University of Amsterdam at TREC 2014: Contextual Suggestion and Web Tracks

Seyyed Hadi Hashemi Jaap Kamps
University of Amsterdam, Amsterdam, The Netherlands
{hashemi|kamps}@uva.nl

ABSTRACT

This paper presents the University of Amsterdam's participation in TREC 2014. For the Contextual Suggestion Track, we experimented with the use of anchor text representations in the language modeling framework, and base our runs either on full ClueWeb12 or the subset of touristic aggregators (e.g., tripadvisor) provided by the organizers of the track. We also look at the effectiveness of priors (in particular, PageRank) and ways of formulating the query based on the context. Our main finding is that the anchor text representation is effective for retrieving candidate attractions, and performs better than a standard document text index. A linear combination of both anchor and document text leads to further improvement. For the Web Track, we continued our experiment with the fusion of anchor text relative to the text-based baseline run. Our main finding is, again, that the combination of anchor and document text leads to improvement, and we demonstrate how the fusion weight can be used as a handle to tune the amount of risk acceptable for the risk sensitive evaluation.

1. INTRODUCTION

In this paper, we present the University of Amsterdam participation in the Contextual Suggestion and Web tracks at TREC 2014 in two relatively self-contained sections. Section 2 discusses the Contextual Suggestion Track, where we define appropriate language models and take advantage of anchor texts of the ClueWeb12 web pages. Section 3 discusses the Web track, in which we indicate how we fuse the baseline with the model based on anchor texts of the ClueWeb12.

2. CONTEXTUAL SUGGESTION TRACK

In this section, we present our participation in the TREC 2014 Contextual Suggestion Track. The main goal of this track is to investigate search techniques for complex information needs that are highly dependent on context and user interests. In each run, participants have to produce up to 50 ranked suggestions for each pair of profile and context based on a given set of profiles, a set of example suggestions and a set of contexts.

Each *profile* corresponds to a user who has judged the example suggestions' description and website of two seed cities (i.e., Chicago, IL and Santa Fe, NM). The user profiles contain a five-point scale rating for each pair of profile and example suggestion. Each example suggestion was rated by users and it includes a title, a description and a URL. The *context* is a city for which the suggestion rankings are going to be generated. This set has 50 randomly selected cities in the United States, and the name, state, latitude and longitude of each city are available in the contexts set.

There are two different kinds of submissions in the TREC Contextual Suggestion Track: 1) open web, or 2) ClueWeb12. Open web submissions usually crawl the aggregator websites such as Yelp, TripAdvisor or WikiTravel or use their API in order to gather potentially high quality and up-to-date suggestion candidates for the given contexts [4, 5]. The two kinds of submissions are evaluated separately and it is clear that providing contextual suggestion ranking based on the ClueWeb12 is harder than giving contextual suggestion ranking based on the open web using aggregator APIs. This motivated us to focus on ClueWeb12, were in principle a reusable test collection is created.

The track presents several challenges. First, retrieving relevant suggestions to the given context from the huge number of webpages on the web or ClueWeb12 (i.e., 733,019,372 web pages) requires working with scalable methods able to cope with this volume of data. Second, the task is a genuine needle-in-a-haystack problem, where many of the pages matching a context (city name as query) present no touristic attraction in the city, requiring suitable features that can function as prior probability to separate potentially interesting suggestion candidates from the rest of the web pages Third, personalizing suggestion rankings based on the user profiles is hard giving that these are based on only few examples. Fourth, for each candidate web page a descriptive title and descriptions need to be extracted, good enough to motivate the users to select the result on a hitlist.

Previous work relied either on the open web or ClueWeb12, and used the textual content of the web pages as document representation. Inspired by the effectiveness of anchor text representations [3, 10], we experiment with the incoming anchor text of suggestion candidates to estimate their relevancy to the contexts as well as profiles.

The rest of this section is organized as follows. In Section 2.1 we review some related work on Contextual Suggestion track at TREC 2012 and TREC 2013. In Section 2.2 we detail our models of Contextual Suggestion, and Section 2.3 is devoted to the experimental setup definition and reporting the experimental results. Finally, we present the conclusions and future work in Section 2.4.

2.1 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 aggrega-

tor 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. However, judging temporal aspect of the context was difficult for the NIST assessors, so it has not been used for the TREC 2013 and TREC 2014 [4].

In the TREC 2013, the participants could use either the open web or the ClueWeb12 dataset, but there were only seven submitted runs out of 34 that were ClueWeb12 runs [5]. 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. In the following, we will discuss more about the previous works on the ClueWeb12 dataset.

The approach of CWI team had two main parts. They retrieved the most relevant documents for the contexts, and then personalized the sub collection based on the cosine similarity between the documents and the user profiles. The Georgetown University team provided two runs. In the first run, they extract venue names of each context from WikiTravel and used them as a query to retrieve the relevant pages to those venues in the ClueWeb12-B collection. Finally, they tried to find the home page of the retrieved venues based on the retrieval score or the similarity between venue name and anchor text of pages. In their second approach, the Georgetown University team created category-based language models and used them to retrieve relevant documents to the category mentioned in the user profiles. IRIT team assigned one or more categories from WordNet and Google Places to each user and used the Terrier to retrieve documents relevant to user categories [5].

2.2 Approach

The Contextual Suggestion Track has two main challenges for ranking suggestions based on the user preferences. The first one is finding a way to answer the "what are the potentially good suggestions for the given context?" question. Moreover, the second main challenge is finding an effective approach to personalize the suggestion candidates based on users' profiles, so that the suggestion rankings could satisfy their needs. The open web submissions usually ignore the first challenge and just either aggregate suggestions of aggregators websites or use suggestions of a aggregator website like Yelp. When using ClueWeb12, this problem cannot be ignored.

Following the language modeling framework, we used the Bayes' Theorem in order to model contextual suggestion problem. As a result, we use this probabilistic model in order to rank the suggestion candidates. Regarding our two submissions, the only difference of them is the documents representative that is indexed for retrieving relevant suggestion candidates to the context and profile pairs.

In our first submission (i.e., *Model-Text*), we extract textual content of the ClueWeb12 web pages and index them to be able to retrieve the relevant suggestion candidates. Our main contribution that is included in our second submission (i.e., *Model-Anchor*) is based on incoming anchor text used to estimate suggestion candidates and contexts as well as suggestion candidates and profiles relevancy.

2.2.1 Model-Text

In this model, we wish to estimate $P(s|p,c)$ effectively and rank the suggestion candidates according to this probability. The Bayes' Theorem is invoked in order to more accurately determine the suggestion candidate relevancy to the given pair of profile and context:

$$p(s|c,p) = \frac{p(s)p(c,p|s)}{p(c,p)},$$

in which, $p(s|c,p)$ represents the relevance score of suggestion candidate s for the given user profile p and context c, $p(s)$ is the prior probability of being relevant for a suggestion candidate s, $p(p,c|s)$ is the probability of the presence of a user profile p and a context c given a suggestion candidate s, and $p(c,p)$ is the prior probability of the presence of a context c and a profile p. Since $p(c,p)$ is a constant for a given context c and profile p pair, it can be ignored for the purpose of suggestion ranking:

$$p(s|c,p) = p(s)p(c,p|s).$$

The probability $p(c,p|s)$ is computed by assuming conditional independence between context c and profile p. As a result, $p(c,p|s) \approx p(c|s)p(p|s)$ and the equation can be simplified as follows:

$$p(s|c,p) \propto p(s)p(c|s)p(p|s),$$

where $p(c|s)$ denotes probability of the presence of a context c given the suggestion candidate s, and $p(p|s)$ is probability of the presence of a user profile p given the suggestion candidate s. As a result, this model has 3 components, namely, the prior probability of a suggestion candidate s (i.e., $p(s)$), the probability of the presence of a context c given a suggestion candidate s (i.e., $p(c|s)$), and the probability of the presence of a user profile p given a suggestion candidate s (i.e., $p(p|s)$).

Prior.

First, $p(s)$ is the prior probability of each suggestion candidate, which could have a significant role on discriminating potentially high quality suggestion candidates from the other candidates. We had an observation upon the judgments of the ClueWeb12 submissions in Contextual Suggestion track at TREC 2013 that shows PageRank scores of the ClueWeb12 web pages judged as interesting or strongly interesting suggestions are usually much higher than the ClueWeb12 web pages judged as uninteresting, strongly uninteresting or could not load ones.

Figure 1 indicates that the average PageRank scores of the ClueWeb12 pages judged interesting is more than 50 times higher than the ones judged uninteresting suggestion candidates at QRel of TREC 2013. In this Figure, ClueWeb12 pages judged strongly interested or interested are counted as interesting suggestions, and the pages judged uninterested, strongly uninterested or could not load as uninteresting suggestions.

This observation motivates us to utilize PageRank of suggestion candidates in order to estimate the prior probability of them in our proposed model, so that we could discriminate high quality suggestion candidates from the ones that probably are not good venues for recommending to the users. Due to the availability of the PageRank of the ClueWeb12

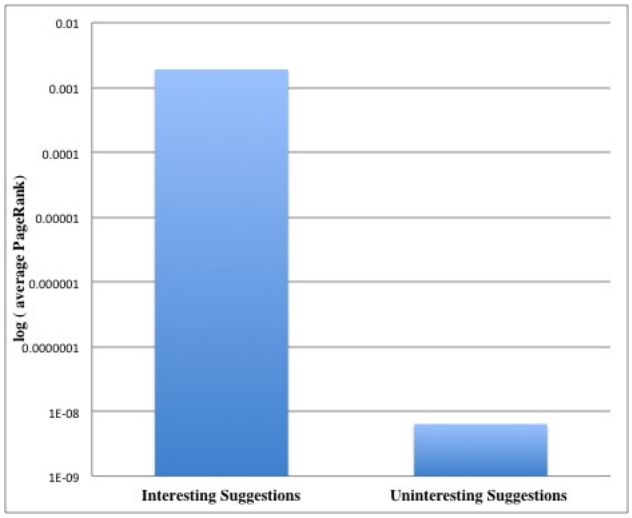

Figure 1: Average Page Rank scores of interesting and uninteresting ClueWeb12 candidates (TREC 2013 qrels, log scale).

dataset at the Lemur project webpage[1], we just normalize these PageRank scores and use it in our model.

Context.

Second, $p(c|s)$ is probability of the presence of a context c given a suggestion candidate s. In our first run at TREC 2014 Contextual Suggestion track, we extract the text content of the ClueWeb12 pages and index them using the Terrier IR platform [11]. Then, the relevance probability of a suggestion candidate s to a context c is estimated by a suggestion language model θ_s of a suggestion candidate s. Specifically, the following language model is used to estimate relevance of each context and suggestion candidate pair in the University of Amsterdam's submissions:

$$p(c|\theta_s) = \prod_{t \in c} p(t|\theta_s)^{n(t,c)},$$

in which, $p(t|\theta_s)$ is the probability of term t given the suggestion language model θ_s, and $n(t, c)$ is the number of times that term t occurs in context c. To avoid zero probabilities, the JM-smoothing [12] is used, so the probability $p(t|\theta_s)$ is estimated as follows:

$$p(t|\theta_s) = \lambda p(t|s) + (1 - \lambda)p(t),$$

where $p(t|s)$ is the maximum likelihood estimation of the occurrence of a term t in a suggestion candidate s, and $p(t)$ is the occurrence probability of the term t in the whole corpus. In our experiments, we use the default smoothing parameter $\lambda = 0.15$.

Usually, it is not easy to retrieve web pages that are interesting for the tourists by only giving the context to the retrieval system. In order to retrieve relatively more interesting suggestion candidates for the tourists, the contexts are expanded by some general touristic terms, which are partly mentioned in Table 1. This query expansion is beneficial to retrieve generally interesting venues like Museum

[1]http://www.lemurproject.org/ClueWeb12/PageRank.php

Table 1: Expansion terms for the context as query

Expanded query based on context (city name) plus ...
Food, cafes, desserts, sandwiches, coffee, ice cream, gourmet, local flavor, tea, bbq, dim sum, sushi, vegetarian, chicken wings, breweries, chocolate, art & entertainment, ticket sales, theater, galleries, arcades, venues, spas, massage, museums, jazz and blues, music venues, mini golf, bowling, yoga, comedy clubs, Pilates, sports teams, Nightlife, bars, wine bars, juice bars, event planning, dance clubs, gastro pub, lounges, jazz and blues, pubs, outdoor, zoos, parks, amusement park, garden, lake, travel service, Shopping, home decor, dept stores, stationery, tobacco shop, gourmet, grocery, shopping center, bookstores, farmers market, flea markets, hobby shops, religious orgs, airport, hotel, tour, landmark, monument, public service, travel services.

web pages related to the context rather than retrieving some uninteresting web pages in this task such as news about that context.

Retrieving relevant suggestion candidates to a given context among the ClueWeb12 data set is not an easy problem. As a result, it is beneficial to find a proper way to filter useless pages. To this aim, we filter the ClueWeb12 web pages based on the ClueWeb12 IDs that were released by the CWI team; therefore, we only consider the ClueWeb12 pages that are in the domains of the "yelp", "tripadvisor", "wikitravel", "zagat", "xpedia", "orbitz", and "travel.yahoo" websites. This subset has the advantage of including many potentially good quality aggregators pages, and the disadvantage of missing a considerable amount of documents that are relevant to the contexts, but are not in the aggregators domains.

Profile.

Third, $p(p|s)$ is probability of the presence of a profile p given a suggestion candidate s. We build positive and negative user profile based on the example suggestions' descriptions, which the user rates interesting and uninteresting, respectively. Finally, the user profile p and suggestion candidate s relevancy is estimated by the same suggestion language model θ_s that is used to estimate relevance of each context and suggestion candidate pair. The following equation indicates how the negative profile as well as the positive profile is considered for estimating this probability:

$$p(p|\theta_s) = w_{pos} p(p_{pos}|\theta_s) + w_{neg} p(p_{neg}|\theta_s),$$

in which, w_{pos} and w_{neg} are, respectively, weight of the positive user profile (i.e., p_{pos}) relevancy and the negative user profile (i.e., p_{neg}) relevancy in the final estimation of the profile relevancy to the suggestion candidate s. According to our experiments on our aggregator domain sub collection, $w_{pos} = 0.75$ and $w_{neg} = -0.25$ are the reasonable weights for suggestion candidates personalization.

Finally, as one of the challenges of the Contextual Suggestion track, we have to generate a title and a description for each of the suggestions. We extract the title of suggestions from the title tag of HTML content of the ClueWeb12 web pages. In order to render more informative sentences as a description, the description was extracted by first looking at

sentences that mention the context. If none of the sentences mentions the context, then we extract the text content of the description tag.

2.2.2 Model-Anchor

Retrieving relevant suggestion candidates to a given context is one of the difficult challenges of the TREC Contextual Suggestion Track. The problem is that there are a lot of irrelevant suggestion candidates to the given context among the retrieved suggestion candidates, which include context's terms in their contents. In fact, in the case that the content of the ClueWeb12 web pages is indexed, due to the size of the text content of the ClueWeb12 pages in comparison to the short query (i.e., context), many of the irrelevant ClueWeb12 pages have the chance of counting as relevant suggestions.

We make two attempts to overcome this problem at TREC Contextual Suggestion track. In our first run, we filter the ClueWeb12 and only consider the ClueWeb12 web pages that are in the aggregators' domains. In our second run, instead of using the whole text content of the ClueWeb12 pages, we decide to consider a better representative of them. In fact, to overcome the mentioned problem, rather than indexing the text content of web pages, it is better to index a good summary of the ClueWeb12 pages, which could be the anchor text of them. Since anchor text is a short summary of webpages, it is helpful for the kind of search that users tend to submit a short query [7]. Similarly, it could be also beneficial to index anchor text of the ClueWeb12 pages for the Contextual Suggestion and retrieve relevant suggestion candidates to the short context queries.

This approach tends to filter those documents that include the given context in their content, but they might only give some unimportant information about the context so that they cannot be a relevant suggestion for the context. Therefore, in our second submission at TREC 2014 Contextual Suggestion track, we index the anchor text of the ClueWeb12 pages that was extracted in [8] and estimate the context and suggestion candidate as well as user profile and suggestion candidate relevance scores based on it. However, the retrieval model of the second submission of the University of Amsterdam is exactly same as the first submission. Specifically, PageRank of each suggestion candidate s is considered as its prior probability (i.e., $p(s)$), and the anchor text of the suggestion candidate s is used to build its suggestion language model θ_s.

2.2.3 Model-LC

As it is mentioned before, *Model-Anchor* tends to have a better precision than *Model-Text*, and it is also expected that the *Model-Text* has a better recall than *Model-Anchor* in retrieving the relevant suggestions. As a result, we decide to use Linear Combination method to fuse the *Model-Text* with the *Model-Anchor* suggestion ranking. To this aim, the following equation is used to combine the two rankings:

$$p(s|c,p,\beta) = \beta p_{Model-Text}(s|c,p)+(1-\beta)p_{Model-Anchor}(s|c,p),$$

where $p(s|c,p,\beta)$ is the relevance probability of a suggestion s to a context c and a profile p based on the weight β and $1-\beta$ given to *Model-Text* and *Model-Anchor* rankings. The performance of this model and the optimal β parameter is discussed in Section 2.3.2.

2.2.4 Model-Anchor-Full

The aggregators sub collection has many good suggestion candidates, but it is a tiny collection (i.e., 175,260 ClueWeb12 pages) and misses many of the suggestion candidates that are useful to be considered as a high quality suggestion candidate. For instance, the home page of suggestions are not included in this sub collection. As a result, we indexed the ClueWeb12-full anchor text, and run our proposed model based on this dataset. This model is exactly the same as *Model-Anchor*, but based on the ClueWeb12-full and without filtering the collection based on the aggregators' domains.

In this run, we personalize the ranking based on the positive, negative and neutral profile. We build the profiles based on the user judgments of example suggestions. Specifically, in the case that the user score either the description or the suggested website more than 2, that description has been added to the positive profile. We also create neutral user profile, and fill in it by example suggestions descriptions that both of theirs website and description were scored 2 in the user profile. Descriptions of the rest of the example suggestions are considered as a negative user profile.

In order to estimate a profile p and a suggestion candidate s relevancy (i.e., $p(p|s)$), we use the suggestion language model θ_s that is introduced in previous section. The following equation indicates how the positive, negative and neutral profiles are applied to the profile and suggestion candidate relevancy estimation:

$$p(p|\theta_s) = w_{pos}p(p_{pos}|\theta_s) + w_{neu}p(p_{neu}|\theta_s) + w_{neg}p(p_{neg}|\theta_s),$$

where w_{neu} is the weight of the neutral user profile (i.e., p_{neu}) relevancy in the final estimation of the profile relevancy to the suggestion candidate s. Due to the fact that the suggestion judgments is sparse, it is not easy to find the optimal weights for this model. However, we find $w_{pos} = 3$, $w_{neu} = 1$ and $w_{neg} = -1$ as reasonable weights for Model-Anchor-Full personalization.

2.3 Experiments

An extensive set of experiments are designed to address the following research questions:

- How efficient is using *PageRank* scores as prior probabilities of suggestion candidates?
- What is the effect of using query expansion in our proposed model rather than only using city name as a query?
- How do our two submissions (i.e., *Model-Text* as a baseline and *Model-Anchor*) perform compared to each other? Is it beneficial to take into account the anchor text of suggestion candidates in order to estimate the relevancy of them to the given context and profile pair?
- How efficient is the linear combination of *Model-Text* and *Model-Anchor*? What is the optimal parameter of the linear fusion?
- How efficient is the *Model-Anchor* in comparison to the *Model-Text* in each context?
- How efficient is the *Model-Anchor* in comparison to the *Model-Text* in each profile?

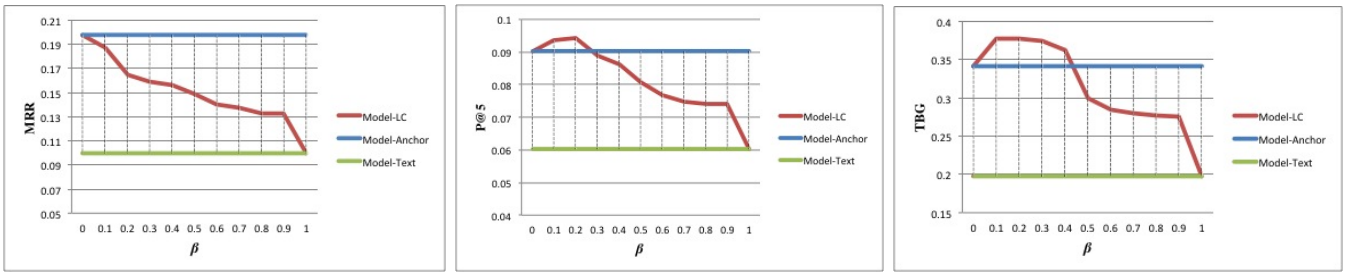

Figure 2: Impact of relative fusion weight (β) on MRR, $P@5$ and TBG.

- Could anchor text be helpful to retrieve suggestion candidates related to context and profile pairs without the help of filtering based on the aggregators domains?
- What is the effect of using neutral, negative, and positive profiles in Model-Anchor-Full suggestion ranking personalization?
- How efficient are our proposed models based on the fraction of suggestions judged in the TREC 2014 judgments?

2.3.1 Experimental setup and metrics

In this section, we describe dataset and evaluation metrics. We build our models based on the ClueWeb12 dataset that was created to support research on Information Retrieval in 2012. This dataset consists of 733,019,372 English webpages and its size is 27.3 TB that shows how big this dataset is. The information about the PageRank of ClueWeb12 web pages is available at the Lemur project website. Therefore, we do not redo the effort for calculating the PageRank and use the PageRank score, which is provided there.

As it is mentioned before, our main idea is using the anchor text of the ClueWeb12 web pages to improve the precision of our proposed Contextual Suggestion model. It is worth mentioning that Djoerd Hiemstra extracted the anchor text of the ClueWeb12 and we reuse this data as an input to our model. The data contains anchor text of about 64 percent of the ClueWeb12-full dataset (i.e., 0.5 billion pages) [8]. Each record of this data consists of "ClueWeb12 ID", "URL" and "Anchor Text".

The users whose profiles were given to the TREC Contextual Suggestion participants and also the NIST assessors judged the output suggestions of our proposed models. The evaluation results of our two submissions (i.e., *Model-Text* and *Model-Anchor*) will be discussed in Section 2.3.2. We also evaluate the performance of the *Model-LC* based on the TREC 2014 official suggestion judgments. However, due to the fact that a small fraction of the submissions are based on the ClueWeb12 dataset, the provided suggestions judgments does not have enough information to judge the *Model-Anchor-Full* that is not submitted for the TREC Conference. In addition, the major part of the ClueWeb12 judged suggestions in TREC 2014 is a subset of the aggregators sub collection. As a result, it is difficult to evaluate suggestion candidates outside of this sub collection. To overcome this problem, we map the judged open web URL to the ClueWeb12 pages, and evaluate the *Model-Anchor-Full* by the expanded ClueWeb12 suggestion judgments.

In order to map the judged open web URLs to the ClueWeb12 pages, we consider exact matching of the URLs and also matching of the normalized URLs (i.e., the URLs that are normalized by removing the "http://", "https://", "www", and the last slash). Finally, we have mapped 1623 out of 15,480 judged open web URLs to the ClueWeb12 pages, but it is still not enough to judge the *Model-Anchor-Full*.

The evaluation of the Contextual Suggestion track is based on the two common Information Retrieval metrics (i.e., $P@N$ and MRR), and also one contextual suggestion specific made metric (i.e., TBG or Time-Biased Gain) [6].

Table 2: Effectiveness of the uniform and PageRank-based prior probability (official qrels)

Method	p@5	MRR	TBG
Model-Text-Uniform	0.0279	0.0707	0.1198
Model-Text-PageRank	**0.0602**	**0.0994**	**0.1969**
Model-Anchor-Uniform	0.0863	0.1858	0.3239
Model-Anchor-PageRank	**0.0903**	**0.1979**	**0.3411**

2.3.2 Experimental Results

To demonstrate how the idea of using anchor text as suggestions representative in estimating suggestion candidates relevancy to contexts or profiles can improve the contextual suggestion performance, we submitted *Model-Text* as a baseline in TREC 2014. We compare various aspects of *Model-Anchor* and *Model-LC* to the baseline (i.e., *Model-Text*) in the following paragraphs.

First of all, we want to test the effectiveness of using PageRank score as a prior probability of a suggestion candidate. Table 2 indicates the evaluation results of *Model-Text* and *Model-Anchor* with uniform as well as PageRank prior probability. As it is expected based on our observation in Section 2.2.1, taking in to account PageRank scores of suggestion candidates as prior probabilities of them is helpful to discriminate potentially relevant suggestion candidates from potentially irrelevant ones.

According to Table 2, using PageRank score as the prior probability of suggestion candidates has more effect on the *Model-Text* suggestion ranking than the *Model-Anchor* suggestion ranking. This observation shows that the retrieved suggestion candidates based on the web pages' anchor text usually have high PageRank scores. As a result, using PageRank score as a prior probability of suggestion candidates has a small effect on the *Model-Anchor* improvement. On the other hand, using PageRank scores of suggestion candidates significantly improves the *Model-Text* suggestion ranking. This experiment shows that using Pagerank scores as prior probabilities generally improves the suggestion rank-

Table 3: Effectiveness of city name versus expanded query (official qrels)

Method	p@5	MRR	TBG
Model-Text-QUnExpanded	0.0328	0.0410	0.0441
Model-Text-QExpanded	**0.0602**	**0.0994**	**0.1969**
Model-Anchor-QUnExpanded	0.0502	0.1373	0.1919
Model-Anchor-QExpanded	**0.0903**	**0.1979**	**0.3411**

Table 4: Effectiveness of combined text and anchor (linear combination, $\beta = 0.2$, official qrels)

Method	p@5	(%)	MRR	(%)	TBG	(%)
Model-Text	0.0602	–	0.0994	–	0.1969	–
Model-Anchor	0.0903	50%	**0.1979**	99%	0.3411	73%
Model-LC	**0.0943**	57%	0.1643	65%	**0.3780**	92%

ing. Therefore, in the rest of the paper, we always use the PageRank score as a prior probability of suggestion candidates in our experiments.

In order to evaluate the effect of query expansion in retrieving relevant suggestion candidates to the given city, we run *Model-Text* and *Model-Anchor* without adding any term to the given city. As it is indicated in Table 3, it has a considerable effect on the suggestion ranking of *Model-Text* as well as *Model-Anchor*. Specifically, expanding the city name with some general tourist attraction terms (e.g., restaurant) helps the proposed model to retrieve relevant suggestion candidates rather than retrieving relevant web pages to the given city, which is not guaranteed to be a proper suggestion. This experiment shows the value of using query expansion in retrieving relevant suggestion candidates. In the rest of the experiments, we always take query expansion into account in our suggestion ranking models.

Table 4 reports the evaluation results of *Model-Text*, *Model-Anchor* and *Model-LC*. In this experiment, it is clear that *Model-Anchor* performs better than the baseline in terms of the average $P@5$, MRR and TBG. The improvement in the MRR proves that although using anchor text might worsen the recall of the relevant suggestion candidates, it definitely improves the precision and the first relevant suggestion in the ranking. Moreover, *Model-Anchor* improves the baseline in terms of the time-biased gain, so the suggestions provided by the *Model-Anchor* is more interesting for the users and they spend more time on exploring the suggestions in comparison to the Model-Text suggestions.

Moreover, we evaluate the performance of the linear fusion of *Model-Text* with *Model-Anchor* rankings and also answer the question "what is the optimal β value for the *Model-LC*?" We test the performance of the Model-LC for different values of $\beta \in [0, 1]$ with 0.1 intervals. Figure 2 indicates that, for $\beta = 0.1$ and $\beta = 0.2$ (hence most weight on the Anchor text), the performance of *Model-LC* is better than *Model-Anchor* in terms of $p@5$ and TBG metrics. However, if you increase the β value, *Model-Text* will add noises to the ranking and worsen the overall performance. On the other hand, the *Model-LC* is not able to improve the MRR of the *Model-Anchor*. This observation also demonstrates the advantage of *Model-Anchor* over the *Model-Text* based on the MRR metric. The performance of *Model-LC* for $\beta = 0.2$ is also compared against the *Model-Text* and the *Model-Anchor* in Table 4.

In order to have a more detail understanding on the evaluation results, we turn to a context-level as well as profile-level analysis of the comparison illustrated in previous experiment. First, we plot the differences in average MRR as well as TBG between *Model-Anchor* and *Model-Text* per context in Figure 3 and 4, respectively.

As it is obvious in the Figure 3, the *Model-Anchor* improves the performance of the baseline in most of the contexts, and worsen it in just 5 out of 50 contexts. Although the *Model-Anchor* worsen the performance of the contextual suggestion in context number 138, which is corresponded to the "Clarksville" city in central Tennessee, more than the other 4 contexts, it is interesting to analyze it and find the reason. In order to find the reason, we look at suggestion rankings of the *Model-Anchor* as well as the *Model-Text* for the context and profile pair that judged the $1st$ suggestion of the *Model-Text* ranking and the $4th$ suggestion of the *Model-Anchor* ranking as relevant ones.

We find out that the city name of this context makes it difficult for *Model-Anchor* to improve the baseline in the tiny aggregators sub collection. As a matter of fact, there are more than 20 cities with the name of Clarksville in different states of the United States. Therefore, it is not easy to find the relevant suggestions of the Clarksville in Tennessee in the top-5 ranks based on the only anchor text of the pages in the tiny aggregators sub collection. For this context, the *Model-Anchor* retrieves the disambiguation page of the wikitravel for Clarksville cities. On the other hand, *Model-Text* provides the wikitravel page of the "Nashville" city in the state of Tennessee as the $1st$ suggestion in the ranking. Due to the fact that the Nashville is just 47.8 miles further than the Clarksville in the state of Tennessee, this page is judged as a relevant suggestion. However, the Clarksville is not mentioned in the anchor text of the Nashville wikitravel page, and it is reasonable that it is not included in the top-5 ranking of the *Model-Anchor*. As illustrated in Figure 4, a similar pattern is observed for the evaluation by the TBG metric.

Figure 5 is a profile-level analysis of the comparison of the *Model-Anchor* against the baseline. According to this figure, *Model-Anchor* improves the average MRR of the baseline for most of the given profiles, but only worsen 8 out of the 299 profiles. We analyze two of the profiles in which the *Model-Anchor* worsen the performance of the baseline more than the other 6 ones.

First, due to the fact that the profile 948 only judged 2 out of the 50 contexts, his/her average judgments is highly affected by the suggestions of the Clarksville TN context that we analyzed it before. Second, the reason of the difference between the average MRR of *Model-Anchor* and *Model-Text* for the profile 700 is his/her judgment in "Kalamazoo MI" context. It is so interesting to know that the *Model-Anchor* suggests the WikiTravel page of the Kalamazoo city that is judged as an irrelevant suggestion in the first rank. On the other hand, the first rank of the *Model-Text* suggestion is the WikiTravel page of the state of Michigan that is judged as a relevant suggestion. It seems that the profile looks for a broader suggestions rather than some specific suggestions for the context. According to the experiments, it is crystal clear that, in comparison to the baseline, the *Model-Anchor* tends to suggest more precise suggestions for the contexts

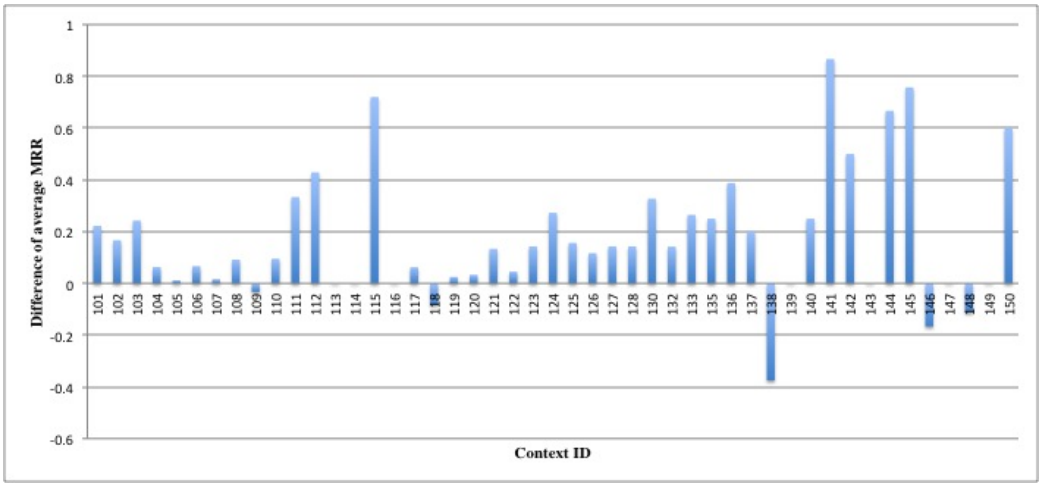

Figure 3: Difference between the average MRR of *Model-Anchor* and *Model-Text* per context.

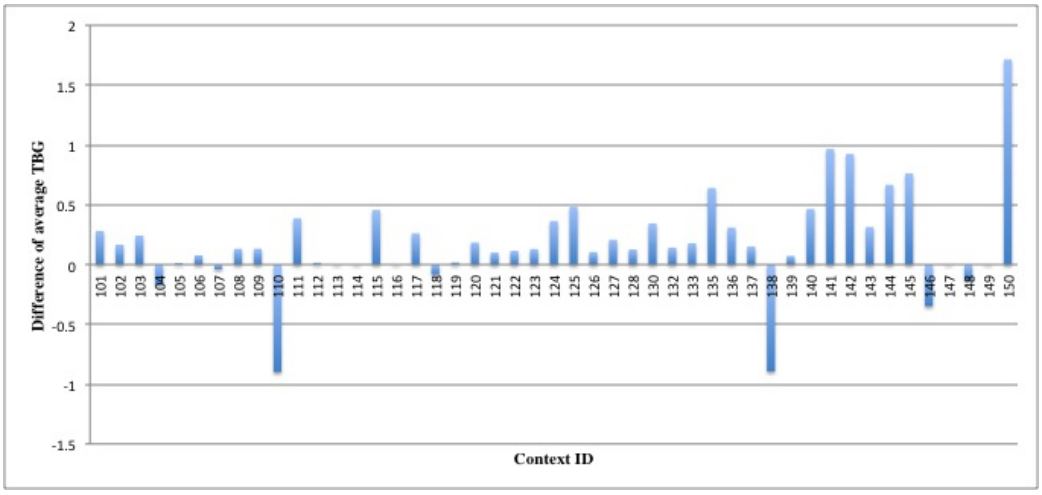

Figure 4: Difference between the average TBG of *Model-Anchor* and *Model-Text* per context.

and profiles pairs, so that it is able to improve the ranking of suggestion candidates.

In this experiment, we analyze the performance of the *Model-Anchor-Full* based on the expanded ClueWeb12 judgments. Due to our observation that only 3 out of 74,750 top-5 suggestions of *Model-Anchor-Full* is judged in the expanded TREC 2014 judgments, it is not possible to compare *Model-Anchor-Full* performance to other models in term of the *precision*@5. In addition, only 14 suggestions of *Model-Anchor-Full* are judged in our expanded ClueWeb12 judgments, which makes it too difficult to evaluate the performance based on any other metric.

However, according to the 14 judged suggestions, Table 5 indicates that the performance of the *Model-Anchor-Full* in providing relevant suggestion candidates is acceptable. As it is shown in Table 5, only 2 out of the 14 suggestions scored less than 2. As a result, this observation demonstrates that the *Model-Anchor-Full* rarely suggest irrelevant suggestion candidates, which is so helpful for improving the performance of the contextual suggestion approaches in terms of *precision* and *MRR*.

As it is mentioned in the previous experiment, the TREC suggestion judgment is sparse, and it is not easy to evaluate the runs that are not submitted to the track. As a result, in the next experiments, we decide to evaluate the proposed models based on the fraction of suggestions, which are judged in the TREC 2014 judgments. In Table 6 and 7, it is indicated that how many percent of the top 5, 10, and 20 suggestions in the given rankings are judged. Moreover, Table 6 and 7 show how many percent of the judged suggestions are relevant. In these experiments, in the case that a suggestion is marginally or precisely geographically appropriate and judged strongly interesting or interesting, it is counted as a relevant suggestion. The rest of the suggestions are considered as irrelevant ones.

Due to the difficulty of the suggestion ranking on the ClueWeb12-Full, we decide to use the neutral profile in order to improve the performance of the *Model-Anchor-Full* in retrieving relevant suggestions. Table 6 shows the effective-

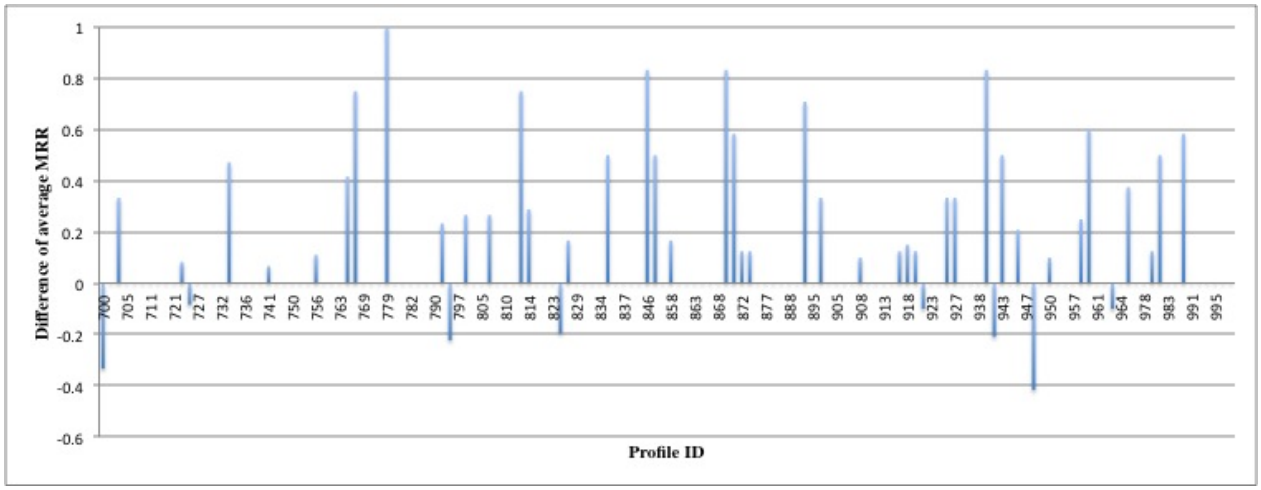

Figure 5: Difference between the average MRR of *Model-Anchor* and *Model-Text* per profile.

Table 6: Fractions of judged pages, and fraction of relevant within the judged pages, for *Model-Anchor-Full* based on different personalization approaches.

Method	Personalization Pos.	Neg.	Neu.	TBG	Judged (%) Top-5	Top-10	Top-20	Relevant Judged (%) Top-5	Top-10	Top-20
Model-Anchor-Full				0.0004	00.73	00.36	00.19	27.27	27.27	27.27
Model-Anchor-Full	√	√	√	**0.0016**	00.20	00.10	00.08	**33.33**	**33.33**	**40.00**
Model-Anchor-Full	√	√		0.0000	00.26	00.20	00.14	25.00	16.66	25.00
Model-Anchor-Full	√			0.0000	00.00	00.00	00.00	00.00	00.00	00.00

Table 7: Impact of qrels expansion (mapping judged URLs to ClueWeb12 IDs) on fractions of judged pages, and fraction of relevant within the judged pages

Method	Qrels	TBG	Judged (%) Top-5	Top-10	Top-20	Relevant Judged (%) Top-5	Top-10	Top-20
Model-Text	(Official Qrels)	0.1969	100.00	54.12	30.43	08.17	08.98	10.42
Model-Anchor	(Official Qrels)	0.3411	100.00	65.98	37.58	11.81	11.10	10.54
Model-LC	(Official Qrels)	**0.3780**	82.27	62.33	41.80	14.79	13.90	12.78
Model-Anchor-Full	(Official Qrels)	0.0016	00.06	00.03	00.05	**100.00**	**100.00**	**66.66**
Model-Anchor-Full	(Expanded Qrels)	0.0016	00.20	00.10	00.08	33.33	33.33	40.00

ness of the idea of using neutral profile in suggestion ranking personalization. In this experiment, we evaluate models based on the expanded suggestion judgment. According to this table, none of the suggestions retrieved by the model that only use positive profile is judged; as a result, it is not possible to compare this model to other variations of the *Model-Anchor-Full*. However, Table 6 indicates that the model whose personalization is based on the positive, negative and also neutral profiles is performing better than the model ignoring neutral profile in the suggestion ranking personalization. In the rest of the experiments, *Model-Anchor-Full* is the one using positive, negative and neutral profiles to personalize the suggestion ranking.

Finally, Table 7 provides a fair metric to compare *Model-Anchor-Full* against our other proposed models. Table 7 indicates that, based on the evaluation that only considers judged suggestions, the *Model-Anchor-Full* is more effective than our other proposed models.

2.4 Conclusion and Future Work

In this paper, we studied Contextual Suggestion problem through proposing a model based on the Bayes' Theorem to retrieve relevant suggestion candidates to the given context and profile pair. We used anchor text of the ClueWeb12 web pages in order to precisely retrieve the relevant suggestions to the given context. We tested our proposed model on the aggregators sub collection of the ClueWeb12 as well as the ClueWeb12-full dataset. We also fuse the proposed model that is based on the anchor text with the model based on the text content of the web pages. The experimental results indicated that the idea of using anchor text is promising in retrieving relevant suggestion candidates, and the linear fusion of the proposed model based on the anchor text with the model based on the text content improves the performance of the contextual suggestion ranking in terms of $p@5$ and TBG. As a future work, we continue to work on defining appropriate language models, investigating other priors

Table 5: Relevant Descriptions and Documents of *Model-Anchor-Full* based on 14 profile, context, suggestion triples (expanded qrels)

Prof.	Cont.	ClueWeb12 ID	Top-5	Des.	Doc.
983	133	ClueWeb12-0007wb-23-21753	Yes	3	3
801	110	ClueWeb12-1601wb-75-03301	Yes	2	2
938	110	ClueWeb12-1601wb-75-03301	Yes	1	1
834	110	ClueWeb12-0000wt-00-11531	No	2	2
748	112	ClueWeb12-1713wb-89-10896	No	2	2
855	112	ClueWeb12-1713wb-89-10896	No	2	2
898	112	ClueWeb12-1713wb-89-10896	No	2	2
976	112	ClueWeb12-1713wb-89-10896	No	3	3
765	133	ClueWeb12-0007wb-23-21753	No	2	2
852	133	ClueWeb12-0107wb-00-20064	No	1	0
918	133	ClueWeb12-0007wb-23-21753	No	3	3
945	133	ClueWeb12-0007wb-23-21753	No	1	2
980	133	ClueWeb12-0007wb-23-21753	No	2	2
726	146	ClueWeb12-0207wb-47-26667	No	3	3

and query expansions including (sub)domain classifiers, and improving the document and anchor text representations by including titles and other sources of annotation, such as can be found in Social Media. Our overall goal is to contribute to the building of a reusable test collection for contextual suggestion.

3. WEB TRACK

In this section, we present our participation in TREC 2014 Web track. The goal of this track is to explore and evaluate retrieval approaches over large-scale subset of the Web (i.e., ClueWeb12 dataset that includes 733,019,372 English web pages) [2]. The Web track organizers provide topics, which were developed with the information extracted from the logs of commercial Web search engines. In order to test different aspects of the approaches, topics contain both broad and specific queries. In this track, the ClueWeb12 dataset is used as a data collection, but participants are also eligible to submit their runs based on the ClueWeb12-B13 dataset, which is smaller than ClueWeb12-full collection.

In this track, we want to answer these research questions:

1. What is the effect of using web pages' anchor texts to estimate their relevance to the given query?
2. What is the performance of the linear combination of our proposed approach with the baseline provided by the Web track organizers?
3. What are the optimal weights of the linear combination of the rankings based on the topics and the judgments of the TREC2013?
4. How does the *Model-LC* perform in the risk-sensitve task?

In order to participate in the Web track, we have used the ClueWeb12-full collection. Specifically, we tackle the Adhoc Retrieval task and propose an approach based on the anchor text of the ClueWeb12-full dataset. We have used the Bayes' Theorem to simplify the problem and take into account the prior probability of web pages. The anchor texts of web pages are indexed by the Terrier IR platform, and the relevance probability of them to the queries are estimated by web pages' language models. We also have used the Linear Combination method to fuse the baseline with the proposed approach. The optimal weights of the linear combination are learned by the logistic regression based on the TREC2013 judgments.

The rest of this section is organized as follows. In Section 3.1 we detail our proposed models for the Web Track Adhoc retrieval task, and Section 3.2 is devoted to a report of the experimental results. Finally, we present the conclusions and future work in Section 3.3.

3.1 Approach

We propose two approaches for the Web Track Ad-hoc retrieval task. The first approach is the model that is used anchor text as webpages' representatives, and the second one is the linear fusion of the baseline with the first proposed model.

3.1.1 Model-Anchor

The main challenge of the Web Track is finding most relevant results among the huge number of candidates (i.e., more than 0.7 billion ClueWeb12 pages). Test topics of the Web Track are relatively short queries that make it difficult to find the relevant results based on the whole text content of the web pages. Since anchor text is a short and informative summary of web pages, it is effective to use it in the kind of search that users tend to submit short queries [7]. As a result, we indexed anchor text of the ClueWeb12 webpages in order to precisely retrieve relevant webpages.

In order to retrieve relevant web pages to the given queries, we have used the following Bayes' Theorem:

$$p(p|q) = \frac{p(p)p(q|p)}{p(q)},$$

where $p(p|q)$ represents the relevance score of web page p for the given query q, $p(p)$ is the prior probability of being relevant for a web page p, $p(q|p)$ is the probability of the presence of a query q given a web page p, and $p(q)$ is the prior probability of the presence of a query q. Since $P(q)$ is a constant for a given query q, it can be ignored for the purpose of web page ranking. As a result, the equation is simplified as follows:

$$p(p|q) = p(p)p(q|p).$$

This model contains prior probability of web pages and probability of the presence of a query q for the given web page p. In our proposed model, the prior probability of the web pages are estimated by their Pagerank scores, which are available at the Lemur project web page. Moreover, the ClueWeb12 anchor text is indexed by the Terrier IR platform [11], and the probability of the presence of a query q in a web page p is estimated by a web page language model θ_p of a web page p. As a matter of fact, the following web page language model is used to estimate this probability:

$$p(q|\theta_p) = \prod_{t \in q} p(t|\theta_p)^{n(t,q)},$$

in which, $p(t|\theta_p)$ is the probability of term t given the web page language model θ_p, and $n(t,q)$ is the number of times that term t occurs in query q. To avoid zero probabilities, the JM-smoothing [12] is used, so the probability $p(t|\theta_p)$ is

estimated as follows:

$$p(t|\theta_p) = \lambda p(t|p) + (1-\lambda)p(t),$$

where $p(t|p)$ is the maximum likelihood estimation of the occurrence of a term t in a web page candidate p, and $p(t)$ is the occurrence probability of the term t in the web page repository. In our experiments, we use the default smoothing parameter $\lambda = 0.15$ in the Terrier.

3.1.2 Model-LC

As we mentioned before, the Web track 2014 topics are relatively short queries, but not all of them. For instance, for the query "how has african american music influence history", it is better to also take advantage of the text content of the web pages. Therefore, we have used the linear combination method to fuse the Model-Anchor ranking with the baseline. Specifically, the following equation is applied to fuse the rankings:

$$p(p|q, w_1, w_2) = w_1 p_{Model-Anchor}(p|q) + w_2 p_{baseline}(p|q),$$

where $p(p|q, w_1, w_2)$ is the relevance probability of a web page p to a query q based on the weight w_1 and w_2 given to *Model-Anchor* and *baseline* rankings. The optimal weights of the linear combination are learned based on the TREC 2013 queries and judgments.

In order to learn a model to estimate the weights, in the preprocessing phase, we filter the documents, which were not retrieved by either the *baseline* or the *Model-Anchor* for each query of TREC 2013 topics. Then, we use relevance variable $c \in \{0, 1\}$ to denote whether a web page p is relevant to a given query q or not. In fact, we want to learn the unknown parameters θ (i.e., w_1 and w_2 in the linear combination of the rankings) of the probability $p_\theta(c = 1|p)$.

For each web page p, we generate training set as $T = \{(p, q, p_l)|p \in D_{qrel}, q \in Q, p_l \in \{0, 1\}\}$, where D_{qrel} denotes the documents judged in the web track 2013 qrel, Q is the TREC 2013 queries and p_l indicates the label of web page p based on the TREC 2013 judgments. In particular, $p_l = 1$ if web page p is relevant to the query q.

The members of set T can be divided into positive and negative instances based on the relevance judgment of p_l of web page p for a give query q. As a result, the likelihood L of the training data is as follows:

$$L = \prod_{n=1}^{|T|} P_\theta(c = 1|p_n, q)^{p_l} P_\theta(c = 0|p_n, q)^{1-p_l}.$$

We model $P_\theta(c = 1|p_n, q)$ by logistic functions on a linear combination of features (i.e., *Model-Anchor* and *baseline* relevance scores). The estimated parameters can then be plugged into w_1 and w_2 weights of the linear fusion of *Model-LC*. According to the learned parameters on the TREC 2013 judgments, the optimal w_1 and w_2 weights are 171.472 and 33.426.

3.2 Experiments

A number of experiments are designed to address the following questions of the proposed research:

- How do our two proposed models (i.e., *Model-Anchor* and *Model-LC*) perform compared to the Terrier baseline?
- What are the optimal parameters of the linear fusion?

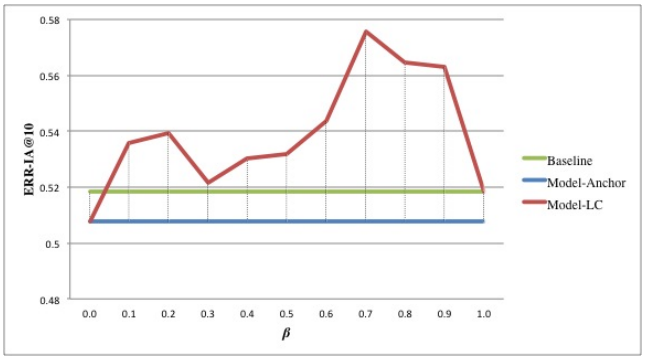

Figure 6: Impact of fusion weight (β) on ERR-IA@10.

- What is the advantage of fusing the *Model-Anchor* ranking with the *basline*?
- What is the optimal parameter of the *Model-LC* in the risk-sensitive task?

3.2.1 Experimental setup and metrics

In this section, we describe dataset and evaluation metrics. We build our models based on the ClueWeb12 dataset that was created to support research on Information Retrieval in 2012. The anchor text of ClueWeb12 web pages that is extracted by Djoerd Hiemstra [8] is used as representatives of the web pages in estimating their relevancy. In addition, We have used the PageRank of ClueWeb12 web pages available at the Lemur project website.

The evaluation of the Web track Ad-hoc rertieval task is primarily based on the intent-aware expected reciprocal rank ($ERR-IA$) [1]. In addition to $ERR-IA$, some standard information retrieval measures such as MAP, $precision$@10 and $NDCG$@10 are reported to evaluate different aspects of the participants' approaches.

3.2.2 Experimental Results

Indexing ClueWeb12 anchor text in *Model-Anchor* is beneficial to improve the precision of retrieving relevant web pages, but it is not as good as baseline in recall of the relevant webpages. Table 8 shows the performance measures of *baseline*, *Model-Anchor* and *Model-LC*. According to this table, the linear fusion of the *baseline* with the *Model-Anchor* ranking improves the *baseline* in terms of $ERR-IA$, $nDCG$, $NRBP$ and $nNRBP$, but does not improve the *baseline* based on $MAP-IA$ and $P-IA$.

The *Model-Anchor* improves the *baseline* of TREC 2013, and because of this, the weight of *Model-Anchor* is about 6 times more than the weight of the *baseline* in the linear fusion of them. However, as it is demonstrated in Table 8, the *baseline* of TREC 2014 is performing better than the *Model-Anchor*, so the learned weights on the TREC 2013 topics could not be the optimal weights for the TREC 2014 queries. As Table 8 shows, although the learned weights are not optimal weights for the TREC 2014 queries, the *Model-LC* improves the *baseline*, which indicates the effectiveness of the linear combination of the *baseline* with the *Model-Anchor*. Figure 6 plots the optimal parameters of *Model-LC* for TREC 2014. In this figure, for simplicity, we consider

Table 8: Effectiveness of baseline (Terrier) and anchor-text in isolation and combination

Method	ERR-IA			nDCG			NRBP	nNRBP	MAP-IA	P-IA		
	@5	@10	@20	@5	@10	@20				@5	@10	@20
baseline	0.5012	0.5183	0.5313	0.5313	0.5697	0.6109	0.4871	0.4931	**0.2043**	**0.4331**	**0.4142**	**0.4131**
Model-Anchor	0.4883	0.5075	0.5152	0.5241	0.5643	0.5909	0.4725	0.4893	0.0694	0.3495	0.2955	0.2356
Model-LC	**0.5263**	**0.5418**	**0.5524**	**0.5623**	**0.5919**	**0.6270**	**0.5139**	**0.5316**	0.0862	0.3737	0.3104	0.2775

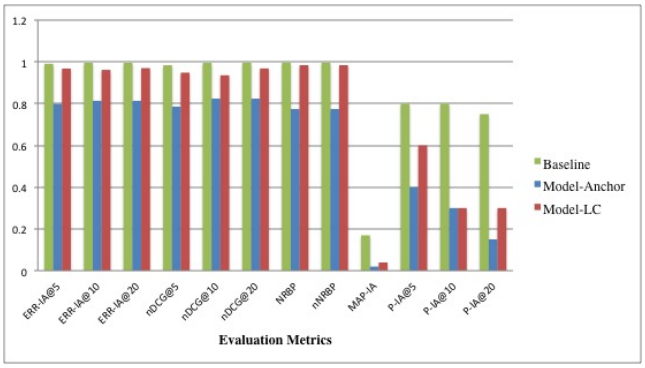

Figure 7: Query 276: "how has african american music influence history."

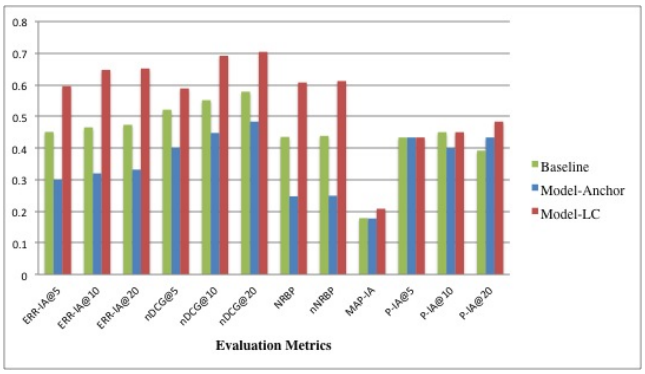

Figure 8: Query 273: "wilson's disease."

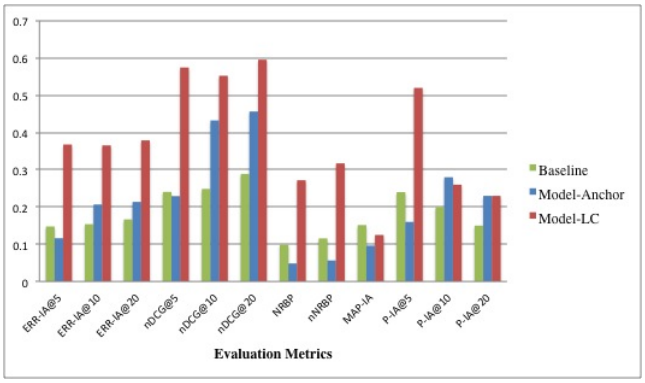

Figure 9: Query 275: "uss cole."

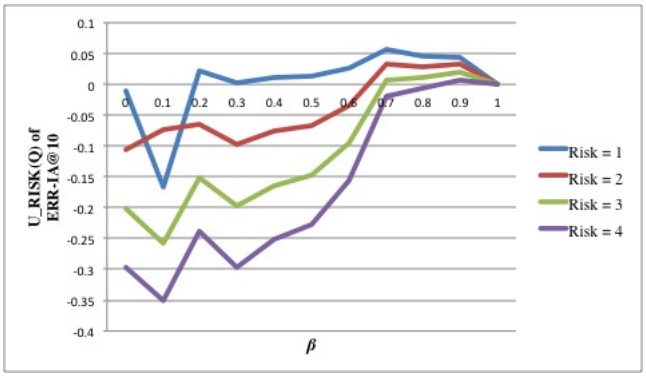

Figure 10: Impact of β on the risk-sensitive task (based on $U_{RISK}(Q)$ of $ERR-IA@10$ metric).

$\beta = \frac{w_2}{w_1+w_2}$; as a result, $w_1 = 1-\beta$ and $w_2 = \beta$. According to Figure 6, $w_1 = 0.3$ and $w_2 = 0.7$ are the optimal weights of the *Model-LC* for TREC 2014.

Figure 7 shows the effectiveness of the *baseline*, *Model-Anchor* and *Model-LC* for a long query "how has african american music influence history". This is a long and focused intent query, which could help us to gain a better understanding of how the linear fusion of *Model-Anchor* with the *baseline* could improve the performance of the *Model-Anchor*. As it is demonstrated in Figure 7, the *Model-LC* takes advantage of the baseline in recall of relevant web pages in order to improve the overall performance of the *Model-Anchor*.

There are also number of examples that indicates the performance of *Model-LC* is better than the *baseline* and the *Model-Anchor*. For instance, Figures 8 and 9 contain two examples of topics that in one of them, *Model-Anchor* performs better than the *baseline* and in the other one, *baseline* performs better than *Model-Anchor*. This experiment indicates that the linear combination of the *baseline* with the *Model-Anchor* could improve the performance of the *baseline* and *Model-Anchor* rankings independent of the information about the performance of them compared to each other.

In Figure 10, we analyze performance of the *Model-LC* having different β parameters in risk-sensitive task. As it is demonstrated in this figure, although we increase value of the risk factor, the *Model-LC* with $\beta \in [0.7, 1.0]$ performs well and minimizes the retrieval losses with respect to the baseline run.

3.3 Conclusion and Future Work

We participated in the Web Track's Ad-hoc retrieval task to evaluate effect of the anchor text in retrieving relevant web pages to the given query from a huge number of candidates in the ClueWeb12-full dataset. We were also eager to investigate the effect of linear fusion of the ranking based on the anchor text with the baseline ranking. We learned the parameters of the linear combination of the rankings by

the logistic regression binary classifier. The experimental results indicate that the fusion of the proposed model with the baseline improved the baseline in terms of the intent-aware expected reciprocal rank ($ERR-IA$), which is the primary metric in evaluating the submissions. We also evaluate performance of the linear fusion of the ranking based on the anchor text with the baseline ranking in risk-sensitive task, which shows the effectiveness of this model in minimizing the retrieval losses. As a future work, we plan to propose further effective data fusion method to combine these two different kinds of ranking, such as mixture language models [9].

Acknowledgments

This research is funded in part by the European Community's FP7 (project meSch, grant # 600851) and the Netherlands Organization for Scientific Research (WebART project, NWO CATCH # 640.005.001; ExPoSe project, NWO CI # 314.99.108; DiLiPaD project, NWO Digging into Data # 600.006.014). The authors thank Thaer Samar for helping in extracting title and description of suggestions, preparing text content of aggregators sub collection, and mapping open web qrels to ClueWeb12 ids.

REFERENCES

[1] O. Chapelle and Y. Zhang. Expected reciprocal rank for graded relevance. In *CIKM'09*. ACM, 2009.

[2] K. Collins-Thompson, P. Bennett, F. Diaz, C. L. A. Clarke, and E. Voorhees. Overview of the TREC 2013 contextual suggestion track. In *Proceedings of TREC 2013*, 2013.

[3] N. Craswell, D. Hawking, and S. Robertson. Effective site finding using link anchor information. In *Proceedings of SIGIR '01*, pages 250–257. ACM, 2001.

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

[5] 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.

[6] A. Dean-Hall, C. L. A. Clarke, P. Thomas, and J. Kamps. Evaluating contextual suggestion. In *Proceedings of EVIA*, 2013.

[7] N. Eiron and K. S. McCurley. Analysis of anchor text for web search. In *Proceedings of SIGIR '03*, pages 459–460, 2003.

[8] D. Hiemstra and C. Hauff. Mapreduce for information retrieval evaluation: "let's quickly test this on 12 tb of data". In *Proceedings of CLEF'10*, pages 64–69. Springer-Verlag, 2010.

[9] J. Kamps. Web-centric language models. In *Proceedings of CIKM '05*, pages 307–308. ACM, 2005.

[10] M. Koolen and J. Kamps. The importance of anchor text for ad hoc search revisited. In *Proceedings of SIGIR '10*, pages 122–129. ACM, 2010.

[11] I. Ounis, G. Amati, P. V., B. He, C. Macdonald, and Johnson. Terrier Information Retrieval Platform. In *Proceedings of ECIR 2005*, pages 517–519, 2005.

[12] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM TOIS*, 22(2), 2004.