# Neural Endorsement Based Contextual Suggestion

Seyyed Hadi Hashemi[1] Nawal Ould Amer[2] Jaap Kamps[1]

[1]University of Amsterdam, Amsterdam, The Netherlands, {hashemi|kamps}@uva.nl
[2]University of Grenoble, Grenoble, France, nawal.ould-amer@imag.fr

## ABSTRACT

This paper presents the University of Amsterdam's participation in the TREC 2016 Contextual Suggestion Track. In this research, we have studied a personallized neural document language modeling and a neural category preference modeling for contextual suggestion using available endorsements in TREC 2016 contextual suggestion track phase 2 requests. Specifically, our main aim is to answer the questions: How to model users' profiles by using the suggestions' endorsements as an additional data? How effective is using word embeddings to boost terms' weights relevant to the given endorsements? How to model users' attraction-category preferences? How effective is using deep neural networks to learn users' category preferences in contextual suggestion task? Our main findings are the following: First, the neural personalized document based user profiling using word embeddings improves the baseline content-based filtering approach based on all the common IR measures including TREC 2016 Contextual Suggestion official metric (NDCG@5). Second, neural users' category preference modeling beats both baseline content-based filtering and the user profiling model using word-embeddings in terms of all the common IR measures.

## 1. INTRODUCTION

In this paper, we present the University of Amsterdam participation in the TREC 2016 Contextual Suggestion Track [9]. 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 a specific context. The user profiles contain a five-point scale rating for each pair of profile and example suggestion. The *context* provided in TREC 2016 is similar to the context being used in the TREC 2015 Contextual Suggestion Track, and it consists of a city name which represents which city the trip will occur in and several pieces of data about the trip.

In particular, in TREC 2016, the contextual suggestion organizers provide a city the trip will occur 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 task. 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.

Moreover, TREC 2016 contextual suggestion track organizers released related tags of each attraction in the qrels file. These endorsements in a way classify suggestions, which is potentially good source of information to improve users' preference modeling. In this paper, we mainly focus on how to build and use the tag preference model in order to build contextual suggestion systems.

In TREC 2016, contextual suggestion track organizers distributed the TREC contextual suggestion web corpus, which is a web archive of the released TREC Contextual Suggestion data collection being used in both TREC 2015 and 2016 [2, 9]. In this study, we have indexed and used the released corpus as a dataset in our experiments.

TREC 2016 contextual suggestion allowed participants to participate in the contextual suggestion phase 1 or phase 2 experiments. In the phase 1 experiment, participants return a list of attraction IDs from the TREC 2016 contextual suggestion collection, but in the phase 2 experiment, participant rank attraction IDs that have been suggested during the phase 1 experiment. In this paper, we discuss our participation in the phase 2 experiment.

In this paper, our main aim is to study the following research question:

1. *How to effectively model users' profiles using neural language modeling?*

2. *How to effectively learn users' preferences using neural category preference modeling?*

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 TREC contextual suggestion results. Finally, we present the conclusions 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

**Algorithm 1** Estimating a Personalized Document LM

**procedure** PDLM$(d, TG_u(d))$
  **for each** $t \in d$ **do**
    $P(t|\theta_d) = \frac{tf(t,d)}{|d|}$
    $P(t|\theta_{d_u}) = \frac{1}{|TG_u(d)|} \sum_{tg \in TG_u(d)} P(t|\theta_d) P(t|tg)$
  **end for**
**end procedure**

out of 31 in 2014 that were ClueWeb12 runs [1]. 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 and 2015, University of Amsterdam experimented with a content based filtering approach using the language modeling framework. In TREC 2014, 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, 7, 8], and they could not test different types of ratings of example suggestions. In TREC2015, they were able to analyze their proposed approach in using both positive and negative profiles in personalization and customization of suggestion candidates by participating in the batch experiment [6]. In this study, we use suggestions' endorsements in creating more effective profiles in the language modelling framework. Moreover, we have done neural category preference modeling for the contextual suggestion task, which has not been done before.

# 3. APPROACH

In this section, we detail two different approaches used in university of Amsterdam (UAmsterdam) submissions. Specifically, in "UAmsterdamCB" submission, we have used a neural language modeling approach to build users' profiles and use it in a content based filtering model. Moreover, we have learned a deep multilayer perceptron to learn category preference of users and used it in the "UAmsterdamDL" submission.

## 3.1 User Profiling Using Word Embeddings

This section studies how to effectively model users' profiles to be used in the content based filtering systems, aiming to answer our first research question: *How to effectively model users' profiles using neural language modeling?*

### 3.1.1 Personalized Document Language Model

In this part, we present how to personalize a document model using user tags. The goal is to estimate a best personalized term distribution for the document according to the tags assigned by a user.

Our approach is shown in Algorithm 1. Given a document $d = \{t_1, t_2, ..., t_n\}$ and his related tags $TG_u(d) = \{tg_1, tg_2, ..., tg_m\}$ assigned by a user u. We first estimate a document model $\theta_d$ as first estimation using maximum likelihood as follow:

$$P(t|\theta_d) = \frac{tf(t,d)}{|d|} \tag{1}$$

**Algorithm 2** Estimating a User Model

**Require:**
  $D_u = \{d_1, d_2, .., d_N\}$ Document preference of user $u$.
  $V_u = \{t_1, t_2, .., t_M\}$ User vocabulary.
**Ensure:**
  $\theta_u$ User Model.
**for each** $d \in D_u$ **do**
  $\theta_{d_u} \leftarrow PDLM(d, TG_u(d))$
**end for**
**for each** $t \in V_u$ **do**
  **for each** $d \in D_u$ **do**
    $P(t|\theta_u) = \frac{1}{|D_u|} \sum_{d \in D_u} P(t|\theta_{d_u})$
  **end for**
**end for**=0

where $tf(t, d)$ is a frequency of term $t$ in the document $d$, and $|d|$ is a document length.

Then, we estimate the personalized document model $\theta_{d_u}$ using tags $TG_u(d)$ as follow:

$$P(t|\theta_{d_u}) = \frac{1}{|TG_u(d)|} \sum_{tg \in TG_u(d)} P(t|\theta_d) P(t|tg) \tag{2}$$

where $P(t|\theta_d)$ is the probability of term in a document as described in Eq.1, $P(t|tg)$ is a probability of selecting a term t given a tag $tg$, and $|TGu(d)|$ number of tag assigned to the document $d$ by a user $u$.

The probability $P(t|tg)$ is computed using the cosine similarities between the two embedded vectors corresponding to term $t$ and tag $tg$ as follow:

$$P(t|tg) = sim(t, tg) \tag{3}$$

### 3.1.2 Constructing User Profiles

We build a user profile based on the user's endorsed documents, following the approach shown in Algorithm 2. Let $Du = \{d1, d2, ..., dn\}$ a set of endorsed documents of user $u$, and $Vu = \cup_{d \in Du} = t1, t2, ..., tp$ a user term vocabulary over his documents. Each document $d$ in $D_u$ is estimated as described in the previous section 3.1.1.

We estimate a user document model denoted $\theta_u$ as raw probabilistic estimation for each term in a user vocabulary. Then for each term t in user vocabulary, we estimate its probability as follow:

$$P(t|\theta_u) = \frac{1}{|D_u|} \sum_{d \in |D_u|} P(t|\theta_{d_u}) \tag{4}$$

where $P(t|\theta_{d_u})$ probability of the term $t$ given a personalized document model 3.1.1.

At last, we have used the created user profiles, based on the personalized document language model, in a content-based filtering engine. Specifically, we have used the KL-divergence of the users' profiles and the suggestion candidates' profiles using standard language modeling as the final score of the "UAmsterdamCB" submission.

## 3.2 Neural Category Preference Modelling

This section studies how to predict relevant attractions to the given user and context using category preference models, aiming to answer our second research question: *How to effectively learn users' preferences using neural category preference modeling?*

Table 1: TREC 2016 Contextual Suggestion Track: Phase 2 results.

| RunID | NDCG@5 | P@5 | MRR | NDCG | MAP | bpref | P@10 | Rprec |
|---|---|---|---|---|---|---|---|---|
| Baseline | 0.1967 | 0.2862 | 0.4440 | 0.6257 | 0.3862 | 0.4332 | 0.3086 | 0.3551 |
| UAmsterdamCB | 0.2730 | 0.4069 | 0.5631 | 0.6499 | 0.4076 | 0.4337 | 0.4000 | 0.3780 |
| UAmsterdamDL | **0.2824** | **0.4448** | **0.5924** | **0.6544** | **0.4168** | **0.4452** | **0.4310** | **0.3881** |

In order to model the contextual suggestion, we cast the context-aware recommendation problem to a binary classification problem, in which relevant suggestions in the users' profiles are labeled 1 and irrelevant ones labeled 0. In this way, we try to learn a model to predict relevant suggestion candidates to the given user profile and the context by the help of users' category preferences. Then, relevance probability of suggestion candidates to the user and context pairs will be used to rank the phase 2 suggestion candidates.

In order to learn the model, a set of features that represent how relevant is each suggestion to each category defined. To this aim, we have created a profile of each given category in the TREC 2016 contextual suggestion requests. Then, we have considered content-based relevance of each category profile to the suggestion as a feature in our both train and test sets. We have found 123 unique categories in the phase 2 requests in total. Therefore, we have defined KL-divergences of 123 category profiles to a suggestion profile as 123 different features for the relevance estimation of the suggestion based on the category preferences.

In order to learn a user preference model, we have used a deep neural network with 4 hidden layers having 478 units. To learn an effective model and avoid over-fitting, we have used a dropout feedforward neural network. Let $l \in \{1, 2, 3, 4\}$ be the index of the hidden layers of the network. Let $z^{(l)}$ be the vector of input to layer $l$ and $y^{(l)}$ be the vector of outputs from layer $l$. We have modeled the dropout neural network as follows for any hidden unit $i$ and $l \in \{0, 1, 2, 3\}$[10, 11]:

$$\begin{aligned} r^{(l)} &\sim Bernoulli(p), \\ \tilde{y}^{(l)} &= r^{(l)} * y^{(l)}, \\ z_i^{(l+1)} &= w_i^{(l+1)} \tilde{y}^{(l)} + b_i^{(l+1)}, \\ y_i^{(l+1)} &= f(z_i^{(l+1)}), \end{aligned}$$

where $r^{(l)}$ is a vector of independent Bernoulli random variables having probability $p$ of being 1, $\tilde{y}^{(l)}$ denotes thinned outputs created by multiplying a sample of $r^{(l)}$ vector by outputs of layer $l$ (i.e., $y^{(l)}$) and used as input for the next layer $l+1$, $w^{(l)}$ is weights at layer $l$, $b^{(l)}$ is biases at layer $l$, and $f$ is an activation function, which is rectified linear units (ReLUs) in our setup. This process is done at each layer.

As earlier research on neural networks reported $p = 0.5$ as a close to optimal value for a wide range of networks in different applications [11], we have also used $p = 0.5$ in our dropout network.

In the learning phase using phase 2 profiles of each request, the derivatives of the loss function are back-propagated through the dropout network. We have used the stochastic gradient descent (SGD) algorithm with mini batches to train the dropout network. The adaptive gradient algorithm (AdaGrad) [3] is used to adjust the learning rates.

For the classification purpose and having probabilities as outputs, we have used Logistic classifier in the last layer. We use variable $c \in \{0, 1\}$ to show relevance of a suggestion to the given user in a context. Specifically, $P_\theta(c = 1|u, c, s)$ is the relevance score of the suggestion $s$ to the user $u$ and context $c$, in which $\theta$ is unknown parameters that are learned using maximum likelihood estimation (MLE) based on the train set, which consists of the users' preferences available in the profile of each request.

Given the relevance judgments $r$ of each suggestion $s_k$ to a user $u_i$ and context $c_j$ in the users' profiles available at each requests, the likelihood $L$ of the train set is as follows:

$$L = \prod_{i=1}^{|U|} \prod_{j=1}^{|C|} \prod_{k=1}^{|S|} P_\theta(c = 1|u_i, c_j, s_k)^r P_\theta(c = 0|u_i, c_j, s_k)^{1-r},$$

in which we assume relevance judgments $r$ are generated independently. We model $P_\theta(c = 1|u_i, c_j, s_k)$ by logistic function on a linear combination of inputs from the last hidden layer units' outputs. Then, the unknown parameters $\theta$ are optimized by maximizing the following log likelihood function:

$$\begin{aligned} \theta^* = argmax_\theta \sum_{i=1}^{|U|} \sum_{j=1}^{|C|} \sum_{k=1}^{|S|} & r log P_\theta(c = 1|u_i, c_j, s_k) \\ & + (1-r) log P_\theta(c = 0|u_i, c_j, s_k). \end{aligned}$$

## 4. RESULTS

In this section, the result of the two approaches detailed in Section 3 is discussed. These results are based on the official TREC 2016 Contextual Suggestion track qrels.

In order to evaluate our proposed models, we have implemented a content-based filtering baseline using standard language model to model users' profiles, and we have used KL-divergence to estimate relevance of the suggestion candidate to the user profile. Table 1 shows our submissions results against the content-based filtering baseline.

As it is shown in Table 1, both the proposed neural approaches beat the baseline in terms of all the common IR measures. The category of the attractions proves to be very useful to include in the contextual suggestion systems, explaining why the "UAmsterdamDL" approach performed better than the two others phase 2 runs.

## 5. CONCLUSION

In this paper, we studied contextual suggestion problem through neural user profiling and neural category preference modeling by the help of suggestions' endorsements. According to the phase 2 results of the TREC 2016 contextual suggestion track, using word embeddings to boost terms' weights related to suggestions' endorsements improves baseline content-based filtering approach in the contextual suggestion problem based on all common IR measures. Moreover, phase 2 results show that neural category preference modeling of the users can lead to even better results than the other tested user modeling approaches in contextual suggestion task. Specifically, the contextual suggestion submitted run based on neural category preference modeling performs better than our user profiling based submission and the content-based filtering baseline in terms of all the common IR measures including the TREC contextual suggestion official evaluation metric (NDCG@5). As a future work, we continue to work on users' preference modeling using category profiles created based on word embeddings.


### *Acknowledgments*

This research is funded in part by the European Community's FP7 (project meSch, grant # 600851), the Netherlands Organization for Scientific Research (ExPoSe project, NWO CI # 314.99.108; DiLiPaD project, NWO Digging into Data # 600.006.014), French Région Auvergne Rhône-Alpes ReSPIr project, and ELIAS (ESF Research Networking Programme).


## REFERENCES


[1] 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.

[2] A. Dean-Hall, C. L. A. Clarke, J. Kamps, J. Kiseleva, and E. Voorhees. Overview of the TREC 2015 contextual suggestion track. In *Proceedings of TREC 2015*, 2015.

[3] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12 (Jul):2121–2159, 2011.

[4] S. H. Hashemi and J. Kamps. Venue recommendation and web search based on anchor text. In *23rd Text REtrieval Conference (TREC)*, 2014.

[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] S. H. Hashemi, M. Dehghani, and J. Kamps. Parsimonious user and group profiling in venue recommendation. In *24th Text REtrieval Conference (TREC)*, 2015.

[7] S. H. Hashemi, C. L. Clarke, A. Dean-Hall, J. Kamps, and J. Kiseleva. Test collection building and maintenance in dynamic domains. In *15th Dutch-Belgian Information Retrieval Workshop (DIR)*, 2016.

[8] S. H. Hashemi, C. L. Clarke, A. Dean-Hall, J. Kamps, and J. Kiseleva. An easter egg hunting approach to test collection building in dynamic domains. In *Proceedings of NTCIR-EVIA 2016*, pages 1–8, 2016.

[9] S. H. Hashemi, C. L. A. Clarke, J. Kamps, J. Kiseleva, and E. Voorhees. Overview of the TREC 2016 contextual suggestion track. In *Proceedings of TREC 2016*, 2016.

[10] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012.

[11] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1): 1929–1958, 2014.