Improving Contextual Suggestions using Open Web Domain Knowledge

Thaer Samar,[1] Alejandro Bellogín,[2] and Arjen de Vries[1]

[1] Centrum Wiskunde & Informatica, Amsterdam, The Netherlands
[2] Universidad Autónoma de Madrid, Madrid, Spain

{samar|arjen}@cwi.nl, {alejandro.bellogin}@uam.es

ABSTRACT

Contextual suggestion aims at recommending items to users given their current context, such as location-based tourist recommendations. Our contextual suggestion ranking model consists of two main components: selecting candidate suggestions and providing a ranked list of personalized suggestions. We focus on selecting appropriate suggestions from the ClueWeb12 collection using tourist domain knowledge inferred from social sites and resources available on the public Web (Open Web). Specifically, we generate two candidate subsets retrieved from the ClueWeb12 collection, one by filtering the content on mentions of the location context, and one by integrating domain knowledge derived from the Open Web. The impact of these candidate selection methods on contextual suggestion effectiveness is analyzed using the test collection constructed for the TREC Contextual Suggestion Track in 2014. Our main findings are that contextual suggestion performance on the subset created using Open Web domain knowledge is significantly better than using only geographical information. Second, using a prior probability estimated from domain knowledge leads to better suggestions and improves the performance.

1. INTRODUCTION

Recommender systems aim to help people find items of interest from a large pool of potentially interesting items. The users' preferences may change depending on their current context, such as the time of day, the device they use, or their location. Hence, those recommendations or suggestions should be tailored to the context of the user. Typically, recommender systems suggest a list of items based on users preferences. However, awareness of the importance of context as a third dimension beyond users and items has increased, for recommendation [1] and search [10] alike. The goal is to anticipate users' context without asking them. This problem – known as *contextual suggestion* in Information Retrieval (`IR`) and *context-aware recommendation* in the Recommender Systems (`RS`) community – is far from being solved. Depending on the type of context taken into account (time, location, group, short-term preferences, etc.), different techniques have been proposed. We use the definition of context stated in TREC's Contextual Suggestion (`CS`) track [5]: a context consists of a geographical location (a city and its corresponding state in the United States). The CS track investigates search techniques for complex information needs that are highly dependent on context and user preferences. Submission based on documents collected from either the Open Web or ClueWeb12 collection has been allowed since 2013, and the goal is to provide a list of ranked suggestions per (user, context) pair. An earlier analysis of the track's empirical results (in 2013 and 2014) has shown that runs based on the Open Web usually achieve higher effectiveness than those based on ClueWeb12 collection [6, 7].

The majority of existing studies have relied on location-based social networks from the Open Web that are specialized in providing tourist suggestions, such as Yelp and Foursquare; focusing on re-ranking the candidate suggestions based on user preferences. The main problem addressed then is to model user interests through content-based recommendation, considering evidence in the form of terms taken from the textual descriptions [12] or categories [14] of suggestions in the user profile and their associated ratings, and approaches to rank suggestions based on their similarity with the user profile. Likewise, in [8] the authors combine various user-dependent and venue-dependent features, including the aforementioned descriptions and category features, in one ranking model. However, using the ClueWeb12 collection as source of attractions requires first the selection of candidate documents, to be ranked later based on user preferences. The selection of candidate documents is a challenging task, since the (potentially) relevant suggestions have to be selected from this large collection.

In this paper, we use domain knowledge inferred from location-based social networks on the Open Web for selecting suggestions from ClueWeb12. We evaluate our contextual suggestion model on two sub-collections of the ClueWeb12 collection. One of the two sub-collections was generated using location-based social networks to annotate the candidate documents from ClueWeb12 collection. We discuss how explicit representation of knowledge about the tourism domain available on the location-based social networks improves the effectiveness of our contextual suggestion model. We show that the same contextual suggestion model for recommendation achieves an order of magnitude difference in effectiveness, depending on the approach used to derive the candidate suggestions from ClueWeb12. We address the following research questions:

RQ1 Can we improve the quality of contextual suggestions based on ClueWeb12 collection by applying domain knowledge inferred from location-based APIs?

RQ2 What is the impact of the type of domain knowledge inferred on recommendation effectiveness?

RQ3 Can we improve the results by modeling the candidate selection process probabilistically?

2. EXPERIMENTAL SETUP

2.1 Dataset and Evaluation

The models and approaches presented in this paper have been evaluated by participating in the TREC 2014 Contextual Suggestion track (CS 2014). The test dataset consists of user profiles and contexts (50 cities situated in the United states), and the task is to provide a ranked list of suggestions for each (user, context) pair. The user profiles were constructed based on the training data, which consists of 100 example suggestions located in two cities, `Chicago, IL` and `Santa Fe, NM`. Each user profile represents the rating given by a crowd-source user to the examples. It consists of two ratings per suggestion, on a 5-point scale; one rating for a suggestion's description (i.e., a snippet), and another rating for its actual content (i.e., once the web page has been visited). In total, 299 (user, context) pairs have been judged. For these pairs, the top-5 documents of every submission have been judged by the assessors (profile owners). Judgments range from 0 (strongly uninterested) to 4 (strongly interested). In order to judge the geographical relevance of the suggestion, assessors were asked to judge whether the suggestion is located in the city it was suggested for. In addition to the crowd-source users, geographical relevance was also judged by NIST assessors. In both cases the geographical judgment ranges from 0 (not geographically appropriate) to 2 (geographically appropriate). Since submissions were allowed to be either from the Open Web or the ClueWeb12 collection, in the relevance judgments suggestions from the Open Web were identified by their URLs, while suggestions from ClueWeb12 collection were identified by their ClueWeb12 ids. For evaluating the performance of submitted runs, Precision@5 (P@5), Mean Reciprocal Rank (MRR), and a modified Time-Biased Gain (TBG) [4] were used as the "official TREC metrics". These metrics consider geographical and profile relevance (both in terms of document and description judgments), taking as thresholds a value of 1 and 3 (inclusive), respectively.

Our initial analysis is based on the two runs that our team submitted for evaluation. Both runs are based on sub-collections of candidate suggestions belonging to the ClueWeb12 collection; the first using the **GeographicFiltered** sub-collection that we describe in Section 3.3.1 and the second one using the **TouristFiltered** sub-collection described in Section 3.3.2. In our analyses, we refer to these runs by the name of the sub-collection that it is based on.

2.2 URL Normalization

A recurring pre-processing step to produce the various results reported in the paper concerns the normalization of URLs. We have normalized URLs consistently by removing their `www`, `http://`, `https://` prefixes, as well as their trailing "forwarding slash" character `/`, if any. In the special case of the URL referencing an `index.html` web page, the `index.html` string is stripped from the URL before the other normalizations are applied.

3. CONTEXTUAL SUGGESTION MODEL

In this section, we formulate the problem and describe a general framework for finding and providing personalized recommendations based on user preferences. Then, we describe the two main components of our model. The first component represents our approach for generating personalized ranked suggestions to the user based on her preferences (Section 3.2). The second component describes our approach for modeling the selection of candidates from ClueWeb12 collection (Section 3.3).

3.1 General Model and Problem Formulation

We assume that we have a set of suggestions – represented by a URL and a description – that have been judged by a set of users. The goal is to provide a ranked list of personalized suggestions for the users in new contexts. We exploit the user preferences and the given suggestion descriptions to model a textual user's positive and negative profiles into a similarity ranking model that is able to regulate the impact of the positive and negative profiles to generate a final scoring. We adopt a standard approach to content-based recommendation to determine a ranked list of suggestions:

$$P_{rel}(u, s) = P(s) \cdot SIM(u, s) \quad (1)$$

$P(s)$ is a probability that estimates how likely it is that suggestion s is relevant to the task, and controls the suggestions considered. We have experimented with different approaches to estimate this probability, described in detail in Section 3.3. Note that $P(s)$ does not necessarily depend on the user (the equivalent to the queries in traditional retrieval models), although it may depend on the context; it can be compared to the "prior probability of relevance" of traditional information retrieval models. If the range of $P(s)$ is restricted to discrete values 0 and 1, then $P(s)$ acts as a Boolean filter that selects candidate suggestions based on some features.

3.2 Personalization

Similarity function $SIM(u, s)$ represents the (content-based) similarity between user interests and candidate suggestions, and determines the personalization of recommendations to the user's interests. We follow an approach to modeling user preferences that has been used widely in the literature on contextual suggestion; consider for example [2, 11, 12]. Descriptions of the previously rated attractions provide the basis to construct two user profiles for each user. The positive profile u^+ represents the attractions that the user u likes, whereas the negative profile u^- represents the attractions that the user u dislikes. We use the value 2.5 (since ratings are on 0 to 4 scale) as a threshold to discriminate between liked and disliked attractions. We compute the similarity score between a candidate suggestion s and a user u as follows:

$$SIM(u, s) = \lambda \cdot SIM(u^+, s) - (1 - \lambda) \cdot SIM(u^-, s) \quad (2)$$

where $SIM(u^+, s)$ is the similarity between user's positive profile and the candidate document, while $SIM(u^-, s)$ is the similarity between user's negative profile and the candidate document. λ is the parameter that regulates the contribution of the $SIM(u^+, s)$ and $SIM(u^-, s)$ to the final score. We used 5-fold cross-validation on training data to find the optimal $\lambda = 0.7$, which was selected from $[0, 1]$ in 0.1 steps. For this experiment, we considered the cosine similarity (based on term frequencies). This has been done after transforming the suggestions and the user profiles from text-representation into a weighted vector-based representation. In this transformation, we filter out the HTML tags from the content of the documents, apply common IR parsing techniques including stemming and stop-word removal.

3.3 Selection Methods of Candidates

The selection of candidate suggestions plays an important role for providing good suggestions to the users. We have already presented how previous works address the contextual suggestion challenge by using a variety of public tourist APIs – including Google Places, WikiTravel, Yelp, and Foursquare – to obtain a set of suggestions. Queries issued are usually related to the target context (location), either given by its name (i.e., *Chicago, IL*) or its latitude and longitude coordinates (i.e., (41.85003, $-$87.65005)). Collecting suggestions from the ClueWeb12 collection poses however new challenges, different from "just" constructing the right query to issue at location-based web services. We formulate the problem of candidate selection from ClueWeb12 as follows. We have a set of contexts (locations) `C` – which correspond to US cities – provided by the `CS` track organizers. For each context $c \in C$, we generate a set of suggestions S_c from the ClueWeb12 collection, which are expected to be located in that context. We investigate two different approaches toward generating S_c. The first approach is to apply a straightforward geographical filter, based on the content of the ClueWeb12 documents. In the second approach, we exploit knowledge derived from external resources available on the Open Web about sites that provide touristic information, and apply this knowledge to ClueWeb12 collection.

3.3.1 Geographically Filtered Sub-collection

Our main hypothesis in this approach is that a good suggestion (a venue) will contain its location correctly mentioned in its textual content. Therefore, we implemented a content-based geographical filter (named `geo_filter`) that selects documents mentioning a specific context with the format `(City, ST)`, ignoring those mentioning the city with different states or those matching multiple contexts. With this selection method we aim to ensure that the specific target context is mentioned in the filtered documents (hence, being geographically relevant documents). The documents that pass this filter form sub-collection, **GeographicFiltered**. In Equation (1), we express this geographic filtering process through probability $P(s)$, which defines the probability of a ClueWeb12 document to be a candidate suggestion. In the simplest instantiation of our model, the probability of any document in ClueWeb12 to be included in the **GeographicFiltered** sub-collection is assigned to 0 or 1 depending on whether it passes the geo_filter:

$$P(s) = \begin{cases} 1, & \text{if } (s) \text{ passes geo\_filter} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

Approximately 9 million documents ($8,883,068$) from the ClueWeb12 collection pass this filter.

3.3.2 Applying Domain Knowledge to Sub-collection

The sub-collection described in Section 3.3.1 only takes the context into account, however, users are not equally satisfied by any type of document when receiving contextual suggestions: they expect those documents to be *entertaining* [4]. This implies that documents about restaurants, museums, or zoos are more likely to be relevant than stores or travel agencies [11]. We incorporate this information into our sub-collection creation process by sampling from the ClueWeb12 collection considering knowledge from the tourist domain. In the following, we present alternative ways to select candidate documents from ClueWeb12 collection using different filters. Each filter represents a domain knowledge about tourist information inferred from the Open Web.

Domain-Oriented Filter.

The first type of domain knowledge depends on a list of `hosts` that are well-known to provide tourist information, and are publicly available. We manually selected the hosts $\mathcal{H}$:= `{yelp, tripadvisor, wikitravel, zagat, xpedia, orbitz,` and `travel.yahoo}`. We consider these hosts as a domain filter to select suggestions from ClueWeb12 collection. The probability of a document in ClueWeb12 to be a candidate is either 0 or 1 depending only on its host. We define the probability $P(s)$ as:

$$P(s) = \begin{cases} 1, & \text{if host}(s) \in \mathcal{H} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

We refer to the set of documents that pass the domain filter defined in Equation (4) as *TouristSites*.

We assume pages about tourist information also have links to other interesting related pages, acknowledging the fact that pages on the same topic are connected to each other [3]. In order to maximize the extracted number of documents from the tourist domain we also consider the outlinks of documents from touristic sites. For each suggestion s $\in$*TouristSites*, we extract its outlinks `outlinks(s)` and combine all of them together in a set $\mathcal{O}$; including links between documents from two different hosts (`external links`) as well as links between pages from the same host (`internal links`). Notice that some of the outlinks may also be part of the *TouristSites* set, because of satisfying Equation (4). Next, we extract any document from ClueWeb12 whose normalized URL matches one of the outlinks in $\mathcal{O}$. The probability of document s to be selected in this case is defined as:

$$P(s) = \begin{cases} 1, & \text{if URL}(s) \in \mathcal{O} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

The set of candidate suggestions that pass this filter is called *TouristSitesOutlinks*.

Table 1: Number of documents for each part of the **TouristFiltered** subcollection.

Filter	Number of documents
TouristSites	175,260
TouristSitesOutlinks	97,678
Attractions	102,604
TouristFiltered	375,542

Attraction-Oriented Filter.

We will now consider a different type of domain knowledge, by leveraging the information available on the Foursquare API [1]. For each context $c \in C$, we obtain a set of URLs by querying Foursquare API. If the document's URL is not returned by Foursquare, we use the combination of document name and context to issue a query to the Google search API e.g., "`Gannon University Erie, PA`" for name *Gannon University* and context *Erie, PA*. Extracting the hosts of the URLs obtained results in a set of $1,454$ unique hosts. We then select all web pages in ClueWeb12 from these hosts as the candidate suggestions, with its probability defined in the same way as in Equation 4. The set of documents that pass the host filter is referred to by *Attractions*.

Together, the three subsets of candidate suggestions *TouristSites*, *TouristSitesOutlinks* and *Attractions* form our second ClueWeb12 sub-collection that we refer to as **TouristFiltered**.

TouristFiltered := *TouristSites*$\cup$*TouristSitesOutlinks*$\cup$*Attractions*

Table 1 shows statistics about the documents that pass each filter.

3.3.3 Candidates Selection Prior Probability

In Sections 3.3.1 and 3.3.2, we introduced probabilities, used as binary filters so far, to decide which documents from the ClueWeb12 collection should be selected as candidates. Each of these filters represents a different kind of knowledge related to tourism inferred from the Open Web. Now, we introduce three different methods to estimate prior $P(s)$ from the **TouristFiltered** sub-collection. Two non content-based priors exploit the correlation between relevance judgments, the depth of URLs, and the filters based on location-based social networks. The third prior is based on the content of the documents found by the best location-based filter. We evaluate the effect of these different estimations $P(s) = P_s^i$, where $i \in \{1, 2, 3\}$, by applying our contextual suggestion model on the **GeographicFiltered** sub-collection.

Previous research has shown that correlations between relevance and non content-based features such as document length can be exploited to improve retrieval results, e.g. [13]. Similarly, the authors

[1] `https://developer.foursquare.com/docs/venues/search`

of [9] presented a general model of embedding non content-based features of web pages (document length, in-link count, and URL depth) as a prior probability in the ranking model. By studying the correlation between the URL depth and the relevance of the webpage, they observed that the probability of being a home page is inversely related to URL depth. Motivated by these studies, we carry out a similar analysis on the URLs of ClueWeb12 documents and the URLs of documents in the `CS` track ground truth. We use the number of slashes in the *normalized* URL to find the depth; a more fine-grained analysis like the four categories used in [9] is deferred to future work. Table 2 shows the depth distribution of URLs in the ClueWeb12 collection. We estimate the relationship between URL depth and the prior probability of relevance by analyzing the ground truth of the Open Web qrels, the ClueWeb12 qrels, as well as the URLs in the Open Web qrels that also exist in the ClueWeb12 collection. We observe in Table 3 that approximately 72% of the documents in the Open Web qrels exist at the top levels of a website (depth zero and one), and that 75% of these are relevant, consistent with findings reported in the literature; we also find that the probability of a document being relevant is inversely related to the URL depth. However, the distribution of URL depth and their corresponding relevance is different for the ClueWeb12 qrels, where the highest percentage of webpages presented (and relevant) in those runs are at depth two, one, and three (in that order).

We can now estimate a prior probability of relevance at each URL depth by combining the statistics derived from the qrels (based on the correlation between URL depth and relevance of the ClueWeb12 ground truth information presented in Table 3 with the URL depth distribution of the complete collection, Table 2):

$$P_s^1 = P_s(depth) = P(rel|URL(depth = d_i)) = \frac{c(Rel, d_i)}{c(d_i)} \quad (6)$$

Similar to how we derive a prior probability of relevance from the URL depth data, we may also use the number of relevant documents generated by each subset filter to inform the prior probability of relevance. In this case, the probability of a document to be relevant considering that it has passed a filter is defined as follows:

$$P_s^2 = P_s(filter) = P(rel|filter_i) = \frac{c(Rel, filter_i)}{c(filter_i)} \quad (7)$$

Here, we use the statistics shown in Table 1 for the total number of documents that pass each **TouristFiltered** subset filter, to normalize the total number of relevant documents in each filter. The outcome is a filter-specific approach to estimate the prior probability of relevance. A document in **GeographicFiltered** sub-collection will get the prior probability of the filter that it passes, and the maximum prior is considered if multiple filters are satisfied. For the rest of the documents in **GeographicFiltered** sub-collection that do not satisfy any filter, they will get a prior estimated by the number of relevant documents in **GeographicFiltered** sub-collection normalized by its total number of documents.

The third prior P_s^3 is a content-based derived prior, where we use a language model constructed from documents that pass the best filter in terms of highest performance values. Specifically, we learn from the documents that pass the *Attractions* filter which were part of the **TouristFiltered** run to compute the prior probabilities. The goal is to boost documents from **GeographicFiltered** sub-collection that are similar to the attraction documents. We construct two different language models. The first is from documents that pass the *Attractions* filter and were judged as relevant. The second is from documents that pass the *Attractions* filter and were judged as not relevant. After that, both sets are processed in a similar way to generate a language model: first the stop words and non-alphabetic words are removed; then, terms are ranked based on their relative frequency in each set.

Table 2: Distribution of ClueWeb12 documents over URLs depth.

Depth	count	%
0	3,726,692	0.5
1	152,584,686	21.0
2	253,913,644	35.0
3	172,258,009	23.7
4	83,629,521	11.5
5	35,464,476	4.9
6	13,495,362	1.9
7	6,756,976	0.9
8	3,693,477	0.5
11	809,692	0.1

Table 4: Performance of **GeographicFiltered** and **TouristFiltered** runs. Analysis per relevance dimension is considered; description (desc), document (doc), and geographical (geo) relevance. We denote with (all) when desc, doc, and geo relevance are considered.

Metric	**GeographicFiltered**	**TouristFiltered**
P@5_all	0.0431	0.1374
P@5_desc-doc	0.2081	0.2222
P@5_desc	0.2828	0.2788
P@5_doc	0.2620	0.2949
P@5_geo	0.1549	0.4808
TBG	0.1234	0.5953
TBG_doc	0.1287	0.6379

4. RESULTS AND ANALYSIS

We now present how we have addressed the three research questions mentioned at the beginning of the paper and the results obtained in each situation. The measures are averaged after running a 5-fold cross-validation.

4.1 Effect of Using External Domain Knowledge for Candidate Selection

In this section we study RQ1: Can we improve the quality of contextual suggestions based on ClueWeb12 collection by applying domain knowledge inferred from location-based APIs? We compare the performance of our contextual suggestion model (see Section 3) used to rank suggestions from the two presented sub-collections **GeographicFiltered** and **TouristFiltered**. We show empirically that the additional information acquired from location-based social networks provides the evidence needed to generate high quality contextual suggestions.

Table 4 summarizes the results from the evaluation, where we are initially only interested in the entries that take all relevance criteria into account, labeled by suffix `_all`. Clearly, the effectiveness using the **TouristFiltered** sub-collection outperforms the **GeographicFiltered** results by a large margin. Also, among the results obtained for the runs submitted in TREC 2014, the former approach was superior to all other submitted ClueWeb12 runs, while the latter ranked near the bottom [6]. We should emphasize that the actual method that ranks the documents is exactly the same in both cases (Section 3.2), and hence, the difference in performance should be attributed to the differences in the candidate suggestions.

We inspect the evaluation outcomes in more detail, by considering relevance dimensions individually. Recall that assessments are made considering geographical and profile relevance independently. For the latter one, the user assessed both the document and the description provided by the method. Considering this information, we recomputed the evaluation metrics while taking into account the geographical relevance provided by the assessors, as well

Table 3: Distribution of URLs depth over the documents in Open Web qrels, documents from Open Web qrels that exist in ClueWeb12 collection, and the ClueWeb12 qrels.

	Open Web runs					overlap					ClueWeb12 runs			
	All		Relevant			All		Relevant			All		Relevant	
depth	count	%	count	%	depth	count	%	count	%	depth	count	%	count	%
0	23,657	66.31	9,271	67.69	0	8,847	87.78	1,891	81.54	0	159	1.79	22	2.53
1	2,113	5.92	636	4.64	1	473	4.69	180	7.76	1	1,856	20.89	208	23.88
2	6,957	19.50	2,758	20.14	2	423	4.20	149	6.43	2	4,537	51.06	479	54.99
3	2,211	6.20	853	6.23	3	210	2.08	52	2.24	3	1,412	15.89	86	9.87
4	434	1.22	113	0.82	4	78	0.77	19	0.82	4	688	7.74	57	6.54
5	179	0.50	47	0.34	5	36	0.36	17	0.73	5	168	1.89	13	1.49
6	52	0.15	5	0.04	6	11	0.11	3	0.13	6	43	0.48	3	0.34
7	61	0.17	6	0.04	7	1	0.01	0	0.00	7	9	0.10	1	0.11
8	14	0.04	8	0.06	8	13	0.13	8	0.34	10	9	0.10	2	0.23
11	1	0.00	13,697			10,079		2,319		13	4	0.05	871	
	35,679										8,885			

Table 5: Effect of domain knowledge filters on **TouristFiltered** run performance. Union means adding suggestions from the subset filter shown in column header of current column to the previous one. The percentage shows the relative improvement in effectiveness due to filter.

	TouristSites	∪ *TouristSitesOutlinks*		∪ *Attractions*		*Attractions*
Metrics	score	score	%	score	%	score
P@5_all	0.0392	0.0518	32.1	0.1374	165.3	0.1057
P@5_desc	0.0917	0.1200	30.9	0.2788	132.3	0.1973
P@5_doc	0.1008	0.1310	30.0	0.2949	125.1	0.2101
P@5_geo	0.2067	0.2659	28.6	0.4808	80.8	0.4667

as the description and document judgments, both separately and combined (that is, a document that is relevant both based on the description and when the assessor visited its URL, denoted with prefix *desc-doc*). Table 4 shows the analysis for each relevance dimension – note that the geographical, description, and document relevance assessments affect in the same way the evaluation metrics. When all the dimensions are considered (*all* prefix), the **TouristFiltered** sub-collection is significantly better than the **GeographicFiltered** one. However, the difference in the performance between the two sub-collections decreases when we look at the relevance of a document and its description, that is, when we ignore the geographical aspect of the relevance. This means that both sub-collections are similar in terms of their appropriateness to the users, where we only consider suitability with respect to the user's profile. At the same time, we observe that the **TouristFiltered** sub-collection is more geographically appropriate, implying that using the domain knowledge to select the candidates improves the performance in that dimension. A similar observation is found when looking at the best relevance dimension: for the **GeographicFiltered** sub-collection, the best performing dimension is the document description, whereas for the **TouristFiltered** sub-collection this is the geographical aspect.

4.2 Impact of Domain Knowledge Filters

In this section, we investigate RQ2: What is the impact of the type of domain knowledge inferred on recommendation effectiveness? We provide a deeper insight on why the domain knowledge-based sub-collection improves so much over the other sub-collection on the different relevance dimensions. Table 5 presents the contribution to the relevance dimensions of each of the **TouristFiltered** sub-collection subsets, where each subset was selected based on a different domain knowledge filter.

Table 6: Effect of using a prior-probability of relevance on the **GeographicFiltered** run performance. *no prior* means applying the general ranking model with $P(s) = 1$ for documents that pass the geo_filter.

Metrics	no prior	depth prior	filter prior
P@5_all	0.0431	0.0660	0.1300
P@5_desc-doc	0.2081	0.1024	0.1912
P@5_desc	0.2828	0.1273	0.2350
P@5_doc	0.2620	0.1468	0.2579
P@5_geo	0.1549	0.3515	0.4842
TBG	0.1234	0.3007	0.5574
TBG_doc	0.1287	0.3281	0.5988

We start modifying the run based on the **TouristFiltered** sub-collection by computing effectiveness based only on suggestions from the *TouristSites* subset (second column), then we add to them suggestions from *TouristSitesOutlinks*, and finally suggestions from *Attractions* are added. The main conclusion drawn from this table is that the larger improvement in performance occurs after adding the candidates from *Attractions* subset. It is interesting to note that the performance of this part alone (last column) is comparable to that of the whole sub-collection.

4.3 Effect of Prior Probability

In this section, we investigate RQ3: Can we improve the results by modeling the candidate selection process probabilistically? In this section, we investigate the effect of adding a prior probability that we discussed in Section 3.3.3 on the performance of the contextual suggestion model. Table 6 shows the effect of depth prior, and the effect of the filter prior when applying the contextual suggestion model on the **GeographicFiltered** sub-collection. As shown in this table, there is a significant improvement on the performance of the **GeographicFiltered** sub-collection after applying the two priors independently. We observe that the domain filter prior has more impact on the performance.

Next, we study the effect of the third prior, which is a content-based derived prior, where we use a language model constructed from documents that pass the *Attractions* filter which were part of the **TouristFiltered** run. We experimented with different cut-offs for selecting the top words to form the language model, precisely the top 500, $1,000$, and $5,000$ words. Without finding a clear relation between cutoff and performance, we present results based on the top $1,000$ terms. Table 7 shows the effect of using the similarity between the language models and the **GeographicFiltered** documents as prior. We observe that the performance is worse than without a prior (compare with first column of Table 6). However, this can also be explained by analyzing the number of documents that have judgments in the rankings generated by each method. We therefore reported also the percentage of judged documents in top-5 as well as the percentage of relevant documents among the judged, and the precision@5 with a condition that the document is judged. We now conclude that the language model generated from the relevant documents improves the performance.

Table 7: Language model constructed from relevant and not relevant documents.

Metrics	$\neg rel$	rel
P@5_all	0.0034	0.0067
P@5_doc	0.0444	0.0694
%judged@5	28.55	46.73
%rel of judged@5	38.18	54.75
P@5_doc(judged)	0.2185	0.4824

5. CONCLUSION

We have presented an approach for improving contextual suggestions based on ClueWeb12 collection. Our approach focused on selecting candidate documents from a large Web crawl (`ClueWeb12`), using tourist domain knowledge inferred from the location-based social networks from the Open Web. First, we presented Boolean filters for modeling selection of candidate suggestions, where each filter represents a different type of knowledge about the tourist domain. The filter is then integrated in the ranking model via a prior probability of relevance. Our empirical evaluation shows that using domain knowledge drawn from location-based social networks improves the performance of the contextual suggestion model when compared to the performance of the same ranking model, using the **GeographicFiltered** sub-collection that is created without any domain knowledge. Second, we found that the two sub-collections have different correlations with the dimensions of relevance considered in the evaluation (geographical and profile relevance), which opens up to investigate more the relation between the filters and the relevance dimension. Third, our analysis shows that filters used to create the **TouristFiltered** sub-collection vary in impact on contextual suggestion effectiveness. We exploit the knowledge of each filter to estimate a probability prior embeded in the ranking model using 5-fold cross-validation analysis. We also consider the correlation between URL depth of the document and its relevance, as an alternative prior. The results of this analysis on the **GeographicFiltered** sub-collection suggest that both priors improved the performance. The domain filter prior has more influence on the performance, suggesting that the domain knowledge filter captures relevance better than the depth prior. In the future, we aim to investigate the effect of the filter prior by incorporating different sources of information, such as the relation between the filter criteria and URL depth, and the relation between filter criteria and the individual dimensions of relevance.

References

[1] G. Adomavicius and A. Tuzhilin. Context-aware recommender systems. In F. Ricci, L. Rokach, B. Shapira, P. B. Kantor, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*, chapter 6, pages 217–253. Springer, Boston, MA, 2011.

[2] M.-D. Albakour, R. Deveaud, C. Macdonald, and I. Ounis. Diversifying contextual suggestions from location-based social networks. In *Proceedings of the 5th Information Interaction in Context Symposium*, IIiX '14, pages 125–134, New York, NY, USA, 2014. ACM.

[3] B. D. Davison. Topical locality in the web. In *In Proceedings of the 23rd Annual International Conference on Research and Development in Information Retrieval (SIGIR 2000*, pages 272–279. ACM Press, 2000.

[4] A. Dean-Hall, C. L. A. Clarke, J. Kamps, and P. Thomas. Evaluating contextual suggestion. In *Proceedings of the Fifth International Workshop on Evaluating Information Access (EVIA 2013)*, 2013.

[5] A. Dean-Hall, C. L. A. Clarke, J. Kamps, P. Thomas, and E. M. Voorhees. Overview of the TREC 2012 contextual suggestion track. In E. M. Voorhees and L. P. Buckland, editors, *Proceedings of The Twenty-First Text REtrieval Conference, TREC 2012, Gaithersburg, Maryland, USA, November 6-9, 2012*, volume Special Publication 500-298. National Institute of Standards and Technology (NIST), 2012.

[6] A. Dean-Hall, C. L. A. Clarke, J. Kamps, P. Thomas, and E. M. Voorhees. Overview of the TREC 2014 contextual suggestion track. In E. M. Voorhees and A. Ellis, editors, *Proceedings of The Twenty-Third Text REtrieval Conference, TREC 2014, Gaithersburg, Maryland, USA, November 19-21, 2014*, volume Special Publication 500-308. National Institute of Standards and Technology (NIST), 2014.

[7] A. Dean-Hall, C. L. A. Clarke, N. Simone, J. Kamps, P. Thomas, and E. M. Voorhees. Overview of the TREC 2013 contextual suggestion track. In E. M. Voorhees, editor, *Proceedings of The Twenty-Second Text REtrieval Conference, TREC 2013, Gaithersburg, Maryland, USA, November 19-22, 2013*, volume Special Publication 500-302. National Institute of Standards and Technology (NIST), 2013.

[8] R. Deveaud, M.-D. Albakour, C. Macdonald, and I. Ounis. On the importance of venue-dependent features for learning to rank contextual suggestions. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, pages 1827–1830, New York, NY, USA, 2014. ACM.

[9] W. Kraaij, T. Westerveld, and D. Hiemstra. The importance of prior probabilities for entry page search. In *SIGIR 2002: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 11-15, 2002, Tampere, Finland*, pages 27–34. ACM, 2002.

[10] M. Melucci. Contextual search: A computational framework. *Foundations and Trends in Information Retrieval*, 6(4-5):257–405, 2012.

[11] A. Rikitianskii, M. Harvey, and F. Crestani. A personalised recommendation system for context-aware suggestions. In *Advances in Information Retrieval - 36th European Conference on IR Research, ECIR 2014, Amsterdam, The Netherlands, April 13-16, 2014. Proceedings*, pages 63–74, 2014.

[12] M. Sappelli, S. Verberne, and W. Kraaij. Recommending personalized touristic sights using google places. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 781–784, New York, NY, USA, 2013. ACM.

[13] A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '96, pages 21–29, New York, NY, USA, 1996. ACM.

[14] P. Yang and H. Fang. Opinion-based user profile modeling for contextual suggestions. In *Proceedings of the 2013 Conference on the Theory of Information Retrieval*, ICTIR '13, pages 18:80–18:83, New York, NY, USA, 2013. ACM.