Finding, Weighting and Describing Venues: CSIRO at the 2012 TREC Contextual Suggestion Track

David Milne, Paul Thomas, Cecile Paris
CSIRO ICT Centre
PO Box 76, Epping, NSW 1710, Australia
{dave.milne, paul.thomas, cecile.paris}@csiro.au

ABSTRACT

We report on the participation of CSIRO[1] in the TREC 2012 contextual suggestion track, for which we submitted four runs. Two submissions were baselines that investigate the performance of a commercial system (namely the Google Places API), and whether the current experimental setup encourages diversity. The remaining two submissions were more complex approaches that explore the importance of time and personal preference. For the former, check-in statistics provided by Foursquare were used to identify which times of day and which days of week venues are more likely or less likely to be frequented. For the latter, textual similarity was used to weight venues with respect to positive and negative examples provided for each profile.

Our submissions all fall either slightly above or slightly below the mean, depending on how they are judged. Interestingly, our baselines consistently outperform our more complex submissions, which suggests that a) venue quality (as given by Google review score) is a more important signal than either time or personal preference, at least in the context of this evaluation, and b) that the evaluation is biased to a specific type of venue, namely pubs.

1. INTRODUCTION

This paper describes CSIRO's participation in the 2012 TREC Contextual Suggestion track. The task tackled here is to suggest venues for an individual to visit, given the context of where they are, what the time is, and what the individual has liked and disliked in the past. We provide an evaluation of two baseline systems that rely on the Google Places API and the user reviews it provides, and two more complex systems that incorporate information from the Foursquare API, and are sensitive to personal preference and time.

The remainder of this paper is structured as follows. The next section describes our approaches for identifying candidate venues, weighting them according to personal preference and time sensitivity, and locating explanatory text to describe them. Section 3 evaluates how this combination of candidate selection, weighting and summarization performs against baseline techniques and the submissions of other participants. Section 4 concludes with some recommendations for future contextual suggestion tracks.

2. IDENTIFYING, WEIGHTING AND DESCRIBING VENUES

The next section describes how candidate venues were extracted from Foursquare and Google Places, and is followed by an explanation of how duplicate venues were identified across the two sources. Section 2.3 describes a function for scoring venues against a profile of suggestions that a person has liked or disliked in the past. Section 2.4 describes a function for scoring venues based on time-sensitivity, to identify times of day and days of week in which it makes sense to suggest them. The track guidelines call for each venue to be described with a textual snippet, and our approach for obtaining this is explained in Section 2.5. These threads are all brought together in Section 2.6, which describes the final runs that were submitted.

2.1 Candidate selection

The basic approach to candidate selection was to use the Yahoo! Placefinder API[2] to establish sensible search bounds for each context city, and then exhaustively mine the Foursquare Venues API[3] and the Google Places API[4] for suggestions within these bounds. We expected these two services to be complementary, with Foursquare providing check-in statistics to identify popular venues and suitable times to recommend them, and Google providing reviews. An additional intuition was that the intersection of these services (i.e., the venues that are known by both) would be a tidier source of suggestions than each individually.

The name and state of the 36 distinct context cities was issued to Yahoo! Placefinder to retrieve suitable bounding areas. The bounds for several of these cities are listed in Table 1. Foursquare was then queried with each bounding rectangle and a filter to exclude venues that did not belong to *Food*, *Arts & Entertainment*, *Great Outdoors*, *Nightlife Spot*, or one of their descendent categories. Unfortunately the foursquare service is not designed for exhaustive search, and so it cannot handle overly large search areas, and will return a maximum of 50 locations for each query. To overcome these limitations, any time the service returned either an *area-too-large* exception or a full list of 50 venues, the search area was split into four quadrants to be issued as further queries. These in turn were split as necessary. This successive quartering of search areas continued until each query returned fewer than 50 venues, or to a maximum of 15 recursions, whichever came first. Even for the largest context cities, the recursion limit translates to a minimum search area of approximately 3m by 3m.

Figure 1 provides a visual demonstration of this recursive search algorithm in action over *Sydney, Australia*. The initial bounding box was split repeatedly, with the boxes predictably becoming smallest over downtown Sydney, where the venues are most densely clustered. There is a distinctive dense vertical bar just

[1] Commonwealth Scientific and Industrial Research Organisation

[2] The Yahoo! Placefinder API is available at *http://developer.yahoo.com/geo/placefinder/*

[3] The Foursquare Venue Search endpoint is described at *http://developer.foursquare.com/docs/venues/search*

[4] The Google Place Search endpoint is described at *https://developers.google.com/places/documentation/search*

Table 1: A sample of context cities, and the candidate venues extracted for them

Location		NE corner		SW corner		Venues		
City	State	lat	long	lat	long	Google	Foursquare	mutual
Akron	OH	41.17	-81.40	41.00	-81.40	1061	3207	584
Albuquerque	NM	35.22	-106.42	34.94	-106.42	678	3213	277
Ann Arbor	MI	42.33	-83.66	42.22	-83.66	794	1947	526
…								
Los Angeles	CA	34.34	-117.93	33.69	-117.93	18947	68227	9660
Mesa	AZ	33.53	-111.58	33.26	-111.58	2639	13267	1492
New York	NY	40.92	-73.69	40.50	-73.69	58458	110428	23994
…								
San Diego	CA	33.11	-116.80	32.51	-116.80	4353	30940	2123
San Francisco	CA	37.85	-122.32	37.70	-122.32	5574	15193	3085
…								
Virginia Beach	VA	37.03	-75.87	36.55	-75.87	1407	5707	762
Washington	DC	39.00	-76.90	38.80	-76.90	5107	18166	3001
					total	157972	477541	72059

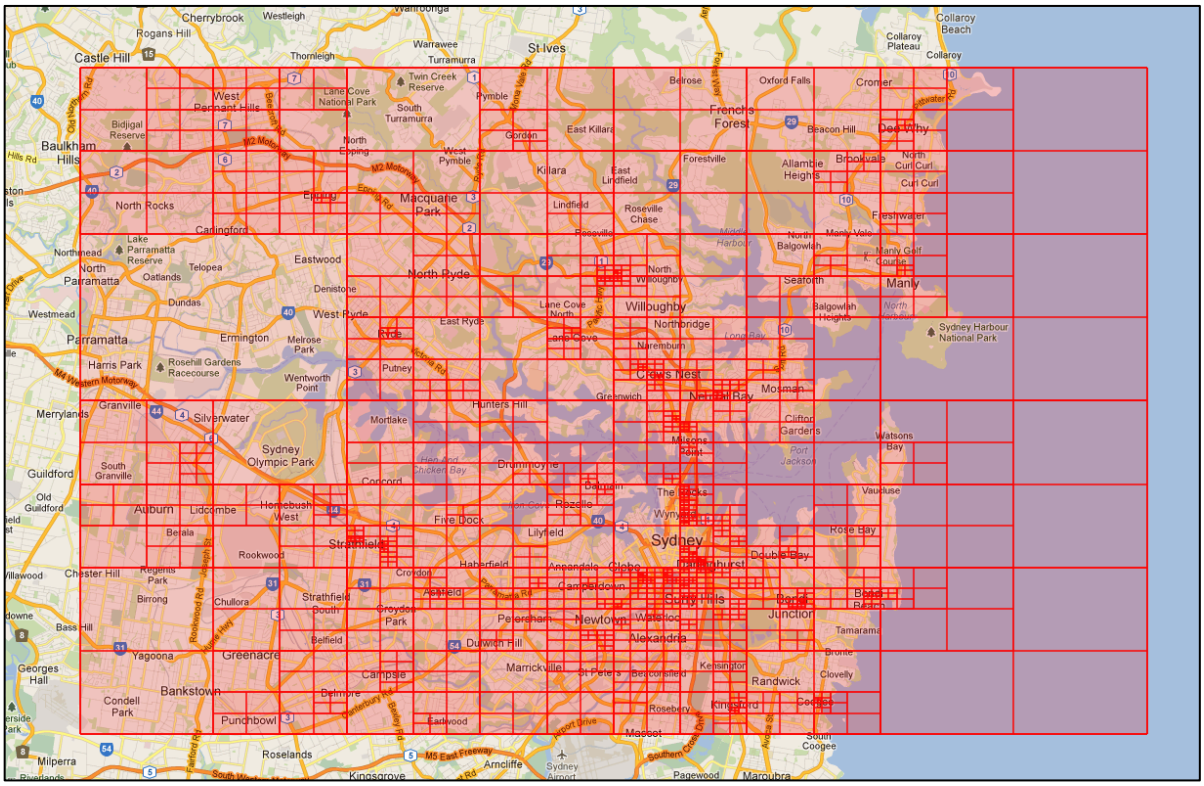


Figure 1: The recursive search algorithm in action over Sydney, Australia.

north of the "Sydney" label, which corresponds to George Street. There are other visible clusters that correspond to Manly, Bondi and other populous areas. There are also a few areas in which the crawling algorithm appears to have broken down or in which venues are unexpectedly sparse, most notably around Parramatta and Darling Harbour. In total, 1411 "leaf" bounding boxes (those that did not need to be split further) were required to capture all of the 15438 relevant venues that Foursquare knew of in the area.

The same recursive process was followed to gather candidate venues from Google Places, with a few small differences. Google does not allow searching with rectangular bounding areas, and so each search box was replaced by a circle centred in the same location with a diameter equal to the square's diagonal. It is also limited to 20 results per query, and consequently the search circles were split more often. Finally, Foursquare's category filters were translated by hand to match Google's place types.

It should be noted that the recursive crawling of these APIs goes against their intended use, and is somewhat wasteful. In Foursquare's case it directly contravenes their terms of service, which unambiguously forbids attempting to obtain exhaustive lists of venues. Permission was obtained directly from Foursquare via email for these experiments; but this should not be seen as permission for the research community as a whole. See Section 4 for our recommendations for future investigations.

Table 2: A sample of terms and preference vectors

Term	User 1	User 2	User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10
bar	0.74	-0.34	3.39	2.87	2.32	-0.87	-0.74	4.3	-0.08	1.66
pub	0.00	-0.31	-0.31	-0.31	0.00	-0.31	-0.31	0.93	0.00	0.93
walk	0.00	1.54	1.54	1.54	1.54	-0.51	1.54	0.00	1.54	0.00
outdoor	0.69	0.44	3.08	1.73	3.08	-0.32	1.79	3.08	0.44	1.42
shrine	0.00	-0.36	1.09	0.00	0.00	-0.36	-0.36	1.09	-0.36	-0.36
seafood	-0.30	0.00	0.00	-0.30	0.00	0.89	0.89	0.00	0.89	0.89

2.2 Disambiguating candidates

Mapping between common listings across the Foursquare and Google Places is a non-trivial task. There are many small variations in both the names and locations of venues. For example, the Google venue *Joey's Delicatessen* in New York is listed by Foursquare as *Joey's Deli*, and there is a distance of 126m between the listings. A common problem is the inconsistent inclusion or omission of words related to the type of venue and its location. For example, *Tope Cocktail Bar Lounge* in Google is listed simply as *Tope* in Foursquare, and *Tarantino's Restaurant* is expanded to *Tarantino's Restaurant at Fisherman's Wharf*. Fortunately, words that are related to type and location are easy to identify, because they occur disproportionately often in the listings while terms like *Tope* and *Tarantino's* are comparatively rare.

For each Google venue, we locate all Foursquare venues within a radius of 500m, and score each potential pairing by the following formula:

$$score(F,G) = \sum_{x \in (F \cap G)} idf(x) - \sum_{x \in (F \cup G - F \cap G)} idf(x)$$

where *F* is the set of tokens in the name of the Foursquare venue, *G* is the set of tokens in the name of the Google venue, and *idf (x)* is the inverse document frequency of the token x, as calculated by the following:

$$idf(x) = log\left(\frac{|T|}{c(x)}\right)$$

where *T* is the multiset of all tokens in all venue titles and *c(x)* is the number of occurrences of token *x* in *T*. In short, each individual token is weighted so that rare tokens are valued highly. A pair of titles is then weighted by their matching and mismatching tokens, with matches adding to the combined score, and mismatches subtracting from it. All negatively scored pairings are thrown away, and the highest weighted pairing for each Google venue is retained. The listings for each context city are treated entirely separately.

Again, we note that the terms of use for the Google and Foursquare APIs may prohibit such aggregation.

2.3 Identifying personal preference

We attempted to match candidates to users' preferences by looking at the text describing each candidate and each example. If a candidate's description is similar to examples a user has liked in the past, it should score well (all else being equal); if a candidate's description is similar to examples a user didn't like, it should score poorly. We used a simple vector-space approach to implement this.

The description of each candidate was treated as a bag of words, with stopwords removed and terms stemmed with the Porter stemmer. Terms were weighted with BM25, and all examples were summed to give a single vector for each profile:

$$p = \beta \sum_{e \in P} e \; - \; \gamma \sum_{e' \in N} e'$$

where β and γ are tuning constants, *P* is the set of positive examples, and *N* is the set of negative ones. Assuming positive examples are more informative than negative ones, we set β=0.75 and γ=0.25. This is equivalent to Rocchio relevance feedback with the initial query a zero vector.

Table 2 provides a sample of terms and preference vectors, and shows the differences in weights from profile to profile. We can see that in general users who (dis)like bars tend to (dis)like pubs (rho=0.46), and those who (dis)like walks tend to (dis)like the outdoors (rho=0.30); on the other hand, there is no correlation between liking bars and liking walks (rho=0.02) and there is wide variation even amongst the first ten profiles.

Armed with a preference vector for each profile, we scored each candidate according to the cosine distance between its description and the preference vector p.

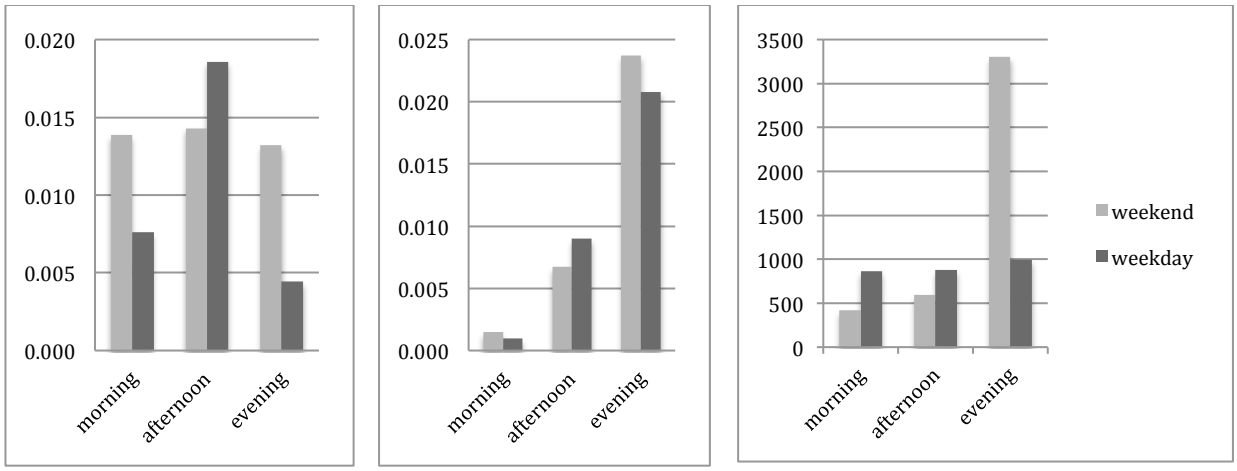
2.4 Identifying time sensitivity

Time is an important factor when suggesting a venue or activity: it makes little sense to suggest a café at 4am, or a water park in the middle of winter. The track's guidelines discuss three levels of time sensitivity: time of day (*morning*, *afternoon* or *evening*), day of week (*weekend* or *weekday*) and season (*spring*, *summer*, *winter*, or *fall*).

Foursquare's check-in statistics are essentially histograms of popularity over time, and as such are an ideal source of data for identifying the time sensitivity of a venue. Unfortunately the API provides public access only to the current check-in statistics for each venue. Historical statistics are only available to the venue's registered owner. The API could be polled to retrieve statistics over time, but that would not be practical for all 477k foursquare candidates our crawling algorithm identified in the 36 context cities (see Table 1). Gathering such statistics even once was a taxing effort for both our systems and for the generously shared APIs. Doing so repeatedly would be pushing one's luck.

Consequently, we generalize to measuring time sensitivity for broad classes of venues rather than individuals. Over the course of approximately two weeks, the city of Toronto was crawled every hour, and check-in counts of each venue were aggregated up into the hierarchy of categories they belong to. With this short timeframe for data collection, any attempt to measure long-term (i.e., seasonal) trends was abandoned, and instead we focus on time-of-day and day-of-week by building histograms of the kind shown in Figure 2.

The graph in Figure 2a was built by calculating the average number of check-ins per hour for all *Theme Parks*, divided by the



a) Theme parks **b)** Movie theatres **c)** All venues

Figure 2: Histograms of time-sensitivity for Foursquare venues.

total number of new check-ins per hour for all venues regardless of category. Figure 2b was built in the same fashion, for *Movie Theaters*. The y-axis in each graph represents the probability that a random check-in occurring at a particular hour would have occurred at the given type of venue.

These graphs demonstrate some interesting patterns. The graph of *theme park*s, for example, is markedly different for weekdays (where most visits occur in the middle of the day) and weekends (where visits are even throughout the day). In contrast, visits to *movie theatres* follow the same pattern (heavily biased towards the evening) regardless of day-of-week.

It is important to point out that each bar in these graphs has been normalised against the total number of visits per hour during each period, so comparing the bars within each graph is potentially misleading. Figure 2c plots the average number of visits per hour to all venues captured by our Toronto crawl, and demonstrates that visits are much more likely in weekend evenings than at any other time. This information is lost in Figure 2a and Figure 2b, which are designed to identify a good type of venue to visit at a particular time of day (i.e., to compare across graphs), rather than a good time of day to visit a particular type of venue (i.e., to compare within each graph).

To calculate a time-based score for a venue at a particular point in time, the relevant histogram is consulted to retrieve the visit-probability. Each probability is individually very small (there are many types of venues), and so they are normalised by the maximum probability of any one type of venue being visited at the given time. Venues belonging to multiple categories are scored using their highest scoring category.

2.5 Describing venues

Only in rare cases did the commercial APIs directly provide textual snippets to describe venues. For all of the remaining venues, snippets were generated in a simple-minded way, by feeding the venue's URL into the Yahoo BOSS search API, retrieving the first result with a matching URL, and extracting Yahoo!'s snippet.

2.6 Runs submitted

We submitted a total of four runs to the contextual suggestion track. Two submissions were baselines: a commercial baseline, which gives a benchmark based on a commercial API; and a "pub-run" baseline, which tests the track's evaluation criteria. The other two submissions were based on different combinations of the preference-based and time-based signals above.

2.6.1 The commercial baseline

The "commercial" baseline (*baselineA*) is output from the Google Places API, a commercial database of places and reviews. It allows us to evaluate how this API performs on the task (admittedly, used simple-mindedly), and to benchmark other runs. Implicitly is also evaluates the importance of personal preference, because this baseline makes absolutely no use of users' profiles.

Each context was split into day, time, and location. For each context, we issued a query to the Google Places API and retrieved the top 20 venues in the given location. This was a single call centered on the context's stated location, rather than a crawl of the type described in Section 2.1. The call was issued at a number of days and times to cover the context: for example, a "weekend morning" would see queries issued from 0800 to 1045, local time, on a Saturday and Sunday. The union of all results was taken, deduplicated, and sorted by Google's own ratings. The top-rated results were kept as suggestions. Profile information was not used in this run because commercial APIs do not expose this for general use.

2.6.2 The pub-run baseline

The "pub-run" baseline (*baselineB*) tests whether the track guidelines are adequate, and especially whether the 2013 guidelines need to consider diversity. On the assumption that our student judges tend to like going to the pub, we just recommend the closest pubs, restaurants, or similar. Any one suggestion like this is probably good: but if a list scores well despite being entirely pubs or restaurants, then we should encourage diversity in future tasks.

The baseline is formed the same way as the commercial baseline: from commercial APIs with no personalisation, but suggestions are restricted to pubs, restaurants, and cafes.

2.6.3 Time-emphasis and preference-emphasis runs

Our other two runs used candidates identified from Foursquare and Google, as described in Section 2.2, and scores for personal preference (Section 2.3) and time (Section 2.4). The scores were combined in a linear fashion:

$$score = \lambda\, score(preference) + (1 - \lambda)\, score(time)$$

Table 3: Performance (P@5) of submitted runs

	commercial baseline	pubrun baseline	preference-emphasis run	time-emphasis run
Description (D)	**0.3031**	0.2444	0.1438	0.1864
Website (W)	**0.4086**	0.2654	0.1531	0.1623
Geolocation (G)	0.7908	0.806	0.758	**0.8096**
Time (T)	0.5694	**0.5883**	0.4734	0.4712

Stone Brewing Company
www.stonebrew.com
Produces the Stone line of microbrews, and offers tours and tastings.

Antique Gas & Steam Engine Museum
www.agsem.com
Depicts life in the early 1900's through exhibits, ongoing restoration projects, and live demonstrations. Events, shows, services, and projects

Legoland California
www.legoland.com
Play your part at LEGOLAND(r). We have more than 60 thrilling rides, shows and attractions to choose from

Museum of Contemporary Art San Diego
www.mcasd.org
With two locations, the Museum of Contemporary Art San Diego (MCASD) is the region's foremost forum devoted to the exploration and presentation of the art of our time ...

Tamarack Surf Beach
www.parks.ca.gov/?page_id=660
California State Parks ... Park Information This San Diego beach features swimming, surfing, skin diving, fishing and picnicking.

Figure 3: Our suggestions for Context 22 (a weekend morning in the winter, at Escondido, CA) to profile 15

The time-emphasis run (*csiroht*) placed more emphasis on suitable times, with λ=0.3. The preference-emphasis run (*csiroth*) put more emphasis on per-user preference with λ=0.7. Note that for both runs there are implicit location, popularity and rating components in addition to the explicit time-based and preference-based signals, because candidates had to be geographically relevant and present in the commercial APIs to be considered.

3. EVALUATION

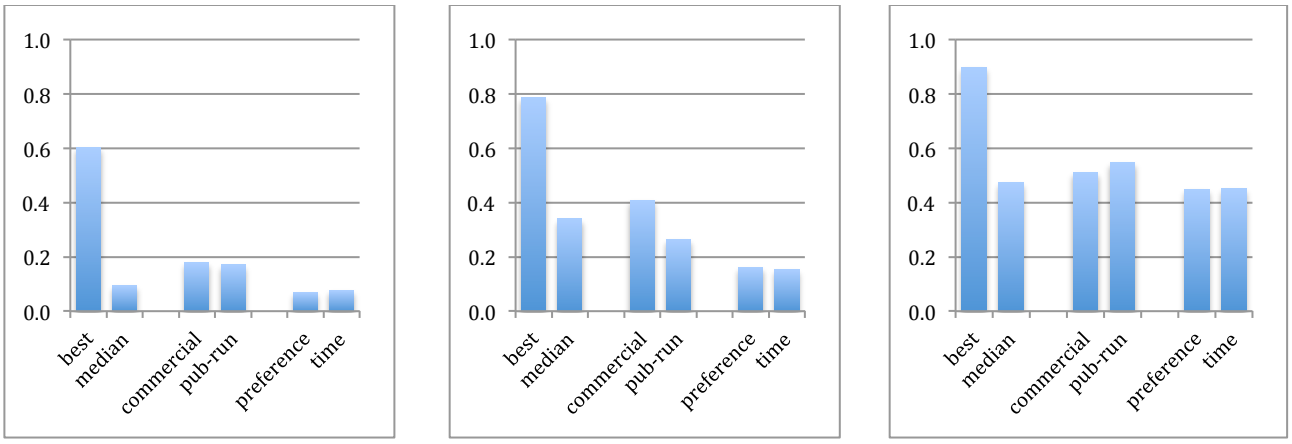
Submissions to the TREC Contextual Suggestion Track were evaluated along four separate dimensions: the interestingness of the venue's descriptive snippet (*D*) and website (*W*), and it's appropriateness given the context's geographical location (*G*) and timeframe (*T*). Table 3 shows the performance of each of our runs with respect to these dimensions, when P@5 scores are averaged across all judged contexts and profiles.

Scores for the *W* and *D* dimensions are low across all runs. Much of the problem may be due to the process described in Section 2.5 breaking down and failing to identify a suitable website for the venue. In these situations we fell back to presenting the Google Places webpage (with a map, short description, and reviews), and simply using the name of the venue as its description. However, spot-checking the results showed that these dimensions where judged somewhat unexpectedly. Evaluation involved presenting the websites and descriptions to the Toronto students who provided the original profiles, and asking them to judge whether they found the suggested venue interesting. For example, in Context 22 we were asked to provide venues close to Escondido CA to visit during a weekend morning in the winter. Figure 3 lists our suggestions for profile 15, the first four of which were judged "not interesting", and the 5th does not appear to have been judged.

One would expect that at least one of these diverse and popular suggestions to be interesting, no matter what individual the suggestions are being personalized for.

Performance is consistently high for the geo-location dimension, with scores ranging from 0.76 for the preference-emphasis run to 0.81 for the time-emphasis run. It is surprising that these scores are not higher across all runs, however. The candidate selection algorithm described in Section 2.1 should have ensured that every venue—even the lowest ranked ones—were appropriate. We have spot-checked the judgments and again found unexpectedly judged candidates. For example, Context 5 asked for suggestions in Los Angeles. For many of the profiles we suggested *Whisky A Go-Go* (a famous nightclub on the Sunset Strip) and the *Honda Center* (an arena hosting concerts and sporting events located 40mins drive from downtown LA). Both venues were judged geographically inappropriate. This is surprising, and suggests tighter guidelines might be needed for geo-location in future; both participants and judges should be able to agree on what makes a reasonable travel time, for example.

Performance for the time dimension ranges from 0.47 for the time-emphasis run to 0.59 for the pub-run baseline. Disappointingly, the naïve baselines that choose highly rated venues regardless of time outperform the runs that are informed by Foursquare's check-in data. Again there are cases of inconsistent judging here: for example, the same Los Angeles venues described above (*Whisky A Go-Go* and *the Honda Center*) were judged inappropriate for a weekday evening, despite foursquare check-in data indicating that they are very popular during these times.



a) website-geo-temporal (WGT) **b)** website (W) **c)** geo-temporal (GT)

Figure 4: Performance (P@5) of baselines and runs against best and median of all submissions.

Figure 4 compares our results to the best, worst and average P@5 scores of all the runs submitted by all participants in the track, again averaged across all contexts and participants that were judged. Here the dimensions are combined into website-geo-temporal (*WGT*), website (*W*) and geo-temporal (*GT*), simply because these were the statistics provided to us. There are no aggregate scores available for the description (*D*) dimension.

As discussed above, our runs all suffer when judged by their representative websites: with the exception of the commercial baseline, every run falls behind the median (0.34) and well behind the best submission (0.79). Performance improves in relation to other participants when geographical and temporal dimensions are considered, with performance in both *WGT* and *GT* following the same pattern: our baselines perform slightly higher than the mean, while our more complex submissions fall slightly below it.

4. DISCUSSION AND CONCLUSIONS

In this paper, we have presented our efforts to recommend venues for people to visit, given where they are, what the time is, and what they have liked and disliked in the past. We have provided an evaluation of two baseline systems that rely on the Google Places API and the user reviews it provides, and two more complex systems that are sensitive to personal preference and time.

The baselines outperform our other submissions, which suggests either that time- and preference-based signals are less important in the current experimental setup than venue rating (i.e., Google review score), or simply that our attempts to capitalize on these signals are faulty. The relatively high performance of the pub-run baseline (which emphasizes one specific class of venue) suggests that the track does not encourage diversity.

This track has presented interesting and challenging research problems, including information retrieval, recommendation, and summarization. We hope it becomes a staple in future TREC conferences. Moving forwards, we would recommend that organizers of the track make efforts to either approach industrial partners to locate more realistic sources of data, or to tailor the task to fit the data they have already released.

There are well-established organizations such as Foursquare and Yelp that already tackle the task of recommending venues. They have very rich data to work with:

- They have a large volume of structured information about venues (which is only available to researchers via the somewhat dubious lengths we discussed in Section 2.1)
- They service a diverse range of people (as opposed to a limited pool of similar participants)
- They have explicit measures of what participants have liked and disliked, sourced from reviews and check-in statistics (as opposed to laborious manual annotation).

Access to such data—Yelp's Academic Dataset is a good example—would resolve the concerns we raised in Section 3 about diversity and consistent annotation. It would also add a new signal to work with: the *people-who-like-this-also-like-this* dimension that powers collaborative filtering and other well-known recommendation techniques.