Summary of the 15th Discovery Challenge
Recommending Given Names

Folke Mitzlaff[1], Stephan Doerfel[1], Andreas Hotho[2,3], Robert Jäschke[3], and Juergen Mueller[1,3]

[1] University of Kassel, Knowledge Discovery and Data Engineering Group, Wilhelmshöher Allee 73, 34121 Kassel, Germany
{mitzlaff, doerfel, mueller}@cs.uni-kassel.de
[2] University of Würzburg, Data Mining and Information Retrieval Group, Am Hubland, 97074 Würzburg, Germany
hotho@informatik.uni-wuerzburg.de
[3] L3S Research Center, Appelstraße 4, 30167 Hannover, Germany
{hotho, juergen.mueller, jaeschke}@l3s.de

The 15th ECML PKDD Discovery Challenge centered around the recommendation of given names. Participants of the challenge implemented algorithms that were tested both offline – on data collected by the name search engine Nameling – and online within Nameling. Here, we describe both tasks in detail and discuss the publicly available datasets. We motivate and explain the chosen evaluation of the challenge, and we summarize the different approaches applied to the name recommendation tasks. Finally, we present the rankings and winners of the offline and the online phase.

1 Introduction

The choice of a given name is typically accompanied with an extensive search for the most suitable alternatives, at which many constraints apply. First of all, the social and cultural background determines, what a common name is and may additional imply certain habits, such as, e. g., the patronym. Additionally, most names bear a certain meaning or associations which, also depend on the cultural context. Whoever makes the decision is strongly influenced by personal taste and current trends within the social context. Either by preferring names which are currently popular, or by avoiding names which most likely will be common in the neighborhood.

Future parents are often aided by huge collections of given names which list several thousand names, ordered alphabetically or by popularity. To simplify and shorten this extensive approach, the name search engine Nameling (see Section 2) allows its users to query names and returns similar names. To determine similarity, Nameling utilizes Wikipedia's text corpus for interlinking names and the microblogging service Twitter for capturing current trends and popularity of given names. Nevertheless, the underlying rankings and thus the search results are statically bound to the underlying co-occurrence graph obtained from Wikipedia and thus not personalized. Since naming a child is a very personal

task, a name search engine can certainly profit from personalized name recommendation.

The task of 15th ECML PKDD Discovery Challenge was to create successful recommendation algorithms that would suggest suitable given names to future parents. The challenge relied on data gathered by Nameling and consisted of two phases, i. e., an *offline* and an *online* challenge. In both phases, participants were asked to provide a name recommendation algorithm to solve the task.

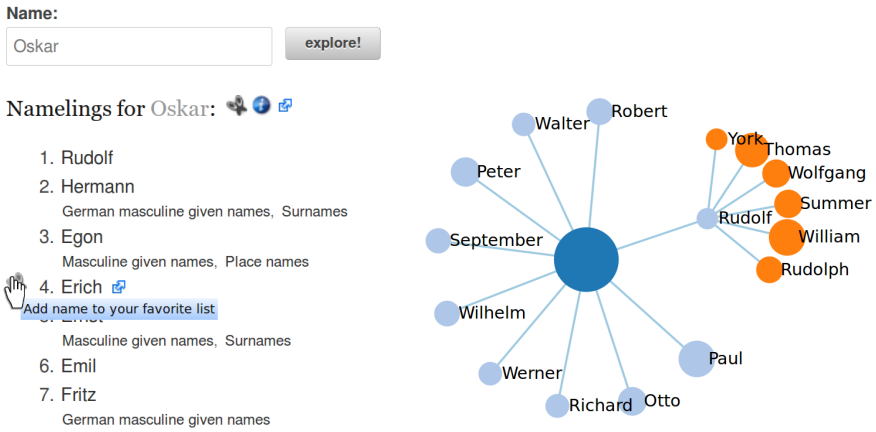
Task 1: The Offline Challenge. In the first phase, recommenders have been evaluated in an offline setting. To train an algorithm, the participants had been provided with a public training data set from Nameling's access logs, representing user search activities. Given a set of names for which a user had shown interest in, the recommender should provide suggestions for further names for that user. A second, private dataset from Nameling contained further search events from users of Nameling. To win the challenge, participants had to predict these search events. Details of the offline phase are discussed in Section 3 where we also summarize the different approaches to solve the challenge as well as the ranking of the participating teams and the challenge's winners.

Task 2: The Online Challenge. The online phase took place after Task 1 had been completed. The participants implemented a recommendation service that could be actively queried via HTTP and would provide names according to the participant's algorithm. These recommendation were shown to actual users of Nameling and their quality was measured by counting user's clicks on recommended names. We elaborate on the online challenge in Section 4.

2 The Name Search Engine Nameling

Nameling is designed as a search engine and recommendation system for given names. The basic principle is simple: The user enters a given name and gets a browsable list of relevant, related names, called "*namelings*". As an example, Figure 1a shows the namelings for the classical masculine German given name "Oskar". The list of namelings in this example ("Rudolf", "Hermann", "Egon", etc.) exclusively contains classical German masculine given names as well. Whenever an according article in Wikipedia exists, categories for the respective given name are displayed, as, e. g., "*Masculine given names*" and "*Place names*" for the given name "Egon". Via hyperlinks, the user can browse for namelings of each listed name or get a list of all names linked to a certain category in Wikipedia. Further background information for the query name is summarized in a corresponding details view, where, among others, popularity of the name in different language editions of Wikipedia as well as in Twitter is shown. As depicted in Figure 1b, the user may also explore the "neighborhood" of a given name, i. e., names which co-occur often with the query name.

From a user's perspective, Nameling is a tool for finding a suitable given name. Accordingly, names can easily be added to a personal list of favorite names. The list of favorite names is shown on every page in the Nameling and can be shared with a friend, for collaboratively finding a given name.



(a) Namelings (b) Co-occurring names

Fig. 1: A user query for the classical German given name "Oskar".

2.1 Computing Related Names

To generate the lists of related names, Nameling makes use of techniques that have become popular in the so called "Web 2.0". With the rise of the latter, various social applications for different domains – offering a huge source of information and giving insight into social interaction and personal attitudes – have emerged that make use of user generated data (e. g., user profiles and friendships in social networks or tagging data in bookmarking systems).

The basic idea behind Nameling was to discover relations among given names, based on such user generated data. In this section, we briefly summarize how data is collected and how relations among given names are established. Nameling is based on a comprehensive list of given names, which was initially manually collected, but then populated by user suggestions. Information about names and relations between them is gathered from three different data sources, as depicted in Figure 2:

Wikipedia: As basis for discovering relations among given names, co-occurrence graphs are generated for each language edition of Wikipedia separately. That is, for each language, a corresponding data set is downloaded from the Wikimedia Foundation[4]. Afterwards, for any pair of given names in our dataset, the number of sentences where they jointly occur is determined. Thus, an undirected graph is obtained for every language, where two names are adjacent if they occur together at least in one sentence within any of the articles and the edge's weight is given by the number of such sentences.

[4] "Database dump progress.", *Wikimedia*. 2012. `http://dumps.wikimedia.org/backup-index.html` (May 1, 2013)

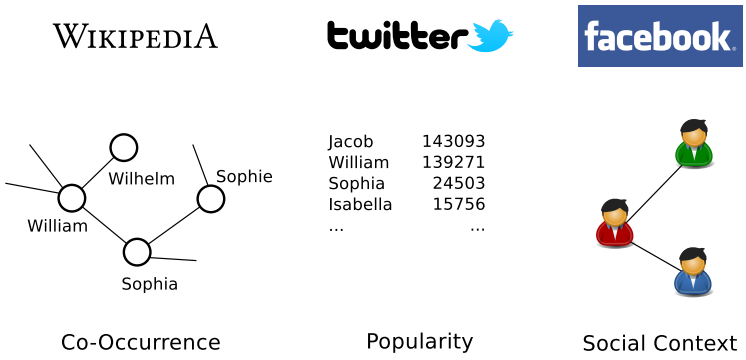


Fig. 2: Nameling determines similarities among given names based on co-occurrence networks from Wikipedia, popularity of given names via Twitter and social context of the querying user via Facebook.

Relations among given names are established by calculating a vertex similarity score between the corresponding nodes in the co-occurrence graph. Currently, namelings are calculated based on the cosine similarity (cf. [2]).

Twitter: For assessing up-to-date popularity of given names, a random sample of status messages in Twitter is constantly processed via the Twitter streaming API[5]. For each name, the number of tweets mentioning it is counted.

Facebook: Optionally, a user may connect Nameling with Facebook[6]. If the user allows Nameling to access his or her profile information, the given names of all contacts in Facebook are collected anonymously. Thus, a "social context" for the user's given name is recorded. Currently, the social context graph is too small for implementing features based on it, but it will be a valuable source for discovering and evaluating relations among given names.

2.2 Research around Nameling

Beside serving as a tool for parents-to-be, Nameling is a research platform too. The choice of a given name is influenced by many factors, ranging from cultural background and social environment to personal preference. Accordingly, the task of recommending given names is per se subject to interdisciplinary considerations.

Within Nameling, users are anonymously identified via a cookie that is, a small identification fragment which uniquely identifies a user's web browser. Although a single user might use several browsers or computers, Nameling uses the simple heuristic of treating cookies identification for users.

[5] Twitter Developers. `https://dev.twitter.com/docs/api/1/get/statuses/sample` (May 3, 2013)

[6] `https://facebook.com/`

The Nameling dataset arises from the requests that users make to Nameling. More than 60,000 users conducted more than 500,000 activities within the time range of consideration (March 6th, 2012 until February 12th, 2013). For every user, Nameling tracks the search history, favorite names and geographical location based on the user's IP address and the GeoIP[7] database. All these footprints together constitute a multi-mode network with multiple edge types. Analyzing this graph (or one of its projections) can reveal communities of users with similar search characteristics or cohesive groups of names, among others. In the context of the Discovery Challenge, the data of Nameling is used to train and to test given name recommendation algorithms.

3 Offline Challenge

The first task of the Discovery Challenge was to create an algorithm that produces given name recommendations – given a list of users with their previous history of name search requests to Nameling. The evaluation of these algorithms was conducted in a classical offline prediction scenario. A large dataset from Nameling was split into a public training dataset and a secret test dataset. In the following we describe details of the task and the dataset. In the second part of this section we summarize the participants' approaches and their results in the challenge.

3.1 Task

In the challenges, we deal with a standard binary item recommendation task. Users of the name search engine Nameling have expressed interest in certain names by searching for them or requesting their details. These interactions with the system are interpreted as (binary) positive feedback to these names, while there is no explicit negative feedback - only names towards which we do not know the user's attitude. A recommender algorithm must determine, which of these names will be of interest to the user.

Participants were given a public dataset to train their algorithms on. For the overall evaluation a second dataset containing only a list of users was given to them. The task in the offline challenge then was to produce for each user in that second dataset a list of 1,000 name recommendations, ordered by their relevance to the user at hand.

For the challenge no further restrictions were made regarding the choice of methodology or additional data. On the contrary, participants were encouraged to make use of any kind of data they might find suitable, e. g., family trees, location information, data from social networks, etc.

[7] "GeoIP databases and web services.", *MaxMind*. `http://www.maxmind.com/en/geolocation_landing` (May 3, 2013)

3.2 Challenge Data

For the challenge, we provided data from the name search engine Nameling, containing users together with their (partial) interaction history in Nameling. A user interaction hereby is always one of the following:

`ENTER_SEARCH` The user entered a name directly into Nameling's search field.
`LINK_SEARCH` The user followed a link on some result page (including pagination links in longer result lists).
`LINK_CATEGORY_SEARCH` Wherever available, names are categorized according to the corresponding Wikipedia articles. The users clicked on such a category link to obtain all accordingly categorized names.
`NAME_DETAILS` The user requested some detailed information for a name using a respective button.
`ADD_FAVORITE` The user added a name to his list of favorite names.

The full dataset contains interactions from Nameling's query logs, ranging from March 6th, 2012 to February 12th, 2013. It contains profile data for 60,922 users with 515,848 activities. This dataset was split into a *public training dataset* and a *secret test dataset*. For this purpose, a subset of users (in the following called test users) was selected for the evaluation. For each such test user, we withheld some of their most recent activities for testing according to the following rules:

- For each user, we selected the chronologically last two names for evaluation which had directly been entered into Nameling's search field (i.e., `ENTER_SEARCH` activity) and which are also contained in the list of known names. We thereby considered the respective time stamp of a name's first occurrence within the user's activities. We restricted the evaluation to `ENTER_SEARCH` activities, because all other user activities are biased towards the lists of names which were displayed by Nameling (see our corresponding analysis of the ranking performance in [2]).
- We considered only those names for evaluation which had not previously been added as a favorite name by the user.
- All remaining user activity after the (chronologically) first evaluation name has been discarded.
- We required at least three activities per user to remain in the data set.
- For previous publications, we already published part of Nameling's usage data. Only users not contained in this previously published data set, have been selected as test users.

With the above procedure we obtained two data sets[8]: The secret evaluation data set containing for each test user the two left out (i. e., `ENTER_SEARCH`) names and the public challenge data set containing the remaining user activities of the test users and the full lists of activities from all other users. The only

[8] Both datasets are available from the challenge's website: `http://www.kde.cs.uni-kassel.de/ws/dc13/downloads/`

applied preprocessing on our part was a conversion to lower case of all names. Additionally to the public training dataset, participants were provided with the list of all users in the test dataset to be used as input for their algorithms. Furthermore, we published a list of all names known to Nameling, which thus included all names occurring in the training or the test data (roughly 36,000 names).

3.3 Evaluation

Given the list of test users (see above), each participant produced a list of 1,000 recommended names[9] for each such user. These lists were then used to evaluate the quality of the algorithm by comparing for each test user the 1,000 names to the two left-out names from the secret test dataset. As usual, it is assumed that good algorithms will rank the left-out names high, since they represent the actual measurable interests of the user at hand.

The chosen assessment metric to compare the lists of recommendations is *mean average precision* (MAP@1000). MAP means to compute for each test user the precision at exactly the ranking positions of the two left-out names. These precision values are then first averaged per test user and finally in total to yield the score for the recommender at hand. While MAP usually can handle arbitrarily long lists of recommendations, for the challenge we restricted it to MAP@1000, meaning that only the first 1,000 positions of a list are considered. If one or both of the left out names were not among the top 1,000 name in the list, they were treated as if they were ranked at position 1,001 and 1,002 respectively. More formally, the score assigned to a participant's handed-in list is

$$\text{MAP@1000} := \frac{1}{|U|} \sum_{u=1}^{|U|} \left(\frac{1}{r_{1,u}} + \frac{2}{r_{2,u}} \right)$$

where U is the set of all test users, $r_{1,u}$ and $r_{2,u}$ are the ranks of two left-out names for user u from the secret evaluation dataset, and $r_{1,u} > r_{2,u}$.

The choice of the evaluation measure is crucial in the comparison of recommender algorithms. It is well-known that different measures often lead to different evaluation results and the choice of the metric must therefore be motivated by the use case at hand. In the case of Nameling, we had already seen that recommending given names is a difficult task [3]. For many users, many recommenders did not produce recommendation rankings with the test names among top positions. Thus, measures like precision@k – with k typically obtaining low values like 5 or 10 – make it hard to distinguish between results, especially for lower cut-off-thresholds k. MAP (Mean Average Precision) is a measure that is suitable for (arbitrarily long) ordered lists of recommendations. Like NDCG (Normalized Discounted Cumulative Gain) or AUC (Area Under the Curve) it evaluates the recommendations based on the positions of the left

[9] 1,000 name sound like a large number but given that parents currently read much longer and badly sorted name lists the number is reasonable (details below).

out items within the list of recommendations. It yields good scores when test items appear on the top positions of the recommendations and lower scores if they are ranked further below (unlike precision@k where lower ranked items are cut off and thus do not contribute to the score).

The main difference to AUC and NDCG is how the ranks of the left-out names are incorporated into the score. While AUC yields a linear combination of the two ranks, MAP takes a linear combination of the reciprocal ranks and NDCG a linear combinations of the (logarithmically) smoothed reciprocal ranks. Among these measures, MAP is the one that discriminates the strongest between higher or lower ranks and therefore was most suitable for the challenge.

Although we have just argued against cut-off-measures like precision@k it is reasonable to cut off lists at some point. In contrast to many other areas where recommendations are used (e.g., friend, movie, book, or website recommenders), in Nameling the time needed to evaluate a recommendation is very short: if you like a name, just click on it. Additionally, the cost in terms of money or time spent for following a recommendation that turns out bad, is very low. At the same time, finding the perfect name for a child is often a process of months rather than minutes (like for finding the next book to read or deciding which movie to watch) or seconds (deciding which tag to use or which website to visit on the net). Thus it is reasonable to assume that parents-to-be are willing to scroll through lists of names longer than the usual top ten – especially, considering that one of the traditional ways of searching for names is to go through first names dictionaries where names are listed unpersonalized, in alphabetical order. In such books usually there are a lot more than 1,000 names that have to be read and therefore it seems reasonable that readers of such books won't mind studying longer name lists on the web.

3.4 Summary of the Offline Challenge

Registered participants (or teams of participants) were given access to the public training dataset and the dataset containing the names of all test users. The offline challenge ran for about 17 weeks, beginning March 1st and ending July 1st, 2013. Every week, participants were invited to hand in lists of recommended names for the test users. These lists were evaluated (using the secret test dataset and the evaluation described above) to produce a weekly updated leaderboard. The leaderboard allowed the participants to check on the success of their methods and to compare themselves with the other participants. Since frequently updated results would constitute an opportunity for the participants to optimize their algorithms towards this feedback or even to reverse-engineer the correct test names, the leaderboard was not updated more often than once a week.

Participants and Winners More than 40 teams registered for the challenge of which 17 handed in lists of recommended names. Of these 17, six teams submitted a papers which are summarized below in Section 3.5. Table 1 shows the final scores of the 17 teams and reveals the winners of the offline phase:

1. place goes to team *uefs.br* for their approach using a features of names.
2. place is won by team *ibayer* using a syntactically enriched tensor factorization model.
3. place goes to team *all your base* for their algorithm exploiting a generalized form of association rules.

Table 1: The final results of the 17 teams in the offline challenge, showing all the participants' team names, together with the achieved MAP@1000 score.

Pos.	Team Name	MAP@1000
1	uefs.br	0,0491
2	ibayer	0,0472
3	all your base	0,0423
4	Labic	0,0379
5	cadejo	0,0367
6	disc	0,0340
7	Context	0,0321
8	TomFu	0,0309
9	Cibal	0,0262
10	thalesfc	0,0253
11	Prefix	0,0203
12	Gut_und_Guenstig	0,0169
13	TeamUFCG	0,0156
14	PwrInfZC	0,0130
15	persona-non-data	0,0043
16	erick_oliv	0,0021
17	Chanjo	0,0016

Figure 3 shows the scores of those six teams that handed in papers in time describing their approach. Only described approaches can be judge and presented at the workshop and therefore, all the other results are not considered in the remaining discussion. Additionally, two baselines (NameRank from [3] and the simple most-popular recommender) are presented in Figure 3. The latter simply suggests to any user those names that have been queried the most often in the past. It is thus unpersonalized and rather ad-hoc. NameRank is a variant of the popular personalized PageRank algorithm [1]. From the baseline results we can already tell that the recommendation problem is indeed hard, as the scores are rather low (between 0.025 and 0.030). On the other hand, we can observe that the simple most-popular is not that much worse than the much more sophisticated PageRank-like approach. The first approaches of almost all participants yielded scores lower or comparable to those of the baselines. However, over the course of the challenge the scores improved significantly and by the end of the challenge all teams had produced algorithms that outperformed both baselines.

To compare the recommenders' performances in greater detail, Figure 4 shows the cumulative distribution of the different ranking positions $(1, \ldots, 1000)$ for the

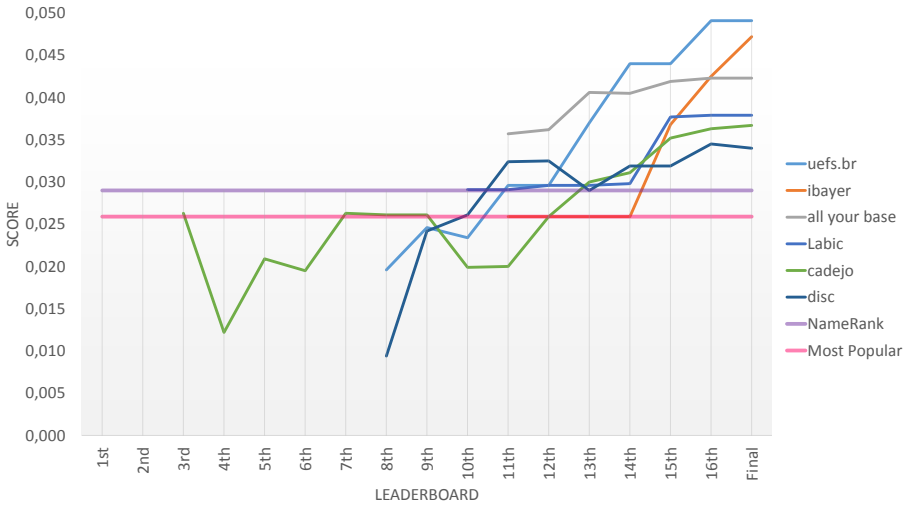


Fig. 3: The scores of the six teams that handed in papers plotted of the 17 weeks runtime of the offline challenge. For comparison, two baselines have been added (the two constant lines): NameRank and Most Popular.

top three algorithms. For each recommender system and every ranking position k, displayed is the number of hold-out names – out of the $8,280$ hold-out names from the $4,140$ test users with two hold-out names each – that had a rank smaller than or equal to k on the list of recommended names. We can observe, that the three curves are very close together, indicating that the distributions of ranks are similar. Comparing the results of the top placed two algorithms (team *uefs.br* and team *ibayer*), we see that the former has predicted more of the hold-out names in its top $1,000$ lists in total. However, the latter succeeded in placing more hold-out names among the top ranking positions ($k \leq 400$). The distribution of the third algorithm (team *all your base*) is almost identical with that of team *uefs.br* over the first 300 ranks, but then falls behind.

3.5 Approaches

In the offline phase of the challenge, six teams documented their approaches in the papers that are included in the challenges proceedings. In the following, the key idea of each approach is summarized. Using the respective team name of each paper's authors, their scores can be identified in Figure 3.

A mixed hybrid recommender system for given names
The paper by Rafael Glauber, Angelo Loula, and João B. Rocha-Junior (team *uefs.br*) presents a hybrid recommender which combines collaborative filtering, most popular, and content-based recommendations. In particular the latter contributes with two interesting approaches (Soundex and splitting

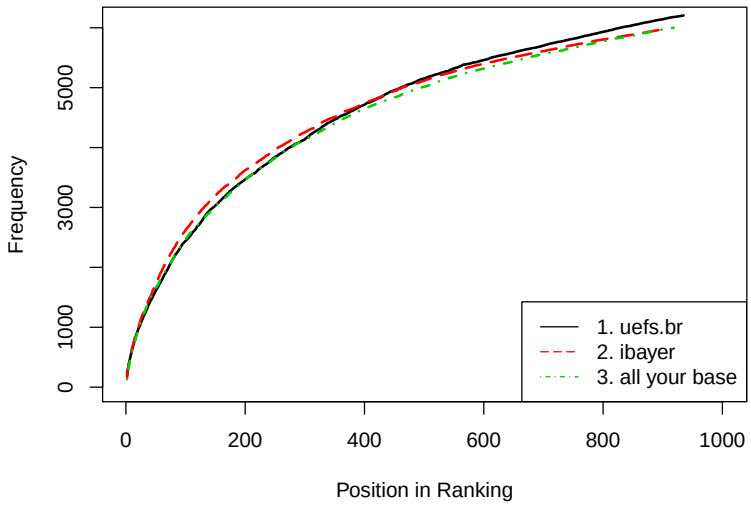


Fig. 4: Cumulative distribution of the ranking positions for the top three recommendation systems described in Section 3.5.

of invalid names) that are fitted to the problem at hand. The combination of the three approaches as a concatenation is likely the reason for the success in the challenge, yielding the highest score on the test dataset and thus winning the offline phase.

Collaborative Filtering Ensemble for Personalized Name Recommendation
Bernat Coma-Puig, Ernesto Diaz-Aviles, and Wolfgang Nejdl (team *cadejo*) present an algorithm based on the weighted rank ensemble of various collaborative filtering recommender systems. In particular, the classic item-to-item collaborative filtering is considered in different, specifically adopted variants (considering only `ENTER_SEARCH` activities vs. all activities, frequency biased name sampling vs. recency biased name sampling), item- and user-based CF as well as PageRank weights for filling up recommendation lists with less than 1,000 elements. The weights of the ensemble are determined in experiments and yield a recommender that outperforms each of its individual components.

Nameling Discovery Challenge - Collaborative Neighborhoods
The paper by Dirk Schäfer and Robin Senge (team *disc*) created an algorithm which combines user-based collaborative filtering with information about the geographical location of the users and their preference for male/female and long/short names. The paper further explores the use of two different similarity measures, namely Dunning and Jaccard, and find that the rather exotic one Dunning yields better recommendations than the Jaccard measure.

Improving the Recommendation of Given Names by Using Contextual Information

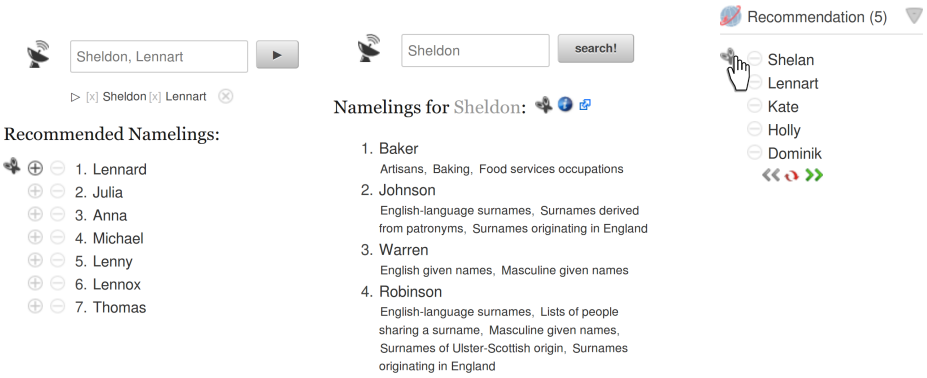
The paper by Marcos Aurélio Domingues, Ricardo Marcondes Marcacini, Solange Oliveira Rezende and Gustavo E. A. P. A. Batista (team *Labic*) presents two approaches to tackle the challenge of name recommendation: item-based collaborative filtering and association rules. In addition, the weight post filtering approach is leveraged to weight these two baseline recommenders by contextual information about time and location. Therefore, for each user-item pair the probability that the user accessed it at a certain context (i.e., time or location) is computed and used to weight the baseline results.

Similarity-weighted association rules for a name recommender system
The paper by Benjamin Letham (team *all your base*) considers association rules for recommendation. The key idea here is the introduction of an adjusted confidence value for association rules, capturing the idea of inducing a bias towards observations which stem from likeminded users (with respect to the querying user). This generalized definition of confidence is additionally combined with a previous approach of the author [4] which accounts for association rules with low support values, by adding in a Beta prior distribution. This recommender system achieved the third place in the challenge's offline task.

Factor Models for Recommending Given Names
The paper by Imannuel Bayer and Steffen Rendle (team *ibayer*) presents an approach using a sequential factor model that is enriched with syntactical name similarity – a prefix equality, called "*prefix smoothing*". The model is trained with a slight adoption of the standard Bayesian Personalized Ranking algorithm, while the final recommendation is obtained by averaging the rankings of different, independently trained models. This recommender system achieved the second place in the challenge's offline task.

4 Online Challenge

Conducting offline experiments is usually the first step to estimate the performance of different recommender systems. However, thus recommender systems are trained to predict those items that users found interesting without being assisted by a recommender system in the first place. In an online evaluation different recommenders are implemented into the running productive system. Their performance is compared in a test, where different users are assigned to different recommenders and the impact of the recommendations is estimated by analyzing the users responses to their recommendation. Like noted in [5] online experiments provide "the strongest evidence as to the true value of the system [...]", but have also an influence on the users as the recommendations are displayed before users decide where to navigate (click) next. In the challenge, the online phase gave the participants the opportunity to test the algorithms, they had created during the offline phase in Nameling. In the following we describe the setup that allowed the teams to integrate their algorithms with Nameling before we discuss this phase's results and winners.



(a) interactive recommendation

(b) sidebar recommendation

Fig. 5: Implemented recommendation use cases in Nameling: A user interactively queries for suitable name recommendations (a) or gets recommended names displayed in the sidebar of some regular Nameling page (b).

4.1 Recommendations in Nameling

The feature of recommendations in Nameling was introduced with the beginning of the challenge's online phase. To every page of the system was added a personalized list of recommended names. This list automatically adapts to the user's activities (e.g., the user's clicks or entering of favorite actions). Users may use this list to further search for names by clicking on one, add a name to his favorite names, or ban a name which they do not want to be recommended again. Additionally, users can visit an interactive recommendation site in Nameling, where they can enter names and will get personalized recommendations related to those names. The latter functionality is very similar to the usual results Nameling shows, the difference being that regular search results are non-personalized. Figure 5 shows how recommendations are displayed in Nameling's user interface.

To integrate their algorithms into Nameling, the participants had to implement a simple interface. The search engine itself provides a framework for the easy integration of recommender systems based on lightweight REST (HTTP + XML / JSON) interaction. Participants could choose to implement their recommender in Java or Python and to run their recommender in a web service on their own or to provide a JAR file to be deployed by the challenge organizers.

The recommender framework that integrates the different third-party recommender systems into Nameling is sketched in Figure 6. When a user of Nameling sends a request, a call for recommendations including the current user's ID is sent to each participating recommender. Recommendations are collected from each such recommender within a time frame of 500 ms, i. e., recommendations produced after that time are discarded. Using an equally distributed random

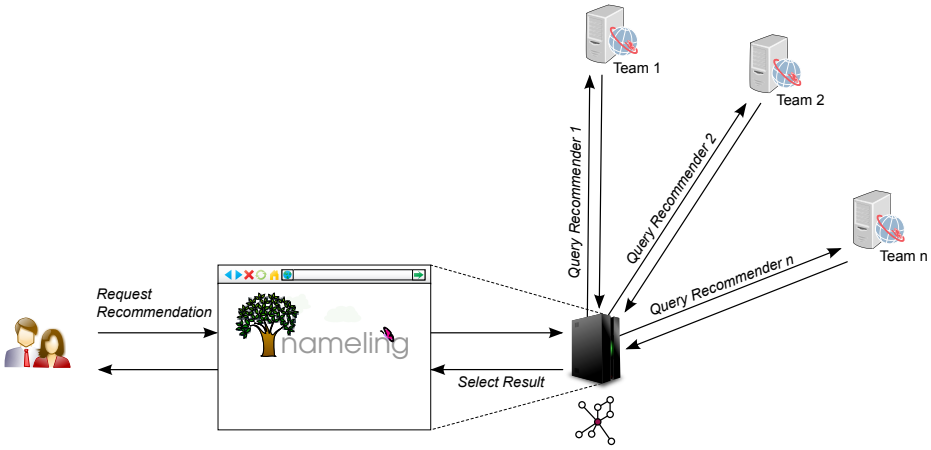


Fig. 6: Schematic representation of Nameling's online recommender framework.

function, one of the recommendation responses is selected and the recommended names are displayed to the current user in the response to their request. Once a recommender has been assigned to a user, this assignment is fixed for the duration of the current session unless the user specifically requests new recommendations.

4.2 Evaluation

Assessing the success of recommendations in a live system requires some quantity that can be measured and that represents the use of the recommendations to the user or to the system. Often used measures include a rise in revenue (e. g., for product recommendations), click counts on the recommended items, or comparisons to ratings that users assigned to recommended items. In the case of name recommendations, no particular revenue is created for the system as there are no products to be sold. Thus to evaluate different recommenders we focused on the interest that users showed in the recommended names. For the challenge we estimated this interest by the combined number of requests users made responding to the recommendations. More precisely, we counted all interactions that could be made on the recommender user interface (i. e., `LINK_SEARCH`, `LINK_CATEGORY_SEARCH`, and `ADD_FAVORITE` events). Here, we excluded the previously mentioned option to ban names as their interpretations is unclear. On the one hand, banning a name is certainly a negative response to that particular recommendation. On the other hand, since users are not bound to react to the recommendations at all, it is a clear sign of interest in the recommendations and could well be interpreted as a deselection of one uninteresting name among from a set of otherwise interesting names. Since the recommenders were assigned to different users using an equally distributed random function, the final measure

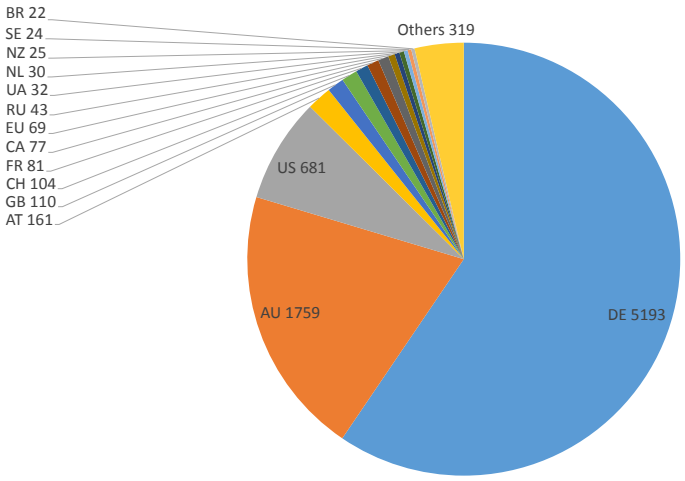


Fig. 7: The countries (according to IP address) of Nameling's visitors during the online phase of the challenge.

was simply the sum of all considered requests in one of the three mentioned categories.

4.3 Summary of the Online Challenge

The online phase ran from August, 1st to September 24th, 2013. During the time of the online phase, more than 8,000 users visited Nameling engaging in more than 200,000 activities there. Figure 7 shows the distributions of the users over their countries (it is to be expected, the home country is an important influence on the choice of names). While most of the requests came from Germany, followed by Austria, the largest number of visitors from an English speaking country came from the US.

Participants and Winners Of the teams that contributed to the challenge proceedings, five entered their algorithms in the online challenge. Of those five teams, four managed to produce recommendations within the time-window of 500 ms: "all your base", "Contest", "ibayer", and "uefs.br". Figure 8 shows for each of these four teams the number of responses – in terms of clicks to one of three categories of links related to the recommended names (see Section 4.2 – to their recommendations. The clear winner of the online phase is team "ibayer": Immanuel Bayer and Steffen Rendle (Paper: Factor Models for Recommending Given Names). Ranks two and three go to teams "all your base" and "uefs.br" respectively. Compared to the offline challenge, we find, the three top teams of the offline phase were the same that constituted the top three of the online

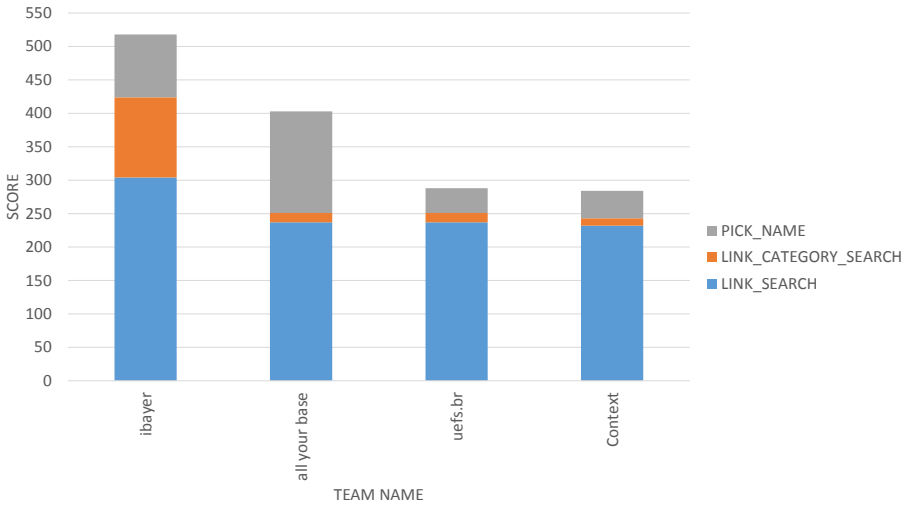


Fig. 8: Schematic representation of Nameling's online recommender framework.

phase. It is however interesting to note that the order of the teams changed – team "uefs.br" fell from rank one to rank three. It is also worth noting that although team "Context" yielded a MAP@1000 score of only 0,0321 in the online challenge, compared to team "uefs.br" with 0,0491, both teams were almost equally successful in the online phase. It thus seems that the offline testing has indeed been a reasonable precursor, yet also that the offline scenario does not fully capture the actual use case.

5 Conclusion

The 15th Discovery Challenge of the ECML PKDD posed the task of recommending given names to users of a name search engine. In the two parts of the challenge, the offline and the online phase several teams of scientists implemented and augmented recommendation algorithms to tackle that problem. In their approaches, participants mainly chose to use well-established techniques like collaborative filtering, tensor factorization, popularity measures, or association rules and hybridization thereof. Participants adapted such algorithms to the particular domain of given names exploiting name feature like gender, a name's prefix, a name's string length, or phonetic similarity. In the offline challenge, six teams entered their approaches and by the end of the phase, each team had produced a new algorithm outperforming the previously most successful recommender NameRank. The achieved scores of the individual recommenders were yet rather low (compared to other domains were recommenders are applied). This shows that there is yet much to be explored to better understand and predict the attitude of users towards different names. Through the challenge, a multitude of

ideas and approaches has been proposed and a straight forward next step will be to explore their value in hybrid recommender algorithms. Hybridization has been used already by several participants with great success.

The online challenge opened the productively running name search engine Nameling to the scientific community, offering the possibility to implement and test name recommendation algorithms in a live system. Results showed that the actual performance varied from that measured in the offline challenge. However, it could also be observed that despite the low scores in the offline phase, the recommendations were perceived by users and were able to attract their attention.

As organizers, we would like to thank all participants for their valuable contributions and ideas.

References

1. S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.
2. F. Mitzlaff and G. Stumme. Relatedness of given names. *Human Journal*, 1(4):205–217, 2012.
3. F. Mitzlaff and G. Stumme. Recommending given names, 2013. cite arxiv:1302.4412Comment: Baseline results for the ECML PKDD Discovery Challenge 2013.
4. C. Rudin, B. Letham, A. Salleb-Aouissi, E. Kogan, and D. Madigan. Sequential event prediction with association rules. *COLT 2011 - 24th Annual Conference on Learning Theory*, 2011.
5. G. Shani and A. Gunawardana. Evaluating recommendation systems. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*, pages 257–297. Springer US, 2011.