Classification and Categorization

Classification

Snorre Ralund, Ph.D Fellow, SoDaS, UCPH

KØBENHAVNS UNIVERSITET

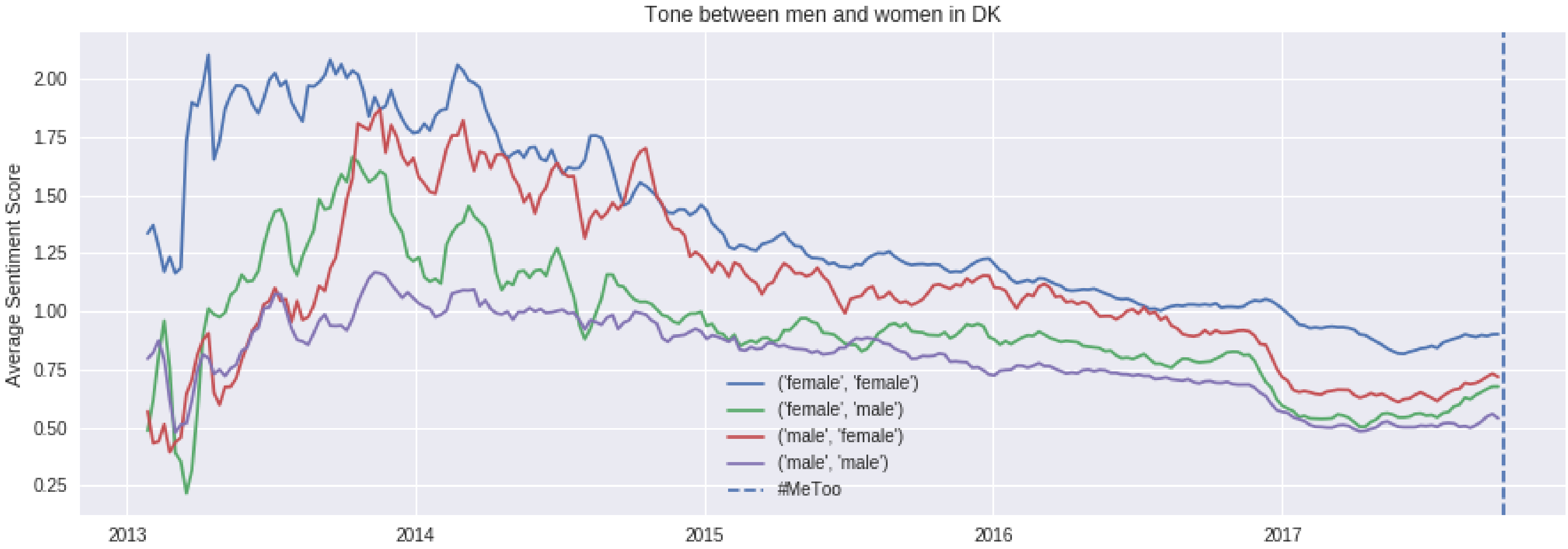
Lexical and Rulebased approaches to Text Classification

Simple Keyword/Phrase Matching rules, are used for topic classification and sentiment analysis.

Pros

- Comes with very low cost.
- Fast and scalable.
- Good for prototyping results.
- Sufficient for certain tasks (e.g. topic classification)
- **Use for Weak Supervision**

300 million documents, more 5 million unique tokens. How to inquire?



Sentiment analysis

- Purely Lexical: Naively Matching positive words.
  - "You are beautiful."
- Rule-based: Can Adopt hard-coded rules to counter more or less simple negations.
  - "You are not particularly beautiful."

Examples

- Afinn (is danish!): http://neuro.imm.dtu.dk/wiki/AFINN
- Liu Hu (lexical):http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html and
- Vader (lexical and rulebased): https://github.com/cjhutto/vaderSentiment

issues with the lexical-based approach (1)

"Pretty. Pretty actresses and actors. Pretty bad script. Pretty frequent "let's strip to our undies" scenes. Pretty fair F/X. Pretty jarring location decisions (the college dorm room looks like a high-end hotel room - probably because it was shot at a hotel). Pretty bland storyline. Pretty awful dialog. Pretty locations. Pretty annoying editing, unless you like the music video flash-cut style.This one isn't a guilty pleasure - this is more an embarrassing one. If you must watch this, pick a good dance/techno album and turn the sound off on the movie - you'll see the pretty people in their pretty black undies, and probably follow the story just fine.The cast may be able to act - I doubt that anyone could look skilled given the lines/plot that they had to deal with."

issues with the lexical-based approach (2)

- **Atomized words: How well can meaning be derived from atomized words?**
  - Not applicable to more complex rulebased versions:
    - e.g. VADER
    - E.g. Argument dictionary phrase-based. (https://mpqa.cs.pitt.edu/lexicons/arg_lexicon/)
- **What is the Recall?**
  - Bad practice using dictionaries without explicit validation.
- **Conglomerates of words = Concept?**
  - What is the theoretical validity of a collection of words, scraped from many sources, validated at some historical point in time, given some score by a number people (students, amazonturks)?

** See notebook lexical_mining.ipynb for a compilation of lexicon classifiers**


https://github.com/ulfaslak/sds_tddl_2020/blob/master/lexical_methods.ipynb

Notebook for collecting and compilling lexicons for text mining

In [ ]:
```
import pandas as pd
import numpy as np
import requests
```

Argument dictionary

Swapna Somasundaran, Josef Ruppenhofer and Janyce Wiebe (2007) Detecting Arguing and Sentiment in Meetings, SIGdial Workshop on Discourse and Dialogue, Antwerp, Belgium, September 2007 (SIGdial Workshop 2007).

In [636]:
```
import os,re
path = '/home/snorre/Dropbox/Forskning/PhD/undervisning/arglex_Somasundaran07/arglex_Somasundaran
07/'
files = [path+i for i in os.listdir(path) if 'tff' in i]
macros = ['modals.tff','spoken.tff',
'wordclasses.tff',
'pronoun.tff','intensifiers.tff']
macro2replace = {}
for macro in macros:
    filename = path+macro
    l = open(filename,'r').read().split('\n')[0:-1]
    for i in l[1:]:
        name = i.split('={')[0]
        if not '@' in name:
            continue
        words = '|'.join(i.split('={')[1].strip('}').split(','))
        macro2replace[name] = words

class2re = {}
for filename in files:
```

** See notebook lexical_mining.ipynb for a compilation of lexicon classifiers**

```
def text2argfeatures(text):
    d = {}
    for name,regex in class2re.items():
        d[name] = len(regex.findall(text))
    return d
#import codecs
string_test = codecs.open(path+'patterntest','r','utf-8').read()

import pickle
pickle.dump([class2re,string_test],open('lexicon_functions/text2arg.pkl','wb'))
class2re,string_test = pickle.load(open('lexicon_functions/text2arg.pkl','rb'))
text2argfeatures(string_test)
```

```
{'inconsistency': 18,
 'conditionals': 8,
 'contrast': 12,
 'emphasis': 30,
 'causation': 38,
 'wants': 6,
 'difficulty': 11,
 'inyourshoes': 4,
 'rhetoricalquestion': 5,
 'assessments': 24,
 'generalization': 5,
 'structure': 3,
 'necessity': 25,
 'doubt': 4,
 'priority': 8,
 'possibility': 21,
 'authority': 1}
```

```
#class="rhetoricalquestion"
do (we|you) (actually|really|still) (need|want)
why not
why don\'t (we|you)
what if
(and )?who (wouldn\'t|doesn\'t) (@EMO1V)
```

```
#class="inconsistency"
except that
except for
with the exception of
however
nevertheless
that said
that having been said
that being said
despite
in spite of
even so
at the same time
still
wait a minute
hold on a second
hold on a sec
it\'s just that
all well and good
as far as it goes
you might think (that)?
you may think (that)?
```

Lexicon and Rulebased classifiers as Noisy labels

- Rules can be more than Keyword/Phrase matching.

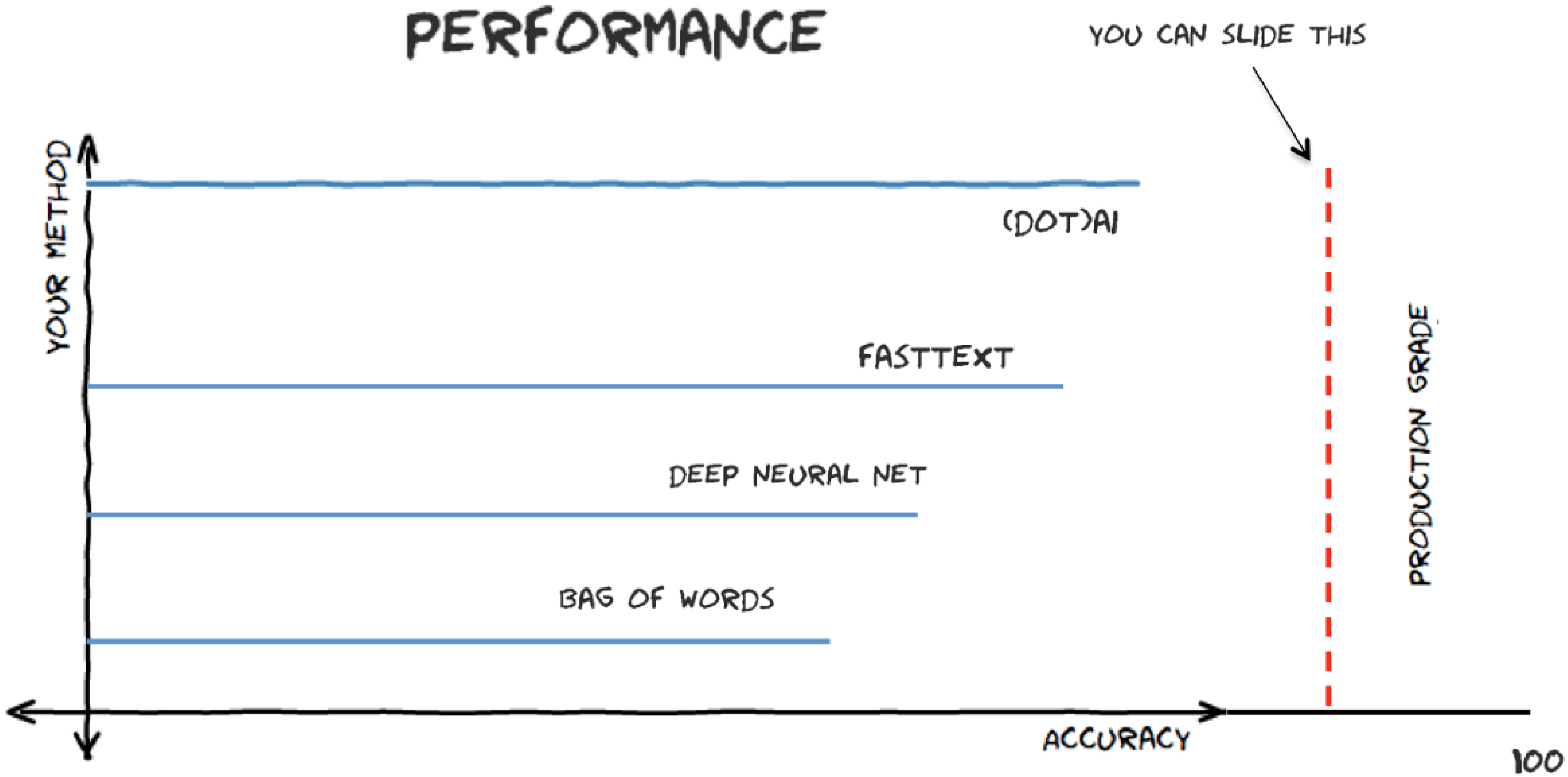
```
(?:
    ^(?:never|no|nothing|nowhere|noone|none|not|
        havent|hasnt|hadnt|cant|couldnt|shouldnt|
        wont|wouldnt|dont|doesnt|didnt|isnt|arent|aint
    )$
)
|
n't
```

- Can include NLP systems for parsing.
  - Detect negative emotion directed at women.
    - Match Negative Emotion from lexicon.
    - Use parser to find documents with woman as objects.

Lexicon and Rulebased classifiers as Noisy labels

- Use for Exploration:
  - Explore variance given many different keywords.
  - Lexicon disagreement as sources of variance.
- Use for Prototyping:
  - Initial Results to lead the research questions.
- Use as Noisy labels / Weak Supervision:
  - E.g. DeepMoji build on the idea that Emoji are labels that carry rich information.
  - Snorkel: Define several noisy classifiers evaluate them and use labels proportional to the accuracy for training a new classifier on full data.

Supervised Text Classification: Baselines



Supervised Text Classification: Baselines

Theoretical value

Setting up a good baseline for understanding progress and task difficulty.

- Baselines and bigrams.
- Comparing prediction between simple BoW models to a more Complex model allow you to understand what the model has "learned".

Practical Value

- Potential bias in more complex NLP systems and Language Models.
- Efficiency → e.g. for human-in-the-loop training i.e. Active Learning.

Active Learning for text data

- Solution to the “Rare case problem” in text data.

Procedure

1. Train a model on a random sample (or use weak supervision e.g. lexical approach).
2. Use the model prediction to sample new data to be labelled if “uncertain”.
3. Retrain model, and repeat step 2. and 3.

Supervised Text Classification: Baselines

Baselines and Bigrams

- Combine the Naive Bayes with an SVM (NBSVM)
- Use Naive Bayes as feature selector, and perform simple regularized linear regression. "NBLOG"

$$\mathbf{p} = \alpha + \sum_{i:y^{(i)}=1} \mathbf{f}^{(i)}$$

$$\mathbf{q} = \alpha + \sum_{i:y^{(i)}=-1} \mathbf{f}^{(i)}$$

$$\mathbf{r} = \log\left(\frac{\mathbf{p}/\|\mathbf{p}\|_1}{\mathbf{q}/\|\mathbf{q}\|_1}\right)$$

Where α is the laplace smoothing parameter and $\mathbf{f}^{(i)}$ is a count of feature i

Supervised Text Classification: Baselines

Baselines and Bigrams

- Combine the Naive Bayes with an SVM (NBSVM), or regularized logistic regression. "NBLOG“
- Naive Bayes as feature selector

- **Why does this ratio help?**

$$\mathbf{r} = \log\left(\frac{\mathbf{p}/\|\mathbf{p}\|_1}{\mathbf{q}/\|\mathbf{q}\|_1}\right)$$

$$\sum_{i}^{n}(y_i - \widehat{y}_i)^2 + \lambda \sum_{j}^{p} \|\beta_j\|$$

β is will be more costly for rare words, even if they distinquish the categories better.

Supervised Text Classification: Baselines

Naive Bayes Ratio as input

- How much more probable is the word "bacon" in the positive reviews, than the negative review?
- **or even better with trigrams**
- How much more probable is the trigram "not enough bacon" in the negative reviews.

Getting Baselines

- Jump to Code in baseline_classification.ipynb