Sentiment Analysis

Text Mining, Transforming Text into Knowledge

Ayoub Bagheri

Last week

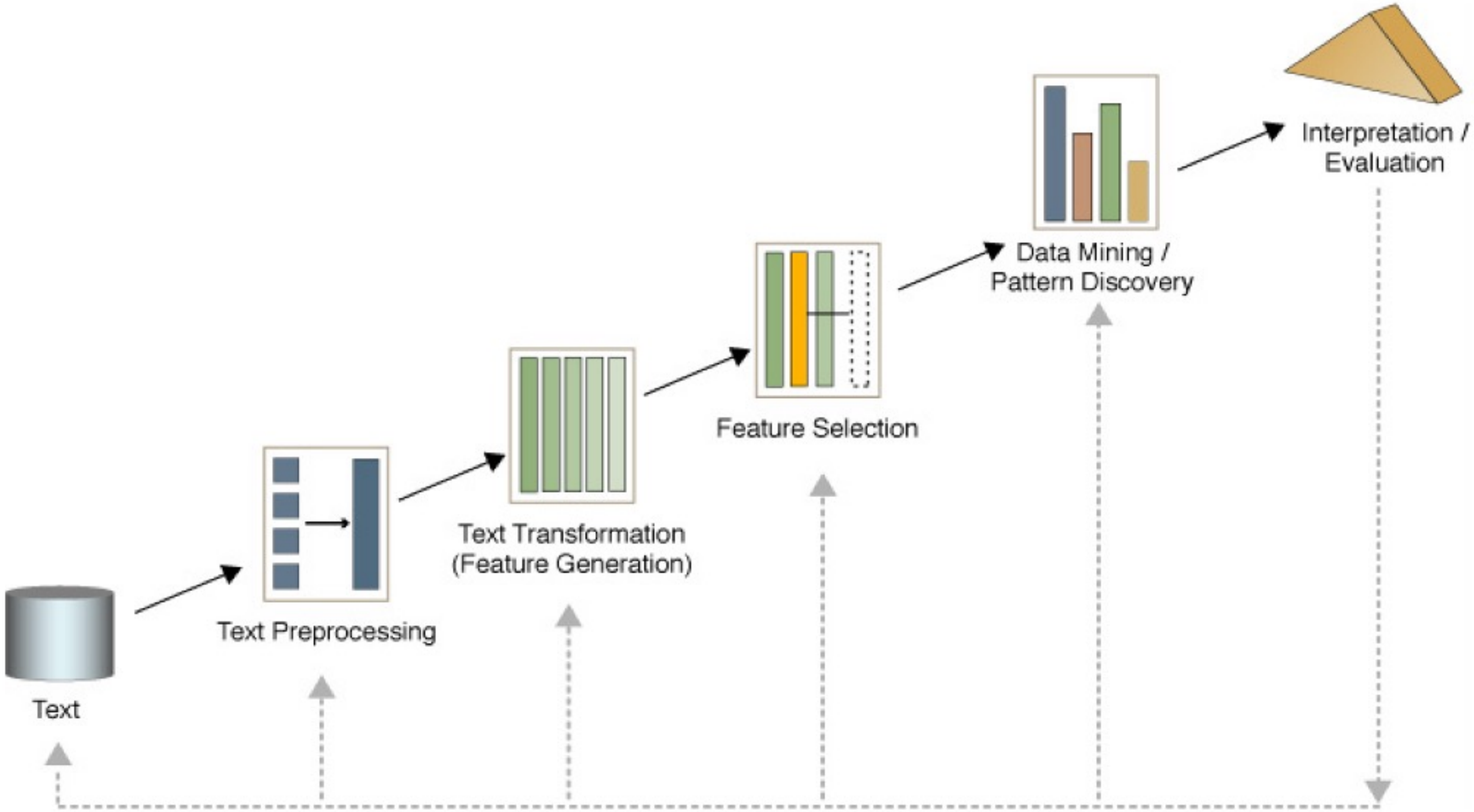
- Deep learning
- Feedforward vs Convolutional vs Recurrent neural network
- Large language models

Today

- Sentiment analysis
  - Lexicon and dictionary-based sentiment analysis
  - Machine learning-based sentiment analysis

- Sentiment classification
  - Document level
  - Sentence level

Text mining process



Text mining process

- **Data: Text**
- **Text Preprocessing:** is the process of cleaning, normalizing, and structuring raw text data into a format suitable for analysis or input into NLP models. (week 2)
- **Text transformation, feature generation:** involves converting text data into a different format or structure, such as numerical vectors or simplified forms, to make it suitable for analysis or modeling. (weeks 1, 2, 3, 6, 7, 8)
- **Feature selection**: is the process of identifying and selecting the most relevant features from a dataset to improve model performance and reduce complexity. (week 4)
- **Data mining, pattern discovery**: is the process of extracting meaningful patterns and knowledge from text. (weeks 3, 5, 7, 8, 9)
- **Interpretation / Evaluation:** is the process of understanding and explaining the model and patterns / is the assessment process to measure performance and quality. (weeks 3-9)

Sentiment Analysis

An example

This is a nice book for both young and old. It gives beautiful life lessons in a fun way. Definitely worth the money!

+ Educational

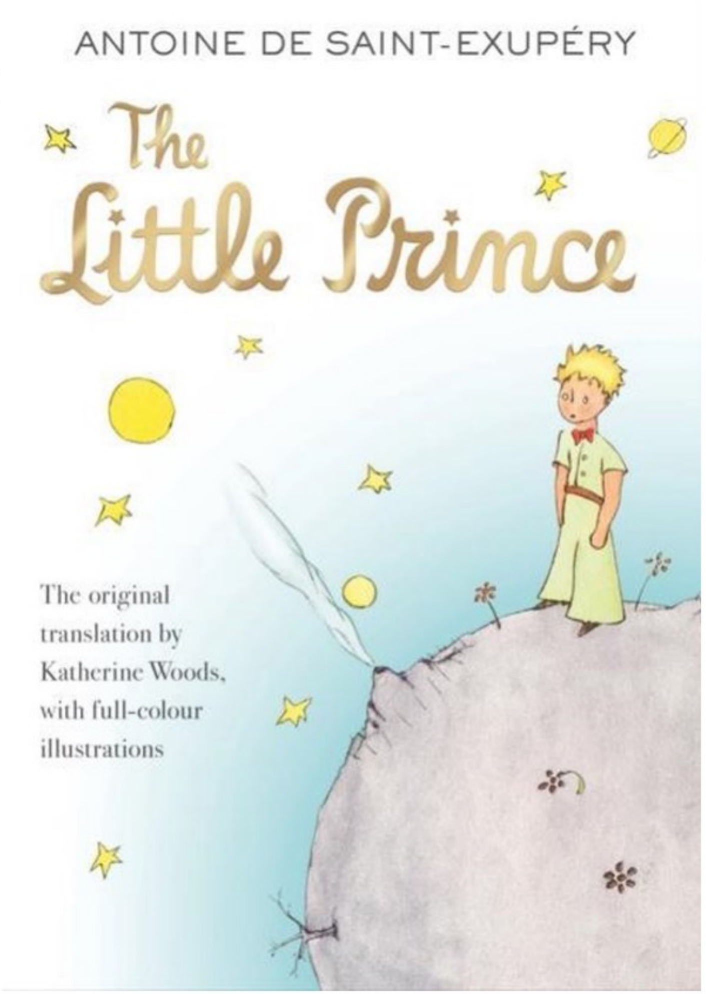
+ Fun

+ Price

Nice story for older children.

+ Funny

- Readability



Sentiment analysis

“task of classifying the polarity of a given text.”

Classify the following Google reviews of UU into

☺, 😐, or ☹:

1. “Great university and great campus”
2. “Overrated university. The facilities for the humanities studies are severely outdated and really poor quality.”
3. “Good school but hideous building”

Sentiment

- Sentiment
  - Feelings, Attitudes, Emotions, Opinions
  - A thought, view, or attitude, especially one based mainly on emotion instead of reason
  - Subjective impressions, not facts

Sentiment analysis

- Sentiment analysis = opinion mining
- Use of NLP, text mining and computational techniques to automate the extraction or classification of sentiment from text.

Practical definition of sentiment/opinion

An opinion is a quintuple

(*entity*, *aspect*, *sentiment*, *holder*, *time*)

where

- ***entity***: target entity (or object).
- ***aspect***: aspect (or feature) of the entity.
- ***sentiment***: +, -, or neu, a rating, or an emotion.
- ***holder***: opinion holder.
- ***time***: time when the opinion was expressed.

Opinion types

- Regular opinions: Sentiment/opinion expressions on some target entities
  - Direct opinions:
    - "The touch screen is really cool."
  - Indirect opinions:
    - "After taking the drug, my pain has gone."
      - Positive or negative? About what/whom?
- Comparative opinions: Comparison of more than one entity.
  - E.g., "iPhone is better than Blackberry."

Sentiment analysis tasks

- Simplest task:
  - Is the attitude of this text positive or negative?
- More complex:
  - Is the attitude of this text positive, negative or neutral?
  - Label the attitude of this text from 1 to 5
- Advanced:
  - Detect the target, source, or complex opinion types
  - Implicit opinions or aspects

Document-level

- Classify a document (e.g., a review) based on the overall sentiment of the opinion holder
  - Classes: Positive, negative (possibly neutral)
    - **Neutral** means no sentiment expressed
    - "I believe he went home yesterday."
    - "I bought an iPhone yesterday"
- An example review:
  - *"I bought an iPhone a few days ago. It is such a nice phone, although a little large. The touch screen is cool. The voice quality is great too. I simply love it!"*
  - Classification: positive or negative?
- It is basically a text classification problem

Sentence-level

- Classify the sentence expressed in a sentence
  - Classes: positive, negative (possibly neutral)
- But bear in mind
  - Explicit opinion: “I like this car.”
  - Fact-implied opinion: “I bought this car yesterday and it broke today.”
  - **Mixed opinion**: “Apple is doing well in this poor economy”

Methods for sentiment analysis

- **Lexicon-based methods**
  - Dictionary based: Using sentiment words and phrases (e.g., good, wonderful, awesome, troublesome, cost an arm and leg)
  - Corpus-based: Using co-occurrence statistics or syntactic patterns embedded in text corpora
- **Supervised learning methods**: to classify reviews into positive and negative.
  - Traditional Machine Learning: Naïve Bayes, Support Vector Machine
  - Deep learning: BERT, GPT

Lexicon-based Methods

Lexicon-based sentiment analysis

- Old-school sentiment analysis
- **Algorithm.** Start with a list of "positive" words and "negative" words, the "*lexicon*". Then count them.

  Sentiment = Total no. positive words − Total no. negative words.

- Popular lexicons are: LIWC, FINN, bing, NRC, …
- Tidytext has AFINN, bing, and nrc
- There are also domain-specific sentiment lexicons, and lexicons for languages that are not English

Basic algorithm

- Detect sentiment in two independent dimensions:
  - Positive: {1, 2,… 5}
  - Negative: {-5, -4,… -1}
- Example: “He is brilliant but boring”
  - Overall sentiment = ?

Basic algorithm

- Detect sentiment in two independent dimensions:
    - Positive: {1, 2,… 5}
    - Negative: {-5, -4,… -1}
- Example: “He is brilliant but boring”
    - Sentiment(‘brilliant’) = +4
    - Sentiment(‘boring’) = -2
    - Overall sentiment = +2

Lexicon-based sentiment analysis

`AFINN` lexicon (Finn Årup Nielsen):

- assigns words with a score that runs between -5 and 5, with negative scores indicating negative sentiment and positive scores indicating positive sentiment
- terms manually labelled for valence by Finn Årup Nielsen between 2009 and 2011.
- Specifically created for sentiment analysis of microblogs such as Twitter

```r
get_sentiments("afinn")
```

```
## # A tibble: 2,477 x 2
##    word       value
##    <chr>      <dbl>
##  1 abandon       -2
##  2 abandoned     -2
##  3 abandons      -2
##  4 abducted      -2
##  5 abduction     -2
##  6 abductions    -2
##  7 abhor         -3
##  8 abhorred      -3
##  9 abhorrent     -3
## 10 abhors        -3
## # ... with 2,467 more rows
```

Lexicon-based sentiment analysis

`bing` lexicon (Bing Liu and collaborators):

- categorizes words into positive and negative categories
- Developed for mining and summarizing customer reviews
- First, adjective words were identified using a natural language processing method. Second, for each opinion word, semantic orientation was determined

```
## # A tibble: 6,786 x 2
##    word        sentiment
##    <chr>       <chr>
##  1 2-faces     negative
##  2 abnormal    negative
##  3 abolish     negative
##  4 abominable  negative
##  5 abominably  negative
##  6 abominate   negative
##  7 abomination negative
##  8 abort       negative
##  9 aborted     negative
## 10 aborts      negative
## # ... with 6,776 more rows
```

Lexicon-based sentiment analysis

`nrc` lexicon (Saif Mohammad and Peter Turney):

- categorizes words into categories of positive, negative, anger, anticipation, disgust, fear, joy, sadness, surprise, and trust
- annotations were manually done by crowdsourcing

```
## # A tibble: 13,901 x 2
##    word        sentiment
##    <chr>       <chr>
##  1 abacus      trust
##  2 abandon     fear
##  3 abandon     negative
##  4 abandon     sadness
##  5 abandoned   anger
##  6 abandoned   fear
##  7 abandoned   negative
##  8 abandoned   sadness
##  9 abandonment anger
## 10 abandonment fear
## # ... with 13,891 more rows
```

Corpus-based sentiment analysis

- PMI between two words:
  - How much more do two words co-occur than if they were independent?

$$PMI(word_1, word_2) = log_2 \frac{P(word_1, word_2)}{P(word_1)P(word_2)}$$

Corpus-based sentiment analysis

- P(word) estimated by `hits(word)/N`
- P(word1,word2) by `hits(word1 NEAR word2)/N^2`

$$PMI(word_1, word_2) = log_2 \frac{hits(word_1 \text{ NEAR } word_2)}{hits(word_1)hits(word_2)}$$

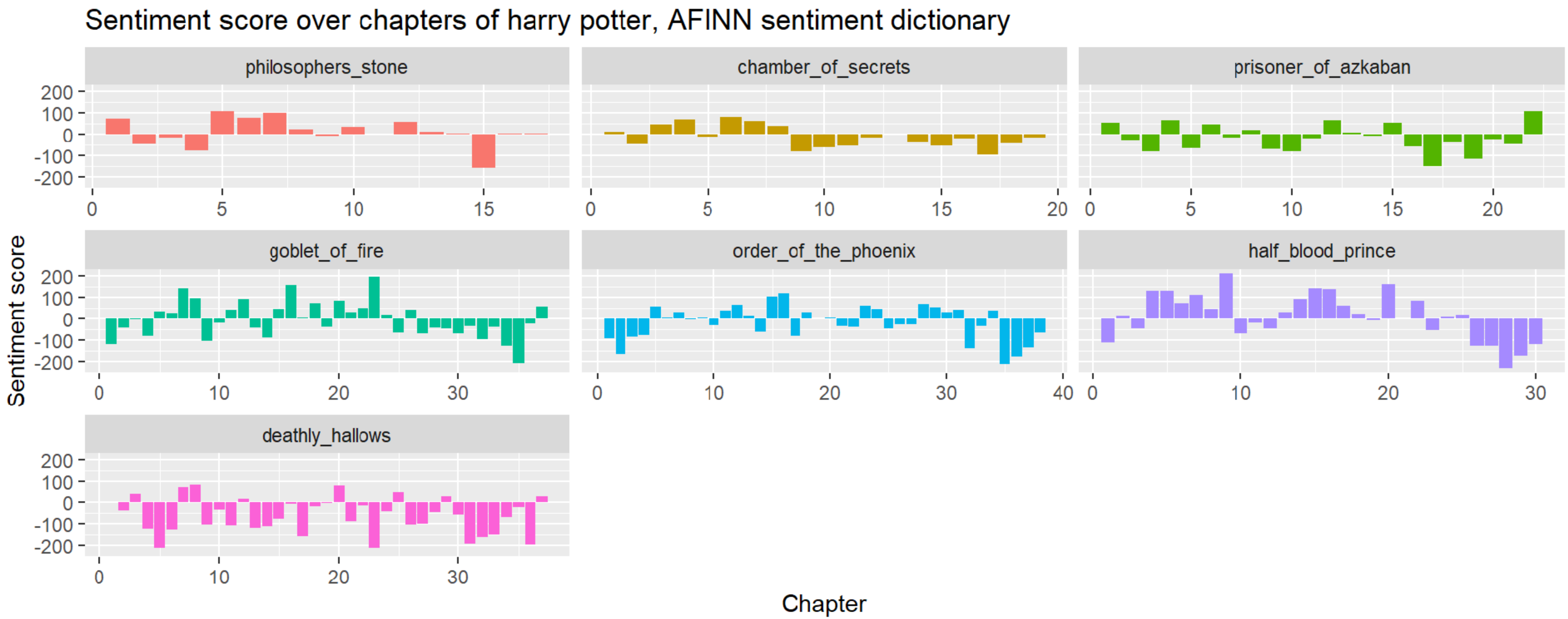
Example using NRC

Most common joy words in Harry Potter

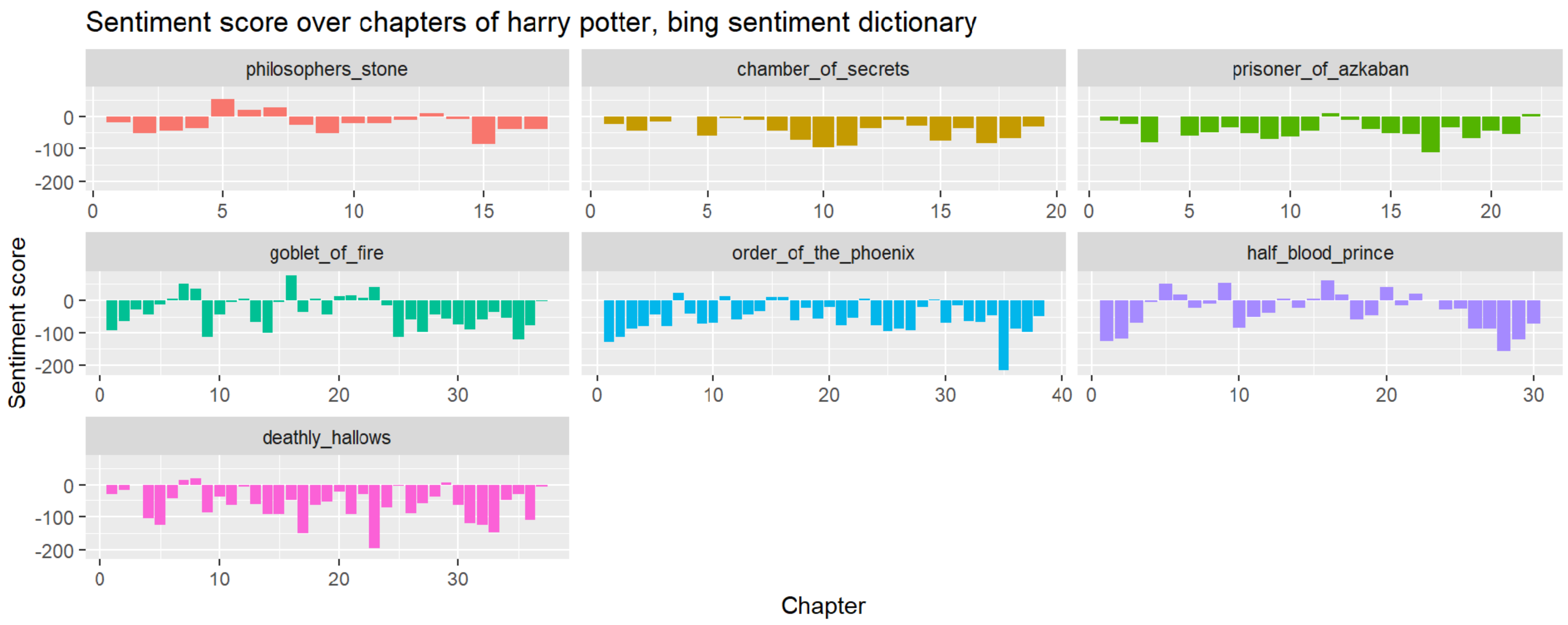
```
## # A tibble: 440 x 2
##    word         n
##    <chr>    <int>
##  1 good      1065
##  2 found      614
##  3 ministry   576
##  4 feeling    391
##  5 magical    380
##  6 white      331
##  7 green      294
##  8 mother     284
##  9 smile      244
## 10 hope       234
## # ... with 430 more rows
```

Most common fear words in Harry Potter

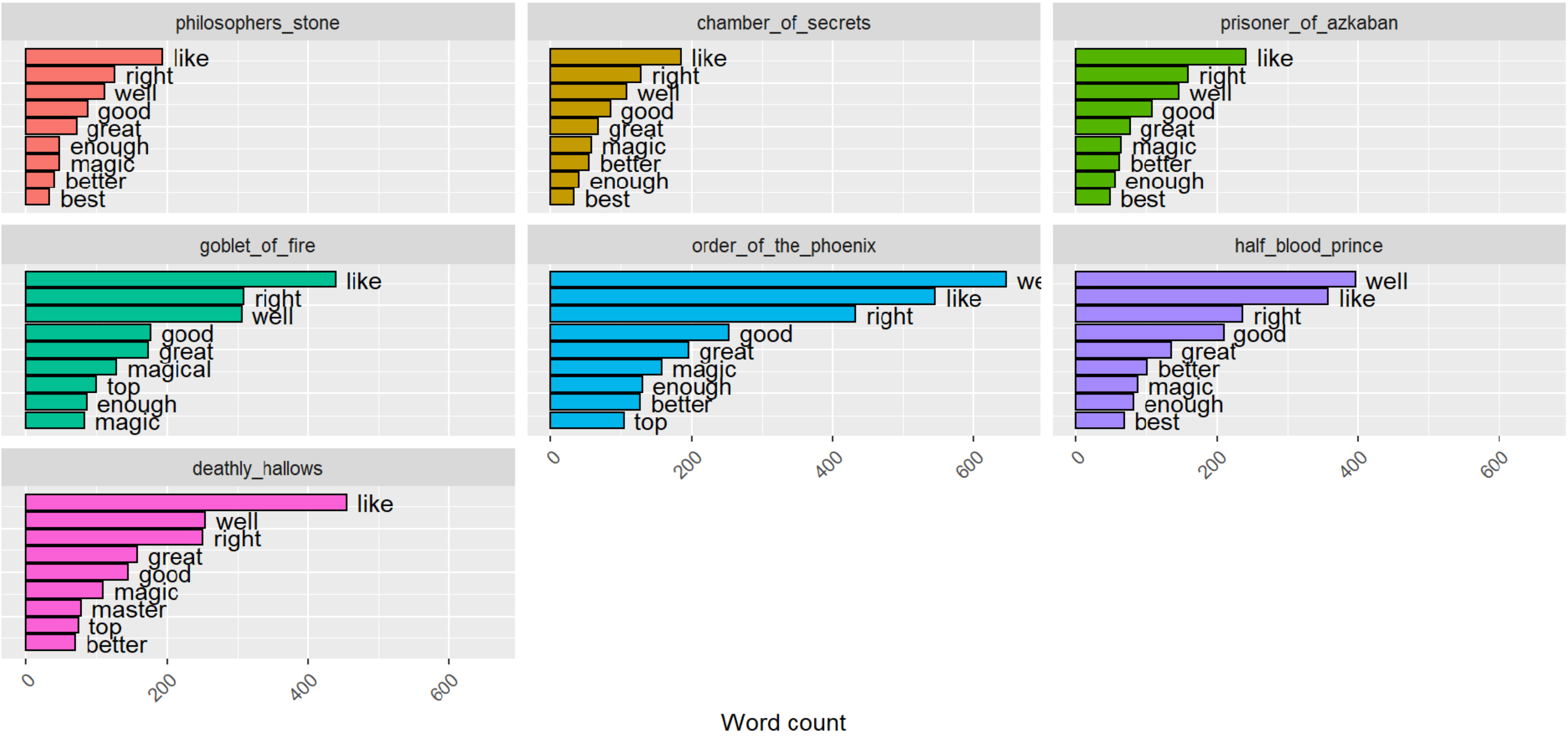
```
## # A tibble: 888 x 2
##    word         n
##    <chr>    <int>
##  1 death      757
##  2 feeling    391
##  3 fire       388
##  4 crouch     297
##  5 shaking    277
##  6 scar       276
##  7 mad        269
##  8 kill       267
##  9 elf        259
## 10 watch      256
## # ... with 878 more rows
```

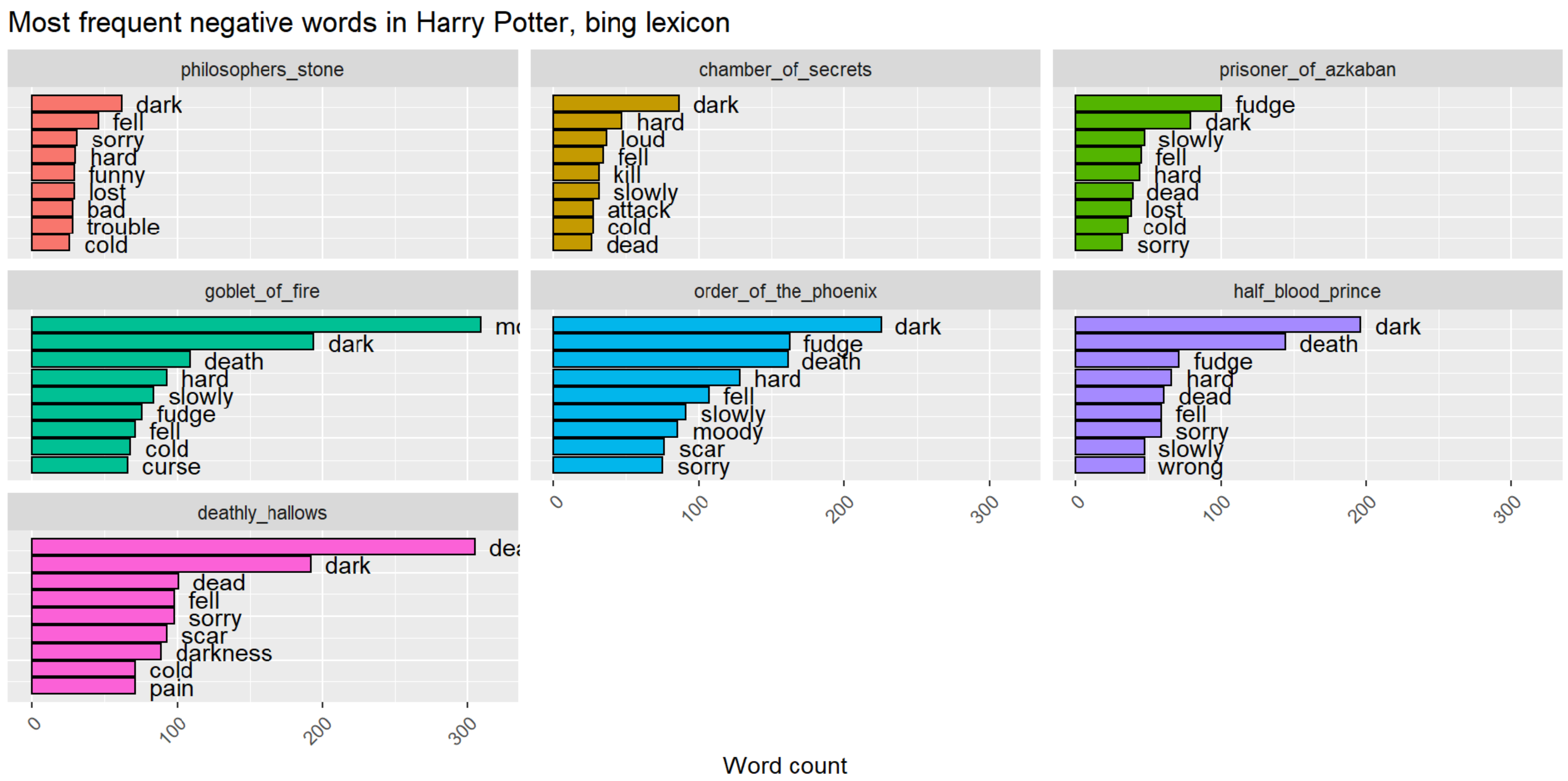


Plot of novel four to six changes towards a negative sentiment towards the end, while the seventh novel has a quite negative sentiment overall.


Sentiment score over chapters of harry potter, bing sentiment dictionary
philosophers_stone
chamber_of_secrets
prisoner_of_azkaban
goblet_of_fire
order_of_the_phoenix
half_blood_prince
deathly_hallows
Sentiment score
Chapter
0
-100
-200

Most frequent positive words in Harry Potter, bing lexicon

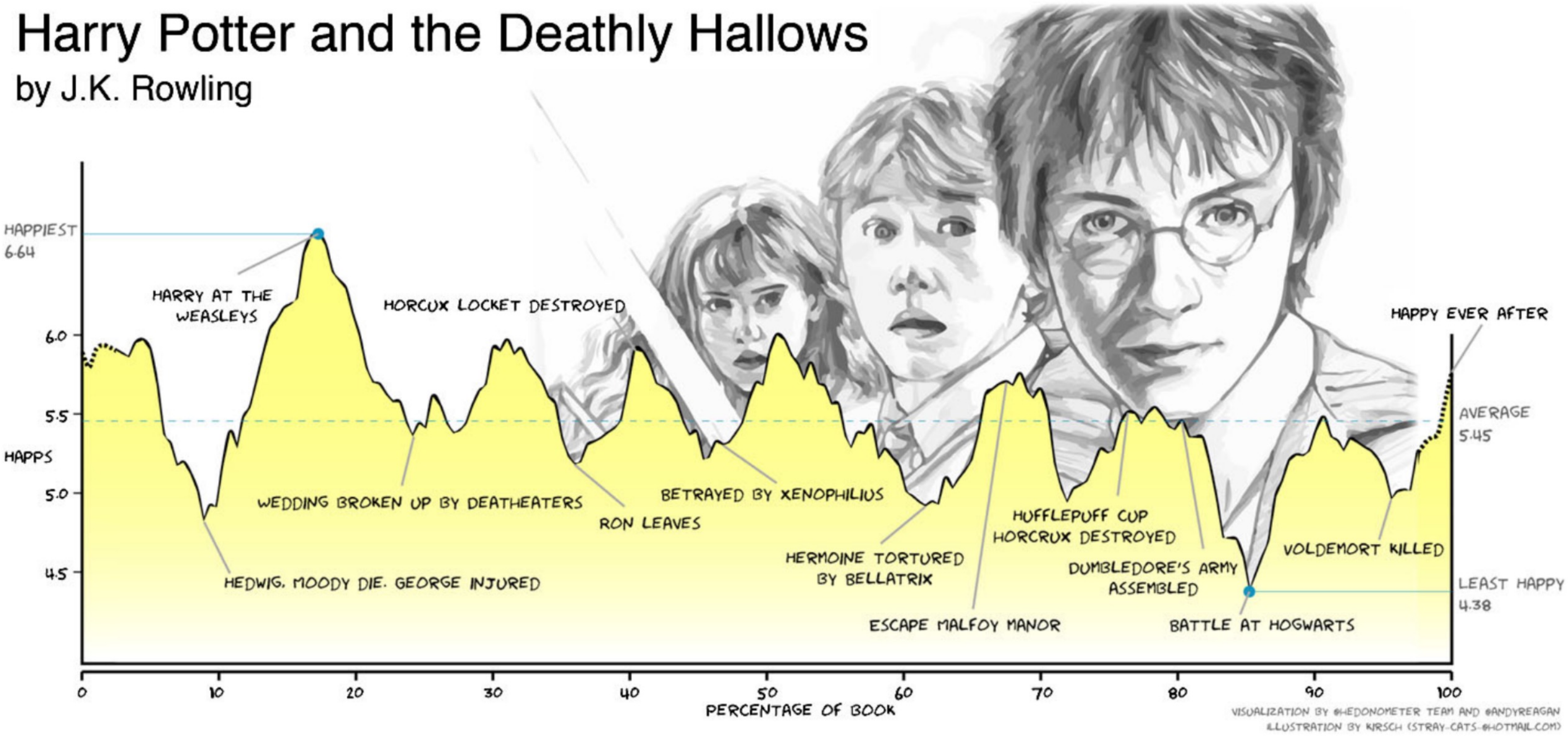




Most frequent negative words in Harry Potter, bing lexicon
philosophers_stone
dark
fell
sorry
hard
funny
lost
bad
trouble
cold
chamber_of_secrets
dark
hard
loud
fell
kill
slowly
attack
cold
dead
prisoner_of_azkaban
fudge
dark
slowly
fell
hard
dead
lost
cold
sorry
goblet_of_fire
dark
death
hard
slowly
fudge
fell
cold
curse
order_of_the_phoenix
dark
fudge
death
hard
fell
slowly
moody
scar
sorry
half_blood_prince
dark
death
fudge
hard
dead
fell
sorry
slowly
wrong
deathly_hallows
dark
dead
fell
sorry
scar
darkness
cold
pain
0
100
200
300
Word count

Harry Potter and the Deathly Hallows

by J.K. Rowling



Reagan et al. (2016). The emotional arcs of stories are dominated by six basic shapes.
http://doi.org/10.1140/epjds/s13688-016-0093-1

Lexicon-based sentiment analysis

- The “old-school” (lexicon-based) method is not great with:
  - Longer texts (why?)
  - Negation
  - Context-dependency in general
- You can also just consider this a **classification task**, where the input data is the text and the target is categorical

Break

Supervised Methods

Basic steps

- Pre-processing and tokenization
- Feature representation
- Feature selection
- Classification
- Evaluation

Sentiment tokenization issues

- Deal with HTML and XML markup
- Twitter mark-up (names, hash tags)
- Capitalization (preserve for words in all caps)
- Phone numbers, dates
- Emoticons
- Useful code:
  - Christopher Potts sentiment tokenizer
  - Brendan O’Connor twitter tokenizer

The danger of stemming

- The Porter stemmer identifies word suffixes and strips them off.
- But:
  - objective (pos) and objection (neg) -> object
  - competence (pos) and compete (neg) -> compet

Features for supervised methods

- The problem has been studied by numerous researchers.
- **Key**: feature engineering. A large set of features have been tried by researchers. E.g.,
  - Terms frequency and different IR weighting schemes
  - Part of speech (POS) tags
  - Opinion words and phrases
  - Negations
  - Stylistic
  - Syntactic dependency

Negation

- Add NOT_ to every word between negation and following punctuation:

```
didn't like this movie , but I

didn't NOT_like NOT_this NOT_movie but I
```

Example

Kiritchenko et al. (2014)

- Features:
  - ngrams
  - character ngrams
  - all-caps: the number of tokens with all characters in upper case
  - POS
  - the number of negated contexts
  - sentiment lexicons
  - the number of hashtags, punctuation, emoticons, elongated words
- Classifier: linear-kernel SVM

Example

Kiritchenko et al. (2014)

System	Accuracy
a. Majority baseline	50.1
b. SVM-unigrams	71.9
c. Previous best result (Socher et al., 2013)	85.4
d. Our system	**85.5**

: Message-level task: The results obtained on the movie review excerpts dataset.

https://www.svkir.com/papers/Kiritchenko-et-al-sentiment-JAIR-2014.pdf

Supervised methods

- Advantages
  - Lead to better performance compared to lexicon-based approaches
  - The output can be explained (most of the times)
- Disadvantages
  - They need training data
  - They can’t capture the context, need more advanced methods; see next slide
  - Based on feature engineering that is a tedious task
  - Not good performance in multiclass classification

Sentiment analysis with LLM

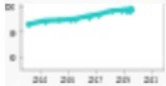
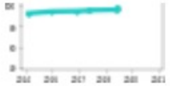
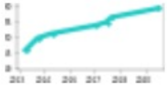
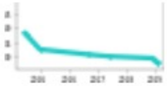
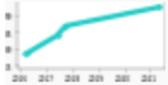
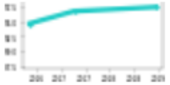
Devlin et al. 2019

- Sentiment analysis was one of the tasks in the BERT paper

System	MNLI-(m/mm)	QQP	QNLI	SST-2
	392k	363k	108k	67k
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4
OpenAI GPT	82.1/81.4	70.3	87.4	91.3
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**

https://paperswithcode.com/task/sentiment-analysis

Benchmarks

Trend	Dataset	Best Model
	SST-2 Binary classification	SMART-RoBERTa Large
	IMDb	NB-weighted-BON + dv-cosine
	SST-5 Fine-grained classification	RoBERTa-large+Self-Explaining
	Yelp Binary classification	XLNet
	Yelp Fine-grained classification	XLNet
	Yelp Fine-grained classification	XLNet
	MR	EFL
	Amazon Review Polarity	BERT large
	Amazon Review Full	BERT large
	User and product information	BiLSTM+CHIM
	CR	EFL

Show all 32 benchmarks

Pre-trained models

https://huggingface.co/blog/sentiment-analysis-python

- Twitter-roberta-base-sentiment is a roBERTa model trained on ~58M tweets and fine-tuned for sentiment analysis (https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment)
- SST-2 BERT: Fine-tuned on the Stanford Sentiment Treebank (SST-2) which consists of sentences from movie reviews. The model is well-suited for general sentiment analysis tasks. (https://huggingface.co/distilbert-base-uncased-finetuned-sst-2-english)
- Bert-base-multilingual-uncased-sentiment is a model fine-tuned for sentiment analysis on product reviews in six languages: English, Dutch, German, French, Spanish and Italian (https://huggingface.co/nlptown/bert-base-multilingual-uncased-sentiment)
- Distilbert-base-uncased-emotion is a model fine-tuned for detecting emotions in texts, including sadness, joy, love, anger, fear and surprise (https://huggingface.co/bhadresh-savani/distilbert-base-uncased-emotion)

Conclusion

Conclusion

- Sentiment analysis
  - Document-level versus sentence level sentiment classification
- Lexicon-based method is not great
- Supervised learning methods improve old school ones
  - Could use BoW and TF-IDF, sometimes better
  - Sometimes BoW similar problems as lexicon-based method
  - Big disadvantage is that you will need (partly) labeled data
- LLMs and deep learning-based methods perform better

Practical
IMDB sentiment classification

Remarks

- **Next week**: Responsible Text Mining & Applications

- Notes about the exam:
  - https://textminingcourse.nl/exams/FinalExam.html

Questions?

a.bagheri@uu.nl