**Documentation Search with Elasticsearch and GENRE**

**Team Softmax**

* Ilia Andreev (captain), iandre3@illinois.edu
* Raj Krishnan, rajk3@illinois.edu
* Sengill Kim, sk100@illinois.edu

**Abstract**

In this project, we will compare the performance between Generative Entity Retrieval(GENRE) and Elasticsearch-based standard full-text search methods. We will use the Python documentation as the knowledge base, documentation sections (e.g., “Instance Objects” or “Expression Lists”) as documents, and StackOverflow-like questions as queries. We will answer these queries and measure the performance in two ways : by retrieving relevant documentation with Elasticsearch and by generating relevant documentation section names.

**Proposal**

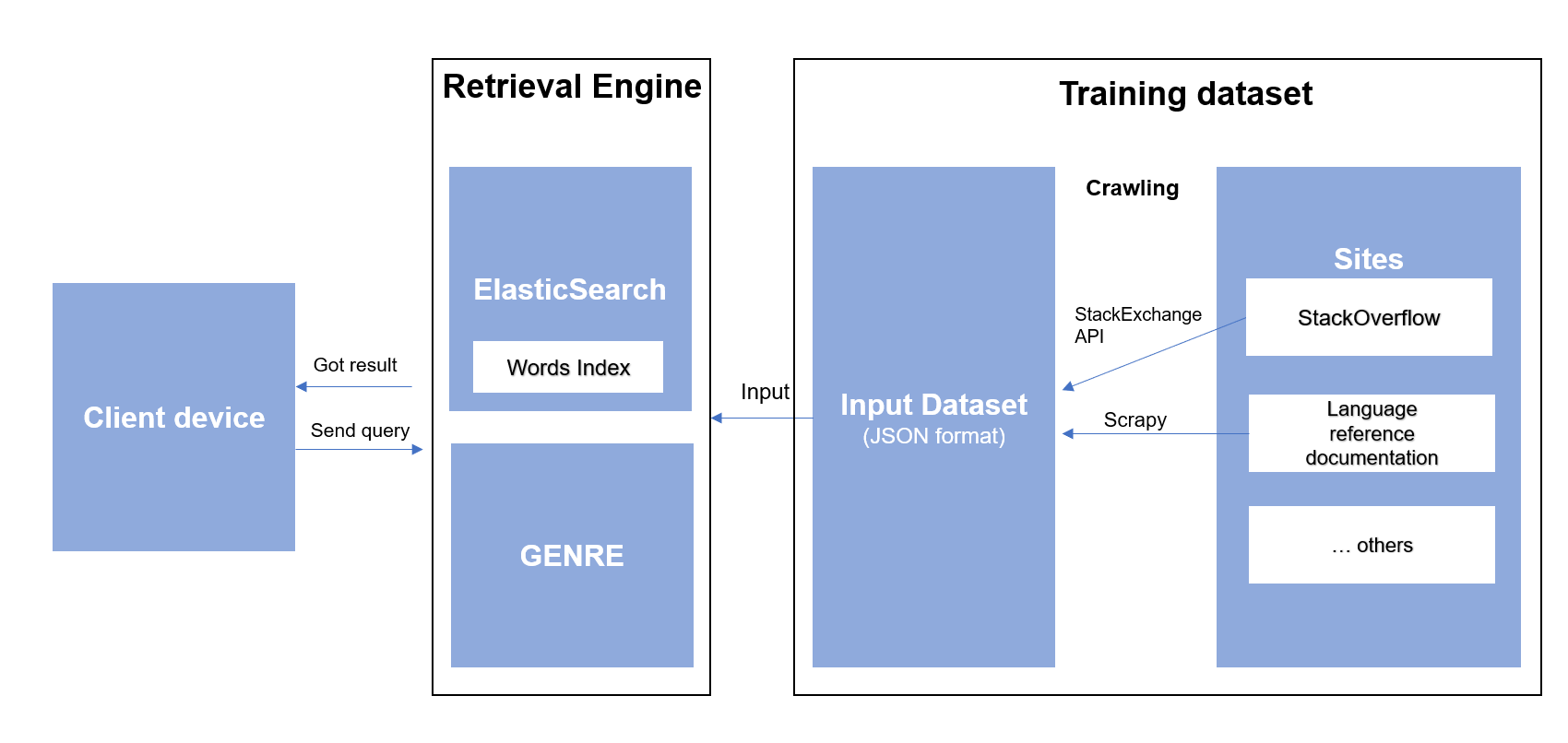
A paper on Autoregressive Entity Retrieval came out in March 2021, along with the accompanying implementation called GENRE. This paper proposes a new method of searching knowledge bases. It uses a sequence-to-sequence model to handle end-to-end entity linking, entity disambiguation, and document retrieval. In practice, this means that GENRE generates names of the knowledge base entries related to the query based on the context in which those entries have been referenced in the training data.

We seek to compare this approach with an industry-standard search solution – Elasticsearch, which is built on Apache Lucene – and see how the two approaches differ for the task of retrieving entities from documentation when presented with StackOverflow questions as queries. We find this work important since it compares some of the classical text retrievals approaches with more novel but less common methods. For this project, we are going to build our own training and test datasets. These datasets will consist of publicly available StackOverflow questions with popular answers linking to specific language documentation sections. We are going to use Python 3 Standard Library and Language Reference as our knowledge base. All systems implemented for this project will be written in Python.

Although this project is suggested as a free topic, it is closely related to other themes. First, creating a training dataset through StackOverflow and language reference itself can be an independent project in theme 4, leaderboard competitions. Furthermore, our project can be considered as a learning system that offers the opportunity to improve learning efficiency. Searching is a fundamental building block on learning nowadays.

**Proposed Implementation**

Below is the conceptual architecture of our project.



1. We will implement both ElasticSearch and GENRE to compare performance.
2. The training dataset will be crawled from StackOverflow and Language reference documentation. We will use StackExchange API and Scrapy for the data crawling.
3. Queries from the client will be StackOverflow-like questions.
4. The performance measurement will depend on the Mean Reciprocal Rank metrics. Each team member will decide the relevance and compare scores.

Based on them, our implementation steps will roughly involve the following:

1. Build a corpus of language documentation.
2. Index the language documentation into Elasticsearch with minimal tuning.
3. Build a corpus of queries from StackOverflow, retrieving user questions with at least one popular accepted answer. Filter the results to only consider the answers linking to Python documentation and split the question-answer pairs into training and test datasets.
4. Use the training dataset from StackOverflow to train the GENRE sequence-to-sequence model.
5. Run the test queries against the model based on GENRE and Elasticsearch, and see how their Mean Reciprocal Rank metrics compare.

**Expected Challenges**

We are aware that this project is challenging. Below are expected challenges during the project, but we also expect more obstacles during the project.

1. Right data format for the training dataset: There may not be a universal data format for both ElasticSearch and GENRE.
2. ElasticSearch optimization: Is it fair to compare the score with the default configuration of ElasticSearch? If not, how much should we invest our time for the optimization?
3. GENRE implementation: We trust the reliability of ElasticSearch. However, does GENRE work well as expected? How is our increased workload to fix bugs?
4. Track issue: How do we track and record issues during the project?

We expect the workload to be distributed as follows, but also expect actual investment will be more than expected during the project.

1. Build a language documentation corpus: 5 hours
2. Index documentation into Elasticsearch: 5 hours
3. Build StackOverflow question-answer dataset: 20 hours
4. Train the sequence-to-sequence model using the StackOverflow dataset: 20 hours
5. Build a test harness to evaluate approaches: 15 hours
6. Compare approaches: 2 hours
7. Total: 67 hour