Learning Profile-Based Recommendations for Medical Search Auto-Complete

Guusje Boomgaard[a], Selene Báez Santamaría[a], Ilaria Tiddi[a], Robert Jan Sips[b] and Zoltán Szlávik[b]

[a] *Vrije Universiteit Amsterdam , De Boelelaan 1111, 1081 HN, Amsterdam, The Netherlands*

[b] *myTomorrows, Anthony Fokkerweg 61, 1059 CP, Amsterdam, The Netherlands*

Abstract

Query popularity is a main feature in web-search auto-completion. Several personalization features have been proposed to support specific users' searches, but often do not meet the privacy requirements of a medical environment (e.g. clinical trial search). Furthermore, in such specialized domains, the differences in user expertise and the domain-specific language users employ are far more widespread than in web-search. We propose a query auto-completion method based on different relevancy and diversity features, which can appropriately meet different user needs. Our method incorporates indirect popularity measures, along with graph topology and semantic features. An evolutionary algorithm optimizes relevance, diversity, and coverage to return a top-k list of query completions to the user. We evaluated our approach quantitatively and qualitatively using query log data from a clinical trial search engine, comparing the effects of different relevancy and diversity settings using domain experts. We found that syntax-based diversity has more impact on effectiveness and efficiency, graph-based diversity shows a more compact list of results, and relevancy the most effect on indicated preferences.

Keywords

Query Auto-Completion, Medical Information Retrieval, Knowledge Graphs, Professional Search

1. Introduction

Despite its promising results in web-search, query frequency or Most Popular Completion (MPC) is not always suitable to provide high-quality auto-complete suggestions [1, 2, 3]. Creating an unbiased Query Auto-Completion (QAC) algorithm can be challenging as MPC can steer users into a specific direction, causing a self-enforcing loop as click frequencies increase [4, 5]. Additionally, QAC suffers from position-bias as users tend to click more often on items that occur more often [6], which could undermine the quality of the auto-complete suggestions, especially in the context of domain-specific search engines.

When QAC is transferred to a medical search engine, three problems occur. Firstly, personalized features that are often included in web-search QAC systems (e.g. demographics, search

In A. Martin, K. Hinkelmann, H.-G. Fill, A. Gerber, D. Lenat, R. Stolle, F. van Harmelen (Eds.), Proceedings of the AAAI 2021 Spring Symposium on Combining Machine Learning and Knowledge Engineering (AAAI-MAKE 2021) - Stanford University, Palo Alto, California, USA, March 22-24, 2021.

g.boomgaard@student.vu.nl (G. Boomgaard); s.baezsantamaria@vu.nl (S.B. Santamaría); i.tiddi@vu.nl (I. Tiddi); r.sips@mytomorrows.com (R.J. Sips); zoltan.szlavik@mytomorrows.com (Z. Szlávik)

https://kmitd.github.io/ilaria/ (I. Tiddi)

context, location) may be unwanted, especially when dealing with sensitive information or when unbiased suggestions are required. Secondly, differences in user expertise are far more widespread than in web-search, i.e. a system often has to answer needs of patients, healthcare providers and pharmaceutical professionals at the same time. Thirdly, the language of different users is also diverse, thus raising problems with processing the user input as well. For example, medical specialists use different language than laymen [7], similarly to native vs. non-native English speakers [8]. These differences in user population come with different requirements, which may be hard to tackle with a single solution.

This research aims to improve the disease auto-completion process of a clinical trial search engine at the e-Health company myTomorrows[1]. The existing QAC method incorporates string similarity, Pubmed and clinical trial statistics, and string length. However, the provided suggestions suffer from redundancy; and due to the vast amount of matches to any short prefix, there is a need for an intelligent selection and ranking of suggested terms. In this paper, we propose to improve the QAC method using a graph-based taxonomy of medical conditions. This structured knowledge source combines semantic information, corpus statistics and graph topology, allowing us to study how different types of relevancy and diversity may aid in avoiding the common problem of suggestion redundancy [9] and to support different user profile needs. We evaluate the effectiveness (recall) and efficiency (tokens saved rate) of each method in finding the intended suggestion with the goal of understanding how to support different user profiles. We also evaluate the set of suggestions presented to the user, both in set length and coverage. As a result, we provide alternative approaches for frequency-based and personalized methods, and recommending different versions based on the requirements for various user profiles.

2. Related Work

In this section, we present generic literature on web and professional user profiling, then focusing on the medical domain. We then present methods to improve auto-complete suggestions, i.e. combining relevancy with diversity, individual tokens in queries, and knowledge graphs.

Web vs. professional search. With the increasing usage of the web, a large body of work has focused on the analysis of query logs both to profile single users and cohorts [10, 11] in web search contexts, revealing that (a) additional external knowledge about users and the search corpus can be relevant for personalization, and (b) different techniques can be relevant for different target type. Several studies [12, 13] have investigated the search practices and preferences in different specific domains (e.g. legal, recruitment, academia, healthcare professionals), showing that challenges such as boolean query formulation, the need of knowledge management and sharing across searches, and the ambivalence of relevance ranking are common despite the domain differences.

Various studies have investigated how to automatically identify medical experts and laypeople using query log data. White et al. [14] developed a general model to predict whether users were domain experts in four different domains, namely medicine, finance, law, and computer science. Knowledge and usage of Pubmed was identified as a salient feature for medical experts. Similarly, Palotti et al. [15] estimates medical expertise by using two query log sources aimed

[1] https://search.mytomorrows.com/search

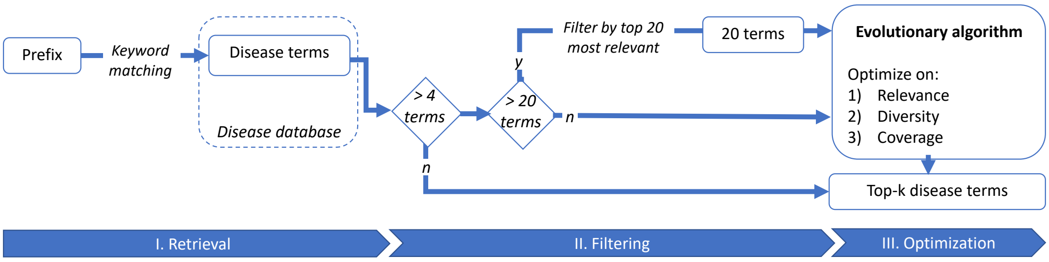


Figure 1: Proposed method for Query Auto-Completion.

at different audiences to differentiate between medical experts and laypeople. Pang et al. [16] also show that people searching for health-related topics often engage in a more exploratory type of search (e.g. visiting multiple websites) in an effort to fill knowledge gaps and receive hints for correct spelling.

Improving recommendations. Relevance and diversity have been identified as requirements for high-quality recommendations [17]. Relevancy in QAC is often defined as an item being popular, since the historical frequency is often a good predictor of the likelihood that the item will be searched again in the future, making MPC [1] a widely used approach. However, MPC tends to overlook long-tail queries and causes redundancy in the list of results [18]. Diversity can be introduced as the counterpart to relevancy, following a Goldilocks principle. This equilibrium is closely related to the balance achieved between exploration, which typically favors long and diverse lists, and exploitation, which is heavily relevance focused. Some studies investigate approaches to combine these bi-criteria [9], a more effective tri-criteria approach is presented by Zhong et al. [19], where *local diversity* (i.e. the dissimilarity between top-k returned items) is distinguished from *global diversity* (i.e. how many different relevant non-returned items are similar to at least one of the returned items).

Term importance. Given a user's input prefix (e.g. `di-`), individual tokens (e.g. a single word like `disease`) may be relevant to a different degree (e.g. looking for `diabetes`, as opposed to `pulmonary disease`). In Groza and Verspoor [20], term importance is applied to improve biomedical concept recognition in texts, using concepts from the UMLS Metathesaurus to create a representation of a document[2]. This document is then used to determine the information gain of specific terms by calculating a Divergence From Randomness score for individual terms.

Semantics. Structured knowledge in the form of (knowledge) graphs allow to identify complex semantic relations, such as concept similarity [21], along with the relational knowledge represented in traditional databases. Concept similarity can serve as an important feature for diversifying query results. Additionally, graphs can accommodate search by synonyms [22].

3. Approach

Our approach produces auto-complete suggestions optimized across three dimensions: relevancy, local diversity, and global diversity (or coverage). In order to produce a top-k list of results, the auto-complete suggestions are matched, ranked, and selected in a three-phased

[2] A collection of various source vocabularies such as ICD10, MeSH and SNOMED CT.

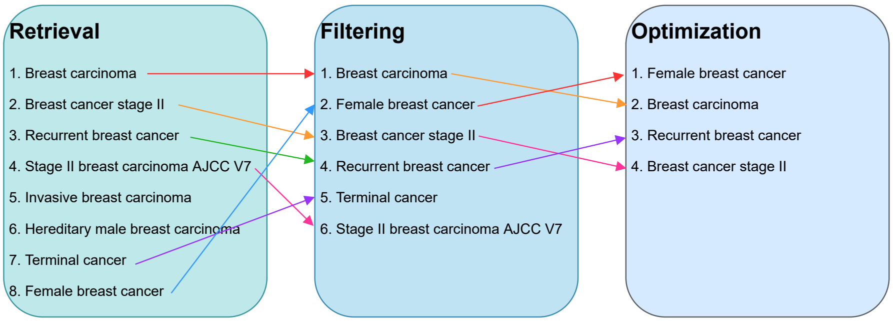

Retrieval
1. Breast carcinoma
2. Breast cancer stage II
3. Recurrent breast cancer
4. Stage II breast carcinoma AJCC V7
5. Invasive breast carcinoma
6. Hereditary male breast carcinoma
7. Terminal cancer
8. Female breast cancer
Filtering
1. Breast carcinoma
2. Female breast cancer
3. Breast cancer stage II
4. Recurrent breast cancer
5. Terminal cancer
6. Stage II breast carcinoma AJCC V7
Optimization
1. Female breast cancer
2. Breast carcinoma
3. Recurrent breast cancer
4. Breast cancer stage II

Figure 2: Auto-completion suggestions across phases.

process (Figure 1). First, in the retrieval phase (1), all disease terms that match the prefix are retrieved from the database. Subsequently, in the filtering phase (2), this set is reduced by taking the 20 most relevant items in order to limit the computational load. In the final optimization phase (3), the optimal subset, according to all three dimensions, is selected using an evolutionary algorithm.

Let us consider the user input: `Breast ca-`. In the retrieval phase, the retrieved keywords would include {`Breast`, `Cancer`, `Carcinoma`}, while the returned disease names are shown in Figure 2. Some disease terms (e.g. `Breast carcinoma`) are more likely to be searched for (i.e. more relevant) than others (e.g. `Hereditary male breast carcinoma`). During the filtering phase, less relevant terms would be ranked at a lower position, below the filtering threshold. Note that `Breast cancer stage II` and `Stage II breast carcinoma AJCC V7` are related terms in the UMLS taxonomy, the latter being a subtype (i.e. a 'child') of the former.

In the optimisation phase, the system identifies this and decide to return only `Breast cancer stage II`, as it implicitly covers its children, too. Consider now the case in which only higher level candidates such as `Breast carci- noma` and `Terminal cancer` are kept. The results would not provide diversity high enough, causing specific but desired disease terms (e.g. `Male breast cancer`) to be omitted. Our optimisation step aims to balance such considerations, resulting in a list of relevant, diverse, and high-coverage terms.

3.1. Data Preprocessing

In order to implement all dimensions in our algorithm and to enable fast retrieval of items, the graph data needs to be pre-processed. Features such as TF-IDF, trial, and paper counts are generated, normalized, and combined into a various scores. Then, keywords are created for every disease node (1) to ensure word-order independent matching of the prefix to a disease term, and (2) to calculate TF-IDF scores for each disease keyword.

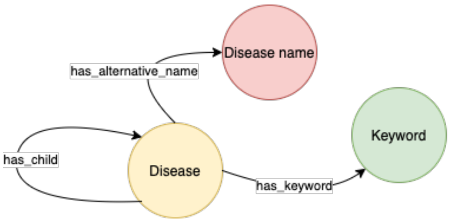


Figure 3: Knowledge graph schema.

Knowledge Graph. Our knowledge graph (stored both as a SQL and a Neo4J database) contains information about disease concepts derived from the different medical vocabularies provided by UMLS. A disease concept consists of a *preferred disease term* and its synonyms. The graph includes three types of nodes (see Figure 3). `Disease` nodes are connected through `has_child` relationships (as per UMLS standards), and `Disease_name` nodes are connected to their corresponding `Disease` nodes through `has_alternative_name` relationships. After the creation of the keywords, we link each `Keyword` node to their corresponding `Disease` nodes using the `has_keyword` relationship. Keywords are generated for each disease concept by tokenizing the preferred disease term and its synonyms. Tokens were normalized to American English.

Features. The features to implement different dimensions can be divided into individual features (i.e. related to individual disease concepts) and group features (i.e. an aggregated score of a group of disease concepts). We use [R], [D], and [C] to indicate whether they refer to relevancy, diversity, or coverage, respectively.

- *Clinical trial count (TC)* [R] refers to the total count of clinical trials related to a disease concept, standardized using a sigmoid transformation around the median. The NER system QuickUMLS[3] was used to detect disease concepts from the title, keywords and conditions sections of clinical trials, collected from the official repositories from the Unites States and Europe[4].
- *Paper count (PC)* [R] is computed per disease concept by processing title and abstract text from Pubmed articles and standardized using the sigmoid transformation.
- *TF-IDF* [R] was calculated for each `has_keyword` relationship using tf-idf = tfn $* \ log(N/\text{df})$, where *tfn* is the normalized term frequency, *N* the number of disease concepts, and *df* the document frequency of a keyword. The score is also standardized as above.
- *Children count (Ch)* [R] represents the number of child nodes connected to a disease node, divided by 100, where a limit is set to a maximum value of 1.
- *Graph-depth$_{median}$* [D] is computed per disease node as the median of the graph depths of its originating sources. Recall that our graph is an aggregate of multiple sources, so disease concepts can have multiple graph-depth values according to their taxonomy of origin.
- *Concept similarity* [D] is measured using the average graph distance, calculated by taking the average of pairwise shortest-path distances of a set of disease nodes, as in [21].
- *Covered items* [C] exploits the hierarchical relationships in the graph to compute the number of children of a returned item. Given a set of relevant items, we compute the amount of relevant non-returned items covered by the returned items.

The above features contribute to one of the following score variants (ranging [0,1]):

1. **Basic Relevance**[5] [R], which combines the *TC* and *PC* features as $\text{Basic}_R = \frac{PC_n+TC_n}{2}$.
2. **TF-IDF Relevance** [R], which combines the former two with the *TF-IDF* feature as $\textit{TF-IDF}_R = \frac{(PC_n+TC_n)*(TFIDF_n+1)}{4}$).

[3] https://github.com/Georgetown-IR-Lab/QuickUMLS

[4] https://clinicaltrials.gov/ and https://www.clinicaltrialsregister.eu/

[5] All aggregated relevance scores were determined by summing the scores while applying a discount for higher ranks: $relevance = \sum_{i=1}^{6} \frac{Rank_i}{1} * relevance_i$. The relevance scores of items below rank 6 were not counted as these are often not looked at by the user [6].

3. **Semantic Relevance** [R], combines all 4 relevance features as $Semantic_R = \frac{(Ch+1)*(PC_n+TC_n)*(TFIDF_n+1)}{8}$.
4. **Distance Diversity** [D], is calculated by taking the average of all pairwise distances, where distance is measured in number of edges, as in [21].
5. **Specificity Diversity** [D], through graph depth is rewarded by calculating the diversity by dividing the unique depth values by the total number of depth values: $\frac{depth_{unique}}{N}$, as in [21].
6. **Coverage** [C], We use the *Covered items* feature as the coverage score of a set of items by counting the number of items covered by the returned set, divided by 10.

3.2. Query Auto-Complete

Prefixes are matched to disease nodes through one or more keyword nodes (fig. 2). The retrieval algorithm matches the prefix with keywords, which are linked to disease nodes. In the filtering phase, this set is reduced to a smaller subset in order to limit the computational load. This is done by taking the 20 most relevant items.

In the optimization phase, a set of disease terms (from now on called items) is selected, such that they are (1) relevant, (2) diverse, and (3) cover-relevant. The problem is approached as a multi-objective optimization task. Evolutionary algorithms have shown to perform well at similar query recommendation problems when the search space is large, such as the selection of topical queries [23]. Therefore, a genetic algorithm is designed to optimize these three objective functions, which we combine in a fitness function by taking the mean of the normalized scores of each dimension. These three dimensions are thus set to have equal impact on the fitness score, in order to obtain a subset where these three dimensions are balanced. The gene pool consists of 20 candidate items, which are sorted by relevance to increase the selection probability of items indexed closer to 1. An array of integers containing these indices represents an individual. Variation operators consist of recombination by one-point crossover and mutation of individual genes (mutated gene value is decreased per 1). Termination criterion is fulfilled if no improvement occurs in at least 20 generations. Items are only selected if they contribute to the overall fitness of the set (i.e. relevance, diversity, coverage). As a result, the number of returned items is dynamic. Parameter tuning was performed to find the optimal values for population size, mating pool size, and mutation factor (cfr. Table 1).

Table 1: Hyperparameters of the Evolutionary Algorithm.

Representation	Integers
Recombination Probability	100%
Mutation	Gene value - 1
Mutation probability	10%
Parent Selection	Roulette wheel
Survival selection	Elitism
Population Size	50
Number of Offspring	100
Initialisation	Random, 50% full length

4. Experimental Evaluation

4.1. Experimental Set-Up

Experimental settings. We carried out experiments with historical query data to evaluate the efficiency and effectiveness of our algorithm. To investigate the role of different feature setups, we compare three variants of relevancy and two of diversity: ‘$Basic_R$ - $Distance_D$’, ‘$TF\text{-}IDF_R$ - $Distance_D$’, ‘$Semantic_R$- $Distance_D$’, ‘$Basic_R$ - $Specificity_D$’, ‘$TF\text{-}IDF_R$ - $Specificity_D$’, ‘$Semantic_R$ - $Specificity_D$’.

Baseline. As baseline, we take the current disease QAC employed in myTomorrows’s clinical trial search engine. This depends on three resources: (a) a mapping of n-gram prefixes to disease name tokens, (b) a mapping of disease name tokens to disease names, and (c) a concept relevance score per disease node in UMLS. The baseline consists of four phases:

1. **Individual string matching:** First, each n-gram in the user input (as separated by spaces) is matched to a list of tokens, which is associated to a list of disease name candidates. We reward user-input identified as a valid disease name tokens (e.g. `BRE`, standing for `Benign rolandic epilepsy`), and penalize them otherwise. The score is then $ind\_score = \frac{len(ngram)}{len(name)} * X$, where X is 1.05 for rewards and 0.7 for penalties.

2. **Aggregated string matching:** Since different input n-grams may lead to the same disease name candidate (i.e. `Breast` and `Canc` are both associated with `Breast Cancer`), an aggregation step is needed to provide with a unique list of disease name candidates. The aggregated score is calculated as $agg\_score = \log_{num(ngram)+1}(\sum(ind\_score) + 1)$. At this point, only candidate names scoring above a certain threshold (set at 0.05) are further processed.

3. **Semantic relevance:** The list of disease names is queried against UMLS to retrieve Concept Unique Identifiers (CUI) and their relevance. The specifics of the relevance metric fall beyond the scope of this paper, but they are similar to the Basic Relevance score (Section 3.1). To ensure unique concepts are shown to the user, we remove any duplicate concepts, while keeping the disease name with the highest string matching score.

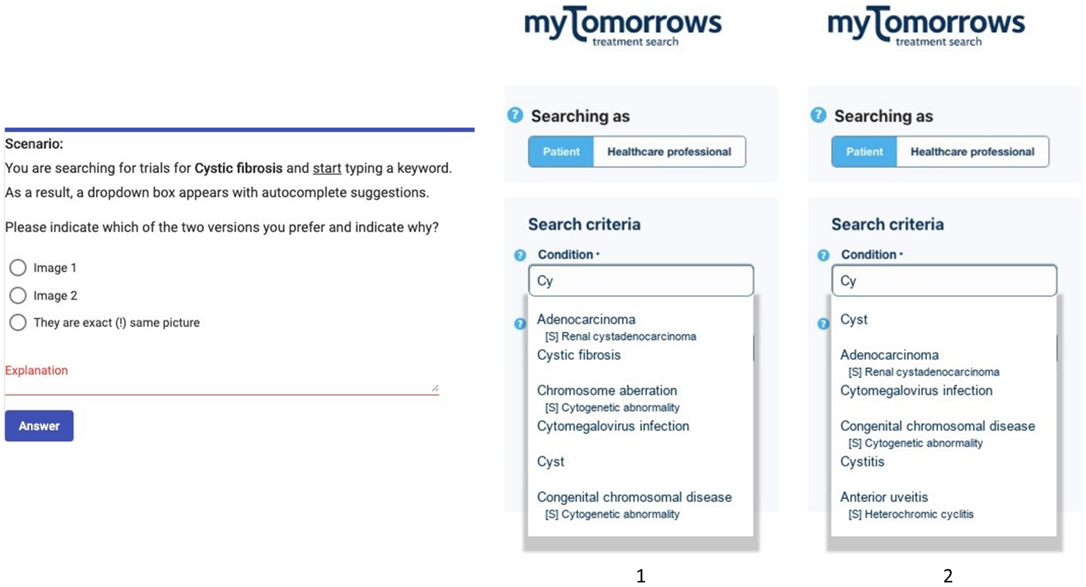
4. **Ranking:** The list of suggestions are ranked according to the average between the string match score and the concept’s relevance i.e. $score = \frac{agg\_score+relevance\_score}{2}$. Finally, the top-10 items are shown to the user[6].

Data. For the quantitative evaluation, myTomorrows provided the 500 most popular queries, consisting of anonymized searched and selected disease terms. The list contained 409 distinct terms. For the qualitative evaluation, the 18 most queried terms from our query log were used. For each query, the corresponding selected term was treated as the intended search term.

Evaluation. Via two quantitative experiments, we compare recall, precision and efficiency scores. In Experiment 1, efficiency was measured as Tokens Saved Rates (TSRs) by increasing the number of characters of the query entered into the QAC until the clicked term was included in the results. To evaluate how different methods ranked the intended item, Experiment 2 compares the item’s rank after input of different query lengths (2, 4, 6, 8, 10 characters). Results were compared by performing pairwise t-tests, with Bonferroni correction applied to accommodate for multiple testing.

Experiment 3 consisted of an offline user-based evaluation. Due to accessibility, target users (experts and non-experts in the medical domain) were simulated by myTomorrows employees

[6] Note that this QAC method was created through informal experimentation, and its behavior has not been thoroughly studied, hence motivating the current work.



Scenario:
You are searching for trials for **Cystic fibrosis** and start typing a keyword.
As a result, a dropdown box appears with autocomplete suggestions.

Please indicate which of the two versions you prefer and indicate why?

○ Image 1
○ Image 2
○ They are exact (!) same picture

Explanation

Answer

myTomorrows treatment search
Searching as
Patient
Healthcare professional
Search criteria
Condition *
Cy
Adenocarcinoma
[S] Renal cystadenocarcinoma
Cystic fibrosis
Chromosome aberration
[S] Cytogenetic abnormality
Cytomegalovirus infection
Cyst
Congenital chromosomal disease
[S] Cytogenetic abnormality
1
myTomorrows treatment search
Searching as
Patient
Healthcare professional
Search criteria
Condition *
Cy
Cyst
Adenocarcinoma
[S] Renal cystadenocarcinoma
Cytomegalovirus infection
Congenital chromosomal disease
[S] Cytogenetic abnormality
Cystitis
Anterior uveitis
[S] Heterochromic cyclitis
2

Figure 4: Example view of user evaluation task.

with a similar split in profiles. Each participant performed 10 tasks per round, and on average completed 22.5 comparison tasks per person. In each task, participants were shown two images with auto-complete results (Figure 4), and were asked to (1) indicate their method preference, and (2) briefly motivate their decision. Forcing participants to make a choice between two shown lists allowed us to make preferences more explicit, and method differences more detectable, similarly to how pairwise preference elicitation works for recommender systems [24].

For each query, a prefix was constructed varying the lengths between queries. In the first 6 conditions, each method is compared to the baseline. Additionally, to evaluate the differences within our methods, another 9 combinations were compared: 6 to compare three relevance scores within each diversity score and 3 to measure the effect of each diversity score within each relevance method.

4.2. Experimental Results

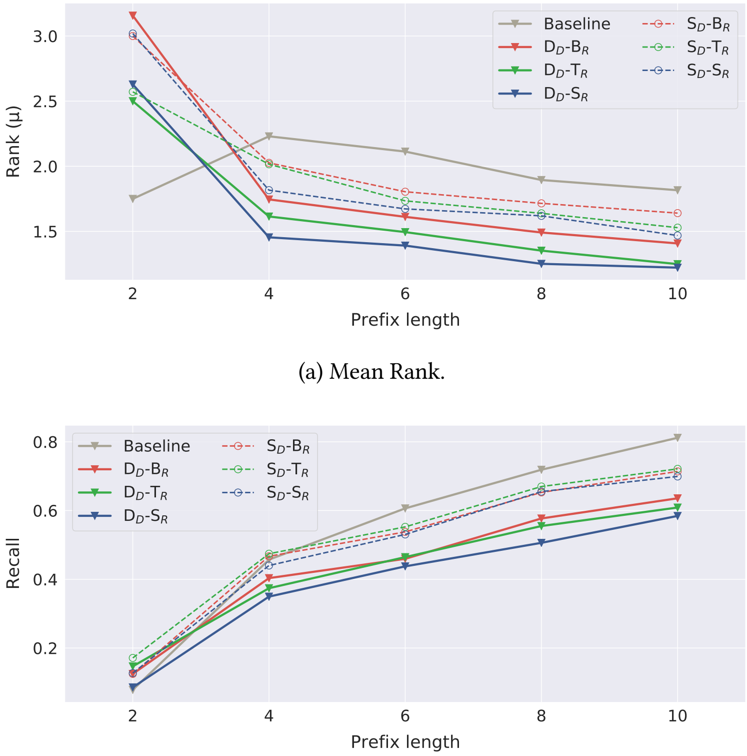
Experiment 1. Table 2 shows the results for Experiment 1. We compare only terms that were returned by all methods (n=354). On average, the length of each query was 15.58 characters (including whitespaces). TSR results show that *Distance*$_D$-*TF-IDF*$_R$ performs similarly to the baseline, while other methods are showing a significantly lower TSR. Regarding the rank of items at the

Table 2: Results of Experiment 1.

	TSR		Rank	Cov.	Terms	NRCI
	%	*tok.*				
Baseline	60%	9.8	2.663	6.44	9.86	0/2
Dist$_D$						
Bas$_R$	52%	8.5^{***}	2.339	6.44	4.16^{***}	19/45
TF-IDF$_R$	52%	8.4^{***}	2.006^{***}	3.93	3.50^{***}	18/20
Sem$_R$	49%	8.0^{***}	1.785^{***}	3.77	2.83^{***}	22/42
Spec$_D$						
Bas$_R$	55%	9.1^{**}	2.598	4.11	5.60^{***}	6/20
TF-IDF$_R$	56%	9.2^{*}	2.470	3.84	5.49^{***}	8/19
Sem$_R$	54%	8.9^{***}	2.499	3.77	5.18^{***}	11/23

* $p < 0.05$, ** $p < 0.01$, $^{***}p < 0.001$, moment they were returned, the *TF-IDF$_R$* and *Semantic$_R$-Distance$_D$* methods show a significantly higher rank than the baseline, while for the *Basic$_R$* and the three *Specificity$_D$* methods no significant difference was found. Additionally, we studied the moment that the intended items were covered by another item for the first time (i.e. the intended item is a child of an item from the results). A complementary effect is found: where TSR improves, more keystrokes are required before an item is covered. All our methods improved in this aspect compared to the baseline, with *Semantic$_R$* requiring the least keystrokes. The mean number of suggestions returned by each method are shown in the fourth column (Terms). All methods show a significant decrease in this aspect compared to the baseline. We also found that although some of the items were not found, they were covered by other items that were returned. The amounts of non-returned but covered items (NRCI) are shown in the last column of Table 2, where the last number indicates the amount of non-returned items.

Experiment 2. While Experiment 1 looked at if and when items get returned, Experiment 2 focuses on the rank at which relevant items get returned over various prefix lengths. Our methods improve on the baseline in terms of ranking, however, not on recall (Figure 5). Additionally, complementary effects are shown on ranking and recall as the baseline initially ranks items high in the list, while starting with a relatively low recall, but these both reverse later on. Overall, *Distance$_D$* shows to consistently have the lowest average rank from length 4 and higher, whereas it seemingly counterpart *Specificity$_D$* shows to have the highest recall.



(a) Mean Rank.

(b) Recall Rate.

Figure 5: Results of the quantitative evaluation of Experiment 2.

Experiment 3. In this experiment, participants were asked to indicate their preferences. Each method received a score of either 1 or 0 per query, based on the majority of votes. Comparing the baseline to our methods, for a majority of the queries the baseline was preferred by users (ranging from 60% to 100% of the queries). In terms of relevance, user preferences are as follows: *TF-IDF$_R$* is the most preferred, followed by *Semantic$_R$*, and finally *Basic$_R$* (Fig-

Table 3

Experiment 3. User agreement is calculated using Fleiss k. Names are shortened for readability.

D	$Dist_D$ (k=.57)		$Dist_D$ (k=.48)		$Dist_D$ (k=.44)	
R	$TF\text{-}IDF_R$	Bas_R	Sem_R	Bas_R	$TF\text{-}IDF_R$	Sem_R
	60%	40%	50%	50%	56%	44%

D	$Spec_D$ (k=.35)		$Spec_D$ (k=.49)		$Spec_D$ (k=.38)	
R	$TF\text{-}IDF_R$	Bas_R	Sem_R	Bas_R	$TF\text{-}IDF_R$	Sem_R
	82%	18%	67%	33%	57%	43%

R	Bas_R (k=.21)		$TF\text{-}IDF_R$ (k=.19)		Sem_R (k=.55)	
D	$Dist_D$	$Spec_D$	$Dist_D$	$Spec_D$	$Dist_D$	$Spec_D$
	62%	38%	62%	38%	44%	56%

ure 3, top two tables). In terms of diversity settings, users seemed to mostly prefer *Distance*$_D$ over *Specificity*$_D$, but we note here that the inter-user agreement is low for both conditions. *Specificity*$_D$ is instead preferred when combined with *Semantic*$_R$ (Table 3, bottom table).

5. Discussion and Lesson Learnt

Baseline vs. Proposed Method. Our results show that our newly introduced methods are precision-oriented, and, when returning the relevant item, they also rank correct results higher than the baseline; while the baseline is recall-oriented, and tends to return longer result lists. Our initial assumptions were that users want to be pointed to a particular item quickly, as opposed to being shown a longer set of alternatives from which it may take more time to choose the intended item. Contrary to this, our users found the baseline preferable to any of our methods. This shows that QAC users in our use-case may be heavily recall and ranking-focused, and they do not prefer short, focused lists of suggestions, as opposed to what findings for web search indicate [6].

There may be several reasons for users preferring the baseline. Primarily, most of our users have already been exposed to the baseline auto-complete method, and they may have expressed their preference towards something familiar to them. To confirm this, we are planning experiments with users who see the search system, and any auto-complete method we want to evaluate, for the first time. Secondly, user feedback is highly dependent on the user experience design, and since our methods provide additional information to the user compared to the "plain" baseline, this angle should also be considered when comparing various methods.

Beyond this, using evaluation data from logs where the baseline method was in operation could be a source of bias (e.g. for recall), and hence our findings are potentially more insightful when comparing our methods to one another.

Diversity Given a set of 20 most relevant items, *Specificity*$_D$ will possibly select more items than *Distance*$_D$. On one hand, those items tend to be closely located in the graph - thus, not diverse according to *Distance*$_D$; on the other hand, they show high variance in graph depth - therefore, high diversity according to *Distance*$_D$. If a user profile requires (a) only a few keystrokes before having the intended term suggested, and (b) there is a preference towards more suggestions, then *Specificity*$_D$ would be the most adequate to use in the QAC system (see

Table 2). For example, non-experts may benefit from this method, since spelling is often an obstacle for them [16], and it could be important that slight spelling variations between different concepts are brought to their awareness by showing more suggestions. However, since Experiment 3 did not show convincing preferences for either *Diversity*$_D$ method, further experiments are needed to confirm this. Experiment 2 showed that *Distance*$_D$ overall returns relatively higher-ranked items in more concise lists. Therefore, user profiles that require fast typing and quick investigating of suggestions would most likely gain more from using *Distance*$_D$.

Relevancy Experiment 1 showed that relevance variations have the most impact on how quickly an item is covered as users type. From *Basic*$_R$ to *TF-IDF*$_R$ and *Semantic*$_R$ the ability to cover items after a few keystrokes shows to increase. Given that TSR and the number of returned suggestions both decrease between these settings, suggestions seem to be more abstract for *TF-IDF*$_R$ and, even more, for *Semantic*$_R$ when compared to both *Basic*$_R$ and the baseline. In Experiment 3, we found that users preferred *TF-IDF*$_R$ over both *Basic*$_R$ and *Semantic*$_R$, which could indicate that they prefer the level of specificity of items selected by *TF-IDF*$_R$ (i.e.more specific than *Semantic*$_R$, but more abstract than *Basic*$_R$). As mentioned before, laypeople might benefit from additional support in concept disambiguation. Therefore, given a user profile where there is a need for awareness of differences between subtypes to be inspected with care, grouped items should be suggested first, while further refinements could be provided after the query is submitted. This type of behaviour may be achieved through either *Semantic*$_R$ or *TF-IDF*$_R$, with an observed user-preference for *TF-IDF*$_R$.

6. Conclusion and Future Work

Often, complex problems cannot have a 'one size fits all' solution. In the context of Query Auto-Completion for medical search, we have found that no one solution fits all users' needs. However, we showed that recommendations could be learned for different user profiles, such as, but not limited to, medical experts (i.e. healthcare providers, pharmaceutical professionals) and laypeople (i.e. patients). We have proposed and investigated a graph-based method that aimed to outperform a currently implemented QAC system. Our method has shown to achieve this in terms of ranking and covering items. We experienced many benefits of using a graph over a traditional database, such as, handling complex queries more time-efficiently and easing the process of tracing descendants while calculating graph distance. Future work will focus on improving recall as well. Furthermore, we will extend our user-based evaluation to assess all returned items' relevance. Ultimately, our work aspires to be used as a step towards accommodating both laypeople and experts and improving the accessibility of health information[7].

References

[1] Z. Bar-Yossef, N. Kraus, Context-sensitive query auto-completion, in: Proceedings of the 20th international conference on World wide web - WWW '11, ACM Press, Hyderabad, India, 2011, p. 107. doi:10.1145/1963405.1963424.

[7] Code and sample dataset are publicly available at https://research.mytomorrows.com/

[2] F. Su, M. Somaiya, S. Mishra, R. Mukherjee, EXOS: Expansion on session for enhancing effectiveness of query auto-completion, in: 2015 IEEE International Conference on Big Data (Big Data), 2015, pp. 1154–1163. doi:10.1109/BigData.2015.7363869.

[3] S. Whiting, J. M. Jose, Recent and robust query auto-completion, in: Proceedings of the 23rd international conference on World wide web - WWW '14, ACM Press, Seoul, Korea, 2014, pp. 971–982. doi:10.1145/2566486.2568009.

[4] B. Rieder, G. Sire, Conflicts of interest and incentives to bias: A microeconomic critique of Google's tangled position on the Web, New Media & Society 16 (2014) 195–211. doi:10.1177/1461444813481195.

[5] Z. Szlávik, W. Kowalczyk, M. Schut, Diversity measurement of recommender systems under different user choice models, in: Fifth International AAAI Conference on Weblogs and Social Media, 2011.

[6] E. Cutrell, Z. Guan, What are you looking for? an eye-tracking study of information usage in web search, in: Proceedings of the SIGCHI conference on Human factors in computing systems, 2007, pp. 407–416. doi:10.1145/1240624.1240690.

[7] A. Rotegård, L. Slaughter, C. Ruland, Mapping nurses' natural language to oncology patients' symptom expressions, Studies in health technology and informatics 122 (2006) 987–8.

[8] M. Dahm, Coming to terms with medical terms – exploring insights from native and non-native english speakers in patient-physician communication, HERMES - Journal of Language and Communication in Business 25 (2017) 79–98. doi:10.7146/hjlcb.v25i49.97739.

[9] J. Carbonell, J. Goldstein, The use of MMR, diversity-based reranking for reordering documents and producing summaries, in: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '98, ACM Press, Melbourne, Australia, 1998, pp. 335–336. doi:10.1145/290941.291025.

[10] A. Alhindi, U. Kruschwitz, C. Fox, M.-D. Albakour, Profile-based summarisation for web site navigation, ACM Transactions on Information Systems (TOIS) 33 (2015) 1–39. doi:10.1145/2699661.

[11] J. Yan, W. Chu, R. W. White, Cohort modeling for enhanced personalized search, in: Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval, 2014, pp. 505–514. doi:10.1145/2600428.2609617.

[12] J. List, The name of the game: Information seeking in a professional context, in: Proceedings of the Integrating IR Technologies for Professional Search Workshop, Moscow, Russia (March 24, 2013), 2013.

[13] S. Verberne, J. He, G. Wiggers, T. Russell-Rose, U. Kruschwitz, A. P. de Vries, Information search in a professional context-exploring a collection of professional search tasks, arXiv preprint arXiv:1905.04577 abs/1905.04577 (2019). arXiv:1905.04577.

[14] R. W. White, S. T. Dumais, J. Teevan, Characterizing the influence of domain expertise on web search behavior, in: Proceedings of the Second ACM International Conference on Web Search and Data Mining - WSDM '09, ACM Press, Barcelona, Spain, 2009, p. 132. doi:10.1145/1498759.1498819.

[15] J. Palotti, A. Hanbury, H. Müller, C. E. Kahn, How users search and what they search for in the medical domain, Inf Retrieval J 19 (2016) 189–224. doi:10.1007/

s10791-015-9269-8.

[16] P. C.-I. Pang, K. Verspoor, S. Chang, J. Pearce, Conceptualising health information seeking behaviours and exploratory search: result of a qualitative study, Health Technol. 5 (2015) 45–55. doi:10.1007/s12553-015-0096-0.

[17] P. Castells, N. J. Hurley, S. Vargas, Novelty and diversity in recommender systems, in: Recommender systems handbook, Springer, 2015, pp. 881–918. doi:10.1007/978-1-4899-7637-6_26.

[18] Z. Huang, B. Cautis, R. Cheng, Y. Zheng, N. Mamoulis, J. Yan, Entity-Based Query Recommendation for Long-Tail Queries, ACM Transactions on Knowledge Discovery from Data 12 (2018) 1–24. doi:10.1145/3233186.

[19] M. Zhong, H. Cheng, Y. Wang, Y. Zhu, T. Qian, J. Li, Towards both Local and Global Query Result Diversification, in: G. Li, J. Yang, J. Gama, J. Natwichai, Y. Tong (Eds.), Database Systems for Advanced Applications, volume 11447, Springer International Publishing, Cham, 2019, pp. 464–481. doi:10.1007/978-3-030-18579-4_28.

[20] T. Groza, K. Verspoor, Assessing the Impact of Case Sensitivity and Term Information Gain on Biomedical Concept Recognition, PLOS ONE 10 (2015) e0119091. doi:10.1371/journal.pone.0119091.

[21] B. Sathiya, T. V. Geetha, A review on semantic similarity measures for ontology, Journal of Intelligent & Fuzzy Systems 36 (2019) 3045–3059. doi:10.3233/JIFS-18120.

[22] A. Jaech, M. Ostendorf, Personalized language model for query auto-completion, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 700–705. doi:10.18653/v1/P18-2111.

[23] R. L. Cecchini, C. M. Lorenzetti, A. G. Maguitman, I. Ponzoni, Topic relevance and diversity in information retrieval from large datasets: A multi-objective evolutionary algorithm approach, Applied Soft Computing 69 (2018) 749 – 770. doi:10.1016/j.asoc.2017.11.016.

[24] S. Kalloori, F. Ricci, R. Gennari, Eliciting pairwise preferences in recommender systems, in: Proceedings of the 12th ACM Conference on Recommender Systems, RecSys '18, Association for Computing Machinery, New York, NY, USA, 2018, p. 329–337. doi:10.1145/3240323.3240364.