Sentiment analysis in the Twitter stream

Alina Sinelnikova, Eirini Ntoutsi, Hans-Peter Kriegel

Institute for Informatics,

Ludwig-Maximilians-Universität (LMU), Munich

The 36th Annual Conference of the German Classification Society on Data Analysis, Machine Learning and Knowledge Discovery, 1-3 August 2012, Hildesheim

Outline

- Introduction

- Data preprocessing

- Sentiment Analysis

- Conclusions & Outlook

Outline

- Introduction
- Data preprocessing
- Sentiment Analysis
- Conclusions & Outlook

Sentiment Analysis

- With the advent of Web 2.0 and its social character a lot of opinion-rich resources have arisen
  - People have the power to influence each other in the decision making process
  - Companies turn into social media monitoring in order to optimize and strengthen their products and brands

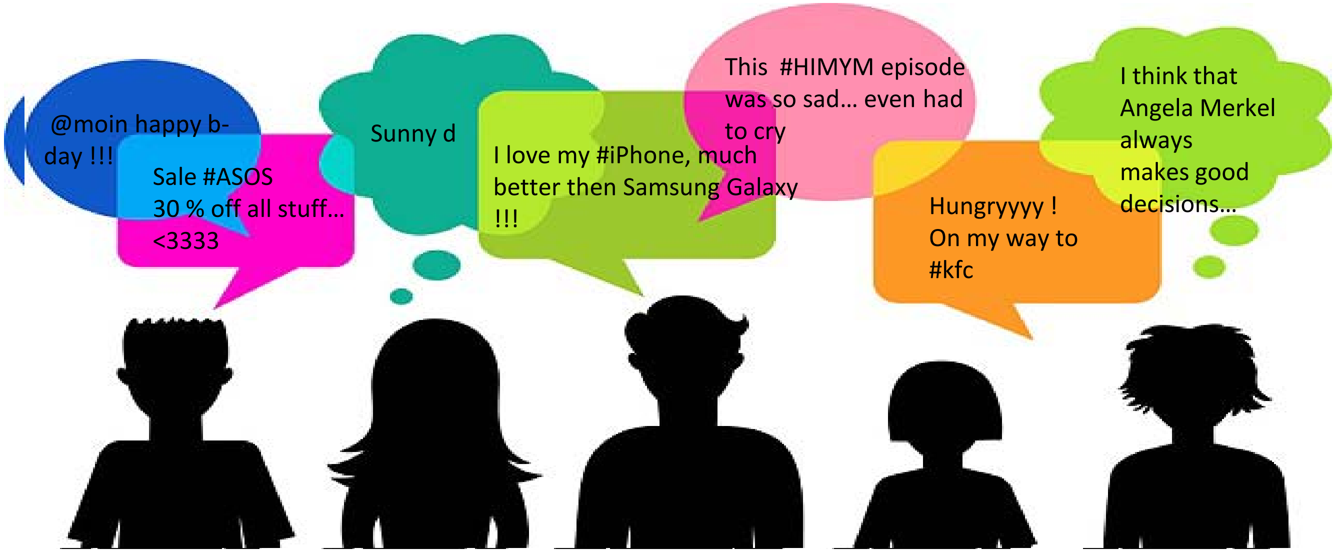
Social Media
Publish
Share
Discuss
Social Networks
Microblog
Lifestream
Livecast
Virtual Worlds
Social Games
MMO
wikia
digg
flickr
last.fm
slideshare
Crowdstorm
phpbb
PHORUM
meebo
skype
talk
myspace.com
facebook
bebo
orkut
hi5
LinkedIn
Ning
twitter
PLURK
Pownce
twitxr
tweetpeek
socializr
friendfeed
socialthing!
justin.tv
USTREAM
livecastr
qik
HABBO
CLUB PENGUIN
neopets
KARTRIDER
CONAN
doof
POGO
KONGREGATE
There
LIFE

- Huge amounts of opinions are generated
- Problem: Impossible to manually deal with all thi amount of data
- Solution: Sentiment analysis or Opinion mining
  - Automatically determine whether some opinion is positive or negative

☺ or ☹ ?

Twitter

- One of the most opinion-rich resources
- 500 million active users
- 340 million tweets per day
- One of the top 10 most visited websites on the Internet


@moin happy b-day !!!
Sale #ASOS 30 % off all stuff… <3333
Sunny d
I love my #iPhone, much better then Samsung Galaxy !!!
This #HIMYM episode was so sad… even had to cry
Hungryyyy ! On my way to #kfc
I think that Angela Merkel always makes good decisions…

Challenges of sentiment analysis in Twitter

- Finding an appropriate data set for training
  - subjective tweet: “love my iPhone“
  - objective tweet: “iPhone 4 has a lot of new functions like facetime "
- The unambiguous identification of sentiment
  - bipolar tweet: "Mcdonalds, was so tasty! But now I'm feeling so fat, like a cow:(!“
  - sarcastic tweet: “Have to wait for 30 minutes! Love it!"
- Dealing with colloquial language
  - tweets containing colloquial slang: "omg luv ths burger arrrr, so damn tastyyy!"
- Dealing with unbalanced data
  - 65% positive, 35% negative tweets
- Dealing with data changes (Twitter stream)
  - online algorithms operating with limited resources, time and memory

Outline

- Introduction

- Data preprocessing

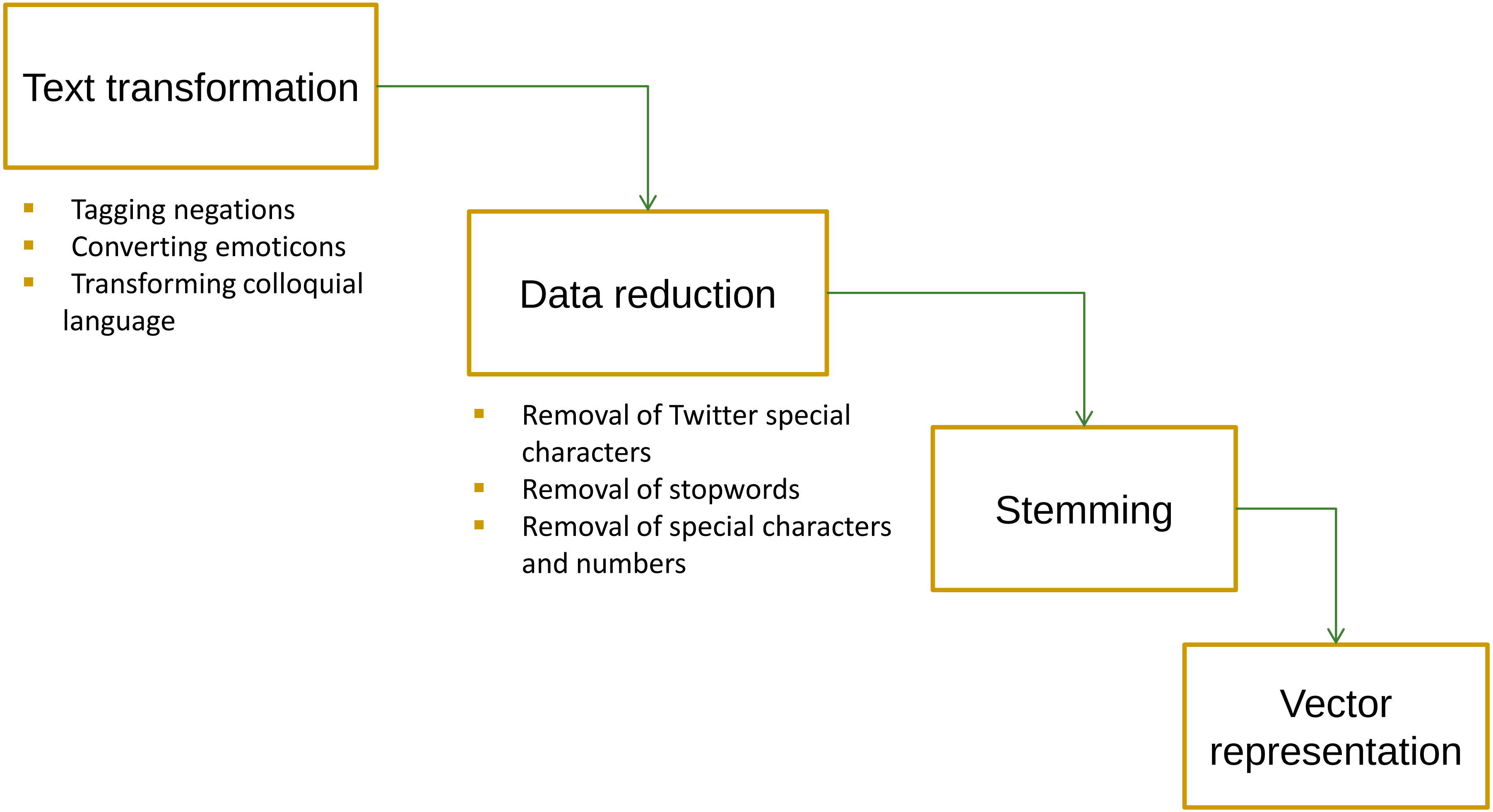
- Sentiment Analysis

- Conclusions & Outlook

Datasets

Dataset	Twitter Sentiment (www.sentiment140.com)	Crawled (LMU)
Topics	mixed	mixed (16 topics)
Labeling	machine *(emoticons based)*	human
Size	1.600.000	4.197
Period	1/4/2009 – 1/7/2009	20/11/2011- 17/12/2011
Positive instances	800.000	2.949
Negative instances	800.000	1.248
Words	21.763.131	55.063

Data preprocessing



Text transformation

- Tagging negations
- Converting emoticons
- Transforming colloquial language

Data reduction

- Removal of Twitter special characters
- Removal of stopwords
- Removal of special characters and numbers

Stemming

Vector representation

Text transformation: tagging negations

- Tagging negations with verbs

```
I do not like  → I NOT_like
It didn't fit  →  It NOT_fit
```

→ 81.348 found negations

- Tagging negations with adjectives
  - 2-part adjective co-occurrences

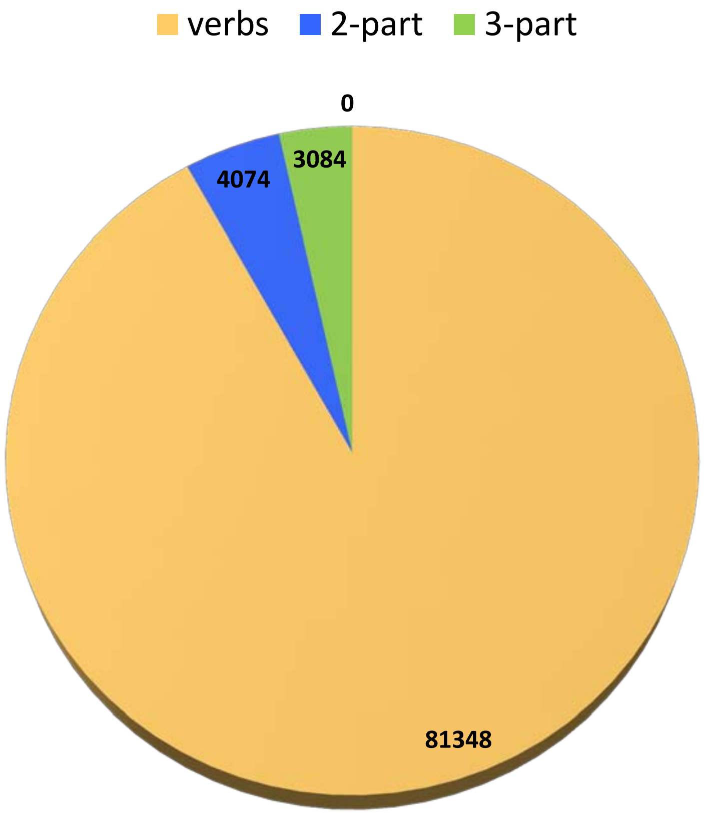
```
not pretty  → ugly
not bad → good
```

→ 4.074 found negations

  - 3-part adjective co-occurrences

```
not very young → old
```

→ 3.084 found negations



Verbs negation list: www.vocabulix.com
Adverbs negation list: www.scribd.com

Text transformation: converting emoticons

Examples:

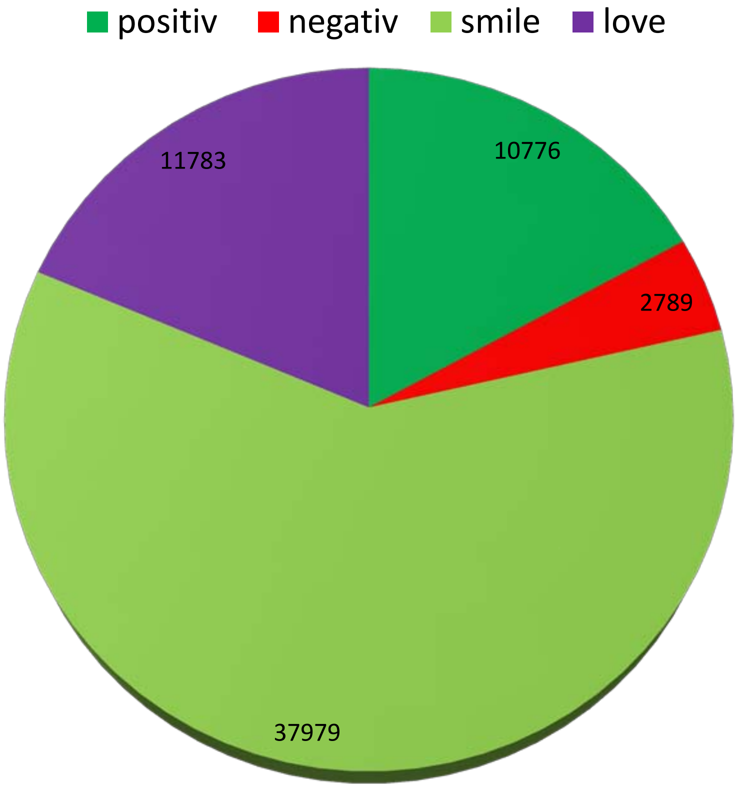
:-) :) :o) =) ;) (: (; (= → “positive”

:(:-(:o(=(;(;-( ):);)= → “negative”

:D :-D :oD =D ;D → “smile”

<3 → “love”

→ 63.327 emoticons found



Text transformation: transforming colloquial language

Examples:

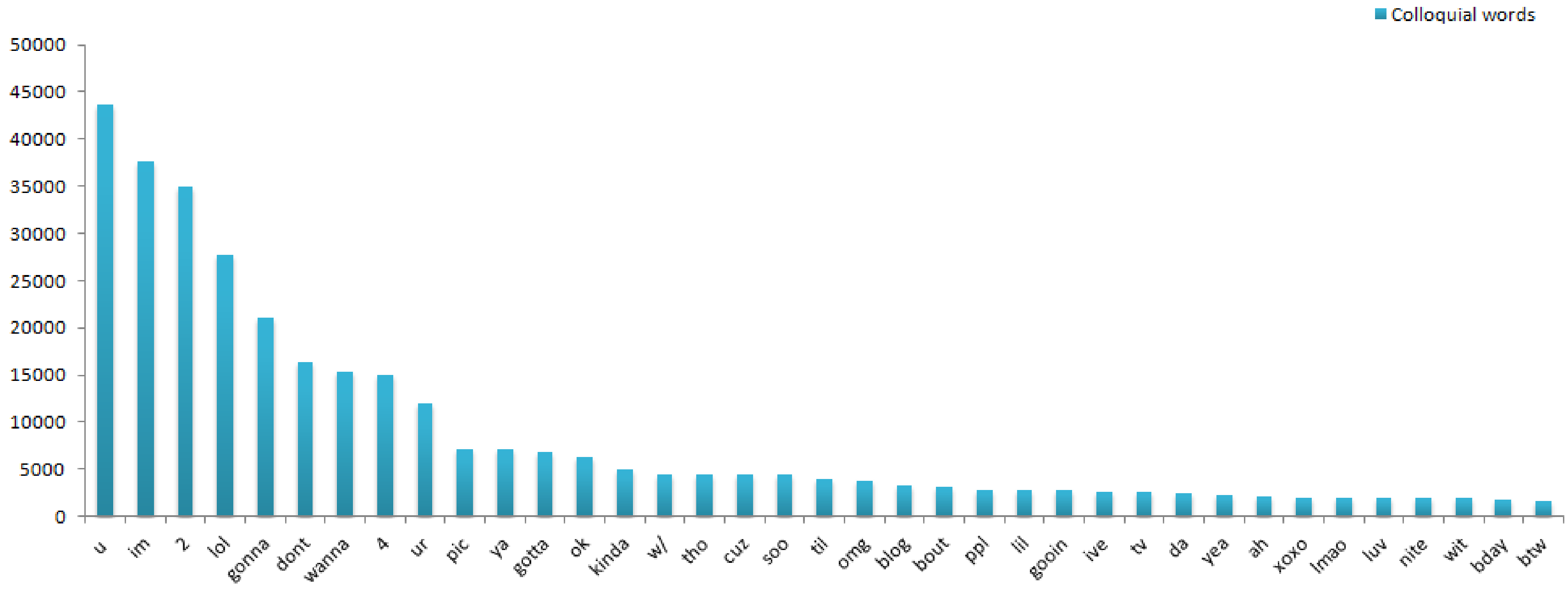
lol → laughing out loud

xoxo → kisses and hugs

u → you

Slang dictionary: www.noslang.com

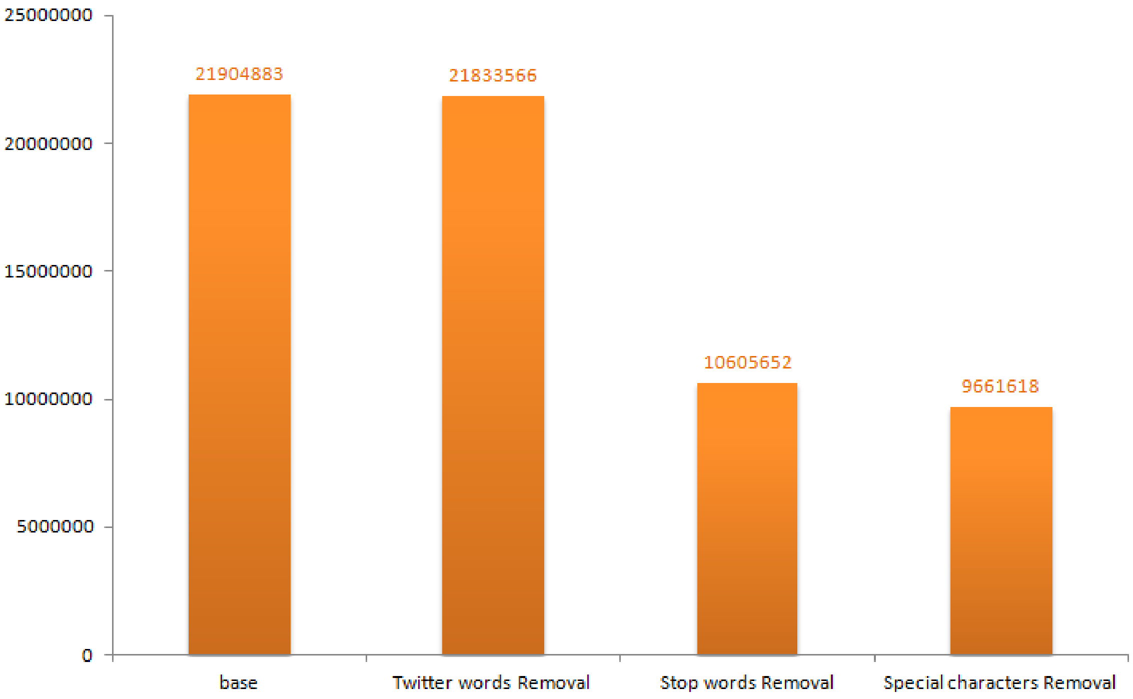
→ 499.576 transformations made



Eirini Ntoutsi

Data reduction: Elimination of superfluous words

- Removal of Twitter special characters (@, #, RT)
- Removal of stopwords (and, for, with, about, you, me, …)
- Removal of special characters and numbers (?, %,!, 1, 2, 3, …)



Stopwords list from WEKA
www.cs.waikato.ac.nz/ml/weka

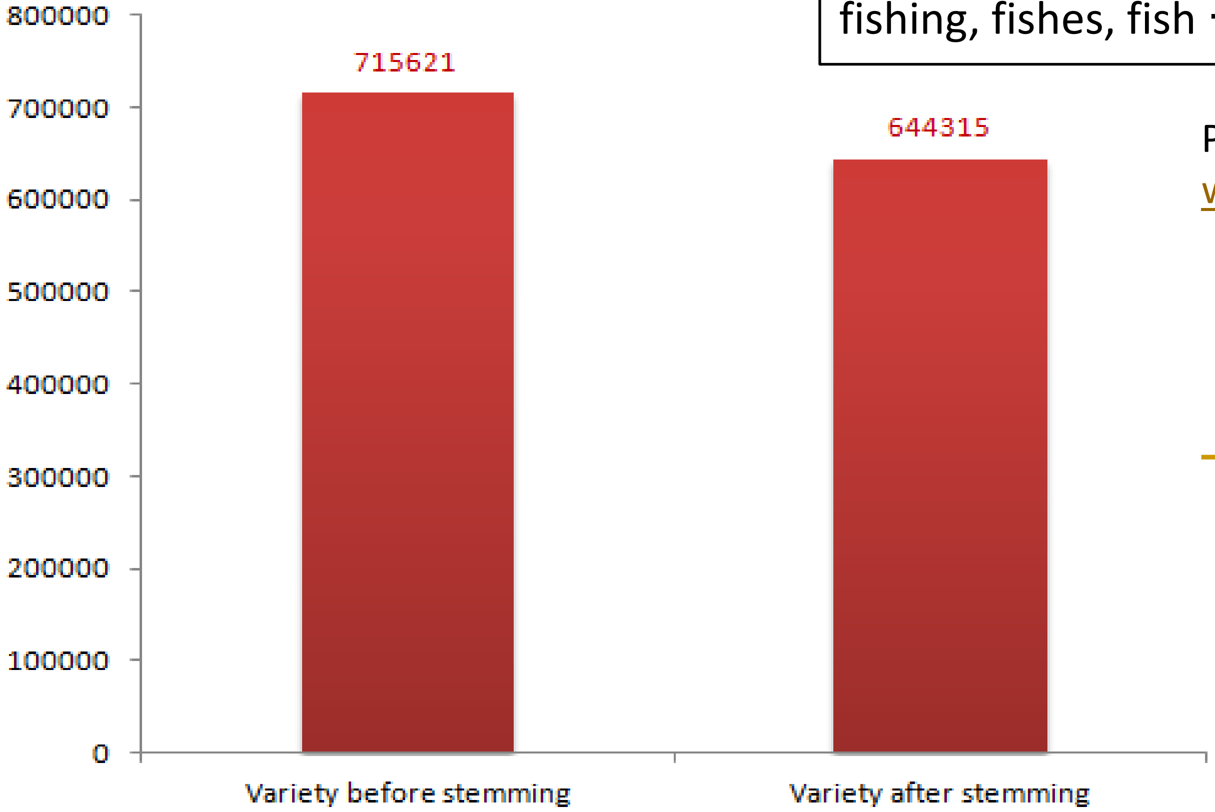
→ 56% corpus size reduction

Data reduction: Stemming

examples:

monitoring, monitored, monitor → monitor

fishing, fishes, fish → fish



Porter stemmer from WEKA
www.cs.waikato.ac.nz/ml/weka

→ 10% variety reduction

Outline

- Introduction
- Data preprocessing
- Sentiment Analysis
- Conclusions & Outlook

Classification models

- We run our experiments in the MOA (Massive Online Analysis) framework
  - Extension of WEKA for data streams (moa.cs.waikato.ac.nz)
- We experimented with several online classifiers
  - Multinomial Naive Bayes (MNB)
    - Naïve Bayes classifiers modeling word occurences
  - Adaptive Size Hoeffding Tree (ASHT)
    - Decision tree with a Hoeffding bound and of limited size
  - Ensemble of Adaptive Size Hoeffding Trees (OzaBag ASHT)
    - Ensemble of different sized ASHT
  - Stochastic Gradient Descent (SGD)
    - A linear classifier optimizing a loss function

Multinomial Naïve Bayes (MNB)

- Naïve Bayes classifiers for document classification
- Bag-of-words model
- Models word frequency in documents
- Class probability for a document

$$P(c|d) = \frac{P(c)\prod_{w \in d} P(w|c)^{n_{wd}}}{P(d)}$$

- Class conditional word probability

$$P(w|c) = \frac{1+\sum_{d \in D_c} n_{wd}}{k+\sum_{w'}\sum_{d \in D_c} n_{w'd}}$$

n_{wd}: # occurrences of word w in document d

D_c: documents of class c

k: vocabulary size

1,k: Laplace correction factors to avoid the 0-probabilities problem

Adaptive Size Hoeffding Tree (ASHT)

- Hoeffding tree
  - A decision tree for data streams
  - Leaf nodes are transformed into decision nodes by splitting on an attribute
  - Hoeffding trees exploit the fact that a small sample can often be enough to choose an optimal splitting attribute
    - Hoeffding bound: With probability 1-δ, the true mean of variable r is at least $r_{\mu}-\varepsilon$

$$\epsilon = \sqrt{\frac{R^2 ln(1/\delta)}{2n}}$$

n: # observations

r: variable

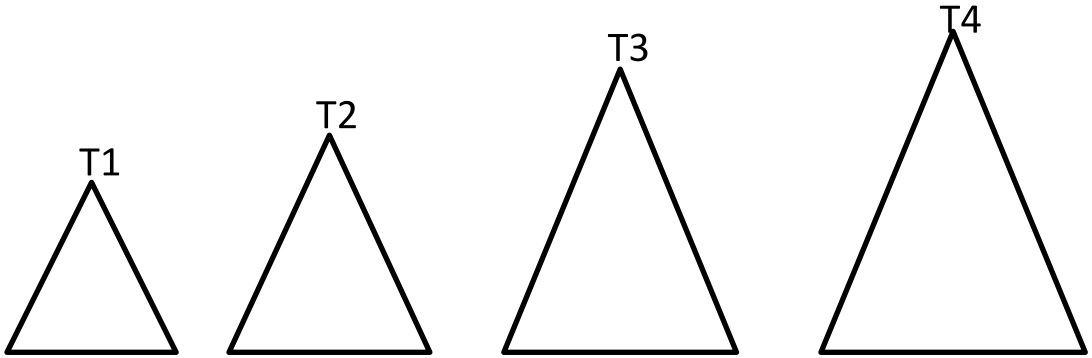
R: range of the variable

r_{μ}: computed mean of r

- Adaptive size Hoeffding tree (ASHT)
  - The tree has a maximum size (# of splitting nodes)
  - After one node splits, if the number of split nodes of the ASHT tree is higher than the maximum value, then it deletes some nodes to reduce its size

Ensemble of Adaptive Size Hoeffding Trees (OzaBag ASHT)

- Bagging using ASHTs of different sizes



  - Smaller trees adapt more quickly to changes
  - Larger trees perform better during periods with no or little change
  - The max allowed size for the nth ASHT tree is twice the max allowed size for the (n-1)th tree.
  - Each tree has a weight proportional to the inverse of the square of its error
  - The goal is to increase bagging performance by tree diversity

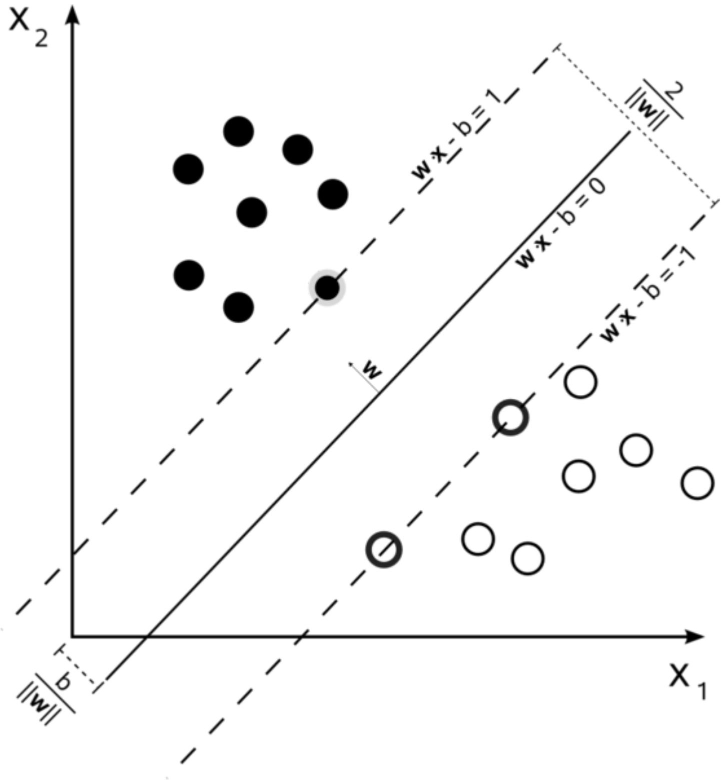
Stochastic Gradient Descent (SGD)

- An approximation to gradient descent methods commonly used in batch learning
  - In standard gradient descent, one computes the gradient of the objective loss function using all training examples, which is then used to adjust the parameter vector in the direction opposite to the gradient. The process is repeated each iteration till convergence.
  - Subgradient methods represent an approximation in which only a subset of all training examples is used in each gradient computation
  - When the size of the subsample is reduced to a single instance, we arrive at stochastic gradient descent.

Stochastic Gradient Descent (SGD)

- Efficient approach to discriminative learning of linear classifiers, minimizing an objective function that is written as a sum of differentiable functions
- Loss function of the linear classifier
  - Training error $\frac{\lambda}{2}||\mathbf{w}||^2 + \sum [1 - (y\mathbf{x}\mathbf{w} + b)]$
- A common choice to find the model parameters is by minimizing the regularized training error
- Approximation of the training error by considering a single training example at a time

$$f(x) = w^T x + b$$



Evaluation methods and measures

- Evaluation methods

  - ❑ Prequential evaluation - “test then train“
    - One dataset for training and testing
    - Models are first tested then trained in each instance

  - ❑ Holdout evaluation
    - 2 separate datasets for training and testing
    - Training set: ~70% - 80% of the dataset
    - Test set: ~20%-30% of the dataset

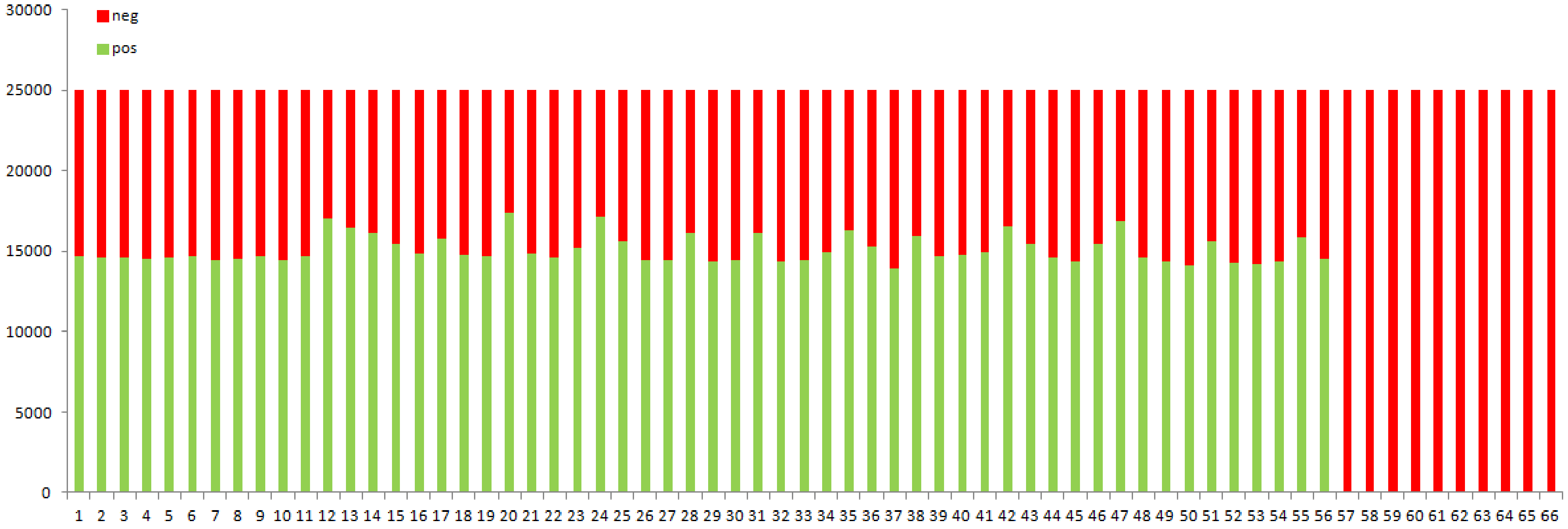
- Evaluation measures

  - ❑ Accuracy, within a sliding window
  - ❑ Kappa measure, within a sliding window
    - normalizes the accuracy of a classifier by of a chance predictor

$$k = \frac{p_0 - p_c}{1 - p_c}$$

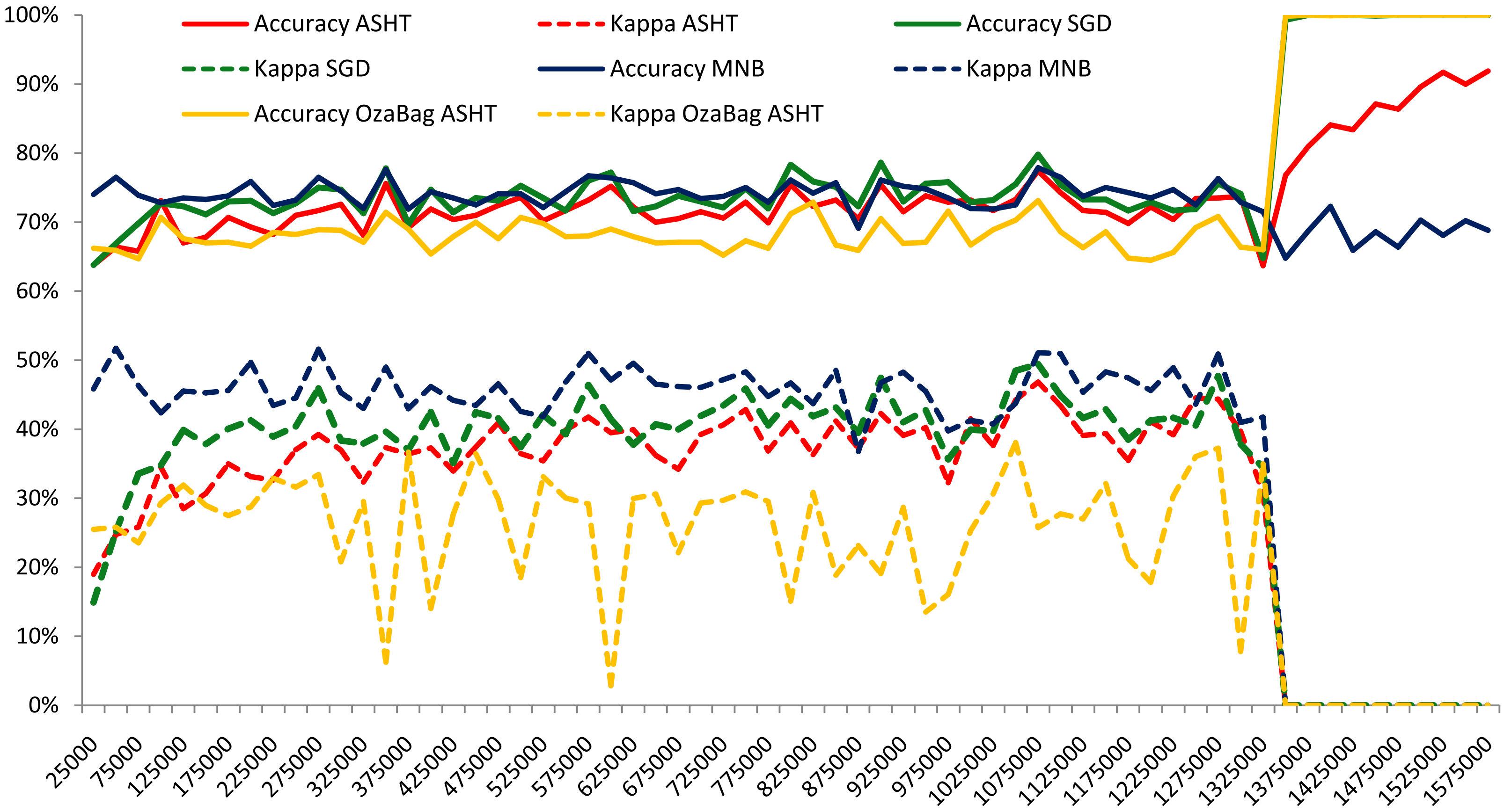
Kappa value	Classifier‘s performance
0%-20%	bad
21%-40%	fair
41%60%	moderate
61%-80%	substantial
81%-100%	(almost) perfect

Class distribution of the dataset



- 1.600.000 instances splited in 67 chunks of 25.000 tweets per chunk
- Balanced dataset (800.000 positive, 800.000 negative tweets)
- The distribution of the classes changes over time

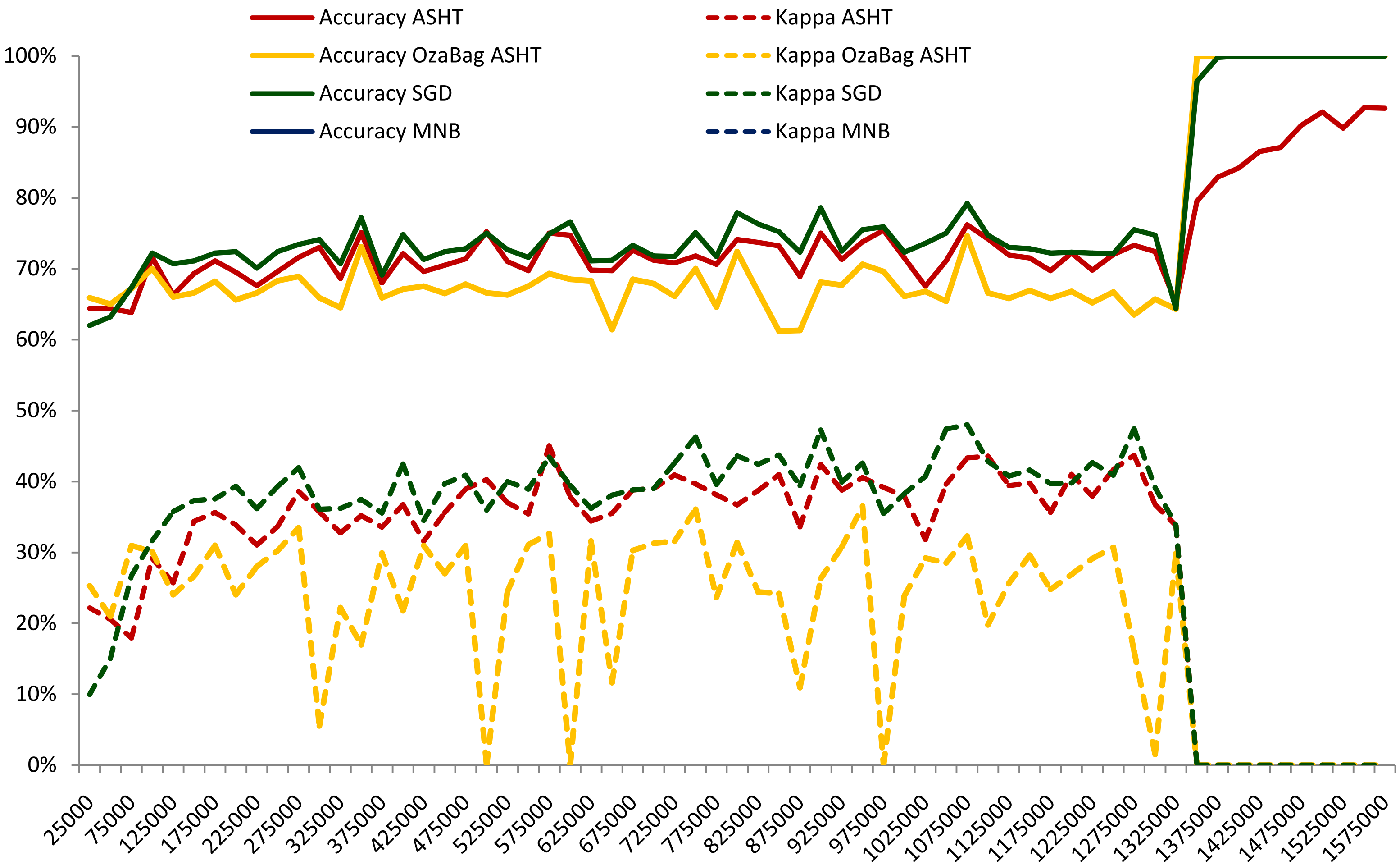
Prequential evaluation - “test then train“ results



→ MNB & SGD reach best results when the class distribution is stable

→ OzaBag ASHT & SGD can deal best with distribution changes

Holdout evaluation results



Conclusions & Outlook

- We presented a study on sentiment analysis in the Twitter stream
  - We applied several preprocessing steps and showed their effect
  - We experimented with a variety of online classifiers
    - SGD & MNB performed best on streams with a constant sentiment class distribution
    - SGD & OzaBag ASHT performed best on streams with changing class distribution
- Outlook
  - Reasoning on sentiment changes
  - Topic-specific classifiers vs generic classifiers
  - Preprocessing improvement, e.g.:
    - handle comparisons (e.g., “LMU is better than TUM”)
    - detect contextual switches (e.g. “My Pizza was so tasty! Now: work and programming, programming, programming!!! “)

Thank you for your attention!

Questions?