Master Thesis

Crowdsourced Product Descriptions and Price Estimations

Steve Aschwanden
Dammstrasse 4
CH-2540 Grenchen
steve.aschwanden@students.unibe.ch
05-480-686

Supervisor
Dr. Gianluca Demartini
C312, Bd de Pérolles 90
CH-1700 Fribourg
demartini@exascale.info

Grenchen, February 18, 2014

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all the principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Steve Aschwanden, 05-480-686

Grenchen; February 18, 2014:

_____

(Signature)

Acknowledgements

I like to acknowledge ...

Abstract

I like to acknowledge ...

Contents

List of Figures

List of Tables

Listings

Chapter 1

Introduction

Contents

1.1 Statement of the problem

The first step of creating an online auction is mostly to take pictures of the corresponding item. This help the buyers to get information about the state and quality of the article. After that the item needs a short and clear description, some properties (category, state) and a starting offer. If the seller wants to create a lot of different auctions, the whole procedure is time consuming and boring. A price estimation of an article can be difficult, because the background knowledge is missing and other auctions to compare aren't available at any time. Machines aren't able to solve all these steps by themself, because the spectrum of the articles is huge and image processing methodes aren't capable to classify them all correctly. To get all the needed parts of an online auction, a human powered approach is necessary. Crowdsourcing platforms provide the possibility to solve tasks, which are difficult to handle for a computer.

1.2 Existing research

1.3 Goals and objectives

The thesis will have the following goals and their corresponding objectives:

- **Collect auction item properties by the crowd**
  - Analyze the composition of an auction item and select the parts which can be crowdsourced
  - Form a ground truth including different auctions created by real online auction platform users
  - Design and publish tasks to gather data from the crowd
  - Evaluate the quality of the generated content
- **Try to improve the initial solution by implementing a hyprid approach**
  - Search for image processing methodes which can simplify and/or support a human intelligence task

  - Implement the methodes and adapt the task design
  - Publish the new tasks on the same crowdsourcing platform
  - Evalutate the results and compare them to the first solution

If the main goals of the thesis are fulfilled, some optional goals can be covered by the thesis:

- **Implement a web application which combines the created subtasks to a complete workflow**
  - Find a web application framework and a crowdsourcing platform which provide APIs in the same programming language
  - Create a workflow which put all the subtasks together to an overall solution
  - The user can manage the items (upload pictures to create new items, edit and remove items) and directly create an online auction

1.4 Evaluation

Appendix A

Some Appendix

A.1 README

```
Fuzzily classify twitter messages using storm and store to cassandra
===

Setup Cassandra (on ubuntu):
---
1. Make sure oracle JDK is installed (1.6+): https://help.ubuntu.com/community/Java#Oracle_Java_7
2. Add the DataStax repository key to your aptitude trusted keys.
> $ curl -L http://debian.datastax.com/debian/repo_key | sudo apt-key add -
3. Install Cassandra:
> sudo apt-get update && sudo apt-get install cassandra
4. Create keyspace and tables:
> cqlsh
> run commands from src/main/resources/createDatabase.txt

Build Runnable jar
---
1. Open a terminal window, navigate to pom.xml directory (project root)
2. Execute the following command:
> mvn clean compile assembly:single
3. In target/, a runnable jar tsfc.jar is created

Run Program
---
> java -jar tsfc.jar <<comma separated list of topics to watch (without whitespace)>>
```