

Lab Note: The Advanced Question Query Generator

Team name: The Query Lab

Members: Mark Henri Mörsheim, Maryam El Ghadioui

In this micro task, we developed an Advanced Question Query Generator to produce realistic, question-like search queries. Our goal was to reflect actual user behavior more closely than traditional generators. We began with a simple heuristic model and iteratively refined it based on empirical observations.

Our system follows a rule-based pipeline with probabilistic elements. We begin by retrieving topic-specific user queries using `get_given_queries()` from the iFind framework. The final query in the list is taken as the *topic seed* for generation. For each new query the generator selects a *wh-word* such as *how*, *what*, or *why*, based on a weighted distribution reflecting real user data. Then, it applies a sentence template containing placeholders for the *wh-word* and *topic*, producing natural-sounding question structures, for example:

- *"{w_word} are the main causes of {topic}"*
- *"{w_word} can governments do about {topic}"*

And to make the queries more realistic, we also control the punctuation at the end: there's a 29% chance of ending with a question mark, 22% with a period, and 49% with no punctuation at all. This generation process is repeated 10 times to create a batch of natural-sounding questions.

Our original baseline was too simple for real search behavior. It just added a random *wh-word* to the beginning of an existing query and placed a question mark at the end. So, we worked with the session group who had clustered real user queries resembling questions. Their analysis helped us spot common traits:

- Only **29%** of queries ended with a **question mark**
- Nearly **50%** had **no punctuation**
- Over **70%** began with a **wh-word**
- Frequent openers included "how to", "what is", and "how does"
- Average length: 79 characters or **~11 words**; most common: **48 characters**

Using this data, we built better templates and adjusted our system to reflect how people actually type queries. We also experimented with using spaCy to extract syntactic structures and part-of-speech patterns, but the results were noisy and less effective than our simpler, template-driven method.

This task showed us that small, data-informed changes can make a big difference in making query generation feel more realistic. In interactive retrieval, users don't just ask one question—they adapt, reformulate, and explore as they go. Our generator doesn't fully capture changing intent yet, but it better reflects how people naturally ask questions in search settings. Looking ahead, evaluation in IR could improve by taking user behavior and session context more seriously. If we combine typical user patterns with simple language rules, we can get much closer to simulating how real search sessions actually work.