestion track. In *Proceedings of TREC 2014*, 2014.
- [4] S. H. Hashemi and J. Kamps. Venue recommendation and web search based on anchor text. In *23rd Text REtrieval Conference (TREC)*.
- [5] S. H. Hashemi, C. L. Clarke, A. Dean-Hall, J. Kamps, and J. Kiseleva. On the reusability of open test collections. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 827–830, 2015.
- [6] D. Hiemstra, S. Robertson, and H. Zaragoza. Parsimonious language models for information retrieval. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 178–185, 2004.
- [7] T. Joachims. Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 217–226, 2006.