Study of n-grams in illness related tweets

Juan Ignacio Gil Gómez

July 27, 2014

1 Description

An study of illness related n-grams in Twitter

We will study the distribution of n-grams (contiguous sequences of n-words) in the illness related tweets.

A broad outline of the study will be:

1. Study and visualization of the distribution of the n-grams[1] of the whole distribution of tweets.
2. Study and visualization of the distribution of n-grams of each group of diseases. Search for the n-grams characteristics of each group (n-grams with a high frecuency relative to the distribution of the whole tweets.
3. Study and visualization of the distribution of n-grams of each individual disease. Search for the n-grams characteristics of each group (n-grams with a high frecuency relative to the distribution of the group of diseases tweets).

By isolating these n-grams with a higher frecuency I expect to identify issues unique to each of the diseases (as symptoms, or social consecuences.)

2 Tools

We will use Python to analyze the tweets, probably with the NLTK libraries, and the Mongo database.

3 Installing and running the Mongo database

To be able to use the python tools, mongodb has to be installed and running in the computer.

[1] It's still to decide if we will study the n-grams to a given n as a unique set, or as a different n sets. We will take this desicion after reading more bibliography and some experimenting.

4 Adding tweets to the database

The function send_tweets_to_mongodb.py sends the tweets in the csv files to the database. It can be used to send a unique csv file as:

```
import send_tweets_to_mongodb as st
path='/Users/cato/programacion/HealthCare_Twitter_Analysis/Twitter Data/Jan to May'
group='Blood'
file='Tweets_BleedingDisorders.csv'
client = MongoClient()
db = client['HealthCare_Twitter_Analysis']
st.process_disease_file(path,group,file,collection)
```

or to navigate a folder with the structure ./Folder/Group/disease.csv as

```
python send_tweets_to_mongodb.py Folder
```

The function check for all individual tweets (represented by their url) if they are yet in the database, so no duplicates will be send, even if the p is run twice on a Folder.

Each tweet is a document in the collection, with the following fields:

```
{
_id
firstpost_date
url
trackback_author_nick
content
score
trackback_permalink
trackback_author_url
group
disease
}
```