A Web-based Platform to Annotate Messages from Twitter with Metadata

Software requirement specifications

Valerio Basile
3/6/2015

Introduction

Purpose

The system responds to the needs of the researcher who needs to perform content analysis on social media. Rather than separate pieces of software, this platform offers an integrated environment to carry out the entire process of analysis, from the collection of relevant material to its computer-aided enrichment with metadata.

Definitions

- Content Analysis: the analysis of the content of a text, performed either by human experts, optionally aided by computer tools, or automatically by a software.
- Coding: the act of assigning to an item of content a set of metadata values.
- Social Media: Web-based platforms where users are linked together in a relationship network and can create and share content with each other and with the public.

Overview

The tool that we want to develop serves as an aid to the researcher that performs content analysis on social media. Specifically, the tool supports two different kind of users, both contributing to the content analysis process. On the one hand, the researcher needs a support to set up content analysis projects, while on the other hand the people performing the actual analysis (coders) need a platform to carry out the coding in the most efficient way.

Description of the software

The tool is Web-based and accessible through a website under the University domain. Upon accessing the first time, the user is presented with a static page with name of the platform, logo and a short description, and a sign in/sign up form.

Roles and Entities

The user of the system are of one of two types: either **coders** or **administrators** (henceforth, admins). For each installation of the software, exactly one administrator is present, its login is “administrator”, and the password is predefined.
If a coder has never accessed the system, a standard registration option is offered. The coder can choose a username or stick to her email address for login, and chooses a password. After inserting the information and clicking the “sign up” button, an automatic email is sent to the provided email (mandatory), with a link to confirm the registration.
Registered coders, after login, access a **page** with a summary of their past activities and links to the **projects** they are assigned to.

Variables

A coding project comprises a set of **variables** to be filled in with values. Variables have a name, an hidden identifier (e.g., V1, V2, …) for use in statistics later, a description and a block of text of instructions. Each variables also has a set of **categories** (i.e., possible values), each one with its name. Variables have a boolean value that decides whether the variable is visible or it has to be toggled by some visibility rule (see below).

Rules

The admin can, for each project, define a set of **visibility rules**. A rule is a set of conditions that toggle the visibility of a variable on the coding screen. The conditions are over the values of other variables, e.g., “true is selected for variable V1”.

Functionalities

Project management

The admin has access to a special page where he can create projects and assign users to the projects. The admin can create as much coding projects as he likes. Each project has a list of users who are the **participants** in the project. The admin can add and remove participants to a project from the **project’s page**. The admin can also email all of the participants of the project at once through a textbox.

Data Collection

Each project starts with a **set of tweets** to be coded. On the data collection section of the interface, the admin uploads a list of Twitter username, specifies a time span (e.g., from two months ago to now), clicks a button and the system starts to collect the tweets in background. The project page displays the status of the data collection, and notifies the admin when the collection is finished.

Data split

The tweets are assigned to the coders in two possible ways: **randomly** or on a **username basis**. In the first case each tweet is assigned to a coder in an uniformly random way. In the second case, the admin manually assigns, in the project page, each Twitter username to a coder. A subset of tweets is also selected to be part of the **test set**, typically a small sample of tweets, also in chronological order.

Coding

Once the data (tweets) for a project has been collected, the coders can start their work. The **coding page** presents to the coder one tweet at a time, followed by boxes representing the coding variables. Each **variable box** contains the name of the variable and a sequence of radio buttons each associated with a category of the variable. At the bottom of the page there are the two buttons “submit” and “undo”. Clicking “submit” sends the tweets and its codes to the database and moves to the next tweet. All the visible variables must be coded before proceeding to the next tweet, otherwise a warning message is displayed when pressing the “submit” button. Clicking “undo” makes the interface go back to the previous tweet.

Modes

There are three modes of coding: coding, test and training. In **coding** mode, the coder is presented with the tweets assigned to her, one at a time, in the **chronological order** based on their timestamp. In **test** mode, all the coders are assigned the same set of tweets, that is, the test set as defined above. The project is considered finished when all the coders have coded all the tweets in both coding and test mode. The **training** mode is only used to familiarize with the interface, and its results are not stored in the database. During the training the tweets can be presented to the coders in random order.

Statistics

During the active phase of the project, the current progress of each coder is shown on a separate page linked from the project page and from the coding interface. When the test set is completely coded, from the project page it is possible to compute the agreement between the coders (Krippendorff's alpha).

Export

When the project is completed, that is, when all the tweets have been coded, from the project page it is possible to download the entire dataset in one comma-separated value text file.

Infrastructure

One installation of the software will run on the University’s servers, both for the web application and for the database. The source code will be published, together with a with a GPL-compatible licence.

Notes

The part of the backend software that search and download the tweets could be based on this Python script: https://github.com/valeriobasile/twittercrawler