QUERY AUTO-COMPLETION (SIGIR 2017)

CS 572: Information Retrieval Final Project Report

Yasoda Sai Ram Kandikonda
ykandik@emory.edu

Venkata Anirudh Pillala
vpillal@emory.edu

Veda Varshith Sai Chidalla
cvedava@emory.edu

Kiran Kumar Nimmakayala
knimmak@emory.edu

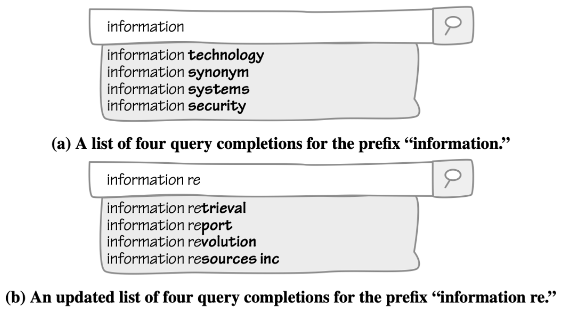
Abstract

Efficiency in Query Auto-Completion (QAC) systems is crucial for improving the user experience in search environments. However, traditional QAC methods often struggle with novel or unique user inputs. This project introduces a novel approach to QAC by leveraging a Large Language Model to predict query completions for previously unseen prefixes. By analyzing variable-length prefixes, the model generates query suggestions by estimating the likelihood of sequence extensions. The proposed model was trained and validated on a comprehensive dataset, resulting in promising improvements over traditional methods, as measured by Mean Reciprocal Rank (MRR). These findings not only highlight the potential of neural language models in QAC systems but also pave the way for future advancements in search technologies, particularly in scenarios with limited historical data.

1 Introduction

Query Auto-Completion (QAC) is an essential feature in modern search engines, enhancing the user experience by reducing the effort required to input queries and by assisting users in query formulation. By predicting and suggesting completions as users type, QAC systems significantly accelerate the search process and can influence the search results displayed. Despite their utility, traditional QAC systems predominantly rely on historical query logs to generate suggestions. This method, while effective in high-frequency scenarios, fails to address less common or novel queries that lack historical data.

The limitation becomes particularly pronounced in specialised search environments, such as academic or niche-market search engines, where unique queries are more frequent and the available query logs are sparse. This project aims to address these limitations by developing a new approach to QAC using a neural language model. This model leverages a transformer-based text-to-text T5 model to learn from past queries and predict possible completions for new, unseen query prefixes.



(a) A list of four query completions for the prefix "information."

(b) An updated list of four query completions for the prefix "information re."

Figure 1: Examples of query auto-completion.

By focusing on this innovative approach, the project seeks not only to enhance the performance of QAC systems in handling rare and novel queries but also to contribute to the broader field of information retrieval by providing insights into the application of neural networks in predictive typing systems.

2 Background and Literature Review

Query Auto-Completion (QAC) has evolved with advancements in information retrieval, employing historical data to suggest completions. Traditional systems, such as frequency-based methods, prioritize commonly entered queries but falter with novel or rare inputs. Contextual methods enhance relevance by incorporating user session data but are similarly limited by their dependence on historical precedents.

Recent shifts toward machine learning, particularly neural networks, offer significant improvements. Transformer networks, for example, excel in sequential prediction tasks critical for QAC, accommodating dynamic user inputs without direct historical references. Research by Aaron Jaech, Mari Ostendorf, Dae Hoon Park, and Rikio Chiba illustrates the efficacy of personalized and context-aware models, underscoring the potential of neural approaches to outperform traditional QAC systems.

3 Methodology

This project leverages a FLAN-T5 transformer-based architecture to develop a neural language model specifically tailored for query auto-completion (QAC). Operating at both character and word levels, the model analyzes text input as users type (the 'prefix') and generates the most likely completion based on learned patterns in query data. Additionally, we incorporate the traditional query auto-completion technique of Most Popular Completion in combination with FLAN-T5 to incorporate popularity. MPC, or Most Popular Completion, uses a trie data structure and is a technique used in query auto-completion systems to suggest the most likely completions based on popularity statistics.

3.1 Model Architecture

The T5 (Text-To-Text Transfer Transformer) architecture is a versatile and powerful transformer-based model developed by Google AI. Unlike traditional sequence-to-sequence models where input is converted into output sequences (like translation or summarization), T5 frames all NLP tasks as a text-to-text problem. This means both the input and output are textual, and the model learns to map one text to another. T5 is built upon the transformer architecture, which consists of an encoder and a decoder with multiple layers of self-attention mechanisms. This architecture allows it to capture long-range dependencies and handle various NLP tasks effectively. T5 utilizes text extracted from the Common Crawl web corpus and applies simple heuristic filtering techniques for preprocessing.

3.2 Training Process

The T5 model undergoes fine-tuning on a comprehensive dataset of prefix-query pairs, which is split into training, validation, and testing sets to ensure robust learning and evaluation. Fine-tuning is conducted at both the character and word levels. Multiple training methodologies are employed throughout the project. Initially, we fine-tuned T5 at the word level on prefix-query pairs. While this method produces a diverse set of suggestions, they tend to lack context. To address this, we incorporate context by utilizing a Named Entity Recognition (NER) model to extract named entities from the queries. These entities are treated as popular words. We concatenate the word embeddings and popular words, treating them as one string separated by a token, and feed this combined input to the model. Additionally, we train a character-level language model to capture the finer details of the text. Finally, we leverage MPC (Most Popular Completion), built with all the extracted named entities and the T5 model, to effectively incorporate popularity into the suggestions. MPC is valuable in query auto-completion systems because it helps users quickly find relevant and commonly searched-for queries. By leveraging the wisdom of the crowd, MPC improves the user experience by suggesting completions that are likely to be useful or interesting to the user.

3.3 Optimization and Regularization Techniques

Dropout: Applied within the network to reduce overfitting by randomly ignoring selected neurons during training, which helps the model generalize better to unseen data.

Beam Search: During the prediction phase, beam search is used to explore multiple prediction paths and their probabilities, allowing the model to generate more accurate query completions by considering a broader context.

3.4 Evaluation Metrics

The model's performance is evaluated using the Mean Reciprocal Rank (MRR). The metric assesses the precision and relevance of the model's predictions, focusing on the ranking of the correct query completions among the suggestions provided by the model.

4 Implementation

The implementation of the Transformer-based model for Query Auto-Completion (QAC) involved setting up a computational environment, training the model, and evaluating its performance. Each of these steps is detailed below.

4.1 Model Setup

Data Preprocessing: The first step involved preprocessing the dataset to convert raw query logs into a suitable format for training. This included tokenization of queries into characters, encoding characters as integers for model input, and splitting the data into training, validation, and testing sets.

Model Configuration: The T5 model was configured with specific parameters, dropout rates, and the dimensionality (128) of the embedding layer.

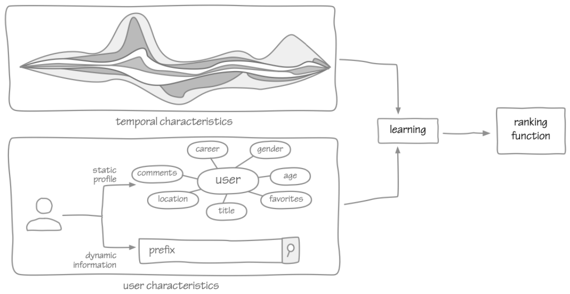


Figure 2: A general framework for learning-based query auto completion approaches

4.2 Training Process

Batch Processing: Queries were fed into the model in batches to optimize memory usage and improve training dynamics.

Epochs: The model was trained with 15 epochs, with each epoch representing a complete pass over the entire training dataset with a learning rate of 1e-6 to prevent overshooting.

Callbacks: Various callbacks were used to monitor the training process, including early stopping to halt training when the validation loss ceased to improve, thereby preventing overfitting.

4.3 Testing and Evaluation

Model Testing: Upon completion of the training, the model was tested on the unseen test set to evaluate its performance using the Mean Reciprocal Rank (MRR).

Result Analysis: The results were analyzed to understand the effectiveness of the model in predicting query completions and to identify any potential areas for improvement.

5 Results

The T5 model for Query Auto-Completion (QAC) was rigorously evaluated using Mean Reciprocal Rank (MRR) to determine its effectiveness in generating query completions, especially focusing on its performance with novel or rare queries.

5.1 Performance Metrics and Outcomes

The results of the model are tabulated below. We used the top 25 results to calculate the MRR. We used MPC on the last word in the prefix to generate context-based results. Below are the results from various configurations of the T5 model:

Model Configuration	**MRR@25**
T5-small	0.199
T5-Large	0.214
T5-Large + NER (popularity)	0.130
T5-Large character level	0.262
T5-Large character level + MPC	0.356

Table 1: Performance of different T5 model configurations.

6 Discussion

The implementation and evaluation of the T5 model for Query Auto-Completion (QAC) at both the character level and the word level yielded several insights and underscored the potential of using advanced neural networks to enhance search functionalities. This discussion highlights the significance of the findings, the model's contributions to the field of information retrieval, and the broader implications.

6.1 Interpretation of Results

The T5 model's performance, as indicated by MRR, validates the hypothesis that neural language models can effectively handle the variability and complexity of human language input better than traditional QAC systems. Specifically, the model's ability to anticipate and complete unseen queries represents a significant advancement, addressing one of the major limitations of existing QAC systems that rely heavily on historical query data.

6.2 Concluding Thoughts

This project underscores the transformative potential of LLMs in enhancing the functionality and user experience of QAC systems. As the field of information retrieval continues to evolve, the integration of such neural models will play a pivotal role in shaping the next generation of search technologies.

7 Conclusion and Future Work

7.1 Conclusion

This project successfully developed and evaluated a T5 for enhancing Query Auto-Completion (QAC) systems. The model demonstrated significant improvements over traditional QAC methods, particularly in handling novel and complex queries. With an MRR of 0.35, the model proved its efficacy in predicting accurate query completions, thereby facilitating a more efficient and user-friendly search experience.

The project's success highlights the robustness of Large language models for understanding and processing human language in a dynamic and interactive context. This approach not only addresses the limitations of traditional QAC systems, which rely heavily on historical data, but also showcases the potential of neural networks to adapt to new and evolving user inputs without the need for extensive historical logs.

7.2 Future Work

RAG (Retrieval-Augmented Generation) represents a pioneering approach that blends retrieval and generation techniques to enhance the performance of natural language generation models. Introduced in the paper "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks" by Lewis et al. in 2020, RAG has demonstrated remarkable improvements over purely generative models across various knowledge-intensive natural language processing (NLP) tasks, including question answering, conversational agents, and summarization. By leveraging external knowledge through retrieval, RAG models can generate more accurate and contextually relevant responses.

While Language Model Generators (LLMs) can produce high-quality but often general suggestions, enhancing the user experience requires considering intent and context. Incorporating historical data and other signals such as popularity, location, and user preferences can significantly enhance the quality of suggestions. These contextual factors provide valuable insights that help tailor suggestions more effectively to the user's needs and preferences.

7.3 Closing Remarks

The integration of Large language models into QAC systems represents a significant step forward in the development of intelligent and responsive search technologies. As the digital landscape continues to evolve, the need for advanced predictive text systems will only grow. This project lays a solid foundation for future explorations and innovations in the field of information retrieval, setting the stage for further advancements that could transform user interactions with technology.

8 References

1. Aaron Jaech and Mari Ostendorf. "Personalized language model for query auto-completion." In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, April 2018.

2. Dae Hoon Park and Rikio Chiba. "A neural language model for query auto-completion." In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1189–1192, ACM, 2017.

3. F. Cai and M. de Rijke. "A survey of query auto completion in information retrieval." *Foundations and Trends in Information Retrieval*, vol. 10, no. 4, pages 273–363, 2016.

4. Shokouhi, M. "Learning to personalize query auto-completion." In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 103–112, ACM, 2013.

5. Georg Buscher, Andreas Dengel, and Ludger van Elst. "Query expansion using gaze-based feedback on the subdocument level." In *Proceedings of the 31st International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 387–394, New York, NY, USA, 2008. ACM.

6. M. Benjamin Dias, Dominique Locher, Ming Li, Wael El-Deredy, and Paulo J. G. Lisboa. "The value of personalised recommender systems to e-business: A case study." In *Proceedings of the 2008 ACM Conference on Recommender Systems*, RecSys '08, pages 291–294, New York, NY, USA, 2008. ACM.