tm.plugin.sentiment
Online Sentiment Analysis using R

Mario Annau

Statistical Natural Language Processing
10.12.2010
Table of Contents

1 Motivation

2 Related work

3 The tm library

4 tm.plugin.sentiment

5 Examples
George just bought a share

...and wants to track its news coverage on the internet
Google News
Search for Microsoft

Results 1 – 10 of about 1,256 for microsoft msft. (0.14 seconds)

MICROSOFT TO PLEASE GAMERS ON BLACK FRIDAY WITH LEADING TITLE DISCOUNTS (MSFT) ✪
Zacks.com - 1 hour ago
Nov 23, 2010 (SmartTrend News Watch via COMTEX) -- Officials from Microsoft Corp. (NASDAQ:MSFT) announced earlier this week that it will give customers Xbox ...
Google News Facts

- Biggest News Aggregator around with about 1 billion clicks per month
- Over 25000 registered publishers worldwide
- 40 different regional editions
- About 4500 publishers providing English content alone
- Google Finance News covers an estimated number of 7000 (mostly unique) Microsoft articles per year, or about 20 articles per day
Who has time to read that?
Motivation for **tm.plugin.sentiment**

- Retrieve content from news sources which are preferably free-of-charge
- Extract the main content from news pages if necessary
- Build up corpus from content which also includes time tags
- Create time series representing the sentiment of news flow over time
We want to extract the news from various sources...
Motivation

Related work

The tm library

Examples

. . . process its content. . .

Microsoft Tag Cloud

1 generated using **snippets** and **tm.plugin.sentiment**, content taken from Yahoo! Boss XML abstracts
... and create some decent sentiment time series
Table of Contents

1 Motivation
2 Related work
3 The tm library
4 tm.plugin.sentiment
5 Examples
Bo Pang and Lillian Lee

- Provide probably best research overview resource for sentiment analysis with their paper

[Pang Lee, 2008]

*Opinion Mining and Sentiment Analysis*


- Their own research focuses on sentiment analysis of online reviews
- Analyzed movie and online product reviews
Paul Tetlock

- Sentiment analysis of popular Wall Street journal column “Abreast of the Market”
  
  [Tetlock, 2007]
  
  *Giving Content to Investor Sentiment: The Role of Media in the Stock Market*
  
  *Journal of Finance 62, 1139-1168, 2007*

- Use bag-of-words model and dictionary from the General Inquirer

- Negative news sentiment category (total: 77) has most predictive power for market prices like the Dow Jones Industrial Avg. (DOW 30)
Lydia/Textmap

- Large Scale Sentiment Analysis Project at Stony Brook University, under supervision of Steven Skiena
- Daily analysis of over 1000 english and foreign language online news sources, blogs, rss feeds, historical archives, and other sources
- Online Analysis possible at http://www.textmap.com/
- Some technical details:
  - Use (adapted) wget for web spidering
  - Currently stores over one terrabyte of text
  - Use Hadoop Map/Reduce for already tagged text using Amazon’s cloud service
Commercial Products

- Thomson One
- Dow Jones News Analytics
- Raven Pack [ravenpack.com] Bayes training, vector classification, word/phrase lists, pattern detection and market response-based analysis are just a few techniques RavenPack deploys in conducting news sentiment analysis.
### Related Open Source Products/Projects

- Rapidminer with Web Extension
- Python with NLTK
Rapidminer with Web Extension

- Rapidminer based on academic project YALE (TU Dortmund)
- Semi–commercial product
- Workflow oriented
- Implemented in JAVA
- Extensions for Textprocessing and Webmining, even R
- Features include web retrieval, content extraction and bag–of–words analysis
Rapidminer with Web Extension (2)

- **Pros:**
  - Large set of operators (esp. through WEKA)
  - Workflow orientation for transparency
- **Cons:**
  - Customization, changes often have to take place in JAVA code
Python with NLTK (1)

- Python is a quite popular scripting language
- Supported by a vast amount of libraries, e.g. the numpy, scipy, matplotlib combination for fast numeric computations
- NLTK is a very popular library for text mining researchers
- User can plug together his own sentiment analysis library using packages like NLTK, feedparser, simplejson, etc.
Python with NLTK (2)

- **Pros:**
  - Excellent scripting language, also interactive
  - Quality of libraries (especially for our purposes: NLTK)
  - Can easily integrated with GUI’s (e.g. PyQt)

- **Cons:**
  - Compared to R: Worse support for stat. functions
Table of Contents

1. Motivation
2. Related work
3. The tm library
4. tm.plugin.sentiment
5. Examples
tm – Short Intro

- Developed by Feinerer as part of his dissertation in 2008
- Provides basic data structures for storing large text corpora
- Abstracts data sources and readers for input
- Includes functions for preprocessing, annotation
- Connectors to various open source libraries like openNLP, KEA
Basic Data Structures

- Text documents, storing texts and individual text meta information
- Corpora, storing collections of Text Documents and meta data

- DMeta(): Store Classification results for each text document
- CMeta(): General Corpus Information like creation date

- Possibility for database storage if corpus does not fit into memory (‘PCorpus’ vs. ‘VCorpus’).
Source–Reader Concept

- Abstract data source and reader functionality
- **Source**: Specifies how to access elements and move forward
- **Reader**: Specifies, how content can be extracted and put in a Text Document data structure.
- Call of function `Corpus()` lets you freely specify Source and Reader: `r <- Corpus(DirSource(reut21578), readerControl = list(reader = readReut21578XMLasPlain)))`
Table of Contents

1 Motivation
2 Related work
3 The tm library
4 tm.plugin.sentiment
5 Examples
Overview

- Web Data Retrieval
- Preprocessing
- Sentiment Analysis
Data retrieval

- Feeds provide meta data of news content about any topic
- Actual content usually resides on external web pages
- Feeds contain urls to content pages
Web Data retrieval

Data retrieval (2)

Therefore a 2-step procedure is necessary:

1. Download meta data feeds: `getFeed()`
2. Download content sites (optional): `getURLPart()`
Sources Available

For ease of use many different news sources have already been implemented in `tm.plugin.sentiment`:

- News Sources from RSS Feeds like Google News, Yahoo News
- APIs (free registrations required) like Reuters Spotlight, NY Times, Yahoo BOSS, Bing
Content Extraction

- News Sites often contain side bars, headers and ads
- We are only interested in the actual news story
- Heuristics are needed in order to get rid of the ‘fluff’
- Best source for a complete overview of HTML extraction techniques:

[Gottron, 2008]

*Content Extraction: Identifying the Main Content in HTML Documents*

*Johannes Gutenberg-University, Mainz, 2008*
Content Extraction (2)

Source: AI Depot
Content Extraction (3)

Implemented in function `extractContentDOM()`:

1. Examine each HTML subnode from top to bottom
2. If \( \frac{textlength}{totallength} < threshold \) then drill down
3. Select text from subnode with longest textlength

Idea stems AI Depot and Jinliang Song’s ExtMainText Python code
After Content retrieval/extraction some standard tm preprocessing steps may be required depending on source/data quality.

- Use of tm transformation functions like `tolower()`, `removePunctuation()`, etc.
- Extract “interesting” part of text content using e.g. `getRelevant()`. Useful for sites like this.
Sentiment Calculation

As a first attempt tm.plugin.sentiment uses bags–of–words model for sentiment calculation. Therefore the following ingredients are needed:

- Document Term Matrix
- Dictionary of Sentiment–laden words like *good*, *happy*, *loose* or *bankrupt*. Available from General Inquirer, NTU Sentiment Dictionary, OpinionFinder’s Subjectivity Lexicon or SentiWordnet

In order to build time series from sentiment scores each document needs to be timestamped.
Sentiment Indicators

\[ polarity = \frac{p - n}{p + n} \]  \hspace{1cm} (1)

\[ subjectivity = \frac{n + p}{N} \]  \hspace{1cm} (2)

\[ pos\_refs\_per\_ref = \frac{p}{N} \]  \hspace{1cm} (3)

\[ neg\_refs\_per\_ref = \frac{n}{N} \]  \hspace{1cm} (4)

\[ senti\_diffs\_per\_ref = \frac{p - n}{N} \]  \hspace{1cm} (5)

\(^2\)taken from the Lydia/Textmap project
Sentiment Visualization

Sentiment MSFT [2010−08−26/2010−09−22]

- Last 24.61
- Sentiment Score (SEMNT polarity): 0.2431925
- Volume (Text Documents): 71
- Volume (millions): 94,283,300
- Subjectivity All(): 0.119
- Pos_Refs_Per_Ref(on = 5): 0.076
- Neg_Refs_Per_Ref(on = 5): 0.044
- Senti_Diffs_Per_Ref(on = 5): 0.032
Table of Contents

1 Motivation

2 Related work

3 The tm library

4 tm.plugin.sentiment

5 Examples
Examples

- Google Finance News — Microsoft (MSFT)
- Yahoo BOSS — Alcoa (AA)
- Reuters Spotlight — Gold Market (GLD)