1: Sentiment Classification

Machine Learning and Real-world Data

Simone Teufel and Ann Copestake

Computer Laboratory
University of Cambridge

Lent 2017

This course

- Machine Learning and Real-world Data
- Three Topics:
  - Sentiment classification – thousands of movie reviews
  - Protein sequence analysis – hundreds of amino acid sequences
  - Social network analysis – thousands of users and links between them
- Practical-based, each with a short lecture introducing the main concepts
- 16 sessions, 12 tasks, 8 ticks

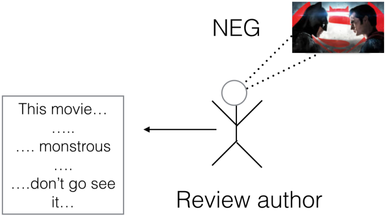
Computer Science as an empirical subject

- The style of solving tasks in this course is *empirical*.
- You will start from a hypothesis or an idea which you will test
- Then you perform some manipulations on your data
- You observe and record the results
- You need a lab book to record your manipulations, observations and measurements
  - physical book or electronic record

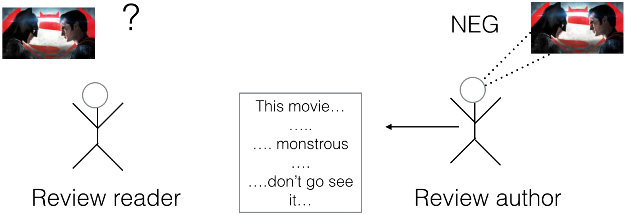
Today: Evaluative language and sentiment classification

- IMDB (= Internet Movie Data Base) has 4 million titles (2015)
- Reviews are written in natural language by the general public
- Sentiment classification = the task of automatically deciding whether a review is good or bad, based on the text of the review
- Standard task in Natural Language Processing

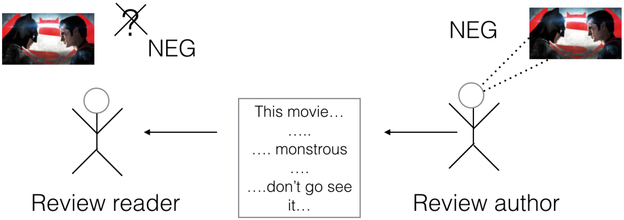
Review sentiment



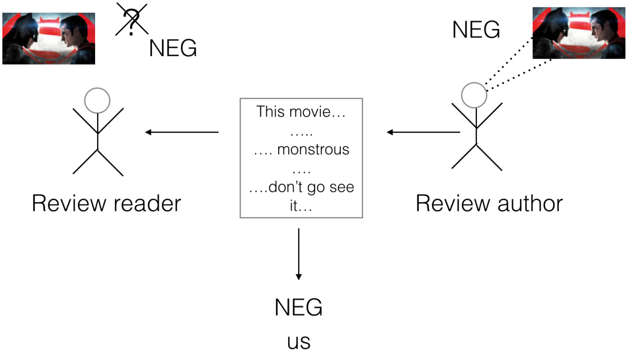
Review sentiment



Review sentiment



Review sentiment



From a good review

... He’s incredible in fights. ... Also his relationship with Irons, who plays Alfred, is just wonderful in general. Irons was exceptional in the role.

A bad review

This movie tries so hard... It completely fails on every single level. The movie is tedious and boring with characters that I just did not care about at all. ...

Today's idea – use hand-selected individual words

- Sentiment Lexicon lists 8222 such words
- Idea: a review that contains more positive than negative such words is positive

```
type=strongsubj len=1 word1=laudably pos1=anypos stemmed1=n priorpolarity=positive
type=strongsubj len=1 word1=laugh pos1=noun stemmed1=n priorpolarity=negative
type=strongsubj len=1 word1=laugh pos1=verb stemmed1=y priorpolarity=negative
type=strongsubj len=1 word1=laughable pos1=anypos stemmed1=n priorpolarity=negative
type=strongsubj len=1 word1=laughably pos1=anypos stemmed1=n priorpolarity=negative
type=strongsubj len=1 word1=laughingstock pos1=noun stemmed1=n priorpolarity=negative
type=strongsubj len=1 word1=laughter pos1=noun stemmed1=n priorpolarity=negative
type=strongsubj len=1 word1=lavish pos1=adj stemmed1=n priorpolarity=positive
type=strongsubj len=1 word1=lavish pos1=verb stemmed1=y priorpolarity=positive
```

Sentiment lexicon words in the good review

... He's incredible in fights. ... Also his relationship with Irons, who plays Alfred, is just wonderful in general. Irons was exceptional in the role.

- incredible positive
- wonderful positive
- exceptional positive

Sentiment lexicon words in the bad review

This movie tries so hard... It completely fails on every single level. The movie is tedious and boring with characters that I just did not care about at all. ...

- try negative
- fail negative
- tedious negative
- boring negative
- care positive

It's not a given that it will work

This movie tries so hard... The ending should be exciting and fun and amazing.. and it just... wasn't. It completely fails on every single level. The movie is tedious and boring with characters that I just did not care about at all. ...

- try negative
- exciting positive
- fun positive
- amazing positive
- fail negative
- tedious negative
- boring negative
- care positive

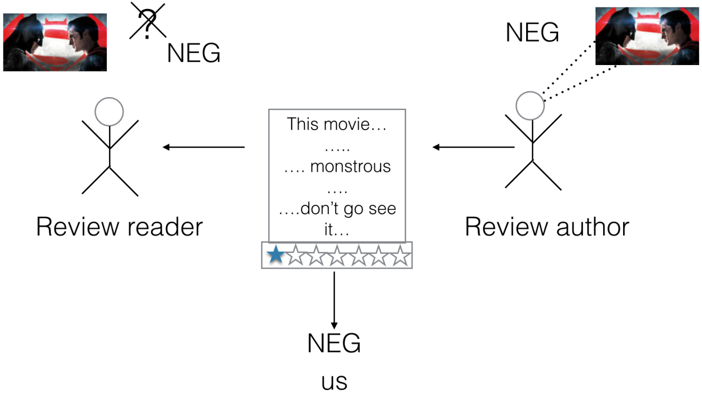
Tokenisation

- You will look up words from your review document in the lexicon.
- Relationship between strings in the review document and items in the sentiment lexicon is not 1:1
- When words are put together to form a text, some typographic transformations occur:
  - For instance, words at the beginning of a sentence appear in upper case.
  - Words occurring before and after punctuation may be directly attached to the punctuation.
- Therefore, splitting on whitespace is not enough.
- Your code will use a well-known tokeniser to undo the most important transformations.

Evaluation

- How do we know when we have got it right?
- (And I don't mean testing your programs for bugs. . .)
- The author of the review told us the truth:
  - Explicitly
  - Numerically
  - Star rating
- This is lucky for us, but it's not always so
- We have harvested the star rating along with the review text
- We will use it to calculate a metric called A (accuracy).

Star rating


NEG
NEG
This movie…
…..
…. monstrous
….
….don't go see it…
Review reader
Review author
NEG
us

Accuracy

- Success can be measured in the number of correct decisions c over all decisions (correct plus incorrect (i)):

$$A = \frac{c}{c+i}$$

- This metric is called A (accuracy).
- We know which decisions are "correct" because we can use the star rating as our definition of truth.

Your tasks for today

Task 1:

- explore the review data (200 documents)
- explore the sentiment lexicon
- write a program that puts the above idea to test
- write a program for using the star ratings to evaluate how well your program is doing

Practicalities

- 16 lectures (approx 20 minutes) [M, F]
- 16 demonstrated sessions in the Intel Lab: from immediately after lecture to 4:30pm [M, F]
- 12 tasks, 8 ticks
- Ticking during demonstrated sessions (normally the session after work on tick completed)
- Catch-up sessions