Exploring LSTMs for Simulating Search Sessions in Digital Libraries

Sebastian Günther, Paul Göttert, and Matthias Hagen

Martin-Luther-Universität Halle-Wittenberg, Halle (Saale), Germany
first-name.last-name@informatik.uni-halle.de

Abstract. We explore the application of long short-term memory models (LSTM) to simulate search behavior in a digital library. Like web search engines, digital libraries update the retrieval backend or the user interface. However, with the typically rather small user base, evaluating the changes based on user behavior analysis is difficult. To improve this process, we explore whether an LSTM-based model can generate realistic user behavior data. Trained on a cleaned version of the SUSS dataset (555,008 search sessions), the LSTM model uses the whole session history to predict the next interaction. Our preliminary experiments show that this approach can generate realistic sessions.

Keywords: Simulation · Search Behavior · User Modeling · LSTM

1 Introduction

Web search engines like Google are able to evaluate and improve their retrieval backend via A/B tests on millions of daily user sessions. Most digital libraries, however, have considerably less traffic—making reliable evaluations via A/B tests much more difficult. Thus, several previous studies suggested to simulate digital library sessions via Markov models or other "classic" machine learning-based approaches. In our study, for the first time, we explore recurrent neural networks (RNN) with a long short-term memory architecture (LSTM) for session simulation. We start by cleaning an existing digital library session log on which we then use Keras and Tensorflow to train and tune LSTM models.

Instead of creating individual simulation models for specific aspects like query reformulation, stopping behavior, or dwell time, we want to evaluate whether a combination of features can be used to directly simulate complex behavior. Our focus is on simulating realistic interaction sequences while abstracting from fine-grained details like, for instance, the exact strings of possibly submitted queries. Besides the LSTM-based simulation approach, we also present and analyze metrics for session similarity and the quality of whole session logs.[1] Our study highlights the importance of not "overfitting" the simulated sessions to be too similar to the original data, but to enable the creation of also somewhat different sessions when utilizing machine learning for simulation.

[1] Code and data: https://github.com/webis-de/tpdl22-lstm-session-simulation

2 Related Work

Search interactions usually fall into a few key types (i.e., query formulation, snippet and document examination, etc.) that Maxwell et al. [5] captured in the Complex Searcher Model (CSM) and implemented in the SimIIR framework [4]. For realistic simulations, past interactions can play an important role as demonstrated by Cheng et al. [2] who used session history in their LSTM-based LostNet model for re-ranking and query prediction. In our study, we will thus try LSTMs to simulate whole sessions of interactions. However, predicting future interactions from historic data is difficult. Kinley et al. [3] run a user study with 50 participants on search tasks and show that the same searcher is likely to have a high variation in behavior for different tasks that a machine learning model without knowledge of the task types might miss.

Besides simulating realistic search behavior, analyzing optimal strategies can also be interesting. Baskaya et al. [1] studied the impact of user behavior factors on the retrieval effectiveness. They concluded that there is no single best strategy for every task type, and that simulated ideal user behavior is not realistic.

An important aspect of realistic or optimal search simulation also is the temporal dimension—with reading as a major factor (snippets, documents). Weller et al. [7] analyzed reading time for different text characteristics (e.g., font type, topic, length). On data of 1,000 study participants, they found that a simplistic text length-based model works very well to predict reading time. We will use regression models to simulate interaction times.

3 Search Session Dataset and Data Preparation

For our study, we use the Sowiport User Search Sessions dataset (SUSS)[2] with 558,008 sessions collected over a 1-year period in 2014. Sowiport [6] was a digital library for the field of social science and was operated until 2017. To prepare the data for LSTM simulation model training and evaluation, we filter out sessions with no search-related interactions, and we detect and remove anomalous interactions from within sessions, as removing the entire sessions would reduce the dataset size substantially. We also identify a small subset of systematic irregularities caused by the logging process (e.g., no duration for the last action of a session), which we "fix" by extrapolating from respective interactions with time information. Lastly, sessions are split after 30 minutes without interaction.

4 Model Training

We train LSTM models on 80% of the data using the open-source library Keras. Each input vector consists of at least two interaction steps from the training data. We initially test two variants: one with five features (action length, action, subaction, origin action, response) and one with six additional features (search

[2] https://data.gesis.org/sharing/#!Detail/10.7802/1380

Table 1. Basic characteristics of real sessions (test data) and simulated sessions.

Data	**Interaction duration**		**Query length**		**Page number**		**Number of results**	
	avg	**sd**	**avg**	**sd**	**avg**	**sd**	**avg**	**sd**
SUSS test data	46.95	354.18	12.52	8.02	1.24	0.59	11.71	5.28
LSTM-simulated	46.82	331.63	12.80	7.88	1.21	0.62	11.68	5.22

term type, search term length, search term complexity, sorted, page, information type). We choose the latter for its slightly improved prediction accuracy.

We normalize continuous values (removing 5% upper outliers) and one-hot encode categorical features. Our rather simple models have two hidden layers with 128 and 64 nodes, sigmoid as the activation function, cross-entropy as the loss function, a learning rate of 0.001, and a batch size of 128. We also set class weights to boost interactions classes that are rare in our training data and conclude training after 20 epochs, as the prediction accuracy does not further improve. The simulation uses regression models to "predict" continuous values (e.g., interaction duration, query length). However, the SUSS data does not contain all the data needed for predictions. An example is interaction time for reading: without document content, it can only be guessed with some randomness. More accurate predictions of reading time or query length require more knowledge about the search intent, the result documents, or the shown snippets.

5 Experiments

Assessing simulated sessions is a difficult task, as there are no established measures and the task is further complicated by the multidimensional nature of the session data. We therefore use three different approaches to assess the simulated sessions, with each approach covering different aspects.

Comparing basic session characteristics. The results in Table 1 show that, on average, the basic characteristics of real SUSS sessions from the test data and of 1,000 LSTM-simulated sessions are very similar. From that perspective, LSTM-based simulation is promising.

Human assessment. In our second assessment, we conduct a manual pilot annotation to evaluate the sessions' "look and feel" from a human perspective. We sample 20 real and 20 simulated sessions each containing a sequence of interactions with durations, number of results, and the usage of pagination or sorting. In random order, one annotator familiar with the SUSS data labeled each session as 'real' or 'simulated'. Afterwards, our annotator told us that their assessments were mostly based on three properties and possible issues of simulated sessions.

(1) Interaction sequence. Search sessions usually follow a cycle of submitting queries and examining results (comparable to the CSM [5]), interleaved with changing parameters like sorting or pagination. Any deviation is an indicator for either a malformed or a multi-browser-tab session.

(2) Interaction duration. Some interactions' duration can indicate abnormal behavior (e.g., assessing a document as relevant after zero seconds). Still, this might also occur when using multiple tabs, refreshing the page, or by misclicks.

(3) Parameters. While most parameter values are plausible, impossible combinations may occur (e.g., examining a result from a zero-result SERP).

Obviously, some of the above properties exploited by our annotator can occur legitimately and may have led to some wrong assessments. In our small pilot annotation, from the 20 real sessions, 16 were correctly identified as real, while 4 were falsely judged as simulated. From the simulated sessions, 8 were correctly identified as simulated, while 12 were convincing enough to be judged as real.

Session novelty. Search session simulation usually has two somewhat conflicting goals: realism (i.e., the sessions should be similar to real ones) but also not just memorization (i.e., not just sampling from the training data). Like Google's reported daily 15% of unseen queries,[3] simulated sessions should also contain "new" interaction sequences. We thus focus on novelty for our third assessment.

Using a new similarity measure for "almost exact matches" (i.e., sessions with the same interactions, in the same order, that take about the same time), we mark sessions as novel that have no match in the SUSS training data in two scenarios. In the first scenario, the LSTM model is trained on the first 80% of the SUSS sessions and the remaining 20% are simulated, while in the second scenario the ratio is 90% to 10%. For both scenarios, we compare the ratio of novel simulated sessions to that of the respectively remaining test data. While in the 80% scenario, about 5.46% of the real SUSS sessions are novel (5.18% @ 90%), the ratio is slightly higher for the simulated sessions at 5.91% (5.74% @ 90%). Also from that perspective, LSTM-based simulation thus is promising.

6 Conclusion and Future Work

We have shown some preliminary results on using LSTM models to simulate search sessions in a digital library. For our study, we filtered and transformed the SUSS dataset to extract suitable search sessions. The interaction histories were compiled into short time series datasets that we used to train models for the predictions. In an experimental analysis, we analyzed basic statistical characteristics of the simulated sessions, manually assessed their plausibility by trying to distinguish real from simulated sessions, and analyzed session novelty compared to the training data. Our results indicate that LSTM-based session simulation is very promising from all three evaluation angles.

In future work, we want to generalize our small-scale experiments by comparing LSTM-based session simulation to other approaches like Markov modeling or the rather simpler approaches implemented in the SimIIR simulation framework.

Acknowledgments This work has been partially supported by the DFG (German Research Foundation) through the project 408022022 "SINIR – Simulating INteractive Information Retrieval".

[3] https://blog.google/products/search/our-latest-quality-improvements-search/

Bibliography

[1] Baskaya, F., Keskustalo, H., Järvelin, K.: Modeling behavioral factors in interactive information retrieval. In: He, Q., Iyengar, A., Nejdl, W., Pei, J., Rastogi, R. (eds.) Proceedings of the 22nd ACM International Conference on Information and Knowledge Management (CIKM 2013), San Francisco, CA, USA, October 27 – November 1, 2013, pp. 2297–2302, ACM (2013)

[2] Cheng, Q., Ren, Z., Lin, Y., Ren, P., Chen, Z., Liu, X., de Rijke, M.: Long short-term session search: Joint personalized reranking and next query prediction. In: Proceedings of The Web Conference 2021 (WWW 2021), Virtual Event / Ljubljana, Slovenia, April 19-23, 2021, pp. 239–248, ACM/IW3C2 (2021)

[3] Kinley, K., Tjondronegoro, D., Partridge, H., Edwards, S.L.: Relationship between the nature of the search task types and query reformulation behaviour. In: Trotman, A., Cunningham, S.J., Sitbon, L. (eds.) The Seventeenth Australasian Document Computing Symposium (ADCS 2012), Dunedin, New Zealand, December 5–6, 2012, pp. 39–46, ACM (2012)

[4] Maxwell, D., Azzopardi, L.: Simulating interactive information retrieval: SimIIR: A framework for the simulation of interaction. In: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2016), Pisa, Italy, July 17–21, 2016, pp. 1141–1144, ACM (2016)

[5] Maxwell, D., Azzopardi, L., Järvelin, K., Keskustalo, H.: Searching and stopping: An analysis of stopping rules and strategies. In: Proceedings of the 24th ACM International Conference on Information and Knowledge Management (CIKM 2015), Melbourne, VIC, Australia, October 19 – 23, 2015, pp. 313–322, ACM (2015)

[6] Mayr, P.: Sowiport User Search Sessions data set (SUSS) (Version: 1.0.0) (2016)

[7] Weller, O., Hildebrandt, J., Reznik, I., Challis, C., Tass, E.S., Snell, Q., Seppi, K.D.: You don't have time to read this: An exploration of document reading time prediction. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J.R. (eds.) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020), Online, July 5–10, 2020, pp. 1789–1794, Association for Computational Linguistics (2020)