Mastering Snowflake: Sentiment Analysis and Performance Experiments

This is the first project for the Advanced Data Systems (ADS) class of 2024. The project is expected to take four weeks to complete.

Goals

1. Become familiar with a modern cloud data platform system, Snowflake.
2. Implement a complex data analysis algorithm using SQL and user-defined table functions (UDTFs) in a procedural programming language.
3. Design and conduct experiments to measure algorithm precision and runtime.

Tasks

Accessing and Understanding Snowflake

Snowflake is a data analysis platform. It is considered an OLAP system, designed to analyze massive amounts of data. The architecture and properties that contributed to Snowflake's success will be discussed in the ADS class on cloud-native databases. Snowflake operates as a service in the cloud.

For this project assignment, you may access Snowflake here:

- https://sfedu02-gyb58550.snowflakecomputing.com

Login credentials will be handed out in class. You will be forced to change the password on the first log in.

Please read the Snowflake Documentation at https://docs.snowflake.com to understand how to work with Snowflake. Importantly, read and follow the user guides on virtual warehouses, databases & tables, and loading data into Snowflake.

Sentiment Analysis in SQL

Sentiment analysis is a technique used to understand the emotions expressed in text, such as whether the sentiment is positive or negative. It analyzes text from sources like social media, reviews, or articles to gauge public opinion and customer feedback. This information helps businesses and organizations make informed decisions by understanding how people feel about a product or service.

In this assignment, you will implement a sentiment analysis algorithm in SQL, specifically, the Naïve Bayes classification algorithm. A good overview of this algorithm can be found in these slides from Stanford University:

- https://web.stanford.edu/class/cs124/lec/naivebayes2021.pdf

The data you will use to train the classifier are sample product reviews from Yelp. The data can be downloaded using the following command. You may need to install Git LFS (Large File Storage) first:

- git clone https://huggingface.co/datasets/Yelp/yelp_review_full

The dataset is freely available for research and teaching purposes, but not for commercial use. Please refrain from publishing this dataset or any of its parts. Use it exclusively for your class project.

You need to upload the data using a command-line tool because the Snowflake UI restricts uploading datasets larger than 250MB. The easiest way is to install the snowsql client on your machine. Connect to your Snowflake account through snowsql using your login credentials. The account name to use for snowsql is 'sfedu02-gyb58550'.

It is recommended to create a database prefixed with your username. For example, if your username is ALPHA, create a database called ALPHA_DB. This way, each of you can work in your own database without impacting others. Create a named stage and use snowsql to upload the training and test data.

Implement the Naïve Bayes algorithm using the training data. Split sentences into words and consider performing additional data cleaning to remove special characters. A UDF in JavaScript can help a lot here. Analyze its precision against the test dataset.

Sentiment Analysis using a UDTF

The Naïve Bayes algorithm is great because it does not require loops. In SQL, using loops can be difficult, if not impossible at times. To address this issue, user-defined functions (UDFs) and user-defined table functions (UDTFs) are available. These allow the implementation of other classification algorithms, such as boosted decision trees or even neural networks. However, you will not explore those in this exercise. Here, you will implement the Naïve Bayes algorithm again using a UDTF. You can choose to use Java, JavaScript, or Python as programming language.

Again, analyze the algorithm's precision against the test data set. Compare both implementations in terms of the amount of work required and the simplicity of the code. Additionally, compare their performance in terms of execution time.

Performance Experiments Using TPC-H

To further dive into performance evaluations, in the final part of this project, you will conduct a standard performance experiment using the TPC-H benchmark. TPC-H is a classic benchmark used to compare the performance of OLAP database systems.

TPC-H data is already loaded in various sizes in Snowflake. You can find the data in different schemas within the SNOWFLAKE_SAMPLE_DATA database. Use only 3 of the 22 queries:

- Query 1, a scan-intensive query.
- Query 5, a join-intensive query.
- Query 18, an aggregate-intensive query (involving many group-by operations).

Use these queries to run performance experiments on four different sizes of TPC-H data (scaling factors SF1, SF10, SF100, and SF1000) and four different sizes of virtual warehouses (XS, S, M, and L). Measure the query runtimes.

Deliverables

The hand-in for the assignment includes:

1. Your code for the sentiment analysis algorithm in SQL and the UDTF. Please provide a link to a GitHub code repository, either from ITU or the public GitHub.
2. Your report.

The report should be approximately 4 pages and should include the following sections:

- **Introduction**. Provide an overview of the report.
- **Experimental Methodology and Results.**
    - Describe your experimental design (e.g., Snowflake configuration, Snowflake client used, data sets used, characteristics of data sets, number of runs to measure performance) to measure the precision and runtime performance of the sentiment analysis algorithms and the runtime performance of the TPC-H queries.
    - Present the results of your experiments. If you use graphs, ensure the x-axis, y-axis, and graph legends have clear titles with corresponding units and explanations.
- **Discussion of Results.** Interpret the results and discuss additional findings, such as the comparison between the two implementations of the sentiment analysis algorithm.
- **Conclusion.** Summarize your main findings.

Feel free to ask us if you have questions. Direct questions to: Martin Hentschel, mhent@itu.dk