Applying Learning to Rank Techniques to Contextual Suggestions

Julia Kiseleva[1] Alejandro Montes García[1] Yongming Luo[1]
Jaap Kamps[2] Mykola Pechenizkiy[1] Paul De Bra[1]
[1] Eindhoven University of Technology, Eindhoven, The Netherlands
[2]University of Amsterdam, Amsterdam, The Netherlands
[1]{j.kiseleva@, a.montes.garcia@, y.luo@, m.pechenizkiy@, debra@win.}tue.nl [2]kamps@uva.nl

ABSTRACT

The Text Retrieval Conference's Contextual Suggestion Track investigates search techniques for complex information needs that are highly dependent on a context and user interests. The goal of the track is to evaluate systems that provide suggestions for activities to users in a specific location, taking into account their historical personal preferences. In this paper, we present our approach for the Contextual Suggestion Track 2014. We suggest to treat the problem of Contextual Suggestion as a Learning to Rank problem. As a source for travel suggestions we use data from four social networks: Yelp, Facebook, Foursquare and Google Places. For our study we train two ranking algorithms: Rank Net and Random Forest. In our experiments, we seek to answer the following research questions: Does the distance between the locations of training and testing contexts impact precision? Which data sources (i.e., Facebook, Foursquare, Yelp, and Google Places) provide more effective training data?

Categories and Subject Descriptors

H.3.4 [**Systems and Software**]: User profiles and alert services

General Terms

Experimentation, Human Factor

Keywords

Context-based recommendation, context-awareness, information retrieval

1. INTRODUCTION

The **T**ext **R**etrieval **C**onference (TREC) investigates techniques for searching complex information. This can be applied for different applications and from different perspectives. In this paper we share our experience of participation in the Contextual Suggestion Track 2014, which has been organized for the third time. The number of participants increases every year because of its relevance in real-world applications. Our aim for the TREC 2014 Contextual Suggestion Track is to use a combination of four social networks: Yelp, Facebook, Foursquare and Google Places.

Contextual Suggestion Track brings an important problem to solve. The report from the Second Strategic Workshop on Information Retrieval in Lorne [1] says:

> *"future information retrieval systems must anticipate user needs and respond with information appropriate to the current context without the user having to enter an explicit query [...] In a mobile context such a system might take the form of an app that recommends interesting places and activities based on the user's location, personal preferences, past history, and environmental factors such as weather and time [...] In contrast to many traditional recommender systems, these systems must be open domain, ideally able to make suggestion and synthesize information from multiple sources ... ".*

The participation in the Contextual Suggestion Track gives an opportunity to test some approaches in a well defined experimental environment. This will help in turn to project solutions into real-world applications.

As an input to the task, the track provides the following data-sources: **(1)** a set of *user profiles*, **(2)** a set of *example suggestions*, and **(3)** a set of *contexts* (geographical locations) for training and testing phases. Each profile corresponds to a single user and indicates the user's preference with respect to each example of the proposed suggestions in a particular context. For example, a suggestion might be to have a beer at the 'Dogfish Head Alehouse'[1] (in Gaithersburg, MD), and the user profile might include a negative preference with respect to this suggestion. Each suggestion in the training set is described using the following triple: a title of attraction, a description of attraction, and an associated URL. In the case of this track, each context corresponds to a particular geographical location (city) (e.g., Gaithersburg, Maryland, USA). However, in general context has a different nature in the literature [9–11]. A more detailed and formalized description of the task is presented in Section 3.

The main goal of this task is to learn user's preferences out of provided examples. Afterwards, we need to return a ranked list of up to fifty ranked suggestions for each pair: `(user profile,context)`. The list of suggestions is ranked based on the user's preferences in the particular geographical location. In order to achieve this goal, we have to formulate the problem setup as a learning to rank problem where we directly optimize the required evaluation metrics, e.g., precision at rank 5 ($P@5$). As a source for contextual suggestions we use data from four social networks namely Facebook, Foursquare, Yelp, and Google Places, which are combined into one dataset.

[1]`http://dogfishalehouse.com/`

In addition to the main task of the Contextual Suggestion Track, we are interested in the following research questions:

Research Question 1: Does the distance between the locations of training and testing contexts impact precision?

Research Question 2: Which data sources (i.e., Facebook, Foursquare, Yelp, and Google Places) provide more effective training data?

The rest of this paper is organised as follows. Section 2 describes background and related work. The formal and detailed description of the task at Contextual Suggestion Track is presented in Section 3. Section 4 describes the process of the data gathering from the selected social networks. Section 5 presents how we model and rank contextual suggestions. In Section 6 we present and discuss obtained results. We summarize our findings and conclude in Section 7.

2. BACKGROUND AND RELATED WORK

According to the TREC Contextual Suggestion Track organizers,"*The contextual suggestion track investigates search techniques for complex information needs that are highly dependent on context and user interests*" [3, 5]. This track has been started in 2012 and it is organized every year since then. However, over the years the track has been going through some changes.

In track from 2012, contexts were defined as a combination of a location and a temporal components: (1) a season, (2) a time of day, and (3) a day of week. Since 2013 contexts are presented only by geographical locations.

In 2012, the work by Hubert and Cabanac [8] achieved the best performance. They implemented a framework that simply retrieves places from the Google Places API.[2] Their framework takes into account a time feature and a list of suggestion is retrieved based on a time related components. In order to learn users' preferences for providing recommendations Hubert and Cabanac [8] followed two approaches. The selected methods separate the positive and negative terms from users' profiles and represented them using two different Vector Space Models[15]. The best approach represents each user' profile as two unique vectors: (1) with the negative terms and (2) with the positive terms. The positive and negative terms are taken from positively and negatively rated reviews respectively.

In 2013, the best performance was achieved by Yang and Fang [17]. Their work was divided into four different modules: The first module is *gathering useful information.* Yelp [3] was crawled to look for useful information such as main page of a business, categories and reviews. The second module is *modeling users' profiles.* In order to model the user profile, they took reviews from Yelp users who gave the same rating as the user being modelled for a specific place. With the terms from those reviews they built a positive and a negative user profile for each user. The third module is *ranking suggestion candidates.* In order to estimate the suitability of a candidate business to a user, they used a similarity measure, more specifically, F2-EXP of axiomatic retrieval model [7]. The fourth and final module is *generating descriptions.* To generate a description that fits the requirements and is interesting for the user, the authors made use of the category of the place, significant reviews and places that the user previously liked.

Most of previously utilized approaches used a 'matching problem setup': given a user' profile (P) and an attraction description (AD) it is needed to provide a score of how AD suits to P. In contrast, we formulate a problem as a learning to rank problem where we directly optimize the evaluation metric, in this case precision at rank 5 ($P@5$).

3. NOTATION FOR CONTEXTUAL SUGGESTION

In this Section we describe some preliminary knowledge about the contextual suggestion track. First, we formalize the task in Section 3.1. Second, we describe the evaluation metrics that are used to estimate the final performance of the solutions for the track in Section 3.2. Third, we present a list of available possibilities for the input data to find the list of attractions that further need to be ranked in Section 3.3.

3.1 Task Description

The Contextual Suggestion Track provides the following three inputs:

1. **Users' Profiles:** a set of user profiles where a profile p_i consists of ratings for a series of attractions $\{a_j\}_{i=1}^n$. An attraction a_j is represented by three entities: a title (t_j), a description (d_j), and an associated website (w_j). Let us denote an attraction as a triple $a_j = (t_j, d_j, w_j)$. Each p_ij consists of three ratings (1) for the attraction title ($R(t_j)$), (2) the attraction description ($R(d_j)$), and (3) the attraction website ($R(w_j)$). Let us denote a user profile as a triple $p_{ij} = (R(t_j), R(d_j), R(w_j))$;

2. **Example Suggestions:** a set of example suggestions where we can find values for each $a_j = (t_j, d_j, w_j)$;

3. **Training and Testing Contexts:** a set of testing contexts $\{c_k\}_{k=1}^m$ where a context corresponds to a particular geographical location. The profiles are given with use of the two training contexts: c_1 = 'Santa Fe, NM' and c_2 = 'Chicago, IL'.

A five-point scale for the ratings is used for the ratings. The scale reflects how interested a user would be in going to the suggested attraction while he is visiting a city from $\{c_k\}_{k=1}^m$. The following list of scores is used:

- **4** – Strongly interested;
- **3** – Interested;
- **2** – Neutral;
- **1** – Disinterested;
- **0** – Strongly disinterested;
- **-1** – Website didn't load or no rating given

The main goal is to generate a ranked list of up to 50 suggested attractions for the user profile p_i in the particular context c_k. A suggested attraction a_i is generated for each

[2] `https://developers.google.com/places/documentation/`
[3] `http://www.yelp.com/`

pair of a user's profile and a context: $\langle p_i, c_k \rangle$. Each suggestion should suit to both the user's preferences expressed in a profile and a context (should be located in a particular geographical locations). A description of a suggestion may be tailored to reflect the preferences of that user. In other words, a description can be personalized.

3.2 Evaluation Metrics

It is important to clarify which evaluation metrics are used to estimate the final results. The evaluation main metric is a Precision at Rank 5 (P@5). The TREC 2014 Contextual Suggestion Track also used addition metrics such as a mean reciprocal rank (MRR), and a modified Time-Biased Gain (TBG) [4] to evaluate the final results.

3.3 Data Sources to Extract Attractions

In this Section we describe a list of data sources that might be used to find attractions. We need the list of attractions to construct contextual suggestions.

The list of contextual suggestions can be derived be either from the **ClueWeb12 dataset**[4] collection or from the open web. In our work we chose to work with the open web. We made this choice because previously the runs that used the open web were showing better performance.

Our final submission for the track was based on a combination of the gathered data from four major social networks. In order to ensure reproducibility and further comparison of our results we made the dataset publicly available.[5]

Next, we will describe our approach for data gathering from the open web.

4. DATA GATHERING PROCESS

In this Section we describe the data gathering process in details. First, in Section 4.1, we show how we collect the data from the four social networks. Then, in Section 4.2 we present how we merge the four sources into one.

4.1 Description of Used Data Sources

Social networks play a major role in the Web because they contain substantial amounts of information [12]. Data from social networks has been successfully exploited in different approaches [6, 12, 14, 16] in order to enhance suggestions.

In our approach, we mined our dataset from well-known social networks APIs for venue recommendation, namely Yelp, Foursquare, Facebook, and Google Places. In the next paragraphs we describe the kind of information that we retrieved from these sources.

Yelp. Yelp is a social network in which users can evaluate and post reviews about local businesses. We queried the Yelp API to extract businesses from the targeted contexts and we extracted the following information for each business:

- `id`: A unique identifier for the site on Yelp.
- `name`: The name of the business.
- `categories`: Categories that the business is associated with.
- `rating`: Rating given by users to the business.
- `location`: The location of the business.
- `url`: URL of the business on Yelp.

Foursquare. Foursquare is another social network similar to Yelp from which we extracted the following information:

- `id`: A unique identifier for the site on Foursquare.
- `name`: The name of the business.
- `categories`: Categories that the business is associated with.
- `description`: Description of the business provided by the owner.
- `rating`: Rating given by users to the business.
- `tips`: Recommendations provided by users about the business.
- `location`: The location of the business.
- `url`: URL of the business provided by the owner.

Facebook. Facebook is a general purpose social network, but its huge number of active users has made it attractive for business to have their own pages also. Therefore, we collected information from Facebook from those places that matched the name of any of the business we have already collected. From Facebook we extracted the following data:

- `id`: A unique identifier for the site in Facebook.
- `name`: The name of the business.
- `likes`: Number of people that liked the place on Facebook.

Google Places. Finally Google Places is a service which also returns information from places. We have not mined any field from the venue from Google Places different from those described above.

4.2 Merging Data Sources

After retrieving information from the four aforementioned sources, we needed to merge our duplicates. To do so we searched places which had a similar name and were located nearby. More specifically, we used the following algorithm:

To calculate the distance between names we used the Levensthein Distance [13], this is the lower number of substitutions required to transform the string **a** into **b**. In our approach we normalized it dividing by the length of the longest string.

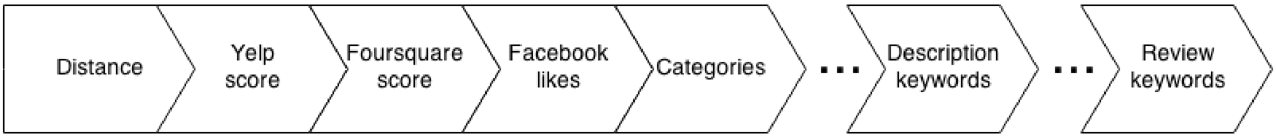
We described how we obtain the dataset for our experimentation that is publicly available. Next, we will present our solution for modeling contextual suggestions using our dataset.

5. MODELING CONTEXTUAL SUGGESTIONS

In this Section we present an overview of the framework developed to obtain a ranked list of contextual suggestions.

[4]`http://www.lemurproject.org/clueweb12.php`
[5]`http://www.juliakiseleva.com/datasets/trec_cs_2014.html`

Figure 1: Vector of features



Data: Set of unmerged places: $\mathcal{P}$
Result: Set of merged places: $\mathcal{R}$
$\mathcal{T} \leftarrow$ threshold;
foreach *place* $\mathcal{P_C}$ *in* $\mathcal{P}$ **do**
  $\mathcal{S} \leftarrow$ draw square around $\mathcal{P_C}$;
  $\mathcal{P}' \leftarrow$ places inside $\mathcal{S}$;
  foreach *place* $\mathcal{P_C}'$ *in* $\mathcal{P}'$ **do**
    if *levDistance(nameOf(*$\mathcal{P_C}'$*), nameOf(*$\mathcal{P_C}$*))* $< \mathcal{T}$
    then
      merge($\mathcal{P_C}'$, $\mathcal{P_C}$);
    end
  end
  $\mathcal{R} \leftarrow$ add $\mathcal{P_C}$
end

Algorithm 1: Merge data from different sources

5.1 Formulating Learning To Rank Problem

In order to achieve the goal of this track we formulated our problem as a learning to rank problem (LRP). LRP usually has the supervised settings: each pair of a query q_i and a document d_j: $\langle q_i, d_j \rangle$ is associated with a score s_k. The set of this pairs is used to train a pair-wise ranker.

The description of the contextual suggestion task, which is presented in Section 3.1, can be perfectly mapped to LRP. We can consider the pair a context c_k and a user profile p_i: $\langle c_k, p_i \rangle$ as a query q_i. An attraction a_j can be considered as a document d_j. We also have the five-point scale labels for the pairs $\langle\langle c_k, p_i \rangle\rangle a_j$ as described before. Therefore we are dealing with a supervised learning setup where one of the learning to rank methods can be applied.

As a ranking method we used the algorithms: RankNet and Random Forest.[6] RankNet employes neural network models as indicated by its name. The obtained ranking model was applied to unlabeled data to derive the final results.

5.2 Deriving Features for LTR

In this Section we will describe a set of features we use to train the rankers. Figure 1 shows a graphic representation of our feature set consisting of: **(1)** 'Distance,' 'Yelp score,' 'Foursquare score,' 'Facebook likes,' and 'Categories' based on the social networks, **(2)** 'Description keywords' from the profiles, and **(3)** 'Review Keywords' based on a sentiment score.

Deriving Features from Social Networks

From the data sources described before, we extract the ratings-like features (or likes in the case of Facebook) as well as categories and the distance to the context location.

[6]We used the following implementation: RankLib library `http://people.cs.umass.edu/~vdang/ranklib.html`, described in details in [2].

Table 1: Official Run Results: RankNet and Random Forest LRT

Run	Method	P@5	TBG	MRR
`tueNet`	RankNet	0.2261	0.9224	0.3820
`tueRforest`	Random Forest	0.2227	0.9293	0.3604

Obtaining Positive and Negative Users Profiles Characteristics

We built a histogram of words with descriptions from business with positive ratings and another histogram for those with negative ratings. From those histograms we took the top-n most frequent words and used them as features as well. We followed the same approach with user reviews.

Incorporating Semantic Characteristics

We are curious whether the sentiment of reviews of a certain place is different from others, and if it has any impact on the overall ranking. Therefore, for each attribute description we generate a sentiment score and use it as a feature for later ranking. This task is easily achieved using the built-in functionality in TextBlob[7].

We presented our solution for the contextual suggestion track where we have formulated it as a LTR problem. We described the set of features that was used to build the model. Next, we will discuss the results, the benefits and the drawbacks of the proposed solution.

6. RESULTS AND DISCUSSION

In this Section we will present results. First, we will concentrate the results obtained by the contextual suggestion track evaluation process. Second, we will concentrate on answering the specific research questions introduced in Section 1.

6.1 Contextual Suggestion Track Results

Regarding the contextual suggestion track, our results are shown in Table 1 based on evaluation metrics presented in Section 3.2. As we can see, the RankNet model is somewhat better than the RandomForest model for P@5 and MRR, and marginally less effective for MRR. Overall, the effect of the two models is very similar.

In comparison to the other open web submissions to the TREC, we find ourselves ranked in the bottom half of the submitted runs: the RankNet model ranks 18th out of 25 submissions on P@5.

There are a number of reasons why our submissions failed to realize the full potential of the LTR approach:

1. First of all, we did not have the chance to perform

[7]`http://textblob.readthedocs.org/`

Table 2: The distribution of $P@5$ of contextual suggestions for the contexts located within a different radius from Santa Fe, NM

Distance (in km) Santa Fe, NM from	P@5
100-500	0.277
500-1000	0.202
1000-1500	0.180
1500-2000	0.261
2000-2500	0.234
2500-3000	0.273
> 3000	0.300

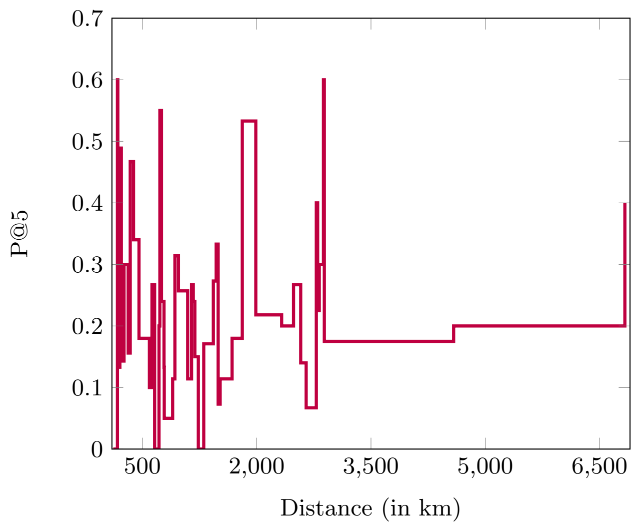


Figure 2: The distribution of $P@5$ of contextual suggestions for the contexts located within a different radius from Santa Fe, NM

any feature selection method due the limitation on the number of submissions;

2. Second, we did not play with a feature normalization process due the same reason;

3. Third, we tried to estimate our performance offline using the five-fold cross-validation but we knew only the labels for the two given contexts: c_1 = 'Santa Fe, NM' and c_2 = 'Chicago, IL'.

6.2 Additional Research Questions

In this Section we will discuss the additional research questions that we mentioned in Section 1.

6.2.1 Research Question 1

Our first research questions was: Does the distance between the locations of training and testing contexts impact precision? As it was mentioned before, we had the two training contexts: Santa Fe, NM and Chicago, IL. The distribution of precision ($P@5$) depending on the distance from Santa Fe, NM to the context (the city from the list of contexts) is presented in Table 2. We measure precision separately for each context from $\{c_k\}_{k=1}^m$ and calculate an average precision within some radius from the initial context. The more detailed distribution is presented in Figure 2.

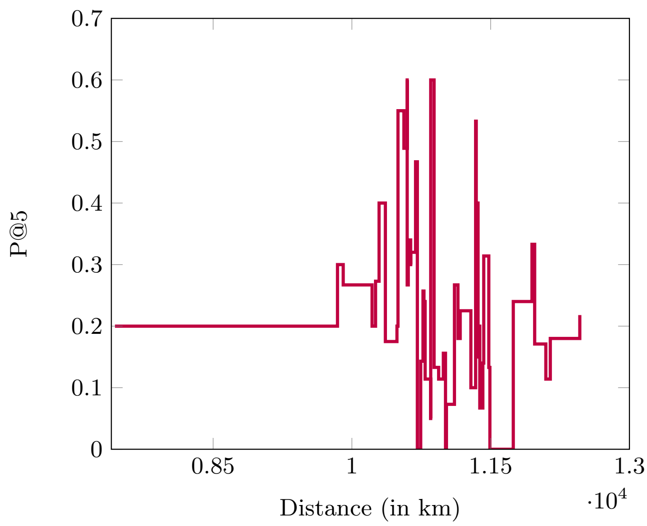


Figure 3: The distribution of $P@5$ of contextual suggestions for the contexts located within a different radius from Chicago, IL.

Table 3: An average contribution of the different data sources per $P@5$

P@5	Foursquare	Facebook	Google Places	Yelp
0.2	0.395	0.222	0.136	0.519
0.4	0.365	0.212	0.115	0.577
0.6	0.625	0.344	0.0	0.500
0.8	0.333	0.333	0.067	0.800
1.0	0.0	1.0	0.0	1.0
Average	0.357	0.422	0.072	0.654

According to these results, we see no clear relation between distance and precision. We can see that at first the precision is getting worse but then it is back to the average again. Therefore, there is no correlation between $P@5$ and how far a city is located from the initial context (such as Santa Fe, NM).

We can think about the following possible explanation the similarity between the two cities is more important than the distance between them. For example, Miami and Santa Fe, NM are both big cities with many attractions. In contrast, Santa Fe, NM is located relatively close to the town called Erie. However, Erie is a small town and probably the list of attractions is not similar to the Santa Fe one.

The same picture we can see if we consider the second initial context: Chicago, IL. The detailed distribution between $P@5$ and the distance from Chicago,IL to other contexts is shown in Figure 3. In the case of Chicago, IL we also cannot see any strong correlation between precision and and how far a city is located from the initial context (such as Chicago, IL).

Therefore we can conclude that the similarity between contexts can be useful information. It can be useful to utilize the description of the contexts, e.g., the population, the size, the number of attractions etc.

6.2.2 Research Question 2

Our second research questions was: Which data sources (i.e., Facebook, Foursquare, Yelp, and Google Places) provide more effective training data?

Let us remind, our final dataset is the combination of data from four social networks: Yelp, Foursquare, Facebook, and Google Places. With respect to an attraction a_j we try to combine information about a_j from these four data-sources using our matching Algorithm 1. However, it is not always possible. For example, a_j can be found in Yelp (Y) and Facebook (F) but it is not presented in Foursquare (FQ) and Google Places (GP). In other words, a_j can represented as a quadruple $\langle Y, F, FQ, GP \rangle$.

In order to answer our research question, we split the obtained results into five groups: **(1)** result that gives us $P@5 = 0.2$; **(2)** result that gives us $P@5 = 0.4$; **(3)** result that gives us $P@5 = 0.6$; **(4)** result that gives us $P@5 = 0.8$; **(5)** result that gives us $P@5 = 1.0$. Then we calculate a proportion of presence of Y, F, FQ, GP per different groups. The result of our investigation is presented in Table 3.

With respect to precision, we can see that Yelp is most effective, with distance, followed by Facebook and Foursquare, and that Google Places is rather ineffective. When broken down over P@5, we see that Yelp performs well throughout, and Foursquare is better than Facebook at lower precision levels, but Facebook excels at the higher precision levels.

We presented the results of our participation in the contextual suggestion tract at TREC 2014. Next, we will conclude our paper.

7. CONCLUSION

In this paper we described our experience of the participation in TREC Contextual Suggestion Track 2014. As a data-source we used a combination of four social networks: Yelp, Facebook, Fourquare, and Google Places.

In contrast to the previous work, we formulated the problem as a learning to rank setup. We described our approach for the feature engineering to train a ranker. For our experimentation we used two ranking algorithms: RankNet and Random Forest.

We experiment with a dataset that is a combination of the data from the four social networks: Foursquare, Facebook, Google Places and Yelp. We also present how we merge data from these different data sources.

Our results were reasonable, but ranked in the lower half of the submissions, likely due to sparse training data available for two contexts. In addition, we asked two research questions. First, we looked into the relation between distance and precision and found no clear correlation. Second, we looked at the relative performance of each of the four data sources, and found that Yelp data performed the best throughout.

REFERENCES

[1] J. Allan, B. Croft, A. Moffat, and M. Sanderson. Frontiers, challenges, and opportunities for information retrieval: Report from swirl 2012 the second strategic workshop on information retrieval in lorne. *ACM SIGIR Forum*, 46(1):2–32, 2012.

[2] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference e on Machine Learning (ICML'2005)*, pages 89–96, 2005.

[3] A. Dean-Hall, C. L. Clarke, J. Kamps, P. Thomas, and E. Voorhees. Overview of the TREC 2012 Contextual Suggestion Track. Technical report, 2012.

[4] A. Dean-Hall, C. L. Clarke, J. Kamps, and P. Thomas. Evaluating contextual suggestion. In *TREC*, 2013.

[5] 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, 2013.

[6] S. Debnath, N. Ganguly, and P. Mitra. Feature weighting in content based recommendation system using social network analysis. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 1041–1042, 2008. ISBN 978-1-60558-085-2.

[7] 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, 2005. ISBN 1-59593-034-5.

[8] G. Hubert and G. Cabanac. IRIT at TREC 2012 Contextual Suggestion Track. Technical report, 2012.

[9] J. Kiseleva. Context mining and integration into predictive web analytics. In *WWW (Companion Volume)*, pages 383–388, 2013.

[10] J. Kiseleva, H. T. Lam, M. Pechenizkiy, and T. Calders. Predicting current user intent with contextual markov models. In *ICDM Workshops*, 2013.

[11] J. Kiseleva, H. T. Lam, M. Pechenizkiy, and T. Calders. Discovering temporal hidden contexts in web sessions for user trail prediction. In *Proceedings of the 22nd international conference on World Wide Web, (Companion Volume, TempWeb@WWW'2013)*, pages 1067–1074. ACM, 2013. doi: http://dl.acm.org/citation.cfm?id=2488120.

[12] I. Konstas, V. Stathopoulos, and J. M. Jose. On social networks and collaborative recommendation. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 195–202, 2009. ISBN 978-1-60558-483-6.

[13] V. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707, 1966.

[14] H. Liu and P. Maes. Interestmap: Harvesting social network profiles for recommendations. In *In Proceedings of the Beyond Personalization 2005 Workshop*, 2005.

[15] G. Salton. Mathematics and information retrieval. *Journal of Documentation*, 35(1):1–29, 1979.

[16] F. E. Walter, S. Battiston, and F. Schweitzer. A model of a trust-based recommendation system on a social network. *Autonomous Agents and Multi-Agent Systems*, 16(1):57–74, 2008. ISSN 1387-2532. doi: 10.1007/s10458-007-9021-x. URL `http://dx.doi.org/10.1007/s10458-007-9021-x`.

[17] P. Yang and H. Fang. An Opinion-aware Approach to Contextual Suggestion. Technical report, 2013.