Information Retrieval and Web Search - Project Phase 1

Lukas Pfahler Tejas Umakanth

February 8, 2014

1 Collaboration Details

We both do everything. clever...asdfa sdkfjalöksdj fölakdjf öaljdf,ajhdf a abdf afas dasfdghalkjdhfa

adsfadsf

sfg asd

sdfg

2 Description

For our project, we decided to crawl and index data found on Instagram[1]. Instagram is a popular[2] social media application that allows users to publish photos and videos and search the media published by other users. Each user is identified by a unique user name. Each media item can have a caption, a list of comments by other users or a location. As known from other popular social networks like Twitter and Facebook, captions and comments can contain hashtags.

[1] `instagram.com`

[2] Instagram reported 150 million users in September, 2013.

3 Crawling

Instagram offers a developer API that allows us to subscribe to real time updates and crawl new media items as soon as they are posted.[3] There are different options for subscriptions: You can either subscripe to a specific user, hashtag or to a geographic area or location. All communication between the crawler and instagram is done using the HTTP protocol: The crawler is itself a http-server and once we subscripe to media updates at instagram.com, their servers start connecting to our crawler. Everytime there is new media, they issue a http POST request. The post does not contain the actual media, but is merely a notification that there has been an update. We then grab the actual data by requesting all recently added media from their servers in a http GET request.

The response is a JSON file containing a list of media items. We save each of those items in a separate JSON file. We also save a thumbnail image for each media item.

Our crawler is based on the software platform Node.js, which allows us to setup a http-server and request data in an asynchronous fashion with very little javascript code.

> Node.js is a platform built on Chrome's JavaScript runtime. [..] It uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.[4]

Instagram allowes developers to issue 5000 requests per hour[5], thus we introduced a politeness factor p: Only every p-th time our crawler is notified we actually request the new data. This has two effects: First, the number of requests is reduced and second, the number of new media items per response is higher. It also means that we probably miss a fraction of

[3] `http://instagram.com/developer/realtime/`

[4] `http://nodejs.org/`

[5] `http://instagram.com/developer/endpoints/`

the data. However, if one would setup a larger system with multiple crawlers the coverage would increase.

Limitation of our own bandwidth.

4 Indexing

blablablablabla concatenation of caption and all comments. remove all hashtagsymbols. remove all occurences of the hashtag used to subscription. index all hashtags separately. index location using lucene.spatial. index usernames.