University of Lugano at the TREC 2013 Contextual Suggestion Track

Andrei Rikitianskii, Morgan Harvey, Fabio Crestani
University of Lugano (USI), Lugano, Switzerland
{andrei.rikitianskii,morgan.harvey,fabio.crestani}@usi.ch

Abstract

We report on the University of Lugano's participation in the Contextual Suggestion track of TREC 2013 for which we submitted two runs. In particular we present our approach for contextual suggestion which very carefully constructs user profiles in order to provide more accurate and relevant recommendations. The evaluations of our two runs are reported and compared to each other. Based on the track evaluations we demonstrate that our system performs very well in comparison to other runs submitted to the track, managing to achieve the best results in nearly half of all runs.

1 Introduction

This paper describes University of Lugano's participation in the 2013 TREC Contextual Suggestion track. This track continued on from the successful 2012 Contextual Suggestion track [2], including more profiles and the possibility of using ClueWeb12 as a source for suggestions. However, this year the organizers decided not to include a temporal (day of week, time of day, season) component for contexts, consisting now of only a location. The task of the track is to recommend places to an individual, given a specific context.

In this work we present a new approach to recommending places to users incorporating geographical information as context and exploiting data from multiple sources. Our approach mostly concentrates on building user profiles very carefully to catch user preferences more accurately. To construct user profiles we exploit simple Natural Language Processing (NLP) technique and Machine Learning (ML) approach. We submitted two runs with slightly different parameters for the learning algorithm. Via analysis of results from TREC evaluations performed by a large group of users we demonstrate the high level of performance delivered by our method, showing that it is able to outperform all other track runs in nearly half of all cases. Over all metrics our system performs considerably better than the median result.

The rest of this paper is organized as follows. We describe our approach for context suggestion in Section 3. Section 4 presents evaluations for the two

submitted runs. Analysis of our results is detailed in Section 5. Finally, Section 6 summarizes the conclusions and provides guidelines for future work.

2 Dataset and Tasks

The TREC Contextual Suggestion Track investigates search techniques for complex information needs that are highly dependent on context and user interests. In this track the goal is to suggest personalized attractions to an individual, given a specific geographic context. The track imagines a traveler in a new city. Given a set of the traveler's preferences for places and activities in their home city, the system should suggest places and activities in a new city that the person may enjoy. In this paper we use the terms "attraction," "place" and "venue" interchangeably.

As input to the task, participants were provided with a set of 635 profiles, a set of 50 example suggestions, and a set of 50 geo contexts in CSV/JSON format. Example suggestions can represent attractions of different types, for example: bars, restaurants, museums, etc. All the attractions are from the Philadelphia area. Each profile corresponds to a single user, and indicates the user's preference with respect to each example suggestion. Each training suggestion includes a title, description, and an associated URL. Each context corresponds to the GPS coordinate of the centre of a number of cities in the United States. The set of cities is quite diverse in terms of population: from small cities such as Beckley, WV (with a population of 17,606) up to much larger cities such as Atlanta, GA and Wichita, KS (with populations in the hundreds of thousands or even millions). Profiles consist of two ratings for a series of attractions, one rating for the attraction's title and description and another for the attraction's website. The ratings are given on a five-point scale, ranging from "strongly disinterested" to "strongly interested", based on how interested the user would be in going to the venue if they were visiting the particular city it is located in.

As output to the task, for each profile/context pairing, the participant should return a ranked list of up to 50 ranked suggestions. Each suggestion should be appropriate to the profile (based on the user's preferences) and the context (according to the location), contains a title, description and attraction's URL. The description of the suggestion may be tailored to reflect the preferences of that user. Profiles correspond to the stated preferences of real individuals, who will return to judge the proposed suggestions. Users were recruited through crowdsourcing sites or are university undergraduate and graduate students. For the purposes of this experiment, it was assumed that users are of legal drinking age for the location specified by the context.

3 A New Approach for Context Suggestion

To generate ranked lists of appropriate recommendations for each user profile and geographical context we developed a geo context-aware system. The system

can be broken down into the following 4 steps:

1. processing geo contexts;
2. inferring user term preferences;
3. building a personal ranking model;
4. ranking suggestions

In the following section we describe these individual steps in more detail.

3.1 Processing Geographical Contexts

Before we can apply any user profile-based personalisation we first need a set of appropriate candidate attractions located within a small radius of the geo context specified. We used the Google Places API [1] to obtain a list of potential suggestions, retrieved based on a query consisting of GPS coordinates and attraction types. We considered only types of venues, as defined by the Google Places API, which were present within the training set. In doing so we retrieved 27 different types, such as: night clubs, amusement parks, libraries, movie theaters, shopping malls, etc. On average, for each geo context, we collected about 350 suggestions.

Google Places only provides a short *title* and a web site *URL* for each suggestion. In order users can evaluate the quality of each suggestion a description of each venue should be provided. To generate these brief descriptions we first queried the Yandex Rich Content API [2] which, given a URL as a query, returns a short static textual description of the page's content. While the Yandex API has generally quite good coverage, there were instances where it was unable to return any information and in these cases we instead queried the Google Custom Search API and used the web site snippet it returned.

3.2 Inferring User Term Preferences

In order to make personalized suggestions for each user we need to be able to compare a new venue with that user's profile to determine how likely it is that the user will like it. We therefore need to have some representation of the user's likes and dislikes based on the training data made available to us, i.e. the venues each user has already rated. In line with previous work on recommender systems [4], we chose to maintain separation between positive and negative preferences using descriptive terms from already rated venues. We used the Natural Language Toolkit (NLTK) software [3] to extract only nouns, adjectives, adverbs and verbs from each description and represented these as binary vectors, indicating the presence or absence of each term. For each user,

[1] Google Places API - `https://developers.google.com/places/documentation/`
[2] Yandex Rich Content API - `http://api.yandex.com/rca/`
[3] Toolkit version 1.0 used, available from `http://nltk.org/`

we extracted positive and negative terms, using them to build separate positive and negative profiles. A positive term is one derived from a positively rated venue, a negative term is one from a venue that was given a negative rating. Venues with a *title and description* rating of more than 2 were considered to be positive, while venues were considered to be negatively rated when it was allocated a rating of less than 2. Terms from neutral suggestions were ignored.

Due to the relative brevity of the descriptions, this approach of using only the terms present is unlikely to result in many exact term matches and will therefore deliver quite unreliable similarity scores. Consider, for example, if the negative profile for a user contains the word "sushi" and this is matched against a description containing the terms "raw" and "fish". Without performing any kind of term expansion these concepts, despite their obvious similarity, would not make any contribution to the similarity score. However, by expanding existing raw terms using similar words we can (at least partially) overcome this vocabulary mismatch. Thus, comparing profiles with venue descriptions, instead of simply using the raw terms, we checked for matches between the synonym lists returned for each term in WordNet [4]. Given a list of synonyms for a term a and a term b, we consider the terms to be matching if the two lists share at least one component (i.e. if there is some overlap).

Using both the positive and negative models, we can estimate what the user's opinion might be about a potential suggestion based on its description. To estimate how positive the description is for user u, we can calculate the cosine distance between vector $\overrightarrow{D_i}$, representing the description of venue i, and a positive user profile $\overrightarrow{M_u^+}$ as follows:

$$cos^+(\overrightarrow{D_i}, \overrightarrow{M_u^+}) = \frac{\overrightarrow{D_i} \cdot \overrightarrow{M_u^+}}{\|\overrightarrow{D_i}\| \cdot \|\overrightarrow{M_u^+}\|}$$

The same formula was applied to estimate how negative the description is, by using the negative user profile. cos^+ and cos^- scores were used in the final ranking model as described in Section 3.3.

3.3 Building a Personal Ranking Model

After obtaining a set of potential suggestions for a given geographical context (described in Section 3.1), we need to rank these potential candidates according to the user's preferences. Because of the variability of individual preferences among users it is nearly impossible to build an accurate global ranking model and given that the task is to provide *personalised* suggestions this would not be suitable anyway. To investigate how varied user preferences were, we measured the level of agreement between all judgments from user profiles by using a standard statistical *overlap* metric [5] where the overlap between two sets of items A and B is defined as:

[4] WordNet - `http://wordnet.princeton.edu`

$$Overlap(A, B) = \frac{|A \cap B|}{min(|A|, |B|)} \quad (1)$$

The mean pairwise overlap between *title and description* ratings is 0.38 and between *website* ratings is 0.39. Both of these overlaps are quite small, suggesting that users have different preferences. Therefore, we decided to build a personal ranking model for each user in order to more precisely adapt suggestion to their own preferences.

3.3.1 Model and Training data.

We consider the choice of suitable candidates as a binary classification problem. We separate relevant and non-relevant suggestions for each individual user, and then rank those classed as relevant based on a confidence score estimated by the classifier. To generate a training data set for each profile we used example suggestion weight calculated as a linear combination of *title and description* and *website* ratings:

$$Weight(S) = \lambda R_{desc,s} + (1 - \lambda) R_{url,s}$$

with $\lambda \in [0, 1]$ and $Weight \in [0, 4]$. In the formula, S indicates the example suggestion from a particular profile; $R_{desc,s} \in [0, 4]$ is the suggestion *title and description* rating; $R_{url,s} \in [0, 4]$ is the suggestion *website* rating. We then assigned a positive label to suggestions with a combined weight of more than a threshold T^+ and negative label to the suggestion with weight less than threshold T^-.

These thresholds T^+ and T^- were tuned to try to balance the number of positive and negative samples in the training set. The degree of imbalance is represented by the ratio of sample size of the small class to that of the large class. We considered a training set to be imbalanced if the ratio was less than 1:5, i.e. if the largest class was more than 4 times greater than the smallest. T^+ and T^- were turned for each profile by using a simple iterative algorithm. If the algorithm was unable to converge (i.e. sufficiently balance the 2 classes) after 3 iterations we consider this particular profile to be unsuitable for classification and use an alternate approach for making recommendations which we outline later.

By default we set uniform weights to the 2 different ratings for each example ($\lambda = 0.5$), meaning that the importance of *title and description* is the same as *website*. It is possible that the true influence of these two factors may not be equal and this may depend on the kind of venue under consideration. For example, for many cafes the description may provide sufficient information upon which to base a decision, whereas for a restaurant the user may wish to browse the website first, perhaps to look at the menu before making a decision. In this track, we organized a simple user study with a small number of participants from our university to estimate these type-dependent values for λ. For each venue type the participants were asked to rate the importance of *title and description* in comparison with importance of *website* on a 9-point scale. A rating value

greater than 5 means that the *title and description* is more important than the *website*, and vise-versa. A rating value of 5 means that both factors have similar importance. In total, each participant evaluated 27 different types of venue. To calculate λ for each type, we rescaled the rating values for this type into the range $[0, 1]$, then these values were averaged across all the participants. In Table 1 we show λ for some types of venue.

In Section 4 we describe in detail how we used default and type-dependent values for λ separately in two submitted runs.

Type of venue	λ
Night club	0.6
Library	0.7
Spa	0.2
Cafe	0.9
Zoo	0.4
Museum	0.7

Table 1: λ for some types of venue

3.3.2 Learning Algorithm and Features.

We chose a Naïve Bayes classifier as our learning algorithm. This is a simple probabilistic classifier based on applying Bayes' theorem and making the assumption that each feature's weight (or in the binary case, presence or absence) is independent of the weights of other features, given the class variable. Although this assumption is unlikely to be entirely true in many cases, it greatly simplifies the model - making it tractable - and does not significantly degrade performance in practice. The motivation for choosing such a simple classifier was that it generally performs better on small data sets than more sophisticated machine learning techniques [1] and does not require any complex parameter tuning or additional learning. In our case, the size of the training data set is never greater than 50 examples; the number of examples for each profile lies in the range 30-49. We use the Weka implementation of the classifier [6] for all of our experiments.

Each suggestion in the training set is represented by a feature vector, consisting of two different types of features: boolean and real-valued. The boolean features were derived based on attraction types, representing the user's preferences with regard to the kind of venue suggested. As described in Section 4.1, the Google Places API returns a simple type for each place, indicating what kind of venue it is. Each place can be assigned to multiple types and as such our binary feature vector encodes the types each suggestion has been assigned to: 1 if it is assigned to that type, 0 otherwise.

The 3 real-valued features were based on the cosine distance between the suggestion description and both user profiles (positive and negative), reflecting user term preferences, and the description length. For a given user u and sug-

gestion i, we calculated two features $cos^+(\overrightarrow{D_i}, \overrightarrow{M_u^+})$ and $cos^-(\overrightarrow{D_i}, \overrightarrow{M_u^-})$, where D_i is a description of suggestion S and M_u^+ and M_u^- are the positive and negative profiles for user u. The description length feature is simply the length of the description in characters. We believe that the length of description may be an important factor when the user explores the suggestions for an attraction as a longer description may provide more detailed information, allowing the user to be more sure of their rating.

3.4 Ranking Suggestions

To rank the potential suggestions for each user, we use their individual personal ranking model as described in Section 4.3. The personal ranking model was first used to determine the 50 most relevant suggestions for each geographical context, in descending order of confidence score as estimated by the classifier. The confidence score, in the case of a Naïve Bayes classifier, is simply the posterior probability that the suggestion belongs to class "relevant" and therefore encodes, in some sense, how likely it is that the user will like the candidate venue. This approach has been demonstrated to work well for ranking [7].

As mentioned in the previous section, there were a few profiles for which it wasn't sensible to build a classifier due to the level of imbalance between the 2 classes in the training data. In this case, potential suggestions were ranked by using only the user term preferences. We ordered the suggestions in descending order of their scores, which were calculated as the difference between cos^+ and cos^-. This is a reasonable, if slightly simplified approach, since it will return a positive value if the similarity between the candidate venue and the positive profile is greater than its similarity compared with the negative profile and vice-versa.

4 Results

In this section we present an overview of the performance of our system. Output suggestions were judged both by the original group of users who supplied the training data and NIST assessors. The user corresponding to each profile judged suggestions in the same manner as for the training examples, assigning a value of 0-4 for each *title and description* and *url*. NIST assessors judged suggestions in terms of geographical appropriateness. In total, 223 of the potential 31750 profile/context pairs were judged, i.e. not all pairs were used for evaluation. The top 5 suggestions for each profile/context pair were taken into account for evaluation.

To evaluate the performance of the model for the problem of Contextual Suggestion, three different measures were used: Precision at Rank 5 (P@5), Mean Reciprocal Rank (MRR) and Time-Biased Gain (TBG). P@5 and MRR are traditional evaluation metrics to measure the overall effectiveness of an IR system in terms of its ability to return a good ranked list. The TBG metric, on the other hand, was developed specially for the contextual suggestion task [3].

As the basis for evaluation, a suggestion was counted as "relevant" if the user liked both the description and the geographically appropriate document. All other suggestions were counted as "non-relevant". P@5 and MRR are calculated by using these definitions for relevant and non-relevant. The TBG metric is more complex and takes into account the impact of descriptions and disliked suggestions, which are ignored by P@5 and MRR. All the metrics were computed for each profile/context pair, and then averaged across all pairs.

We submitted two runs to the TREC 2013 Contextual Suggestion Track: *simpleScore* and *complexScore*. For the *simpleScore* run we used $\lambda = 0.5$ for all venue types, while for *complexScore* we used type-dependent values for λ as described in Section 3.3.1. The overlap of the top 5 suggestions between the two runs is about 41%, i.e. on average both of two runs contain 41% of the same suggestions among top 5 ranked suggestions for each profile/context pair. At the top rank the overlap is 40%, meaning that the top suggestions from two runs are generally different.

Table 2 shows evaluation results for our runs and the median score, which is calculated based on the results from all 34 runs submitted to the TREC track.

Run	**P@5**	**MRR**	**TBG**
simpleScore	**0.4332**	**0.5871**	**1.8374**
complexScore	0.4152	0.5777	1.8226
median	0.2368	0.3415	0.8593

Table 2: Results for our two runs and median scores.

The results show that the *simpleScore* run slightly outperforms the *complexScore* one: P@5 +4.2%, MRR +1.7%, TBG +0.8%. The difference can be explained by two main factors. First of all, the type-dependent values for λ may not reflect user behavior when making a decision, i.e. both the *title and description* and the *website* influence the user equally. Another reason is perhaps that the user study, which we organized with 7 participants, isn't a reliable methodology for estimating these values. They could perhaps be learned from the training data, however we leave this for future work.

The results of our runs demonstrate that our two runs greatly outperform the median score with P@5 +45%, MRR +41%, TBG +53% for our best run (*simpleScore*). Table 3 and 4 present more detailed analys of our results.

Run	**P@5**	**MRR**	**TBG**
simpleScore	21.97%	48.43%	13.90%
complexScore	21.97%	48.43%	15.70%

Table 3: Share of user/context pairs where particular run returned the best result over all entrants.

According to the MRR metric, our system was able to return the best result over all entrants for 48.43% of user/context pairs. When considering P@5, our system returned the best result in 22% of cases and was better than the median

Run	P@5	MRR	TBG
simpleScore	61.88%	53.36%	66.37%
complexScore	55.15%	52.47%	66.82%

Table 4: Share of user/context pairs where particular run returned better result better than the median over all entrants

score 61% of the time for *simpleScore* run. However according to TBG metric *complexScore* run performs slightly better than *simpleScore* in terms of share of user/context pairs with the best results and results which were higher than the median. We found that about 1.5% of all suggestions had a website which could not be loaded during the assessment procedure. Removing these suggestions from the top 5 leads to performance improvements of: P@5: +0.5%, MRR: +1% and TBG: +1.5% for both runs.

5 Analysis

Besides final/max/min/median results for each profile/context pair, the organizers also provided all judgments (description, website, geographical relevance) for each suggestion from the 223 profile/context pairs which were judged. Using an evaluation script provided for the track and these judgments we performed a more detailed analysis of our results.

Table 5 shows how the geographical relevance(G), description(D) and the website(W) ratings contributed to the P@5 and MRR scores. According to this statistic, almost all documents retrieved by our system were geographically appropriate to the context, suggesting that the approach of pre-filtering candidates is effective. The website of each suggestion and its description appear to contribute equally to final result quality. We found that there was a small correlation (0.271) between the number of candidates returned by the Google Places API for a city and the performance of the system, suggesting that it is easier to make good recommendations when there is a wide variety of possible candidates. The 4 context cities with the smallest population also have the worst performance in terms of P@5 and, unsurprisingly, there is a strong correlation between the population of a city and the number of candidates returned for it by the API (0.693).

In general, assessors judged 1115 suggestions from the Top 5 suggestions for 136 different profiles. These suggestions represent 772 unique venues, i.e. some venues were recommended for different profiles at the same time. We explored types of venues which were represented by these suggestions, amounting to a total of 24. Figure 1 presents a distribution over different types of venues for two runs: restaurants (23%), museums (12%) and parks (11.7%) are the most common types of venues. For each venue type we calculated a popularity score, which is the fraction of "relevant" suggestions over all of the suggestions made for that type. Figure 2 demonstrates that the most popular venues for assessor

	simpleScore			
	G	**W**	**D**	**Final(WDG)**
P@5	0.9363	0.5776	0.5381	0.4332
MRR	0.9675	0.7149	0.6700	0.5871

	complexScore			
	G	**W**	**D**	**Final(WDG)**
P@5	0.9381	0.5668	0.5283	0.4152
MRR	0.9720	0.7050	0.6773	0.5777

Table 5: Geographical relevance(G), description(D) and website(W) contributions into P@5 and MRR metrics for both runs.

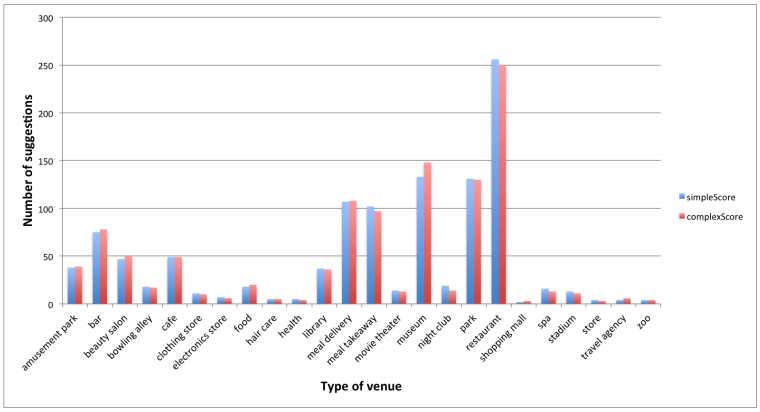

Figure 1: Distribution of numbers of suggestion for 24 different types of venues for both runs.

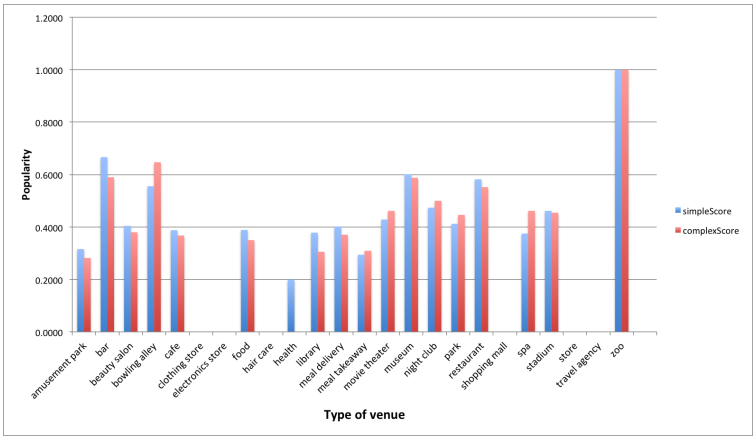

Figure 2: Popularity of 24 different types of venues for both runs.

were zoos, bars, museums and restaurants. This can perhaps be explained by the fact that there were few zoos recommended to users, and all of them were counted as "relevant". Restaurants, bars and museums are often suggested and are highly popular because they are very common tourist attractions and their overall popularity is perhaps not strongly affected by a visitor's interests. The popularity of venues such as travel agencies, shopping malls and electronics stores is 0, likely because these types of places are not particular attractive to tourists and are more likely to be frequented regularly by people who live in the area. In terms of distribution over different types of venues and popularity of venue's types, both runs are very similar.

6 Conclusions and Future Work

In this paper we have described a new system designed to take part in the TREC Contextual Suggestion track, for making context-sensitive recommendations to tourists visiting a new city. Based on analysis of results obtained from the same users who contributed the training data we have shown that the method is very effective for this problem and, when compared to the 34 other runs in the track, delivered results which were well above the median. In nearly half of all contexts our approach was able to deliver the best set of results, confirming that the choices made during the development of the system were sensible and beneficial. Our method is based on quite a simple strategy of using the descriptions of previously rated places to build user profiles, however we introduce a number of novel additions which have clearly lead to improved performance. Evaluation results of two submitted runs has shown that the runs are similar in terms of performance, although they have considerable different suggestions at the top 5 ranked ones.

There are several directions for future work. Our ranking model could be easily extended by adding new features to the classifier. For example, in the current ranking model we did not use information about the distance between the geolocation specified for each context and the venue, the venue rating (provided by content system) or the cuisine type of restaurants, cafes and bars. We believe that new features based on this information could allow the ranking model to reflect user preferences more precisely and that weighting suggestions by their distance from the user's location could lead to better acceptance of the recommendations made. It would also be interesting to use other content systems (such as Foursquare and TripAdvisor) to expand the list of potential candidates and the brief descriptions could perhaps be improved or tailed to the user's interests by also considering user reviews or comments from social networks. Finally, instead of estimating type-dependent values for λ via a user study, we could learn specifics weights for the λ parameter in our model from the training data. We leave all these directions to future work.

References

[1] D. Brain and G. Webb. On the effect of data set size on bias and variance in classification learning. In *Proceedings of the Fourth Australian Knowledge Acquisition Workshop (AKAW '99)*, pages 117–128, Sydney, Australia, 1999.

[2] A. Dean-Hall, C. L. A. Clarke, J. Kamps, P. Thomas, and E. Voorhees. Overview of the trec 2012 contextual suggestion track. In *Text REtrieval Conference (TREC)*, 2012.

[3] K. J. Dean-Hall A., Clarke CLA. and T. P. Evaluating contextual suggestion. In *The Fifth International Workshop on Evaluating Information Access (EVIA)*, pages 45–48, Tokyo, Japan, 2013.

[4] M. Harvey, B. Ludwig, and D. Elsweiler. You are what you eat: Learning user tastes for rating prediction. In O. Kurland, M. Lewenstein, and E. Porat, editors, *String Processing and Information Retrieval*, volume 8214 of *Lecture Notes in Computer Science*, pages 153–164. Springer, 2013.

[5] E. M. Voorhees. Variations in relevance judgments and the measurement of retrieval effectiveness. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR'98)*, pages 315–323. ACM Press, 1998.

[6] I. Witten and F. E. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan and Kaufmann, San Francisco, CA, USA, second edition, 2005.

[7] H. Zhang and J. Su. Naive bayesian classifiers for ranking. In *Proceedings of the 15th European Conference on Machine Learning (ECML2004)*, pages 501–512, Pisa, Italy, 2004. Springer.