A New Approach to Contextual Suggestions Based on Word2Vec

Yongqiang Chen[1], Zhenjun Tang [1], Xiaozhao Zhao [1], D.Song[1,2],P.Zhang[1,2]

Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin University, China
{cyq ,tangzhenjun,zxz,pzhang }@tju.edu.cn,dawei.song2010@gmail.com

Abstract. We report our participation in the contextual suggestion track of TREC 2014 for which we submitted two runs using a novel approach to complete the competition. The goal of the track is to generate suggestions that users might fond of given the history of users' preference where he or she used to live in when they travel to a new city. We tested our new approach in the dataset of ClueWeb12-CatB which has been pre-indexed by Luence. Our system represents all attractions and user contexts in the continuous vector space learnt by neural network language models, and then we learn the user-dependent profile model to predict the user's ratings for the attraction's websites using Softmax. Finally, we rank all the venues by using the generated model according the users' personal preference.

Keywords: Contextual Suggestions,Word2Vec,user model

1 Introduction

The contextual suggestion tracktrac [2] investigates search techniques for complex information needs that are highly dependent on context and user interests. In other words, the track focus on a situation like that: a user traveled to a new city and we only have got the personal history preference data where he or she used to live in, for example the user is fond of watching movies or visiting parks, how do we recommend some places in the new city that the user may interested in and give the places a short description for the user referring to. Similarly, the track give us three files: the 100 example venues include title, description and the corresponding URL for users to evaluate, the contexts that contain 50 cities to serve as new cities and the user profile which is consist of users' judgment to the 100 example venues based on their personal preference. We proposed a novel approach to provide users with suggestions that they may be keen on given the condition that the user stay at different cities.

We presents a new neural network architecture that using unified vector from down to up for contextual suggestions. Instead of exploiting man-made input features carefully optimized for user model, we represent all attractions and user contexts in the continuous vector space learnt by neural network language models (NNML [1])as input layer. We use word2vec [3] to generate vectors for

each venue that given by TREC in the example.csv. Right after this, we mix the users' preference score for the website and the description by different weights and use the mixed score we classify the users' preference degree into 5 categories, range from -1 to 4, i.e., from least interested to the most enjoyed. At last, we take all the vectors that represented the venues as input and take the 5 different like degrees as label to train models for each user using Softmax. Finally, we rank all the venues by using the generated model according the users' personal preference.

We take the word2vec technique as a tool to digitize all the venues. Given the description of the specific venue, we can use the word2vec to output a hundreds of dimensions of vector to represent it. Immediately following, we use the vectors generated by the wrod2vec and the users' preference degree label to train the preference model by employing the Softmax algorithm and then generate the final suggestion. The rest of the paper is organized as follows: we briefly introduce the word2vec technique in Section 2. In Section 3, we first describe our experiment setup procedure. We introduce the detailed process of user modeling in Section 4 and conclusions are provided in Section 5.

2 A Brief Introduction to Train Word Vectors

To avoid the inaccuracy caused by classifying the example into several categories given by TREC manually, we take the word2vec to represent all attractions and user contexts in the continuous vector space learnt by neural network language models. The base of NNML is using neural networks for the probability function. The model learns simultaneously a distributed representation for each word along with the probability function for word sequences, expressed in terms of these representations. Training such large models, we propose continuous bag of words as our framework, and soft-max as the active function. So we use the word2vec to train wikitravel corpus and got the word vector.

To avoid the curse of dimensionality by learning a distributed representation for words as our word vector, we define a test set that compare different dimensionality of vectors for our task using the same training data and using the same model architecture. In Table 1 below, it can be seen that adding more dimensions will increase improvement at the first stage, and then after some point, it provides diminishing improvement. As Tomas' paper,we observe that train the word vectors dimensions on relatively amount of data , So we chose dimensions as 200.

Table 1. Statistics of test collections and topics.

Dimensionality	50	100	150	200	250
Precision Accuracy [%]	9.7	13.6	20.4	33.8	19.5

3 Experiments

In this section, we first give an overview of test collections and the external knowledge base used in our experiments. Then we describe a novel metric to evaluate retrieval model's effectiveness and stability simultaneously. Finally, we demonstrate that expansion information from external sources is valuable for improving the overall performance of retrieval system.

3.1 Data Collection

Wikitravel dataset contains numerous famous cities all over around the world and each page corresponding to a specific city gives us a brief introduction to the city. The city homepages are structured according some general rules, with clearly separated sections for sightseeing, eating around, shopping and having fun. All the attractions listed on the city homepages are given by displaying its name, characteristic, open time and telephone number and so on. In this case, we extracted all the city homepages according the contexts given by TREC from the Wikitravel dataset and we clean all the extracted pages by removing the noisy data such as the History section or the Get in section. We only keep the See, Do, Buy, Eat, Drink and Sleep section to extract the places as candidate venues to be suggested for each city. We use $H_i(i = 1, 2, 50)$ to represent each extracted city homepage andV_i is a set of all the venues in H_i.

As is known, we should provide suggestions for every user-context pair with up to 50 venues which contains title, description and ClueWeb docID. The venues extracted from Wikitravel for each city apparently have no docIDs because the Wikitravel dataset is not part of CluWeb12-CatB. To better solve this problem, we reserve all the URLs of the venues extracted from the city homepage in Wikitravel. Hence, we have all the venues in V_i with corresponding URLs. In the ClueWeb12 dataset, there is a file map almost 7 billion URLs to ClueWeb docIDs, so we can easily map all the venues' URLs in V_i to docIDs and we use $V_i^{'}$ to represent all the venues with docIDs in H_i. Apparently, $V_i^{'}$ is a pure subset of V_i.

However, having considered not all the URLs could be found in the file, we figured out another way to ensure there are enough venues to be recommended. This track request all the participants to make suggestions to the travelers in a new city that has never been. That is to say the suggestions we made most are correlated with travel, so we download a travel attraction ontology, which contains many kinds of tour categories such as eating, sightseeing, playing and so on. We use each category C_i in as input to word2vec to calculate the most relevant words. For example, we use food category as input, then we gained a ranked list of synonyms displaying by order like that: Hamburg, Bread, Restaurant etc. Each similar word in the ranked list we marked it asT_i. We combined the venues that without URLs which can be clearly represented by Δ_i=V_i-$V_i^{'}$ and T_i as a query $Q_i(\Delta_i,T_i)$ to find more venues in the pre-indexed ClueWeb12-CatB dataset. The detailed flow of the extract process could be described as below:

Algorithm 1 Extract Venues

```
for all C_i such that C_i ∈ Ω do
  for all T_i such that T_i ∈ C_i do
    Query the ClueWeb12-CatB by Q_i(Δ_i,T_i)
    Add the extracted venues into H_i
  end for
end for
```

3.2 Data Preprocessing

So far, we have generated all the venues which contains title, description and docID for each user-city pair to be suggested. The next step is how to digitize all these venues. We use $d_i^j(m)$ to represent the mth description of the venues in the city θ_j given user u_i. The character v_{T_i} represents the vector of term T_i. The digitize process of the venues can be depicted as below: The symbol $v_i^j(m)$ stands for the vector of the m th venue in the city θ_j given user u_i and U and Θ represents for the 562 users and 50 cities, respectively. We simply add up the vectors of term T_i appear in the venue's description. For example, for the venue "Ploy Restaurant", there are terms like "Beef", "Pizza", and "Bread" in its description, we add up all these three terms' vector to form a new vector to represent the "Ploy Restaurant".

Up to now, we have digitized all the venues to be suggested, in the following section we will introduce how we take the $v_i^j(m)$ as our user model input and rank the venues according users' personal preference.

So far, we have generated all the venues which contains title, description and docID for each user-city pair to be suggested. The next step is how to digitize all these venues. We use $d_i^j(m)$ to represent the mth description of the venues in the city θ_j given user u_i. The character v_{T_i} represents the vector of term T_i. The digitize process of the venues can be depicted as below:

Algorithm 2 Data Collection Preprocessing

```
for all u such that u ∈ U do
  for all θ_j city such that θ_j ∈ Θ do
    for all T_i such that T_i ∈ d_i^j(m) do
      Calculate how many T_i is destributed in d_i^j(m)
      Each venue's vector v_i^j(m) = Σ v_{T_i}
    end for
  end for
end for
```

4 User modeling

As mentioned above, we classify the user preference into 5 categories according their score corresponding to the venues' description and the URL in the example given by TREC. We use the supervised machine learning algorithm as the core to train user model. We assume the users' likeability to the venues obey Gaussian distribution and the 5 categories obey the Multiple Bernoulli distribution. Hence, taking the Softmax algorithm to do the multiple classify task will be much appropriate for meeting our needs.

We generated an N dimensions vector for each term T_i and we simply add up linearly all the ${}_(T_i)$ of term T_i that contained in the example venue's description to form a new N dimensions vector ${}_(example(i))$. There are 50 example venues in total, so the matrix ${}_example$ on behalf of the 50N digitized example venues. The degree of user preference to each venue, i.e., the 5 integer preference categories range from -1 to 4, serve as supervised train label l_i, and the 50 dimensions column vector $L_example = [l_1, l_2, l_50]^T$ stands for the user label matrix. Similarly, the ${}_(all - venues(i))$ on behalf of the vector of the venue that to be suggested and ${}_(all - venues)$ represent all of them.

The user modeling can be illustrated clearly by Figure 1:

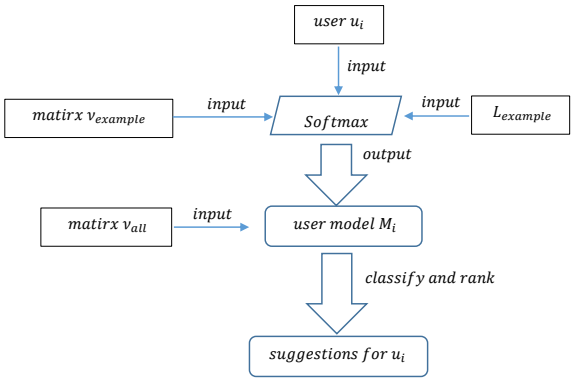

Fig. 1. the flow chart of user modeling

As we know, the algorithm Softmax is good at multi-classify. Given an independent variable as input, it outputs the different probabilities that decides which category the variable belongs to. Finally, the variable is classified to the category that has the highest probability. At the ranking procedure, all the venues are categorized to the 5 categories, we first rank all the venues according to their categories in descending order and then rank the venues through referring their probability belong to one specific category.

5 Results and Analysis

We submitted our run to TREC, the RunA use Softmax and RunB use KNN .our result is not as satisfied as we expected. As the TREC not publish all the ranked list and other participants' results, we cannot make it clear what is our rank on earth. However, comparing our performance with the other teams which focus their work in ClueWeb12 in 2013, our performance get the 6th rank position and is worth mentioning having considered the corpus we use is ClueWeb12-CatB which is a very small part of ClueWeb12. To the best of our knowledge, this is

Table 2. the performance of RunA and RunB.

Runid	P@5	MRR	TBG
RunA	0.0488	0.0889	0.1662
RunB	0.0247	0.0518	0.0581

the first attempt to use word2vec approach to digitize the venues in order to calculate its attraction to a specific user given his or her history preference data. Though the performance of our model is not as so good as others, we proposed a new method and open up a new path to make suggestions for users.

There are several factors account for the defect of our method. To begin with, the corpus we used is ClueWeb12-CatB which is a very small part of the whole ClueWeb12 and the directly consequence affected by this is that we couldn't find more precise or enough data to make suggestions according to the users' personal preference regardless of whatever the user model we use. Due to the objective factor, we have to leave testing our user model in the whole ClueWeb12 in the next year and we believe our performance will be improved given the whole dataset. In addition, we simply add up all the $_{T_i}$of terms T_i that appear in the venues' descriptions which may lack of considering the correlation among the term T_i . And whether add them up linearly have some semantic meaning still remain explored. Lastly, the final factor contribute to influence our preference lays in the ranking procedure which acts as the decisive role. As is well known, the Softmax algorithm is good at classifying other than ranking. We make a bold attempt to use its output probability to rank the venues that belong to the same category. However, the facts proves what have done do not play well and we will modify it in the next year trec.

References

1. Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155, 2003.
2. A. Dean-Hall, C. L. Clarke, J. Kamps, P. Thomas, N. Simone, and E. Voorhees. Overview of the trec 2013 contextual suggestion track. Technical report, DTIC Document, 2013.
3. T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.