Natural Language Processing 1

Katia Shutova and Wilker Aziz

ILLC
University of Amsterdam

29 October 2024

Lecture 1: Introduction

Taught by...

Katia Shutova
Lecturer

Wilker Aziz
Lecturer

Vera Neplenbroek
Lab coordinator

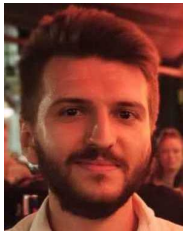
Pieter Pierrot
Senior TA

Teaching assistants


Charlotte Pouw


Mina Janicijevic


Ivo Verhoeven


John Gkountouras


Wafaa Mohammed


Seth Aycock


Adrian Sauter


Milan Miletic


Baohao Liao


Pedro Ferreira


Evgenia Ilia

Overview of the course

- Introduction and broad overview of NLP
- Different levels of language analysis (word, sentence, larger text fragments)
- A range of NLP tasks and applications
- Both fundamental and most recent methods:
  - rule-based
  - statistical
  - deep learning
- Other NLP courses go into much greater depth

Assessment

1. Practical assignments (**50%**)
   - Work in groups of 2
   - Implement several language processing methods
   - Evaluate in the context of a real-world NLP application — sentiment classification
   - Assessed by two reports (20% + 30%)
     - Practical 1: Mid-term report, deadline **13 November**
     - Practical 2: Final report, deadline **13 December**
2. Pen-and-paper exercises (individual work) (**20%**)
   - throughout the course
   - feedback from TAs
3. Exam on 17 December (**30%**)
4. Resit on 12 February (**100%**)

Also note:

Course materials and more info:
`https://cl-illc.github.io/nlp1-2024/`

Contact

- Main contact – your TA (email on the website)
- Practicals – Vera: `v.e.neplenbroek@uva.nl`
- Organisational – Pieter: `p.j.pierrot@uva.nl`

Subject line should have **NLP1-24**

Sign up to groups by Fri, 1 November by emailing your TA

- names of the students
- their email addresses

Course Materials

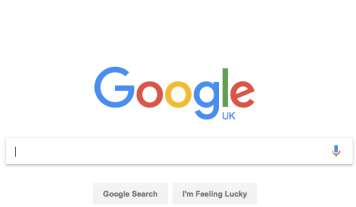
- Slides, further reading, assignments posted on the **website**
- but... assignment submission will be via **Canvas**.
- **Ed** platform for questions and discussion: `https://edstem.org/eu/courses/1755/discussion/`
- **Book:** Jurafsky & Martin, *Speech and Language Processing (2nd edition)*

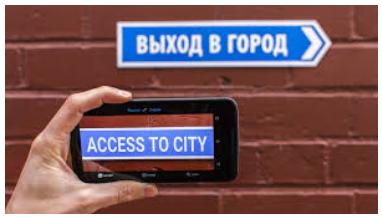
  3 edition (unofficial) at `https://web.stanford.edu/~jurafsky/slp3/`

What is NLP?

NLP: the computational modelling of human language.

Many popular applications





...and the emerging ones

Machine Translation

- Translate from one language into another
- Earliest attempted NLP application
- Early systems based on transfer rules, then statistical and now neural MT
- High quality with typologically close languages: e.g. Swedish-Danish.
- More challenging with typologically distant languages and low-resource languages

Retrieving information

- **Information retrieval**: return documents in response to a user query (Internet Search is a special case)
- **Information extraction**: discover specific information from a set of documents (e.g. companies and their founders)
- **Question answering**: answer a specific user question by returning a section of a document:

  What is the capital of France?
  Paris has been the French capital for many centuries.

Opinion mining and sentiment analysis

- Finding out what people think about politicians, products, companies etc.
- Typically done on web documents and social media
- More about this later today

More recent applications

Automated fact checking

- classify statements and news articles as factual or not
- in an effort to combat misinformation

Abusive language detection

- automated detection and moderation of online abuse
- hate speech, racism, sexism, personal attacks, cyberbullying etc.

Other areas in which NLP is relevant

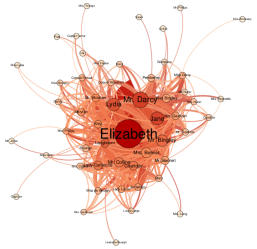
NLP and computer vision

- Caption generation for images and videos

The dog chewed at the shoes

Digital humanities

- e.g. social network in *Pride and Prejudice*



Computational social science

- analyse human behaviour based on language use (deeper than sentiment)

NLP and linguistics

1. Morphology — the structure of words: lecture 3.
2. Syntax — the way words are used to form phrases: lectures 2, 3 and 4.
3. Semantics
   - Lexical semantics — the meaning of individual words: lecture 5.
   - Compositional semantics — the construction of meaning of longer phrases and sentences (based on syntax): lecture 6.
4. Pragmatics — meaning in context: lecture 7.

Why is NLP hard?

Ambiguity: *same strings can mean different things*

- Word senses: bank (finance or river?)
- Part of speech: chair (noun or verb?)
- Syntactic structure: I saw a man with a telescope
- Multiple: I saw her duck

Finally, a computer that understands you like your mother!

Ambiguity grows with sentence length, sometimes exponentially.

Why is NLP hard?

Ambiguity: *same strings can mean different things*

- Word senses: bank (finance or river?)
- Part of speech: chair (noun or verb?)
- Syntactic structure: I saw a man with a telescope
- Multiple: I saw her duck

Finally, a computer that understands you like your mother!

Ambiguity grows with sentence length, sometimes exponentially.

Why is NLP hard?

Ambiguity: *same strings can mean different things*

- Word senses: bank (finance or river?)
- Part of speech: chair (noun or verb?)
- Syntactic structure: I saw a man with a telescope
- Multiple: I saw her duck

Finally, a computer that understands you like your mother!

Ambiguity grows with sentence length, sometimes exponentially.

Why is NLP hard?

Ambiguity: *same strings can mean different things*

- Word senses: bank (finance or river?)
- Part of speech: chair (noun or verb?)
- Syntactic structure: I saw a man with a telescope
- Multiple: I saw her duck

Finally, a computer that understands you like your mother!

Ambiguity grows with sentence length, sometimes exponentially.

Why is NLP hard?

Ambiguity: *same strings can mean different things*

- Word senses: bank (finance or river?)
- Part of speech: chair (noun or verb?)
- Syntactic structure: I saw a man with a telescope
- Multiple: I saw her duck

Finally, a computer that understands you like your mother!

Ambiguity grows with sentence length, sometimes exponentially.

Real examples from newspaper headlines

Iraqi head seeks arms

Stolen painting found by tree

Teacher strikes idle kids

Real examples from newspaper headlines

Iraqi head seeks arms

Stolen painting found by tree

Teacher strikes idle kids

Real examples from newspaper headlines

Iraqi head seeks arms

Stolen painting found by tree

Teacher strikes idle kids

Why is NLP hard?

Synonymy and variability: different strings can mean the same or similar things

Did Google buy YouTube?

1. Google purchased YouTube
2. Google's acquisition of YouTube
3. Google acquired every company
4. YouTube may be sold to Google
5. Google didn't take over YouTube

Example from "Combined Distributional and Logical Semantics", Lewis & Steedman, TACL 2013

Wouldn't it be better if . . . ?

The properties which make natural language difficult to process are essential to human communication:

- Flexible
- Learnable, but expressive and compact
- Emergent, evolving systems

Synonymy and ambiguity go along with these properties.

Natural language communication can be indefinitely precise:

- Ambiguity is mostly local (for humans)
- resolved by immediate context
- but requires world knowledge

Wouldn't it be better if . . . ?

The properties which make natural language difficult to process are essential to human communication:

- Flexible
- Learnable, but expressive and compact
- Emergent, evolving systems

Synonymy and ambiguity go along with these properties.

Natural language communication can be indefinitely precise:

- Ambiguity is mostly local (for humans)
- resolved by immediate context
- but requires world knowledge

World knowledge...

"Knowledge is knowing that a tomato is a fruit"

BUT

"Wisdom is knowing not to put it in a fruit salad"

- Impossible to hand-code at a large-scale
- *either* limited domain applications
- *or* learn approximations from the data

Opinion mining: what do they think about me?

- Task: scan documents (webpages, tweets etc) for positive and negative opinions on people, products etc.
- Find all references to entity in some document collection: list as positive, negative (possibly with strength) or neutral.
- Fine-grained classification:
  e.g., for phone, opinions about: overall design, display, camera.

LG G3 review (Guardian 27/8/2014)

The shiny, brushed effect makes the G3's plastic design looks deceptively like metal. It feels solid in the hand and the build quality is great — there's minimal give or flex in the body. It weighs 149g, which is lighter than the 160g HTC One M8, but heavier than the 145g Galaxy S5 and the significantly smaller 112g iPhone 5S.

The G3's claim to fame is its 5.5in quad HD display, which at 2560x1440 resolution has a pixel density of 534 pixels per inch, far exceeding the 432ppi of the Galaxy S5 and similar rivals. The screen is vibrant and crisp with wide viewing angles, but the extra pixel density is not noticeable in general use compared to, say, a Galaxy S5.

LG G3 review (Guardian 27/8/2014)

The shiny, brushed effect makes the G3's plastic ***design*** *looks deceptively like metal. It feels solid in the hand and the build quality is great — there's minimal give or flex in the body. It weighs 149g, which is lighter than the 160g HTC One M8, but heavier than the 145g Galaxy S5 and the significantly smaller 112g iPhone 5S.*

The G3's claim to fame is its 5.5in quad HD display, which at 2560x1440 resolution has a pixel density of 534 pixels per inch, far exceeding the 432ppi of the Galaxy S5 and similar rivals. The screen is vibrant and crisp with wide viewing angles, but the extra pixel density is not noticeable in general use compared to, say, a Galaxy S5.

LG G3 review (Guardian 27/8/2014)

The shiny, brushed effect makes the G3's plastic ***design*** *looks deceptively like metal. It feels solid in the hand and the* ***build quality*** *is great — there's minimal give or flex in the body. It weighs 149g, which is lighter than the 160g HTC One M8, but heavier than the 145g Galaxy S5 and the significantly smaller 112g iPhone 5S.*

The G3's claim to fame is its 5.5in quad HD display, which at 2560x1440 resolution has a pixel density of 534 pixels per inch, far exceeding the 432ppi of the Galaxy S5 and similar rivals. The screen is vibrant and crisp with wide viewing angles, but the extra pixel density is not noticeable in general use compared to, say, a Galaxy S5.

LG G3 review (Guardian 27/8/2014)

The shiny, brushed effect makes the G3's plastic ***design*** *looks deceptively like metal. It feels solid in the hand and the* ***build quality*** *is great — there's minimal give or flex in the body. It* ***weighs*** *149g, which is lighter than the 160g HTC One M8, but heavier than the 145g Galaxy S5 and the significantly smaller 112g iPhone 5S.*

The G3's claim to fame is its 5.5in quad HD display, which at 2560x1440 resolution has a pixel density of 534 pixels per inch, far exceeding the 432ppi of the Galaxy S5 and similar rivals. The screen is vibrant and crisp with wide viewing angles, but the extra pixel density is not noticeable in general use compared to, say, a Galaxy S5.

LG G3 review (Guardian 27/8/2014)

The shiny, brushed effect makes the G3's plastic ***design*** *looks deceptively like metal. It feels solid in the hand and the* ***build quality*** *is great — there's minimal give or flex in the body. It* ***weighs*** *149g, which is lighter than the 160g HTC One M8, but heavier than the 145g Galaxy S5 and the significantly smaller 112g iPhone 5S.*

The G3's claim to fame is its 5.5in quad HD ***display****, which at 2560x1440 resolution has a pixel density of 534 pixels per inch, far exceeding the 432ppi of the Galaxy S5 and similar rivals. The screen is vibrant and crisp with wide viewing angles, but the extra pixel density is not noticeable in general use compared to, say, a Galaxy S5.*

Sentiment classification: the research task

- **Full task:** information retrieval, cleaning up text structure, named entity recognition, identification of relevant parts of text. Evaluation by humans.
- **Research task:** preclassified documents, topic known, opinion in text along with some straightforwardly extractable score.
- Pang et al. 2002: *Thumbs up? Sentiment Classification using Machine Learning Techniques*
- Movie review corpus: strongly positive or negative reviews from IMDb, 50:50 split, with rating score.

Sentiment analysis as a text classification problem

- *Input*:
  - a document x
  - a fixed set of classes $\mathcal{Y} = \{1, \ldots, C\}$
- *Output*:
  - a predicted class $y \in \mathcal{Y}$

Classes can be 'named' (e.g., positive, negative, neutral), but we index them by integers (e.g., 1, 2, 3).

IMDb: An American Werewolf in London (1981)

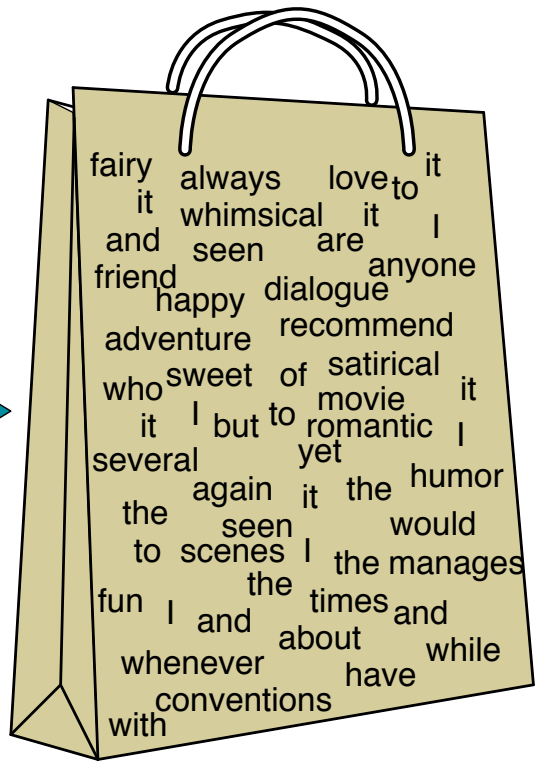
Rating: 9/10

Ooooo. Scary.
The old adage of the simplest ideas being the best is once again demonstrated in this, one of the most entertaining films of the early 80's, and almost certainly Jon Landis' best work to date. The script is light and witty, the visuals are great and the atmosphere is top class. Plus there are some great freeze-frame moments to enjoy again and again. Not forgetting, of course, the great transformation scene which still impresses to this day.
In Summary: Top banana

Bag of words representation

Treat the reviews as collections of individual words.

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!



it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
…	…

Bag of words representation

- Classify reviews according to positive or negative words.
- Could use word lists prepared by humans — sentiment lexicons
- but machine learning based on a portion of the corpus (training set) is preferable.
- Use human rankings for training and evaluation.

Supervised classification

- *Input*:
  - a document x
  - a fixed set of classes $\mathcal{Y} = \{1, \ldots, C\}$
  - a training set of N hand-labeled documents $(x_1, y_1), ..., (x_N, y_N)$
- *Output*:
  - a learned classifier $\gamma : x \rightarrow y$

Classes can be 'named' (e.g., positive, negative, neutral), but we index them by integers (e.g., 1, 2, 3).

Classification methods

Many classification methods available

- Naive Bayes
- Logistic regression
- Decision trees
- k-nearest neighbors
- Support vector machines
- ...

Naive Bayes classifier

For any document x and any of the C possible classes $y \in \mathcal{Y}$, we learn to assign probability to y given x: $P_{Y|X}(y|x)$.

For that, we design a probability model of labels and documents $P_{XY}(x, y) \triangleq P_Y(y)P_{X|Y}(x|y)$.

- The document-conditioned class probability is a conditional of this model $P_{Y|X}(y|x) = \frac{P_Y(y)P_{X|Y}(x|y)}{\sum_{c=1}^{C} P_Y(c)P_{X|Y}(x|c)}$.

Whenever we want to classify a document x, we output the *most probable class* under the model: $\hat{y} = \operatorname{argmax}_{c \in \mathcal{C}} P_{Y|X}(c|x)$.

Naive Bayes classifier

For any document x and any of the C possible classes $y \in \mathcal{Y}$, we learn to assign probability to y given x: $P_{Y|X}(y|x)$.

For that, we design a probability model of labels and documents $P_{XY}(x, y) \triangleq P_Y(y)P_{X|Y}(x|y)$.

- The document-conditioned class probability is a conditional of this model $P_{Y|X}(y|x) = \frac{P_Y(y)P_{X|Y}(x|y)}{\sum_{c=1}^{C} P_Y(c)P_{X|Y}(x|c)}$.

Whenever we want to classify a document x, we output the *most probable class* under the model: $\hat{y} = \operatorname{argmax}_{c \in \mathcal{C}} P_{Y|X}(c|x)$.

Naive Bayes classifier

For any document x and any of the C possible classes $y \in \mathcal{Y}$, we learn to assign probability to y given x: $P_{Y|X}(y|x)$.

For that, we design a probability model of labels and documents $P_{XY}(x, y) \triangleq P_Y(y)P_{X|Y}(x|y)$.

- The document-conditioned class probability is a conditional of this model $P_{Y|X}(y|x) = \frac{P_Y(y)P_{X|Y}(x|y)}{\sum_{c=1}^{C} P_Y(c)P_{X|Y}(x|c)}$.

Whenever we want to classify a document x, we output the *most probable class* under the model: $\hat{y} = \operatorname{argmax}_{c \in \mathcal{C}} P_{Y|X}(c|x)$.

Naive Bayes classifier

For any document x and any of the C possible classes $y \in \mathcal{Y}$, we learn to assign probability to y given x: $P_{Y|X}(y|x)$.

For that, we design a probability model of labels and documents $P_{XY}(x, y) \triangleq P_Y(y)P_{X|Y}(x|y)$.

- The document-conditioned class probability is a conditional of this model $P_{Y|X}(y|x) = \frac{P_Y(y)P_{X|Y}(x|y)}{\sum_{c=1}^{C} P_Y(c)P_{X|Y}(x|c)}$.

Whenever we want to classify a document x, we output the *most probable class* under the model: $\hat{y} = \operatorname{argmax}_{c \in \mathcal{C}} P_{Y|X}(c|x)$.

Naive Bayes classifier - Model components

The model has two types of factors, which we need to estimate:

- C class probabilities $P_Y(y)$
- class-conditioned document probabilities $P_{X|Y}(x|y)$

Problem: as x is high-dimensional, we would need a very, very large corpus to estimate a useful $P_{X|Y}(x|y)$.

Conditional independence assumption ('naive'): segment x into ℓ 'feature values' (e.g., words) $\langle w_1, \ldots, w_\ell \rangle$ and assume these are independent of one another, given the class y.

$$P_{XY}(x, y) \overset{\text{ind.}}{=} P_Y(y) \underbrace{\prod_{i=1}^{\ell} P_{F|Y}(w_i|y)}_{\text{doc as a 'bag of words'}}$$

Naive Bayes classifier - Model components

The model has two types of factors, which we need to estimate:

- C class probabilities $P_Y(y)$
- class-conditioned document probabilities $P_{X|Y}(x|y)$

Problem: as x is high-dimensional, we would need a very, very large corpus to estimate a useful $P_{X|Y}(x|y)$.

Conditional independence assumption ('naive'): segment x into ℓ 'feature values' (e.g., words) $\langle w_1, \ldots, w_\ell \rangle$ and assume these are independent of one another, given the class y.

$$P_{XY}(x, y) \overset{\text{ind.}}{=} P_Y(y) \underbrace{\prod_{i=1}^{\ell} P_{F|Y}(w_i|y)}_{\text{doc as a 'bag of words'}}$$

Naive Bayes classifier - Overview

Choose most probable class given document $x = \langle w_1, \ldots, w_\ell \rangle$:

$$\hat{y} = \operatorname*{argmax}_{c \in C} P_{Y|X}(c|x)$$

$P_{Y|X}(c|x)$ inferred from $P_Y(c) \prod_{i=1}^{\ell} P_{F|Y}(w_i|c)$, via Bayes rule:

$$P_{Y|X}(c|x) = \frac{P_Y(c) \prod_{i=1}^{\ell} P_{F|Y}(w_i|c)}{P_X(x)}$$

For any x, $P_X(x)$ is constant for prediction:

$$\hat{y} = \operatorname*{argmax}_{c \in C} P_Y(c) \prod_{i=1}^{\ell} P_{F|Y}(w_i|c)$$

Naive Bayes: Learning the model

The model has two types of factors, which we need to estimate:

- C class probabilities $P_Y(y)$
- $C \times V$ class-conditioned 'feature' probabilities $P_{F|Y}(f|y)$ (we regard words as 'features')

Maximum likelihood estimation: use frequencies in the data

$$P_Y(c) \overset{\text{MLE}}{=} \frac{\text{count}_Y(c)}{N}$$

$$P_{F|Y}(w|y) \overset{\text{MLE}}{=} \frac{\text{count}_{YF}(y, w)}{\sum_{f \in \mathcal{F}} \text{count}_{YF}(c, f)}$$

$\mathcal{F}$ is the set of all features (e.g., all words in training data).

Problem with maximum likelihood

What if we have seen no training documents with the word ***fantastic*** and classified as **positive**?

$$P_{F|Y}(\text{fantastic}|\text{positive}) \overset{\text{MLE}}{=} \frac{\text{count}_{YF}(\text{positive}, \text{fantastic})}{\sum_{f \in \mathcal{F}} \text{count}_{YF}(\text{positive}, f)} = 0$$

Zero probabilities cannot be conditioned away, no matter the other evidence!

$$\underbrace{P_Y(c) \prod_{i=1}^{\ell} P_{F|C}(w_i|c)}_{=0 \text{ whenever any factor is } 0}$$

Laplace smoothing for Naive Bayes

Smoothing is a way to 'handle' data sparsity

Laplace (also called "add 1") smoothing:

$$P_{F|Y}(w|c) \overset{\text{add1}}{=} \frac{\text{count}_{YF}(c, w) + 1}{\sum_{f \in \mathcal{F}} (\text{count}_{YF}(c, f) + 1)}$$
$$= \frac{\text{count}_{YF}(c, w) + 1}{|\mathcal{F}| + \sum_{f \in \mathcal{F}} \text{count}_{YF}(c, f)}$$

Log space

Use log space to prevent arithmetic underflow.

- Multiplying lots of probabilities can result in floating-point underflow
- sum logs of probabilities instead of multiplying probabilities

$$\log(a \times b) = \log a + \log b$$

- class with the highest log probability score is still the most probable

$$\hat{y} = \underset{c \in C}{\operatorname{argmax}} \left(\log P_Y(c) + \sum_{i=1}^{\ell} \log P_{F|Y}(w_i|c) \right)$$

Test sets and cross-validation

Divide the corpus into

- training set — to train the model
- development set — to optimize its parameters
- test set — kept unseen

or...

use **cross-validation** over multiple splits

- divide the corpus into e.g. 10 parts
- train on 9 parts, test on 1 part
- average results from all splits

If you are worried about *statistical evaluation*, check https://jmlr.org/papers/v18/16-305.html.

Evaluation

Accuracy:

$$Accuracy = \frac{Number\ of\ correctly\ classified\ instances}{Total\ number\ of\ instances}$$

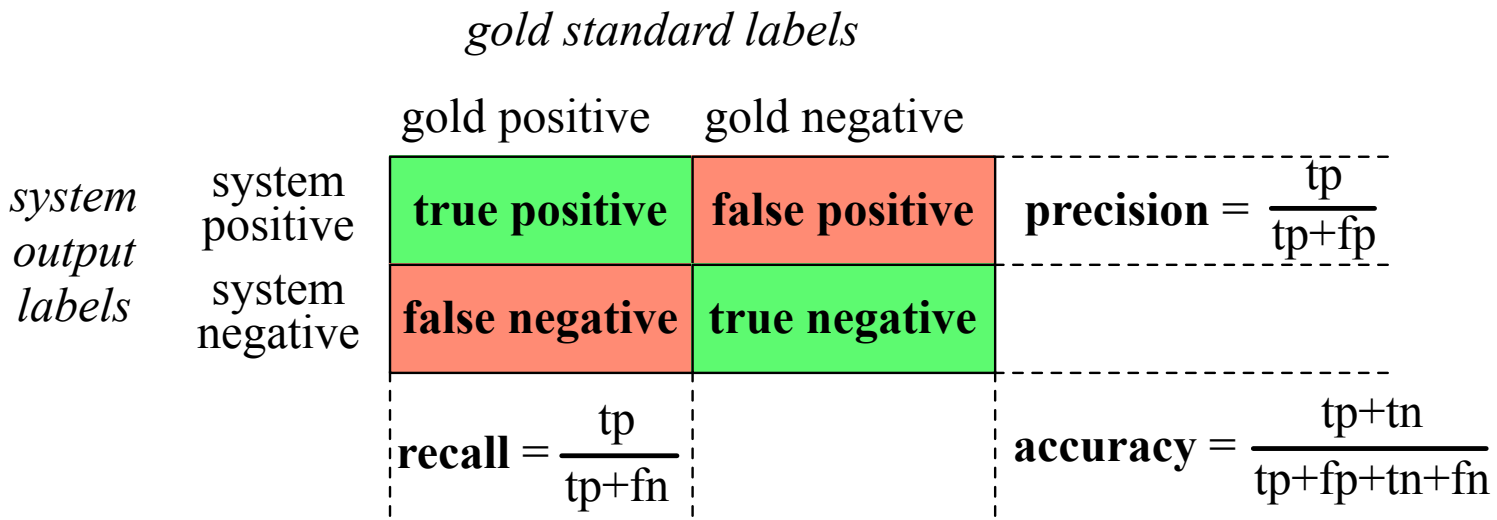
Pang et al. (2002):

- The corpus is artificially balanced
- Chance success is 50%
- Bag-of-words achieves an accuracy of 80%.

Precision and Recall

What if the corpus were not balanced?

- **Precision**: % of selected items that are correct
- **Recall**: % of correct items that are selected



The statistics tp, fp, fn, tn form what we call a 'confusion matrix'.

F-measure

Also called *F-score*

$$F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

β controls the importance of recall and precision

$\beta = 1$ is typically used:

$$F_1 = \frac{2PR}{P + R}$$

With more than 2 classes, we typically compute F_β per class, then report average of per-class F_β with uniform weights (macro average) or using class proportions as weights (weighted average).

Error analysis

Bag-of-words gives 80% accuracy in sentiment analysis

Some sources of errors:

- Negation:
  Ridley Scott has never directed a bad film.
- Overfitting the training data:
  e.g., if training set includes a lot of films from before 2005, *Ridley* may be a strong positive indicator, but then we test on reviews for 'Kingdom of Heaven'?
- Comparisons and contrasts.

Contrasts in the discourse

This film should be brilliant. It sounds like a great plot, the actors are first grade, and the supporting cast is good as well, and Stallone is attempting to deliver a good performance. However, it can't hold up.

Feature View

Say we know V features, indexed by $[V] = \{1, \ldots, V\}$.

We can *represent* a document x as a feature vector $\phi(x)$ in a V-dimensional space. For example, where $\phi_f(x)$ is the number of occurrences of the feature $f \in [V]$ in x.

We can re-express $P_{X|Y}(x|y)$ as

$$P_{X|Y}(x|y) = \prod_{f=1}^{V} P_{F|Y}(f|y)^{\phi_f(x)}$$

We can then generalise the *feature function* $\phi(\cdot)$. For example, it could segment documents into *word pairs*, annotate words with negation scope and discourse structure, etc.

Feature View

Say we know V features, indexed by $[V] = \{1, \ldots, V\}$.

We can *represent* a document x as a feature vector $\phi(x)$ in a V-dimensional space. For example, where $\phi_f(x)$ is the number of occurrences of the feature $f \in [V]$ in x.

We can re-express $P_{X|Y}(x|y)$ as

$$P_{X|Y}(x|y) = \prod_{f=1}^{V} P_{F|Y}(f|y)^{\phi_f(x)}$$

We can then generalise the *feature function* $\phi(\cdot)$. For example, it could segment documents into *word pairs*, annotate words with negation scope and discourse structure, etc.

Feature View

Say we know V features, indexed by $[V] = \{1, \ldots, V\}$.

We can *represent* a document x as a feature vector $\phi(x)$ in a V-dimensional space. For example, where $\phi_f(x)$ is the number of occurrences of the feature $f \in [V]$ in x.

We can re-express $P_{X|Y}(x|y)$ as

$$P_{X|Y}(x|y) = \prod_{f=1}^{V} P_{F|Y}(f|y)^{\phi_f(x)}$$

We can then generalise the *feature function* $\phi(\cdot)$. For example, it could segment documents into *word pairs*, annotate words with negation scope and discourse structure, etc.

Doing sentiment classification 'properly'?

- Morphology, syntax and compositional semantics
  - model relationships between words in the sentences, compose the meanings of phrases, negation, tense ...
- Lexical semantics
  - are words positive or negative in this context? Model word senses (e.g., *spirit*)
- Pragmatics and discourse structure
  - relationships between sentences; what is the topic of this section of text; co-reference resolution.
- Getting all this to work well on arbitrary text is hard.
- But can we do well enough for NLP to be useful?

Human translation?

Human translation?

I am not in the office at the moment. Please send any work to be translated.

Sentiment analysis practical: Part 1

Sentiment classification of movie reviews

1. Sentiment classification with a sentiment lexicon
2. Implement Naive Bayes classifier with bag-of-word features
3. Model grammar: word order and part of speech tags
4. Experiment with support vector machines (SVM) classifier
5. Evaluate and compare different methods

Assessed by the mid-term report, deadline 13 November

Sentiment analysis practical: Part 1

Sentiment classification of movie reviews

1. Sentiment classification with a sentiment lexicon
2. Implement Naive Bayes classifier with bag-of-word features
3. Model grammar: word order and part of speech tags
4. Experiment with support vector machines (SVM) classifier
5. Evaluate and compare different methods

Assessed by the mid-term report, deadline 13 November

Sentiment analysis practical: Part 2

- Experiment within a deep learning framework
- Include (more sophisticated) syntax and semantics
- Model the meaning of words, phrases and sentences
- Evaluate in the sentiment classification task
- Propose your own research question or type of analysis

Assessed by the final report, deadline 13 December

Sentiment analysis practical: Part 2

- Experiment within a deep learning framework
- Include (more sophisticated) syntax and semantics
- Model the meaning of words, phrases and sentences
- Evaluate in the sentiment classification task
- Propose your own research question or type of analysis

Assessed by the final report, deadline 13 December

Acknowledgement

Some slides were adapted from Ann Copestake, Dan Jurafsky and Tejaswini Deoskar