User Modeling for Contextual Suggestion

Hua Li, Rafael Alonso

Leidos, Inc.

{Hua.Li, Rafael.Alonso}@leidos.com

ABSTRACT

This paper describes our work on the Contextual Suggestion Track of the Twenty-Third Text REtrieval Conference (TREC 2014). The key to our approach is user interest modeling. By building explicit models of user interests and information needs, we are able to make suggestions relevant to the user. We extended our Reinforcement and Aging Modeling Algorithm (RAMA) to create user interest models using the rated examples in a user profile as explicit relevance feedback. Two models, one for specific interests and the other for general interests, are built for each user profile. To ensure that the recommendations are contextually appropriate, we have also built a simple model to capture contextual relevance of a recommendation. Candidate suggestions are retrieved from the Yelp®[1] website using its application programming interface. For each candidate, we calculate three component scores based on the specific interest model, the general interest model, and the context model, respectively. Final scoring and ranking are computed as a weighted linear combination of the component scores. We hypothesize that the relative weighting of the components may affect the performance of our system. To test the hypothesis, we have submitted two runs with different weighting schemes. In particular, RUN1 has a specific interest priority whereas RAMARUN2 has a general interest priority. TREC evaluation reveals that both runs performed significantly better than the median of all submitted runs (i.e., the Track Median) on three performance metrics. In addition, RAMARUN2 has a slight performance edge over RUN1. The effectiveness of our approach is evidenced by the TREC evaluation result that RAMARUN2 and RUN1 ranked #2 and #6 out of the 31 runs submitted by the 17 participating teams from around the world.

Keywords

User Modeling, Contextual Suggestion, TREC, Recommendation System, General Interest, Specific Interest, Context, Location, Yelp, Cosine Similarity

1. INTRODUCTION

This is our first participation in the TREC Contextual Suggestion Track. We are interested in this track because of its connection to the area of user modeling. In our view, contextual suggestion is about modeling user's interests and preferences with user relevance feedback and using the models to make smart recommendations. In the past, we have worked on similar problems in a number of research programs. For example, in an IARPA (The Intelligence Advanced Research Projects Activity) program, we modeled the information needs of intelligence analysts and used the models to guide swarming digital ants and combat cognitive biases (Alonso and Li, 2005a, 2005b). In a U.S. Army research program, we dynamically built interest models of operators for discovering relevant information snippets in real-time from a large number of chat rooms (Li et al., 2012). In a DARPA (The Defense Advanced Research Projects Agency) program, we used adaptive interest models to help mobile nodes to combat network disruptions by prefetching information (Li et al., 2014).

For the contextual suggestion task, a sensible suggested attraction should consider user's personal interests on the one hand and the contextual appropriateness on the other. To this end, we have extended our user modeling algorithm to model user's general and specific interests separately. Both models are used to assess the relevance of a suggested attraction. In addition, we have built a simple context model to capture the contextual relevance of a given attraction. The final scoring and ranking of a suggestion involve combining contributions from relevance factors captured in these models. As such, our approach differs from other participating teams in at least three important ways: a) the creation of explicit user interest models; b) the separation of general and specific interests for a user; and c) the explicit capture of contextual relevance.

2. OUR APPROACH

The contextual suggestion architecture shown in Figure 1 involves five key sequential steps: 1) User Interest Modeling, which builds both general and specific interest models for a given user from the information in the provided user profile; 2) Yelp API, which is used to identify candidate suggestions for a given context, i.e., location from the Yelp website[2]; 3) Component Scoring, which generates three component scores: general interest score, specific interest score, and context score for a given candidate suggestion; 4) Component Score Aggregation, which combines the three component scores to produce a

[1]Yelp is a registered trademark of Yelp, Inc. in the United States and/or other Countries.

[2] http://www.yelp.com/

single score for a candidate suggestion; and 5) Suggestion Ranking, which sorts and ranks all candidate suggestions to generate TREC submissions.

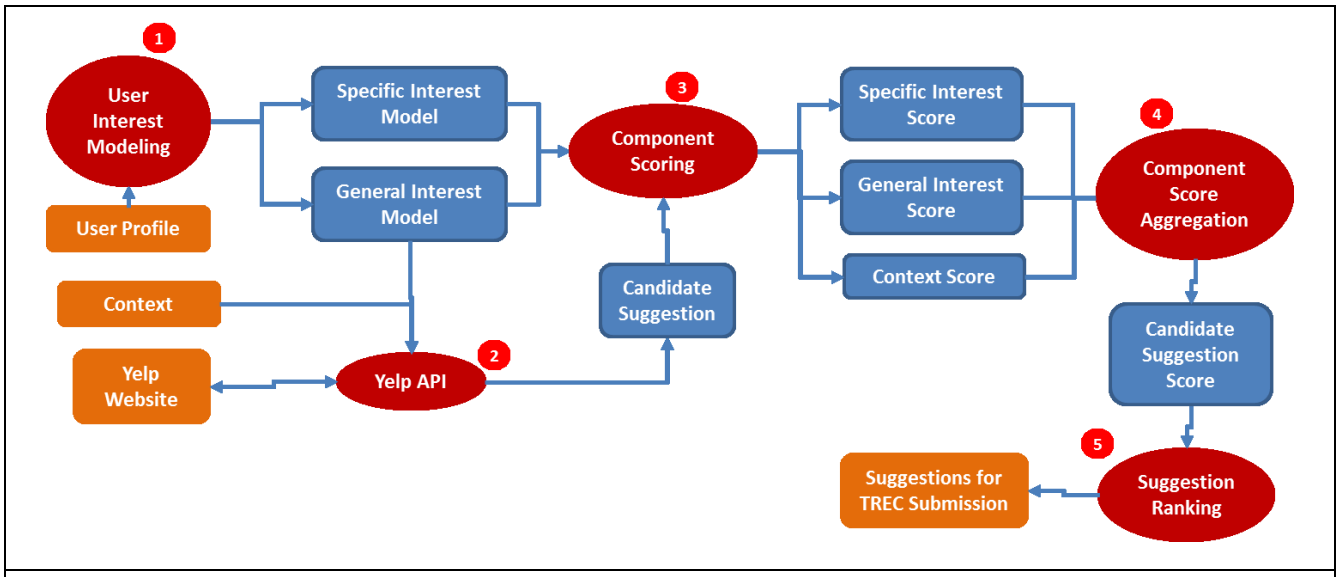


Figure 1. Contextual suggestion architecture diagram

2.1 User Interest Modeling

User Interest Modeling is the key step to our approach for contextual suggestion. Here our user modeling algorithm, i.e., RAMA (Reinforcement and Aging Modeling Algorithm) is applied to build explicit user interest models, which is then used to score candidate suggestions. We describe user modeling in detail in the USER MODELING section below.

2.1.1 User Profile

National Institute of Standards and Technology (NIST) has provided 299 user profiles. For each user profile, 70 to 100 example points-of-interest from both Chicago, IL and Santa Fe, NM are rated on a scale from strongly uninterested (0) to strongly interested (4). Each point-of-interest contains a title, description, and URL. From a user modeling perspective, each rated example is a user event that provides explicit user relevance feedback. One user model is built for each profile with all available rating events in the profile using the RAMA algorithm.

The RAMA algorithm processes one user rating event at a time. Interest elements (content words and interest categories) and their associated ratings are extracted first from the event. Interest elements are extracted from title and description of profile examples using Lucene®[3,4], WordNet™[5,6], and Yelp category taxonomy[7]. Lucene's StandardAnalyzer is used to extract terms from the text inputs. The non-noun words are removed using WordNet. Both the extracted interest elements and the ratings are used in the model adaptation steps of the RAMA algorithm. RAMA has two key functions in weight adaptation: reinforcement and decay. We did not apply the decay function (achieved by setting the attenuation factor to 0) for the TREC Contextual Suggestion track. This is because the track did not provide information regarding the sequential ordering of the user rating events. It is reasonable to assume that all ratings were done in a relatively short time-frame and the user's interests remained unchanged. The last step of the RAMA algorithm involves inserting new interest elements into the model.

2.1.2 General Interest Model

The general interest model captures the user's interests in terms of categories (e.g., museums, landmarks, and galleries). For each rating event in a user profile, we extract the Yelp categories (see Yelp API section for how this is done) and feed them into the RAMA algorithm to build the model. The general interest model for user 814 is shown as a word cloud and a table in Table 1.

---

[3] Lucene is a registered trademark of the Apache Software Foundation in the United States and/or other countries.

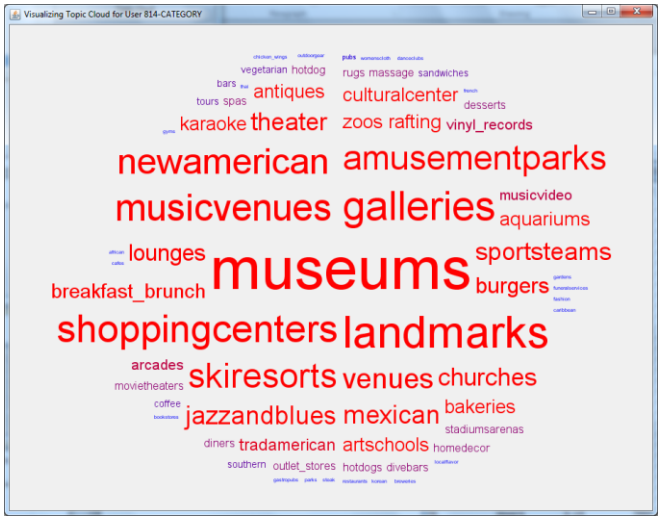
[4] http://lucene.apache.org/

[5] WORDNET is a trademark of the Trustees of Princeton University in the United States and/or other countries.

[6] http://wordnet.princeton.edu/

[7] http://www.yelp.com/developers/documentation/v2/all_category_list

Table 1. Visualizing General Interest Model for User 814



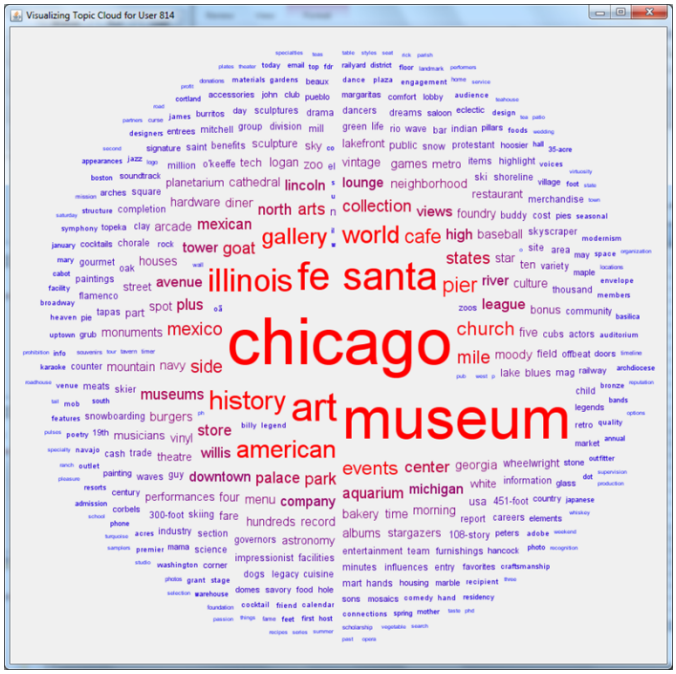
Visualizing Expertise Table View for User 814-CATEGORY

ID	ELEMENT	WEIGHT	TYPE
1	museums	0.917	TERM
2	landmarks	0.672	TERM
3	galleries	0.605	TERM
4	shoppingcenters	0.562	TERM
5	musicvenues	0.549	TERM
6	newamerican	0.527	TERM
7	amusementparks	0.5	TERM
8	skiresorts	0.5	TERM
9	venues	0.398	TERM
10	mexican	0.37	TERM
11	jazzandblues	0.37	TERM
12	churches	0.344	TERM
13	sportsteams	0.344	TERM
14	theater	0.344	TERM
15	lounges	0.312	TERM
16	burgers	0.298	TERM
17	breakfast_brunch	0.265	TERM
18	aquariums	0.25	TERM
19	culturalcenter	0.25	TERM
20	karaoke	0.25	TERM
21	rafting	0.25	TERM
22	zoos	0.25	TERM
23	antiques	0.234	TERM
24	artschools	0.234	TERM

2.1.3 Specific Interest Model

The specific interest model captures the user's interests in terms of content words (e.g., Chicago, history, and pier). For each rating event in a user profile, we extract the noun words from the title and description texts and feed them into the RAMA algorithm to build the model. Note, some of the content words may also be considered categories, but the majority of the words are not. The specific interest model for user 814 is shown as a word cloud and a table in Table 2.

Table 2. Visualizing Specific Interest Model for User 814



Visualizing Expertise Table View for User 814

ID	ELEMENT	WEIGHT	TYPE
1	chicago	0.302	TERM
2	museum	0.269	TERM
3	art	0.174	TERM
4	fe	0.17	TERM
5	santa	0.17	TERM
6	illinois	0.131	TERM
7	history	0.109	TERM
8	world	0.098	TERM
9	american	0.096	TERM
10	gallery	0.086	TERM
11	pier	0.083	TERM
12	cafe	0.073	TERM
13	events	0.072	TERM
14	church	0.072	TERM
15	mile	0.068	TERM
16	mexico	0.065	TERM
17	side	0.063	TERM
18	goat	0.06	TERM
19	arts	0.06	TERM
20	collection	0.059	TERM
21	park	0.058	TERM
22	states	0.058	TERM
23	center	0.055	TERM

2.2 Yelp API

Yelp API is a restful web service provided by Yelp for searching business review and rating information for a particular geographic region or location. We used it to collect candidate suggestions.

2.2.1 Yelp Website

We choose Yelp.com as the only source of candidate suggestions because of its comprehensive coverage of local business[8]. As of the 3rd quarter in 2014, Yelp had 139 million monthly visitors and 67 million reviews. It covers a variety of businesses that are considered attractions, including shopping, restaurants, arts and entertainments, nightlife, etc. Yelp API 2.0 allows applications to programmatically query local business using keywords, locations, or both. It permits 25,000 API calls per day without charge. Each Yelp search query can return up to 20 results. The results are in JSON (JavaScript Object Notation) format and contain business name, distance from context location, business category labels, number of reviews, average rating, review snippets, and other information (see Table 3 for an example Yelp API call).

Table 3. Example Yelp API Call: Query and Results for Context Erie, PA

 # Yelp Query PARAMETERS

{limit=20, sort=2, category_filter=beer_and_wine, ll=42.12922,-80.08506}

 # Yelp Query RESPONSE in JSON Format

{"region": {"span": {"latitude_delta": 0.011431090000002087, "longitude_delta": 0.009826799999881587}, "center": {"latitude": 42.124028150000001, "longitude": -80.089528400000006}}, "total": 2, "businesses": [{"is_claimed": false, "distance": 1370.2337399293274, "mobile_url": "http://m.yelp.com/biz/wine-and-spirits-stores-erie-3", "rating_img_url": "http://s3-media3.fl.yelpcdn.com/assets/2/www/img/34bc8086841c/ico/stars/v1/stars_3.png", "review_count": 1, "name": "Wine \u0026 Spirits Stores", "snippet_image_url": "http://s3-media2.fl.yelpcdn.com/photo/ozfwBwzMDGUy532WUKF_WQ/ms.jpg", "rating": 3.0, "url": "http://www.yelp.com/biz/wine-and-spirits-stores-erie-3", "location": {"city": "Erie", "display_address": ["Liberty Plaza Shop Ctr", "Erie, PA 16501"], "postal_code": "16501", "country_code": "US", "address": ["Liberty Plaza Shop Ctr"], "state_code": "PA"}, "phone": "8148665793", "snippet_text": "They have a better selection of wine at this location as opposed to the Peach Street store which did not carry any French wines over $20. I was able to find...", "categories": [["Beer, Wine \u0026 Spirits", "beer_and_wine"]], "display_phone": "+1-814-866-5793", "rating_img_url_large": "http://s3-media1.fl.yelpcdn.com/assets/2/www/img/e8b5b79d37ed/ico/stars/v1/stars_large_3.png", "id": "wine-and-spirits-stores-erie-3", "is_closed": false, "rating_img_url_small": "http://s3-media3.fl.yelpcdn.com/assets/2/www/img/902abeed0983/ico/stars/v1/stars_small_3.png"}, {*<details for 2nd result omitted>* }]}

2.2.2 Candidate Suggestion

Candidate suggestions are collected using Yelp API 2.0 for each one of the 50 contexts (i.e., locations) specified by the TREC task. In order to do this, we first identified the Yelp category labels associated with each individual user profile example by querying Yelp with the title and the location of the example. We then computed the union of all the categories from 100 examples. We found a total of 72 business categories associated with one or more examples, as shown in Figure 2. Finally, for each context, we issued one query per category to retrieve up to 20 results with highest ratings. In this way, we have collected a total of 11,641 candidate suggestions, which are grouped by context.

The resulting details of each candidate are parsed and used for component scoring. In particular, the name and snippet text fields are used for computing specific interest score whereas the categories field is used for computing the general interest score.

"parks", "steak", "mexican",
"newamerican", "landmarks", "divebars", "burgers", "hotdogs",
"hotdog", "tradamerican", "diners", "museums", "artschools",
"gastropubs", "breakfast_brunch", "amusementparks", "southern",
"desserts", "bakeries", "vinyl_records", "musicvideo", "arcades",
"lounges", "musicvenues", "bars", "thai", "korean",
"chicken_wings", "jazzandblues", "pubs", "sandwiches", "venues",
"wineries", "beer_and_wine", "aquariums", "zoos", "culturalcenter",
"funeralservices", "galleries", "churches", "shoppingcenters",
"movietheaters", "theater", "gardens", "stadiumsarenas",
"sportsteams", "danceclubs", "karaoke", "localflavor",
"vegetarian", "rafting", "skiresorts", "antiques", "tours", "gyms",
"outdoorgear", "womenscloth", "cafes", "breweries", "french",
"outlet_stores", "homedecor", "coffee", "bookstores", "rugs",
"spas", "massage", "horsebackriding", "fashion", "restaurants",
"caribbean", "african"

Figure 2. Yelp business categories for user profile examples.

[8] http://www.yelp.com/factsheet

2.3 Component Scoring

Relevance scores for general interest, specific interest, and context are computed separately for each candidate suggestion.

2.3.1 Cosine Similarity

Both general interest and specific interest scoring involve the calculation of cosine similarity between the respective user interest model and the candidate suggestion. The cosine similarity metric based on the vector space model has been widely used for comparing similarity between search query and document in the information retrieval literature (Salton et al., 1975). To apply this metric, we converted the user interest model into a vector representation with all weighted interest elements in the model. In addition, we extracted a vector representation for the candidate suggestion from its associated information.

We want to mention one novel use of the cosine similarity in this work. In the information retrieval literature, the cosine similarity has been typically used in the 0 to 1 range because the vector weights typically come from *tfidf* (term frequency inverse document frequency)[9] which also has a value from 0 to 1. In our work, we used the full range of the cosine similarity, which is between -1 and +1, inclusive, with +1 indicating the two identical vectors and -1 indicating two opposite vectors. This full range results naturally from the fact that our user models allow the interest elements to have weights from -1 to +1 to represent the full spectrum of interest intensities from hate to love.

2.3.2 General Interest Score

In order to compute the general interest score for a candidate suggestion, the categories are fed into the RAMA algorithm to build a general interest model for the suggestion. This model is then converted into a vector representation as mentioned above. The general interest score is the cosine similarity between the user general interest model and the suggestion model in terms of their vector representations.

2.3.3 Specific Interest Score

Similarly for calculating the specific interest score, both the name of the suggestion and the review snippet text are fed into the RAMA algorithm to produce a specific interest model for the suggestion. The resultant model is also converted to a vector representation as described above. The specific interest score is the cosine similarity between the user specific interest model and the suggestion model in terms of their vector representations.

2.3.4 Context Score

The context score is based the geographic distance from the user's location, the number of reviewers, and the Yelp rating of the candidate suggestion. It is computed as the linear combination of a distance score, a rating score, and a review count score (Formula 1). Distance score has a value between -1 and +1 mapped from the user's distance from the suggested location (Formula 1a). The closer it is, the higher the score. The rating score maps the Yelp rating of 1 to 5 stars to a value between -1 to +1 (Formula 1b). The review count score maps the number of reviews to a value between -1 to +1 (Formula 1c). The weights for the three scores should add up to 1. As a result, the context score has a range of -1 to 1.

ContextScore = Wd * distanceScore + Wr * ratingScore + Wc * review_countScore	**(1)**
distanceScore = – 2 * (distance / DISTANCE_LIMIT_METERS) + 1	**(1q)**
ratingScore = 0.5 * rating – 1.5	**(1b)**
review_countScore = 2.0 * (review_count / REVIEW_COUNT_LIMIT) – 1	**(1c)**
Where: Wd = 0.6; Wr = 0.3; Wc = 0.1; Note: Wd + Wr + Wc = 1 *DISTANCE_LIMIT_METERS = 8000 meters; (about 5 miles)* *REVIEW_COUNT_LIMIT = 100; (the review count at which the rating is regarded as stable)*	

2.4 Component Score Aggregation and Candidate Suggestion Score

Component score aggregation produces the relevance score for the candidate suggestion as a weighted linear combination of the three component scores (Formula 2). Note that the total weights for the three components should equal to 1. The final score has a range of -1 to 1, inclusive, since each component score also ranges from -1 to 1, inclusive.

CandidateSuggestionScore = Wg * GeneralInterestScore + Ws * SpecificInterestScore + Wc * ContextScore	**(2)**
Where: 0<=Wg<=1; 0<=Ws<=1; 0<=Wc<=1; Wg + Ws + Wc = 1	

The weighting scheme for the components affects the quality of the suggestions. Two different weighting schemes were used for the two runs we submitted to TREC for evaluation.

[9] http://en.wikipedia.org/wiki/Tf%E2%80%93idf

2.5 Suggestion Ranking

Suggestion ranking is done by sorting all candidate suggestions based on their relevance score and then assigning an integer rank value. The ranks are consecutive integers. The suggestion with the highest score has a rank value of 1.

2.6 Suggestions for TREC Submission

For a given user and context pair, all candidate suggestions for the context are ranked and the top 50 ranked candidates are formatted for TREC submission. The generated suggestion contains identification information along wtih the rank, title, description, and URL of the recommendation. The descriptions of candidate suggestions come from the original review snippet text generated by Yelp. An example of a formatted submission is shown in Figure 3.

Group ID RAMA
Run ID RAMARUN2
Profile ID 843
Context ID 118
Rank 1
Title Shreveport Railroad Museum
Description The Shreveport Railroad Museum is on the grounds of the Shreveport Water Works Museum just outside of the Central Business District. Run and staffed by the...
URL/Doc ID http://www.yelp.com/biz/shreveport-railroad-museum-shreveport

Figure 3. An example of a formatted suggestion.

3. USER MODELING

3.1 User Model Representation

In the TREC context, the user model represents a dynamic and cohesive set of interest elements that a user may be interested in. It is automatically created and continually adapted by the RAMA adaptation algorithm. A user model is comprised of one or more facets, which describe different aspects or time-sensitive phases of a user's interests. A facet consists of a set of interest elements, each of which represents some dimension of a user's interests. Each element carries a weight to indicate its degree of importance. The interest element may take different forms including terms, named entities, ontological entities, topics and relationships. In particular, a term represents a user interest dimension corresponding to a word or phrase. For the TREC tasks, we have only extracted terms due to limited resources. The weight of an interest element ranges from -1 to +1, inclusive, where -1 indicates dislike and irrelevance, and +1 highest level of like and relevance.

Table 4 shows the XML of an example user interest model generated by applying RAMA to the rating events in the user profile. In this example, the user model contains one facet with a total of 72 term elements (e.g., "museums" with positive weight of 0.92 and "wineries" with negative weight of –0.08). The user model also contains a number of metadata items. For example, the item "intervals" captures the timestamps of the first and last user events used in the current facet. The item "numReportedEvents" keeps count of total user events for building the model. The item "pedigree" records the events

Table 4. General Interest Model for User 814: XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
- <ns2:UserModel xmlns:ns2="http://model.um.saic.com">
    <identifier>814-CATEGORY</identifier>
  - <facets>
      <identifier>1</identifier>
    - <elements xsi:type="ns2:Term" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <identifier>0</identifier>
        <weight>0.9167270660400391</weight>
        <eventTimeMap />
        <text>museums</text>
      </elements>
    - <elements xsi:type="ns2:Term" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <identifier>0</identifier>
        <weight>0.671875</weight>
        <eventTimeMap />
        <text>landmarks</text>
      </elements>
    - <elements xsi:type="ns2:Term" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <identifier>0</identifier>
        <weight>0.605224609375</weight>
        <eventTimeMap />
        <text>galleries</text>
      </elements>
    - <elements xsi:type="ns2:Term" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <identifier>0</identifier>
        <weight>0.5625</weight>
        <eventTimeMap />
        <text>shoppingcenters</text>
      </elements>
      ...
    - <elements xsi:type="ns2:Term" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <identifier>0</identifier>
        <weight>-0.08333333333333333</weight>
        <eventTimeMap />
        <text>wineries</text>
      </elements>
    - <elements xsi:type="ns2:Term" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <identifier>0</identifier>
        <weight>-0.25</weight>
        <eventTimeMap />
        <text>horsebackriding</text>
      </elements>
    + <intervals>
    + <pedigree>
      <retired>false</retired>
    </facets>
    <numReportedEvents>100</numReportedEvents>
  </ns2:UserModel>
```

processed. The user interest model can be visualized as a word cloud or a table (Table 1). With the word cloud, the larger the font and the warmer the color, the higher the weight for the interest element is.

3.2 The User Modeling Algorithm

The user modeling algorithm is used for modeling the user's interests. The algorithm is based on RAMA, which was developed in our previous research programs and was demonstrated to be effective in a formative evaluation study conducted by NIST (Alonso et al., 2010).

Table 5. User Modeling Algorithm Description

1) ***Initialize the user model****: create an empty interest model for the user if it does not exist.*
2) ***Extract interest elements from a user event****: identify terms and categories from rating events in the user profile.*
3) ***Age the current model****: apply a decay function to all interest elements in the interest model.*
4) ***Reinforce interest elements recurring in user event****: increase or decrease the weights of these elements in the model, depending on whether the rating is favorable or not.*
5) ***Insert new interest elements****: incorporate unseen interest elements from the event into the interest model with a default weights.*
6) ***Continue steps 2-5 for each incoming user event***

The user modeling algorithm is described in Table 5. To capture changing user interests, the adaptation algorithm continually updates the user model with incoming user events by applying a reinforcement mechanism and a decay function. For the contextual suggestion track, the weight adjustment for the interest elements are primarily based on a) the ratings of the source profile examples; and b) their occurring frequencies in the profile examples.

3.2.1 Extract Interest Elements from User Event

Interest elements (e.g., content words and interest categories) are extracted from the data associated with the user event (e.g., user ratings of an attraction) using tools such as NLP (natural language processing). We recognize that certain user events tell you a lot about the user's interests whereas others may tell you very little. We use the term **event relevance** to indicate the extent the event reflects the user's interests (e.g., see Table 6). Event relevance ranges from -1 to +1, where -1 indicates negative interests (i.e., dislike), +1 indicates positive interests (i.e., like), 0 indicates neutral interests. It is used by the user modeling algorithm for reinforcement and insertion of new interest elements. For each user rating of an example, the ratings for both the title/description are mapped into event relevance (Table 6).

Table 6. Mapping Ratings to Event Relevance

Title/Description Rating	Event Relevance
4	1.0
3	0.5
2	0
1	-0.5
0	-1
-1	0

3.2.2 Age the Current Model

This aging step captures the observation that user interests tend to gradually decrease over time. The aging is performed as follows: with every new user event the decay function shown in formula (3) below is allied to all interest elements in the current facet of the model. The attenuation factor is a configurable parameter that controls the rate of decay. Its value ranges between 0 and 1 depending on the problem domain. Normally a non-zero value is used to cause interest weights to go down.

newWeight = oldWeight * (1 - attenuationFactor)	**(3)**

3.2.3 Reinforce Interest Elements Recurring in User Event

This reinforcement step is based on the observation that the greater number of interest mentions in times and contexts, the stronger the interest element is. The reinforcement mechanism processes user events differently based on the polarity of the event, i.e., positively or negatively reflecting user's interests. The polarity is indicated by the signage of event relevance (see Table 6 above). With positive events (e.g., high ratings in user profile) that express user's interests, the reinforcement will increase the importance or weight of the interest elements contained in them. With negative events (e.g., low ratings in user profile) that indicate user's lack of interest, the weight of the contained interest elements will decrease. The reinforcement mechanism is expressed in formula (4) below. The old weight is the weight of an interest element in the current facet of the user model prior to reinforcement where the new weight is the result of the reinforcement. The label Math.abs denotes the absolute value function, which is necessary because the old weight can be negative. The mention frequency is the number of occurrences of the interest element in the textual content of the user event, i.e., title and description of the example attraction in the case of this TREC work. The reinforcement factor is a configurable parameter that controls the rate of the reinforcement. The parameter has a value ranging from 0 to 1 depending on the problem domain. For this TREC work, the value is set at 0.5.

newWeight = oldWeight + (eventRelevance * reinforcementFactor * mentionFrequency * (1 - Math.abs(oldWeight)))	**(4)**

3.2.4 Insert New Interest Elements

With this insertion step, new interest elements are incorporated into the current facet of the user interest model. The initial weight for newly inserted interest element is decided using formula (5) below. By comparing with formula (4), it is noted that the initial weight is equal to the size of one reinforcement application.

initialWeight = eventRelevance * reinforcementFactor * mentionFrequency	**(5)**

4. EVALUATION

NIST has conducted an evaluation for the contextual suggestion track submissions. Seventeen (17) teams from academia and industry worldwide have submitted a total of 31 runs.

4.1 Data

NIST provided 299 user profiles with preference ratings for 70 to 100 points-of-interest from two contexts. Fifty (50) contexts are also given by NIST. We have used the open web as the source of our suggestions. Using the Yelp API 2.0, we collected a total of 11641 candidate suggestions for the 50 contexts.

4.2 Evaluation Metrics

Three metrics are used to rank runs: 1) Precision at Rank 5 (P@5); 2) Mean Reciprocal Rank (MRR); and 3) a modified version of Time-Biased Gain (TBG). P@5 is the main measure.

4.3 TREC Submissions

A TREC submission is required to provide up to 50 suggestions for each profile-context pair. We have submitted two runs for evaluation: RUN1 and RAMARUN2. They differ in weighting scheme for component score aggregation. RUN1 put priority on specific interests (i.e., high Ws). In contrast, RAMARUN2 favors general interests (i.e., high *Wg*) (Table 7). Note, we generated a third run with priority on context (i.e., high *Wc*) but did not submit this run due to the submission limit of two runs.

Table 7. Weighting Scheme for Our Runs

	Wg	*Ws*	*Wc*
RUN1	0.09	**0.9**	0.01
RAMARUN2	**0.9**	0.09	0.01

4.4 Results

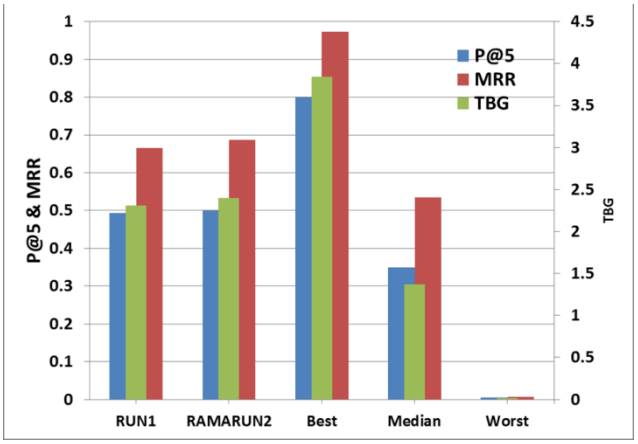


Figure 4. RAMA performance with the open web.

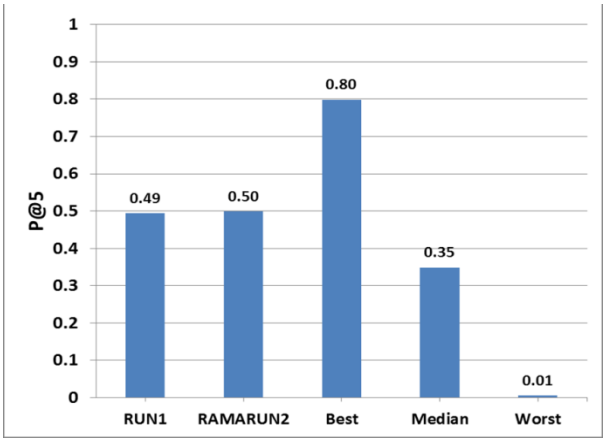


Figure 5. RAMA performance on metric P@5.

NIST judged the accuracy of our two RAMA runs for all the user-context pairs on three performance metrics, P@5, MRR, and TBG. In addition, NIST also provided the Best, Median, and Worst results from all participants who used Open Web as the source for suggestions. Figure 4 shows the average of our results against the average Track-wide results. Note that metrics P@5 and MRR are using the Y-axis on the left while TBG is using the Y-axis on the right. Overall, RAMA performed significantly better than the average Track Median on all three metrics. Since the P@5 is the primary metric for evaluation, we looked closely at it in Figure 5. RUN1 and RAMARUN2 have P@5 values of 0.49 and 0.50, respectively, both significantly better than the average Track Median at 0.35.

4.4.1 Comparison with Average Track Median

Table 8 compares the two RAMA runs with the Track Median on all three performance metrics. Both runs perform significantly better than the Track Median on all three metrics.

Table 8. RAMA performance compared to Track Median

Run	P@5	MRR	TBG
Track Median	0.35	0.54	1.37
RUN1	0.49	0.67	2.31
RAMARUN2	0.50	0.69	2.39

Table 9. RAMA performance improvement over Track Median

Improvement	P@5	MRR	TBG
Track Median	-	-	-
RUN1	0.41	0.24	0.68
RAMARUN2	0.43	0.28	0.75

The improvement in performance over the Track Median is shown in Table 9. Across the three metrics, RAMA shows the most improvement on TBG, medium on P@5 and the least on MRR. RAMARUN2 has the largest improvement (75%) on TBG whereas RUN1 on MRR shows the least improvement (24%).

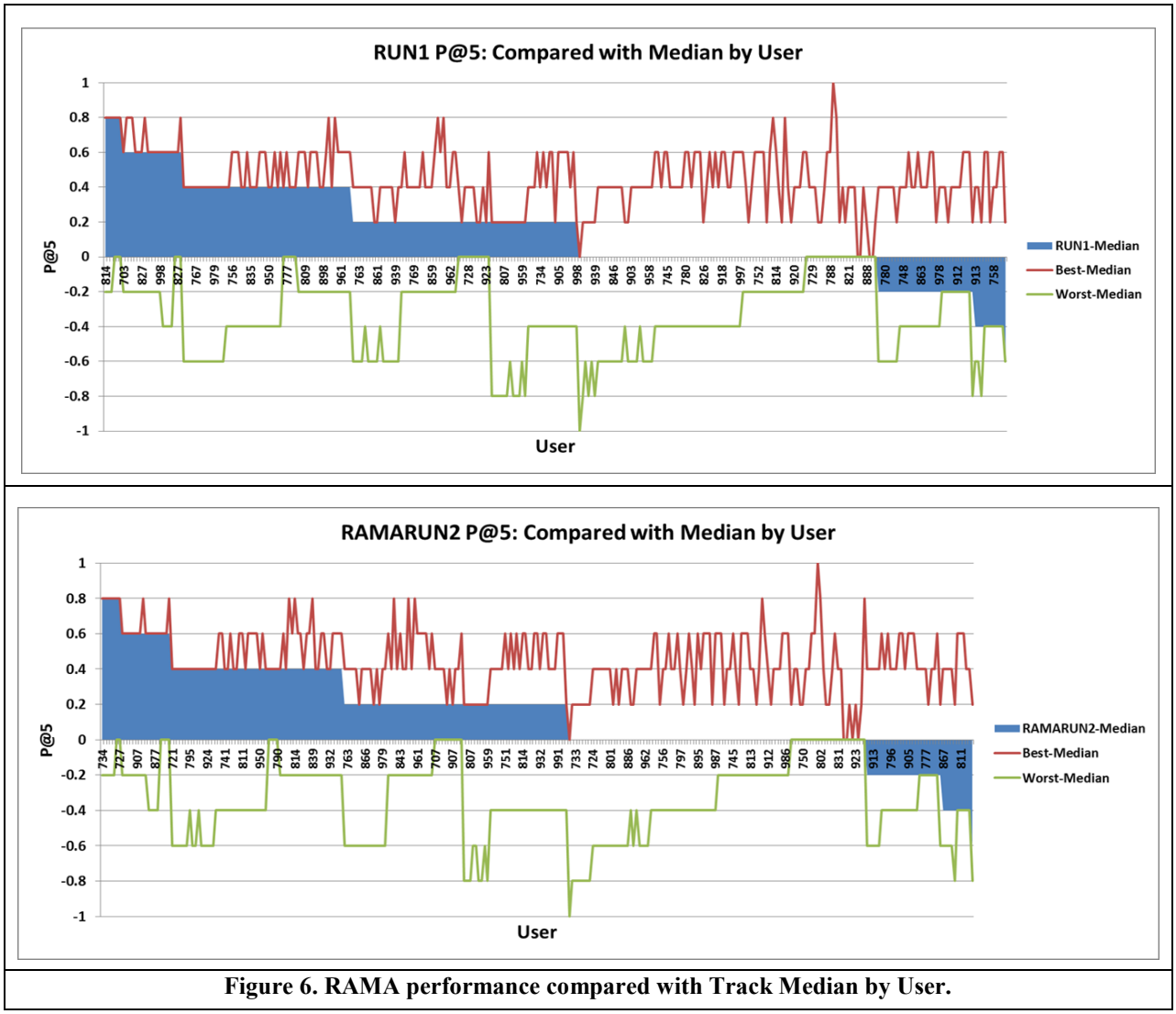


Figure 6. RAMA performance compared with Track Median by User.

4.4.2 Comparison with Track Median by User by Context Pair

To understand the performance for individual user-by-context-pair, we plotted the P@5 score delta from the Track Median for our runs, the Track Best, and the Track Worst. For RUN1 (Figure 6, top panel), the number of user-by-context-pair has a pseudo-normal distribution over the P@5 score delta, i.e., pairs with extremely good or poor performance are in the minority whereas the majority of pairs have more moderate performance. In other words, the run improvement over the Track Median

is distributed across the population, rather than confined to some subclass of super performers. Similar results are found in RAMARUN2 (Figure 6, bottom panel).

4.4.3 Comparison between Two RAMA Runs

The two RAMA runs have very similar performance; with RAMARUN2 having a slight edge over all three metrics (Table 8). The performance improvement of RAMARUN2 over RUN1 ranges from 1% to 4% (**Error! Reference source not found.**).

Table 10. RAMARUN2 performance improvement over RUN1

Improvement	P@5	MRR	TBG
RUN1	-	-	-
RAMARUN2	0.01	0.03	0.04

5. CONCLUSIONS

This is our first participation in the NIST Contextual Suggestion Track. We have leveraged our prior work on user modeling to this task. In particular, we have extended our RAMA algorithm to model user's general and specific interests. These models were used to compute interest-specific relevance scores for a candidate suggestion. In addition, we built a simple context model to calculate a context-specific relevance score for the candidate. Different weighting schemes are used for our two runs to combine the relevance scores and rank the candidate suggestions collected from the Open Web (i.e., Yelp). RAMARUN2 emphasizes general interest whereas RUN1 gives specific interests priority. TREC evaluation shows that both runs performed significantly better than the average Track Median on all three metrics. We learned at the TREC conference in November, 2014, that our RAMARUN2 and RUN1 ranked #2 and #6, respectively, out of the 31 runs submitted by the 17 participating teams.

REFERENCES

[1] Alonso, R., P. Bramsen, and H. Li, Incremental user modeling with heterogeneous user behaviors. *International conference on knowledge management and information sharing, KMIS 2010*: 129-134.

[2] Alonso, R., and H. Li, Model-guided information discovery for intelligence analysis, *Proceedings of the 14th International Conference on Information and Knowledge Management, CIKM 2005a*: 269-270

[3] Alonso, R., and H. Li, Combating cognitive biases in information retrieval, *Proceedings of the First International Conference on Intelligence Analysis Methods and Tools*, 2005b.

[4] Dean-Hall, A., C. LA Clarke, J. Kamps, P. Thomas, and E. Voorhes. Overview of the TREC 2014 Contextual Suggestion Track. 2014 (notebook paper).

[5] Li, H., R. Costantini, D. Anhalt, R. Alonso, M.-O. Stehr, C. Talcott, M. Kim, T. McCarthy, and S. Wood. Adaptive Interest Modeling Enables Proactive Content Services at the Network Edge. *Military Communications Conference (MILCOM 2014)*.

[6] Li, H., J. Lau, and R. Alonso, Discovering Virtual Interest Groups across Chat Rooms, *International Conference on Knowledge Management and Information Sharing (KMIS 2012)*.

[7] Salton, G., A. Wong, and C.S. Yang, A Vector Space Model for Automatic Indexing, *Communications of the ACM*, vol. 18, nr. 11, pages 613–620, 1975.