# A Living Lab Study of Query Amendment in Job Search

Bahar Salehi
The University of Melbourne
Melbourne, Australia

Damiano Spina
RMIT University
Melbourne, Australia

Alistair Moffat
The University of Melbourne
Melbourne, Australia

Sargol Sadeghi
SEEK Ltd.
Melbourne, Australia

Falk Scholer
RMIT University
Melbourne, Australia

Timothy Baldwin
The University of Melbourne
Melbourne, Australia

Lawrence Cavedon
RMIT University
Melbourne, Australia

Mark Sanderson
RMIT University
Melbourne, Australia

Wilson Wong
SEEK Ltd.
Melbourne, Australia

Justin Zobel
The University of Melbourne
Melbourne, Australia

## ABSTRACT

Errors in formulation of queries made by users can lead to poor search results pages. We performed a living lab study using online A/B testing to measure the degree of improvement achieved with a query amendment technique when applied to a commercial job search engine. Of particular interest in this case study is a clear "success" signal, namely, the number of job applications lodged by a user as a result of querying the service. A set of 276 queries was identified for amendment in four different categories through the use of word embeddings, with large gains in conversion rates being attained in all four of those categories. Our analysis of query reformulations also provides a better understanding of user satisfaction in the case of problematic queries (ones with fewer results than fill a single page) by observing that users tend to reformulate rewritten queries less.

## CCS CONCEPTS

• **Information systems → Information retrieval query processing**; **Users and interactive retrieval**;

**ACM Reference Format:**
Bahar Salehi, Damiano Spina, Alistair Moffat, Sargol Sadeghi, Falk Scholer, Timothy Baldwin, Lawrence Cavedon, Mark Sanderson, Wilson Wong, and Justin Zobel. 2018. A Living Lab Study of Query Amendment in Job Search. In *SIGIR '18: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, July 8–12, 2018, Ann Arbor, MI, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3209978.3210082

## 1 INTRODUCTION

Many job seekers use special-purpose online search systems to find employment opportunities. Some of these systems not only gather job advertisements, but also provide a service that enables job seekers to immediately lodge applications. Despite the popularity of such job search engines, there has been relatively little research on job search. Saha and Arya [10] propose a personalized job-search learn-to-rank method, and Spina et al. [12] suggest that job search (on the part of job seekers) and talent search (on the part of employers/head hunters) have different aims to standard web or enterprise information search. One common issue with web and enterprise search is mismatch between query and document collection, due to, for example, spelling errors or vocabulary differences.

The way in which users amend (reformulate) their queries while searching on the web [2], together with techniques to categorize such amendments [7], has been much studied. Means of automatically aiding query amendment have also been extensively examined, including query spelling correction [5] and expansion [4]. In this study we have investigated the impact of query amendment on a commercial job search engine. Analysing user logs and the reformulations they contain, we identified a set of 276 common queries that had small or empty answer sets, and for which we had high confidence that amendment would not confuse the users (including, for example, a set of spelling corrections). For each we selected a single reformulation that captured the intent of the original query and was likely to match a greater number of position descriptions.

Evaluation of amendment algorithms typically employs offline methodologies, such as inferring user behavior from historical query logs [3, 5], or, more commonly, measuring effectiveness on conventional test collections [4]. Online evaluation – the "evaluation of a fully functioning system based on implicit measurement of real users' experiences" [6] – provides another option. Our case study is tested using a *living lab* methodology based on A/B testing. With live deployment of the proposed approach and a significant user base, we are able to judge the extent to which the amendment benefits users.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

*SIGIR '18, July 8–12, 2018, Ann Arbor, MI, USA*

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-5657-2/18/07…$15.00
https://doi.org/10.1145/3209978.3210082

In online evaluation, impact is commonly measured using clicks on search results, and via *conversions*, moments at which the user takes some form of action (for example, purchasing an item), implying satisfaction with the result [13]. In job search, a conversion signal arises if the user lodges applications for one or more of the identified jobs. In contrast to product search, for which conversion is typically a single purchase, job seekers may lodge multiple applications in a session spanning one or many searches. Applications in job search are also more effective than clicks in training learn-to-rank algorithms with weak supervision [10]. In our case study we make use of clicks, applications, and subsequent query formulations. Query reformulations have been effectively used as a mechanism to predict user satisfaction in web search [1, 11]. We analyze query reformulations in the context of problematic queries, ones with fewer results than would fill a single page.

By identifying common misformulations, we found that amendment can lead to dramatic increases in conversions. On average, the reformulations led to a quadrupling of the number of clicks on job advertisements in that subset of the queries, and to an approximately six-fold increase in the number of applications lodged. We also noted a substantial reduction in the number of subsequent reformulations in search sessions. Underpinning our work, we ask two research questions: (i) *Does query amendment improve user outcomes in job search in cases for which only a few results are returned by the original query?*; and, if so, (ii) *Are those improvements reflected by changes in subsequent query reformulations?*

## 2 METHODOLOGY

We now explain our technique for semi-automatically identifying a set of problematic queries, and coupling each with a corresponding amended query.

**Candidate query pair generation.** Candidate pairs of queries that were manually re-formulated by users were mined from a query log (containing ≈20M sessions) of SEEK (`seek.com.au`), a major job search engine.[1] The log was treated as a contiguous sequence of queries within a session (defined as all system interactions by a given user in a single day), and we extracted the candidate pairs based on the following requirements:

- Result imbalance: the first query returned $\leq 20$ results (20 is the size of a search results page) and the second query returned more results than would fit on a single page, i.e. $> 20$.
- Sufficient user support: the first query was observed among the 10,000 most common queries across user sessions in the log.
- Semantic similarity: the two queries in each pair were semantically similar.

The first of these requirements was applied as a constraint, and then the set of matching query pairs was sorted in descending order of semantic similarity, and in the case of ties, sub-sorted in descending order of user support.

To measure semantic similarity, we pre-trained word embeddings over a corpus of job ads containing a total of 422 million tokens (unigrams). These provide a mechanism for identifying terms that are likely to be similar based on their usage. Specifically, we used the CBOW variant of word2vec [9] and generated a 100-dimensional representation, leaving all other hyperparameters as the default.

For a given query, we looked up the set of word embeddings for the component terms (ignoring any terms with low frequency in our job ad corpus, which lack pre-trained word embeddings), and use the embeddings to calculate the Word Mover's Distance ("WMD") [8] between the two sets. WMD is an instance of Earth Mover's Distance, and is based on the minimum cumulative distance that each of the terms in the first query needs to "travel" in the embedding space to map onto one of the terms in the second query. We also compared the ranking of reformulations generated by WMD to a frequency-based ranking, and we found a low correlation. Given that our experiments involve real users of a commercial system, the intrinsic evaluation of our semantic similarity approach is out of the scope of the experiments presented in this work.

A total of 317 query pairs were then considered from the top of the decreasing WMD-ordered list. We manually inspected each of those pairs, and removed instances of query generalization (for example, `spanish speaker` → `spanish`) and pairings of companies in similar sectors. We kept only a small subset of generalizations in our experiments, given that this type of reformulation is more likely to confuse users if they are not explicitly informed of the amendment.[2] This process resulted in the removal of 41 queries, and a final set of 276 query pairs, made up of misspellings (for example, `apprentis` → `apprentice`), multiword variants (for example, `fire fighter` → `firefighter`), synonyms (for example, `governess` → `nanny`), and generalizations.

The final set of query amendments contained 171 *spelling corrections*; 49 *multiwords with spaces added or removed*; 46 *synonyms*; and 10 *generalizations*. While a larger set of query pairs could have been generated, we opted to be conservative during this first instantiation of automatic query correction within a live system.

**A/B testing.** Users in the test stream were assigned to one of two classes based on a hash derived from their unique identifiers, and had all of their subsequent queries processed by one of the following two job search mechanisms:

- *Champion*: One of the current ranking regimes used in the company's job search engine; and
- *Challenger*: A system which performs the corresponding query amendments if any of the 276 trigger queries were observed, and on all other queries acts exactly the same as *Champion*.

The goal was to have approximately half of the users in each stream, so that outcomes could be compared in a controlled setting.

**Post-query actions.** The recorded post-query actions and query reformulation options that we analyze are described in Table 1.

On the one hand, the "gain", or increase of user's success, can be measured by recording the number of hits, clicks and applications obtained using *Champion* and *Challenger*. On the other hand, problematic queries are less likely to be useful to users; hence, if such a query is not automatically amended, users will need to amend it manually, causing more reformulations "cost".[3] A successful query

[1] No personally identifiable information was made available or used in these experiments, and this investigation was compliant with RMIT University's Research Ethics Approval process.

[2] This manual validation process would not be needed if users were informed of the amendment in the interface.

[3] For instance, users tend to reformulate their queries as text queries when initial voice queries are misrecognized by automatic speech recognizers [11].

**Table 1:** Interactions observed after an original (problematic) query from the query set is submitted.

| Measurement | Description |
|---|---|
| *Post-query actions* | |
| *#Hits* | Number of job advertisements retrieved by the query. |
| *#Clicks* | Number of received clicks to view the details of the job advertisements in the retrieved set. |
| *#Apps* | Number of job applications started. |
| *Query reformulation options* | |
| *Repeat* | Repeat exactly the same search query. |
| *Repeat kw* | Repeat the keyword query but change other search filtering criteria available in the interface such as location or work type. |
| *Amendment* | The manually reformulated query corresponds to the amendment proposed by our solution, i.e., the most similar in the word embedding space. |
| *None* | The query is the last in the session, and there is no reformulation. |
| *Other* | The reformulated query is none of the above. |

**Table 2:** Query set and observations obtained from the online experiment; mean and standard deviation in parentheses.

| Category | *# Queries* | *# Obs. Champion* | *# Obs. Challenger* |
|---|---|---|---|
| Misspell. | 171 | 3,134 (18.33 ± 21.22) | 3,037 (17.76 ± 18.13) |
| Multiwd. | 49 | 1,700 (34.69 ± 63.55) | 1,638 (33.43 ± 54.75) |
| Synonyms | 46 | 1,814 (39.43 ± 32.42) | 1,830 (39.78 ± 31.71) |
| Generaliz. | 10 | 346 (34.60 ± 40.88) | 357 (35.70 ± 36.72) |
| All | 276 | 6,994 (25.34 ± 35.95) | 6,862 (24.86 ± 31.95) |

amendment should be able to increase the gain while decreasing the users' cost.

## 3 RESULTS AND ANALYSIS

The number of query observations for the *Champion* and *Challenger* systems is shown in Table 2, with the observations evenly distributed (50.5% versus 49.5%, respectively). The majority of the observations (more than 99%) come from unique users.

Based on these observations, we analyze differences between *Champion* and *Challenger* in terms of clicks and job applications (*gain*) and also in terms of changes in subsequent query reformulations to analyze potential changes in users' effort (*cost*).

**Clicks and Applications.** Table 3 reports the performance of *Challenger* in terms of relative improvement against *Champion.* The two columns labeled "Overall change" report the relative growth of *Clicks* and *Apps* in aggregate over all instances of all of the queries, i.e., as a micro-averaged improvement factor across all *Champion* and *Challenger* observations.

The results show substantial improvement over all target queries during the test period, with 5.9 times more applications (i.e., conversions) resulting overall when the amendments are applied compared to when they are not made, and with increases across the majority of the 276 queries. In particular, the increase in applications for synonyms is substantially higher than the other three categories (8 times more applications than in *Champion*). The improvement in synonyms may indicate that we are not only fixing queries, but also providing users with novel results that they may not have retrieved previously, such as job ads retrieved by the queries `linesman` and `linesperson`, which refer to the same job type. An overall relative improvement in terms of clicks is also observed (4.1 times more), but to a lesser extent than applications, suggesting that these two signals are not surrogates for each other.

The four columns headed "Per-query changes" reflect averages computed on a per-query basis, with the three columns marked $\delta^+(\cdot)$ indicating the number of queries for which the corresponding average measurement was numerically greater for *Challenger* than it was for *Champion*; that is, these three columns reflect macro-averaged improvement counts. The per-query results show that this improvement is consistent across the query set, and the number of queries improved in both clicks and applications is statistically significant, based on the sign test. Surprisingly, improvements are also observed for generalizations, although there is no statistical significance in terms of clicks.

One may wonder if these improvements are only reflected *locally* with actions after submitting the original query. However, the user may achieve the same success in *Champion* by manually reformulating the query (identifying the misspelling and manually correcting it). Therefore, we compared the changes in clicks and application within daily sessions, considering all actions performed by a user on the same day. Overall, *Challenger* obtains 1.1 times more clicks and 1.2 times applications than *Champion.* In the case of synonyms, improvements in terms of applications goes up to 1.9 times. These results suggest that our intervention also increases user success levels at the session level.

**Subsequent Query Reformulations.** The previous results measure *gains* from query amendments in terms of clicks and applications. A complementary measurement is to observe the *cost* of interactions to users. The number of query reformulations can be used as a proxy for cost, and we would expect fewer reformulations if the amendment is successful.

Table 4 provides an analysis of user actions in the next query issued after a target query is processed, with a notably different distribution occurring after the amended queries. To the best of our knowledge, no previous work combines the measurement of gain in terms of conversions, and cost in terms of query reformulations, in online experiments.

The table shows that the four different query categories follow the same tendency. Not surprisingly, the *Generalization* category does not result in substantial differences between *Champion* and *Challenger* in terms of query reformulations.

The increase of *Repeat* and *Repeat kw* actions in *Challenger* shows that users submit the original query more often as a follow-up action. This may be a positive signal, especially for *Repeat kw*, given that users in *Challenger* may obtain satisfactory results when submitting the problematic query – which is automatically amended – without needing to reformulate.

**Table 3:** Relative improvement of *Challenger* against *Champion* resulting from query amendment. In the last four columns, the superscript annotations indicate statistical outcomes from a two-sided sign test, with $^{\triangle}$ and $^{\blacktriangle}$ reflecting significance at $p < 0.05$ and $p < 0.01$, respectively.

| Category | Overall change | | Per query changes | | | |
|---|---|---|---|---|---|---|
| | *rClicks* | *rApps* | *n* | $\delta^{+}(Hits)$ | $\delta^{+}(Clicks)$ | $\delta^{+}(Apps)$ |
| Misspellings | 8.2× | 5.5× | 171 | $161^{\blacktriangle}$ | $154^{\blacktriangle}$ | $96^{\blacktriangle}$ |
| Multiwords | 2.7× | 5.4× | 49 | $41^{\blacktriangle}$ | $44^{\blacktriangle}$ | $28^{\blacktriangle}$ |
| Synonyms | 2.4× | 8.0× | 46 | $35^{\blacktriangle}$ | $34^{\blacktriangle}$ | $18^{\triangle}$ |
| Generalizations | 1.5× | 2.7× | 10 | 6 | 7 | $6^{\triangle}$ |
| All | 4.1× | 5.9× | 276 | $243^{\blacktriangle}$ | $239^{\blacktriangle}$ | $148^{\blacktriangle}$ |

**Table 4:** Distribution of query reformulations following original queries for *Champion* and *Challenger*, expressed as percentages summing to 100% across each row.

| Category | | *Repeat* | *Repeat kw* | *Amend-ment* | *None* | *Other* |
|---|---|---|---|---|---|---|
| Misspellings | *Chmp.* | 0.5% | 16.4% | 28.9% | 15.6% | 38.6% |
| | *Chall.* | 1.9% | 21.8% | 11.5% | 33.7% | 31.1% |
| Multiwords | *Chmp.* | 0.8% | 14.6% | 11.1% | 31.5% | 42.0% |
| | *Chall.* | 1.1% | 19.9% | 4.8% | 38.0% | 36.2% |
| Synonyms | *Chmp.* | 0.7% | 12.2% | 8.0% | 33.9% | 45.2% |
| | *Chall.* | 1.0% | 16.3% | 4.9% | 39.5% | 38.3% |
| Generaliz. | *Chmp.* | 2.0% | 6.9% | 11.3% | 56.4% | 23.4% |
| | *Chall.* | 2.2% | 10.7% | 7.6% | 60.2% | 19.3% |
| All | *Chmp.* | 0.7% | 14.3% | 18.3% | 26.3% | 40.4% |
| | *Chall.* | 1.5% | 19.3% | 7.9% | 37.6% | 33.7% |

We qualitatively analyzed a set of 20 randomly selected *Repeat* reformulations from both *Challenger* and *Champion*. We found that half of the cases are due to the generation of an extra (identical) record in the database when users log in, and do not correspond to explicit reformulation actions of the users. The other half however correspond to a *refresh* user behavior: users are coming back and clicking on the search button after more than 30 minutes. We also looked at a small random sample of *Repeat kw* reformulations and found that in *Champion* users tend to *relax* their search criteria by removing filters, probably in the hopes of obtaining more search results. A further analysis of how filters are changed is needed in order to better understand how users specify or generalize their information requests.

The decrease of the *Amendment* category is consistent with the intuition that users do not need to “fix” the original query manually. Similarly, the increase of the *None* group suggests a reduction in cost; that is, users reformulate less which, combined with the increase of gain, is a positive indication; and the decrease of *Other* also indicates less query reformulation within a session.

We also analyzed the percentage of abandonment (*None*) where users applied for a job before they stopped. For *Challenger*, users applied for a job before they abandon 7.9% of the time, 2.7 times more than for *Champion* (2.9%). This increase suggests that when queries are automatically amended, users are satisfied earlier with more applications for the original query and less reformulation.

## 4 CONCLUSIONS

We have performed a living lab experiment to explore the impact of pre-determined amendments to queries in a job search engine. Our experiments demonstrated large improvements in performance, with a more than five-fold boost in conversions. Perhaps surprisingly, the number of conversions rose more than the number of clicks, demonstrating that these two signals are not surrogates for each other in job search. An analysis of subsequent query reformulations has shown that users reformulate less when queries are automatically amended, suggesting that the increase in conversions is followed by a reduction in user costs. Significant improvements were observed not only in query correction but also when the proposed amendment consisted of semantically related queries such as synonyms. These gains are unsurprising when amending misspellings and multi-word expressions, but less obvious in the case of synonyms and generalizations. Moreover, our proposed evaluation methodology showed a substantial reduction in the number of subsequent reformulations.

**Acknowledgment.** This research was partially supported by Australian Research Council Project LP150100252 and SEEK Ltd.

## REFERENCES

[1] A. H. Awadallah, X. Shi, N. Craswell, and B. Ramsey. 2013. Beyond clicks: Query reformulation as a predictor of search satisfaction. In *Proc. CIKM.* 2019–2028.

[2] P. D. Bruza and S. Dennis. 1997. Query reformulation on the internet: Empirical data and the hyperindex search engine. In *Proc. RIAO.* 488–499.

[3] F. Cai and M. de Rijke. 2016. Selectively personalizing query auto-completion. In *Proc. SIGIR.* 993–996.

[4] C. Carpineto and G. Romano. 2012. A survey of automatic query expansion in information retrieval. *ACM Comp. Surv.* 44, 1 (2012), 1.

[5] J. Gao, X. Li, D. Micol, C. Quirk, and X. Sun. 2010. A large scale ranker-based system for search query spelling correction. In *Proc. COLING.* 358–366.

[6] K. Hofmann, L. Li, and F. Radlinski. 2016. Online evaluation for information retrieval. *Found. & Trends in IR* 10, 1 (2016), 1–117.

[7] J. Huang and E. N. Efthimiadis. 2009. Analyzing and evaluating query reformulation strategies in web search logs. In *Proc. CIKM.* 77–86.

[8] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger. 2015. From word embeddings to document distances. In *Proc. ICML.* 957–966.

[9] T. Mikolov, K. Chen, G. Corrado, and J. Dean. 2013. Efficient estimation of word representations in vector space. In *Proc. ICLR Wrkshp.*

[10] A. Saha and D. Arya. 2017. Generalized mixed effect models for personalizing job search. In *Proc. SIGIR.* 1129–1132.

[11] M. Shokouhi, R. Jones, U. Ozertem, K. Raghunathan, and F. Diaz. 2014. Mobile query reformulations. In *Proc. SIGIR.* 1011–1014.

[12] D. Spina, M. Maistro, Y. Ren, S. Sadeghi, W. Wong, T. Baldwin, L. Cavedon, A. Moffat, M. Sanderson, F. Scholer, and J. Zobel. 2017. Understanding user behavior in job and talent search: An Initial Investigation. In *Proc. SIGIR Wrkshp. eCommerce.*

[13] J. Wang, W. Zhang, and S. Yuan. 2017. Display advertising with real-time bidding (RTB) and behavioural targeting. *Found. & Trends in IR* 11, 4-5 (2017), 297–435.