Uppsala University

Applied Cloud Computing

1TD265 Autumn 2020

Assignment 3

– Celery Workers and the RabbitMQ Broker

Copyright authors:

Henrik Schulze

29th October 2020

ABSTRACT

In this assignment a prototype system is built to analyze a dataset of Twitter tweets using Twitter's datastream API. The tweets have been collected beforehand. The task is to provide counting of some Swedish pronouns as a service.

1 Task-1.

The Twitter tweets are available from the following link:

https://uppsala.box.com/s/qiiggdjd98241wm7rl3lqehfhyjomg4y

The dataset consists of a number of files containing line-separated tweet entries. Every second line is a blank line.

Each tweet is a JSON document http://en.wikipedia.org/wiki/JSON. JSON is one of the standard Markup formats used on the Web. While it would not be considered your typical "scientific data", familiarity with JSON quickly becomes essential in applied cloud computing and data analytics. For the specific case of Twitter tweets, you can read about the possible fields in the JSON documents here: `https://dev.twitter.com/overview/api/tweets`

This particular Twitter dataset was collected by filtering the stream of tweets to store those containing the Swedish pronouns "han", "hon", "den", "det", "denna", "denne" and the gender neutral, new pronoun "hen". Your task is now to construct a compute-service based on the distributed task queue of Celery, using 'RabbitMQ as the broker. The service should be capable of analyzing the dataset on demand. In particular, as an example of an analysis, your solution should count the total number of mentions of the above mentioned pronouns in the dataset.

Minimal Requirement for a passing grade on Task 1

To complete Task 1, your solution should meet the following criteria.

- A backend comprising of Celery/RabbitMQ deployed on a single VM (one Celery worker).
- The service should be exposed to an end-user via a simple REST API, capable (only) of returning a single JSON document containing the pronouns and their respective counts. For a very simple example of a REST API, see the Flask app. from Lab 1.
- A Visualization of the frequencies (number of mentions normalized by the total number of unique tweets) of the pronouns, for example as a bar chart. In the analysis, only unique tweets should be taken into account, i.e. 'retweets' should be disregarded.
- Upload a short report to the student portal, describing how you designed your solution, displaying e.g. a screenshot of the REST API call showing the returned result. Also include the visualization (see above point), and the design of your service. You need to write half a page about the tools you will use and the architecture of your service. Basically the description of your analysis pipeline.

Answer.

I have a backend consisting of the RabbitMQ broker and one Celery worker on a single VM. The service is started by a short and simple command: `docker-compose up -d` [1] [2] [3][1]. The design is inspired by tutorials available online [4] [5]. A schematic drawing of the architecture is shown in figure 1 [6].

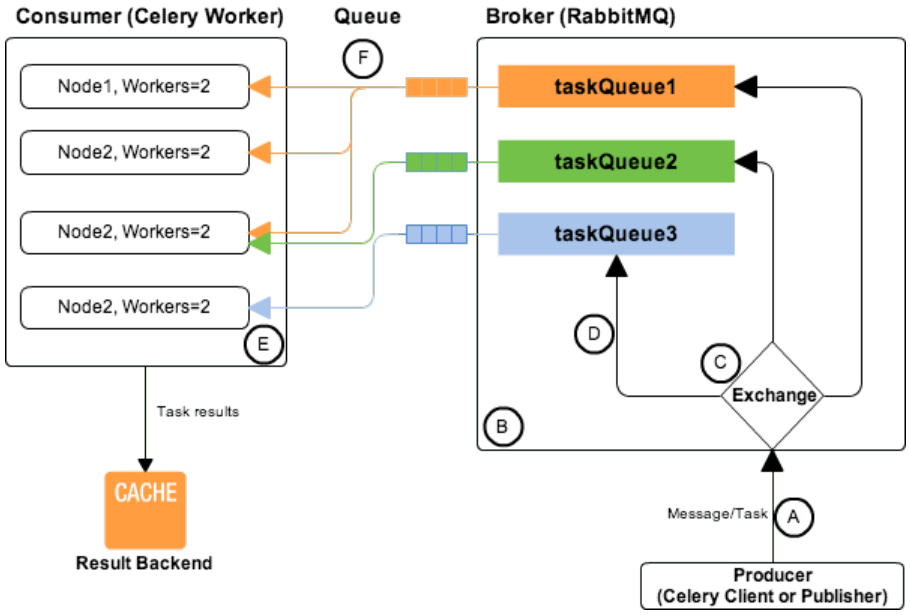

Figure 1: schematic description of the architecture.

The end-user reaches the service via the simple Flask app (a REST API). The service returns a single JSON document containing the pronouns and their respective counts, along with the total number of tweets. See the screenshot in figure 2.

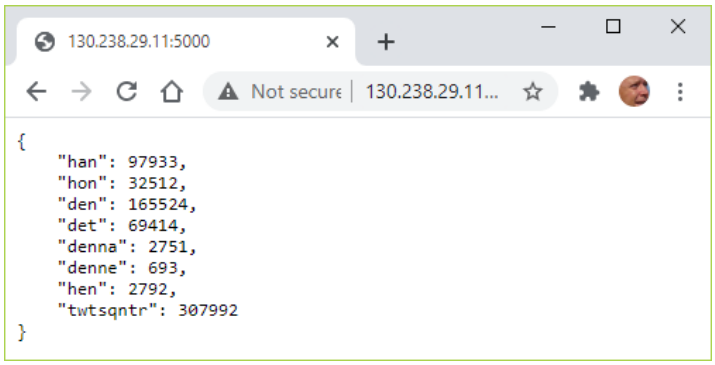

Figure 2: screenshot of the result when calling the service.

[1] In reference [3] I needed to comment out the row `RUN apk add -no-cache gcc musl-dev linux-headers` in order to be able to run the `Dockerfile`. Also, there are quite a few dependencies that need to be installed, which the tutorial does not mention.

Figure 1 shows the result graphically. The result shows that – even – the most common Swedish noun, "den", occurs less than once per tweet on average.

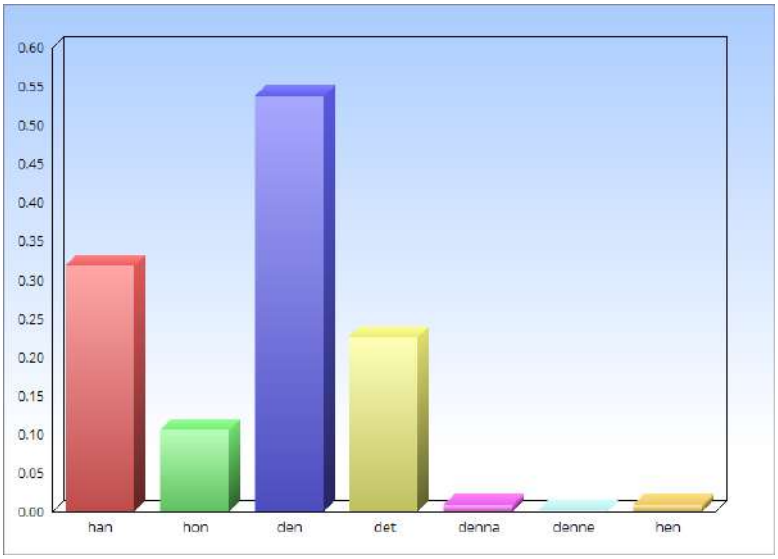

Figure 3: occurrence of seven Swedish nouns in relation to the number of tweets.

References

[1] "How To Install and Use Docker on Ubuntu 18.04."
https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-18-04

[2] "How To Install Docker Compose on Ubuntu 18.04."
https://www.digitalocean.com/community/tutorials/how-to-install-docker-compose-on-ubuntu-18-04

[3] "Get started with Docker Compose."
https://docs.docker.com/compose/gettingstarted/

[4] "First Steps with Celery."
https://docs.celeryproject.org/en/latest/getting-started/first-steps-with-celery.html

[5] "Using RabbitMQ."
https://docs.celeryproject.org/en/latest/getting-started/brokers/rabbitmq.html

[6] "AMQP, RabbitMQ and Celery - A Visual Guide For Dummies."
https://www.abhishek-tiwari.com/amqp-rabbitmq-and-celery-a-visual-guide-for-dummies/?fbclid=IwAR3A4ax85-bbr796slmoT17RyaxX8Vxuis1zvKvgKKcNMyliPH4DFPkuCro#routingkeys