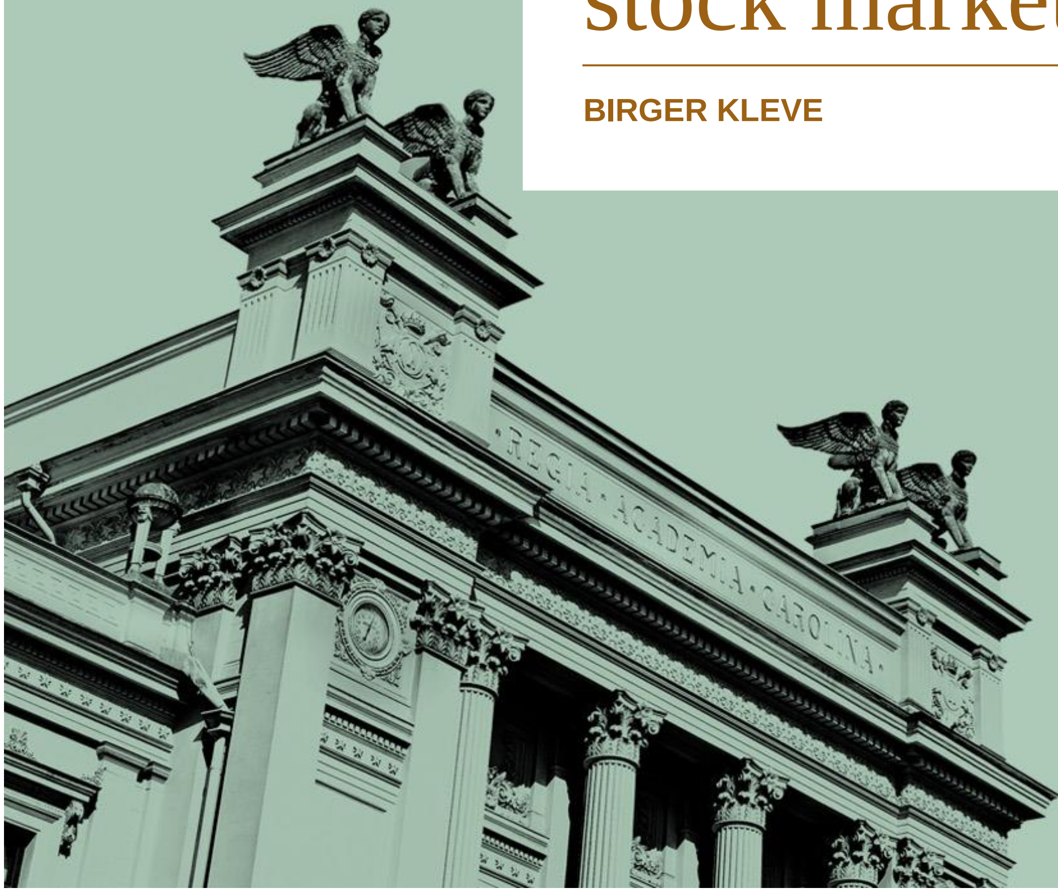
LUND UNIVERSITY

Using sentiment analysis for stock market prediction

BIRGER KLEVE



Project Goals

- Increase Machine Learning knowledge
  - Learning real world practice
  - Facing real world problems
  - Optimize algorithm parameters

Project Definition

Hypothesis:

There is a correlation between tweet sentiment from certain people and a stocks movement.

System:

1 Find tweets mentioning stocks

2 Classify sentiment of the tweet

3 Predict stock movement by processing stock data and tweet sentiment

LUND UNIVERSITY

Availability of Financial data on Twitter

stt2318
@stt2318

TWEETS	FÖLJER	FÖLJARE	FAVORITER	LISTOR
43,8 tn	80	18,9 tn	1 369	1

stt2318 @stt2318 · 22 maj

$CALL long setup

1 2

stt2318 @stt2318 · 22 maj

$TNK nice breakout today... updated chart

2

Project Redefinition

- Drop the financial aspect of the project and only focus on the sentiment of tweets

Sentiment Analysis

- Keyword spotting
  - E.g. Happy, sad, bored
- Lexical affinity
  - Affinity (swe: samhörighet) to a certain probability of polarity
- Statistical methods
- Concept-level techniques
  - Semantic analysis of text

Cambria, E. An introduction to Concept-Level Sentiment Analysis. National University of Singapore

Pang & Lee

- Thumbs up? 2002
- Movie reviews
- Presence of Unigram + Bigram w/ negation

Pang, B. Lee, L. Shivakumar, V. Thumbs up? Sentiment classification using Machine Learning Techniques. Cornell University, IBM Almaden. 2002

Social Media Features

- Words entirely in caps
- Prolonged words like angryyyyy
- Positive/negative emoticons
- Amount of hashtags

- Frequency of different POS tags

Sentiment lexicon

- Look up each word in a sentiment lexicon.
- Lexical affinity
- Use Features:
  - Highest score
  - Total score
  - Mean score

Tokenization and negation

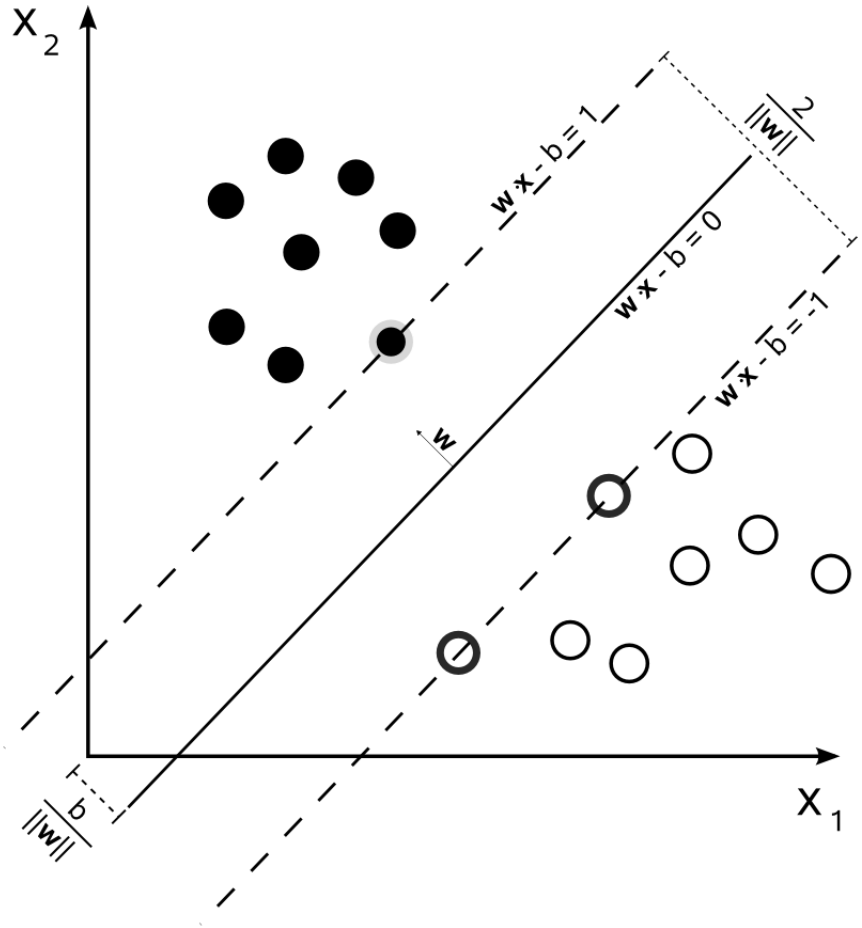
- Change usernames, URLs, hashtags etc. into normalized tokens
- Tag certain words with negation. E.g.

”This horse is not that bad” => ”This horse is not that_NOT bad_NOT”

”not quite as great” => ”not quite_NOT as great”

- Use the presence of each unigram as a feature

Classifier



- SVM with Linear kernel
- Parameters: C

Training

- Tokenize and collect each unique word in the training data and save it as a vocabulary.
- Fit SVM to the entire training set
- Optimizing parameter C
  - 3-fold Cross Validation
  - Grid Search
  - Test the final classifier against a separate test set

Data

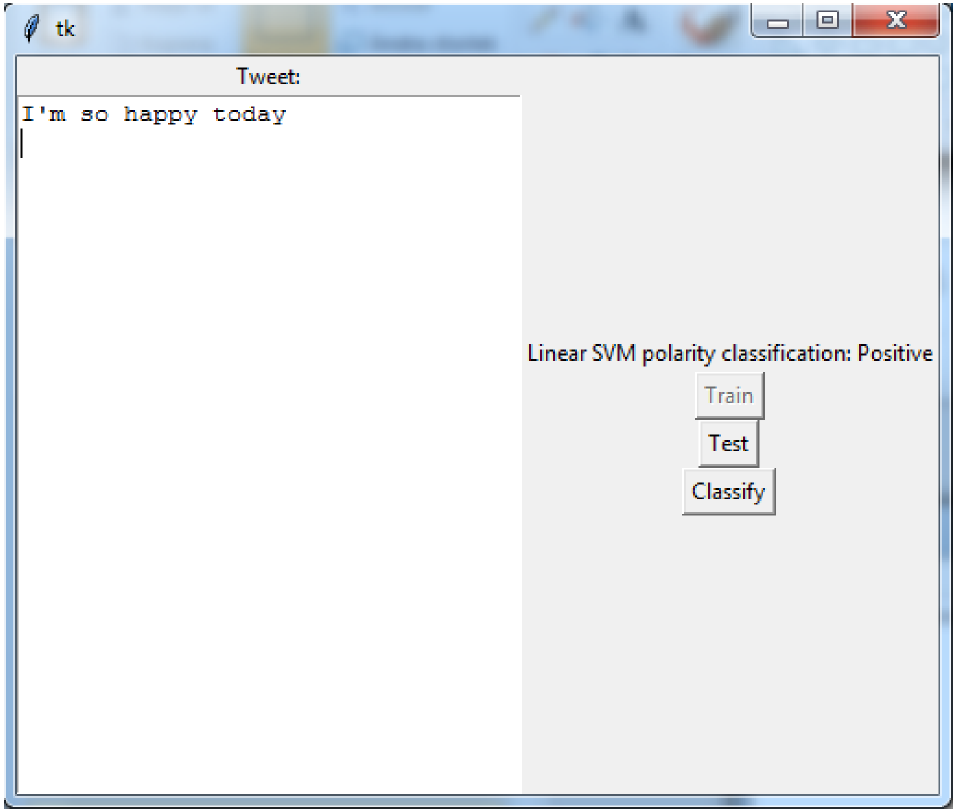
- Training set 1 600 000 automatic classified tweets
  - w/ Keyword search
  - 2 classes: Negative & Positive
- Test set 357 manually classified tweets

Go, A., Bhayani, R., & Huang, L. Twitter sentiment classification using distant supervision. Tech. rep., Stanford University, 2009.

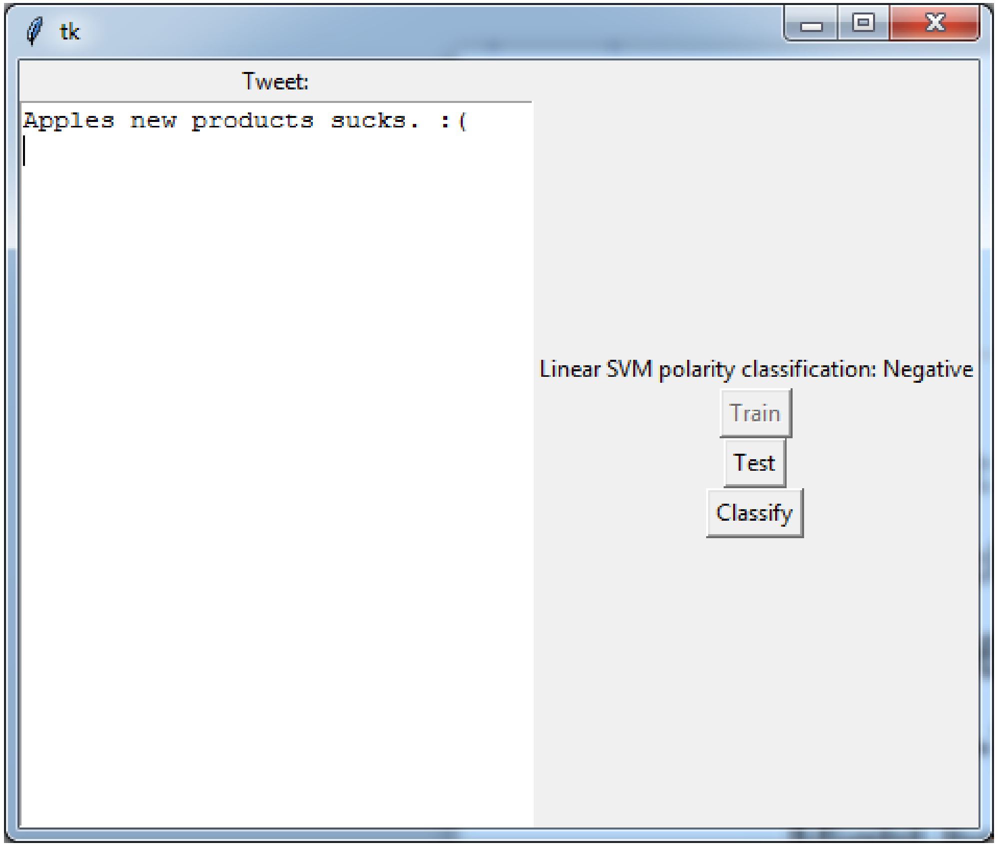
- Sentiment lexicons:
  - Lexical affinity

Kiritchenko, S., Zhu, X., Mohammad, S. Sentiment Analysis of short Informal Texts. Journal of Artificial Intelligence Research, 2014

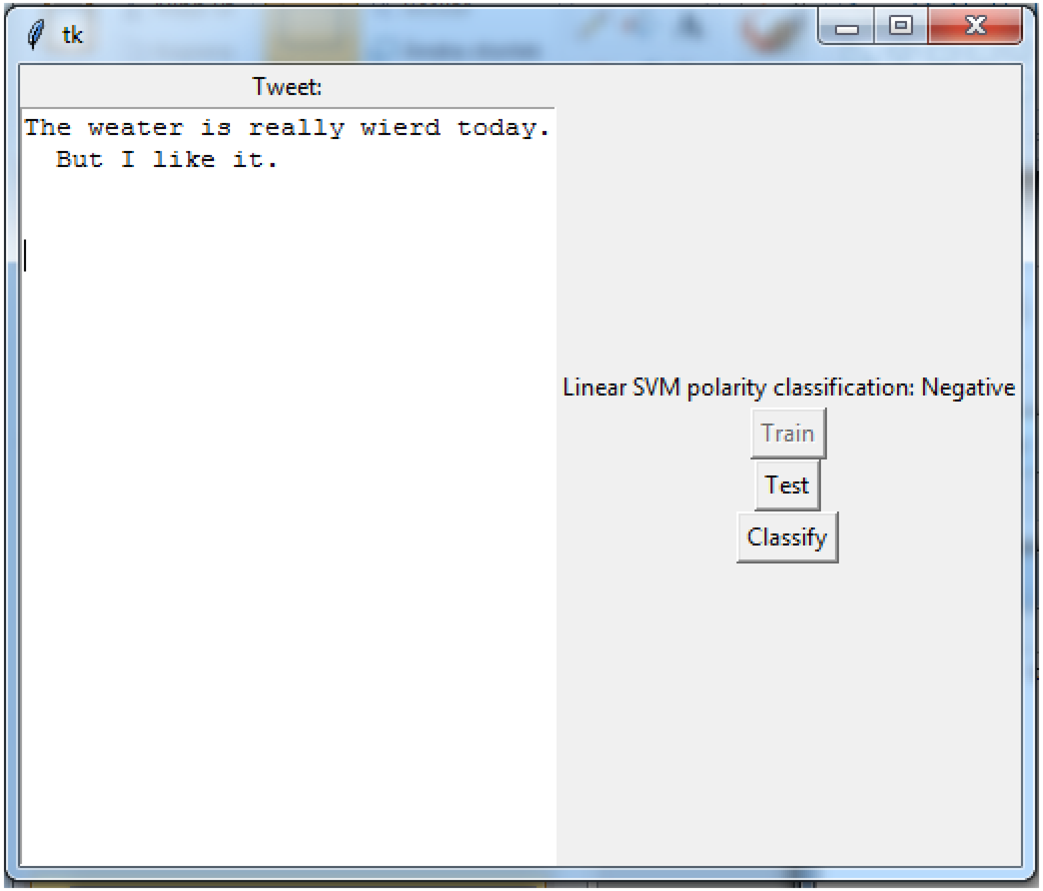
Result


tk
Tweet:
I'm so happy today
Linear SVM polarity classification: Positive
Train
Test
Classify

Result


tk
Tweet:
Apples new products sucks. :(
Linear SVM polarity classification: Negative
Train
Test
Classify

Result


tk
Tweet:
The weater is really wierd today.
But I like it.
Linear SVM polarity classification: Negative
Train
Test
Classify

Result

- Using 1.6% of the training data(25600 samples):
  - 54981 features
  - > 12 hours of optimizing
    - » DNF
  - 1 hour final training
  - Sparse features => enormous RAM allocation

Result

- Human test: ~80%
- Expected: close to 79%
- My baseline: ~65%
- My Improved: ~75%
  - Might be higher

LUND UNIVERSITY

Tools

- Python's Scikit-learn
- NLTK – for POS tagging (as features and to negate context)

What I have learned

- Pitfalls of data collection
- Handling LARGE amount of data
- Using popular machine learning tools
- (SVM, its kernels and their parameters)

LUND
UNIVERSITY