Fundamentals of Machine Learning
Week 6: Text mining

Jonas Moons

Intro

Topics

- Natural Language Processing (NLP)
- Modeling text in English and similar languages
- Classification with Naïve Bayes

Natural language processing

- Production
  - Natural language generation
  - Text-to-speech
  - Chatbots

- Recognition
  - Speech recognition
  - Understanding written text

- Translation

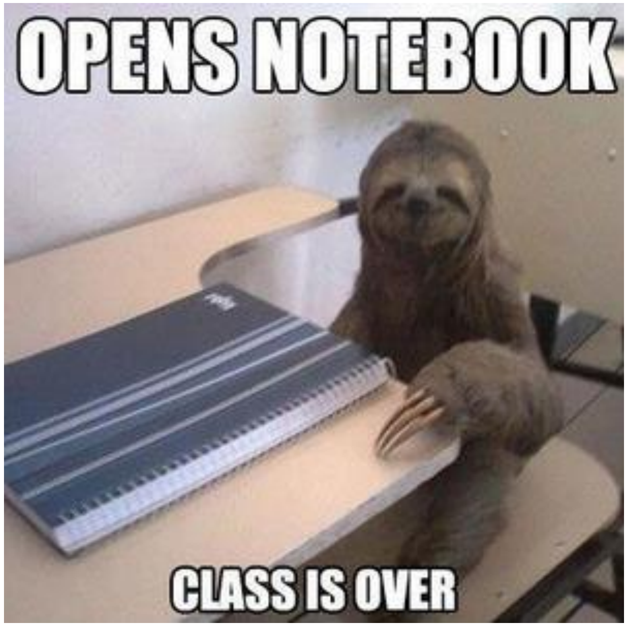
Challenges

- Requires huge corpora and tagged databases
- Specialized algorithms and linguistic expertise
- Context is everything
- Diversity of languages

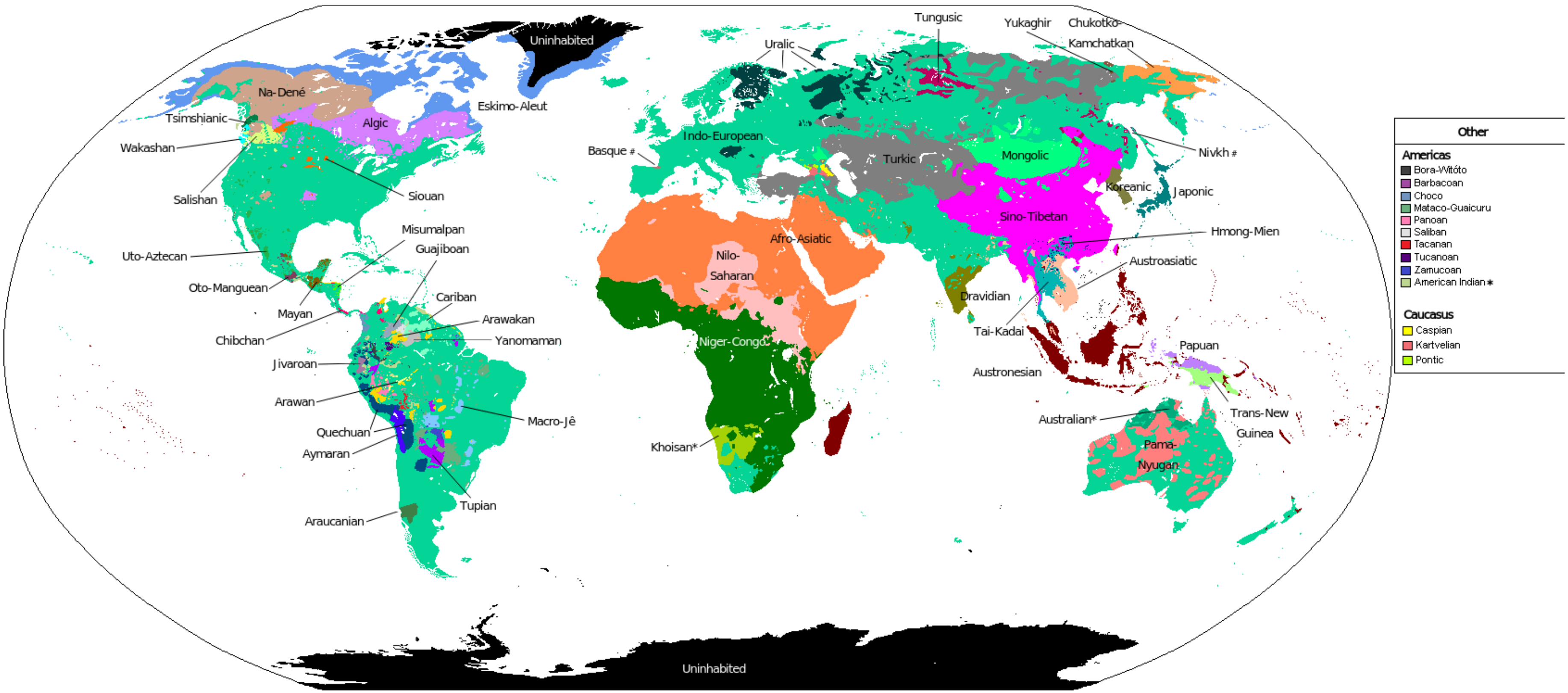
Language is complex…

“What are you doing in this classroom?”

- Morphology (how words are formed)
  → ‘do-ing’, ‘are’
- Syntax (sentence construction, word order)
  → what are you …
- Semantics (what words mean)
  → 'classroom': room in a school
- Pragmatics (meaning in context)
  → ‘get out!’



Languages of the world



Diversity challenges for NLP

- Most work done on English
  - Also some on Mandarin Chinese, Arabic, French, others

- Availability of corpora, text mining libraries, tagging algorithms…

- Different scripts

- Characteristics of English that many languages don't share
  - Easy phone-based writing system
  - Clear what is a word from text (spaces)
  - Simple morphology (word form)

- Bender rule: state which language you're working in

Topics

- Natural Language Processing (NLP)
- Modeling text in English and similar languages
- Classification with Naïve Bayes

Bag of words

- The 'bag of words' model treats a document as a collection of words, and a count for each

- It ignores semantics, syntax (word order), morphology and pragmatics (e.g., irony)

- Very simple but often effective for many languages

Word	Count
You	8
I	6
And	5
Will	5
Be	4

Tokenizing

- Tokenizing is about breaking text up into units (‘words’)

- Relatively easy in English…

  (But what about ‘New York’, ‘ice cream’)

- A lot harder in ‘agglutinative’ languages like Turkish or Finnish

Document-feature matrix

Features / words / variables

	flouncy	flow	flower	flowery	flowey	flowier
Doc1	0	0	0	1	0	0
Doc2	0	0	0	0	0	0
Doc3	0	0	2	0	0	0
Doc4	0	0	0	0	0	0
Doc5	0	1	0	0	0	0

High-dimensional & sparse: almost entirely empty

Exercise 1: building a text model

During this week, you are going to work with dialog lines from the Simpsons. Your task during this week is to build a model that distinguishes Bart's dialogue from Lisa's dialogue.

For this exercise, you can use the *text_mining* Example Notebook

1. Load in the .csv file and have a look at the data set. What will be challenging when building a predictive model?
2. We will only work with Bart's and Lisa's lines. Make the relevant selection in the data set. Tip: this is a selection based on two conditions. Use brackets () around each condition and use | instead of *or* in Pandas. If you are stuck, ask!
3. Create a document-feature matrix of the text data.
4. Print a selection of the features/words.
5. The next step should not freeze your computer, but just in case: save your files. Try and make a regular matrix out of the sparse matrix and add it to the data frame. How much memory does Python use now? (Windows: Task manager / Taakbeheer. Apple: Activity Monitor)

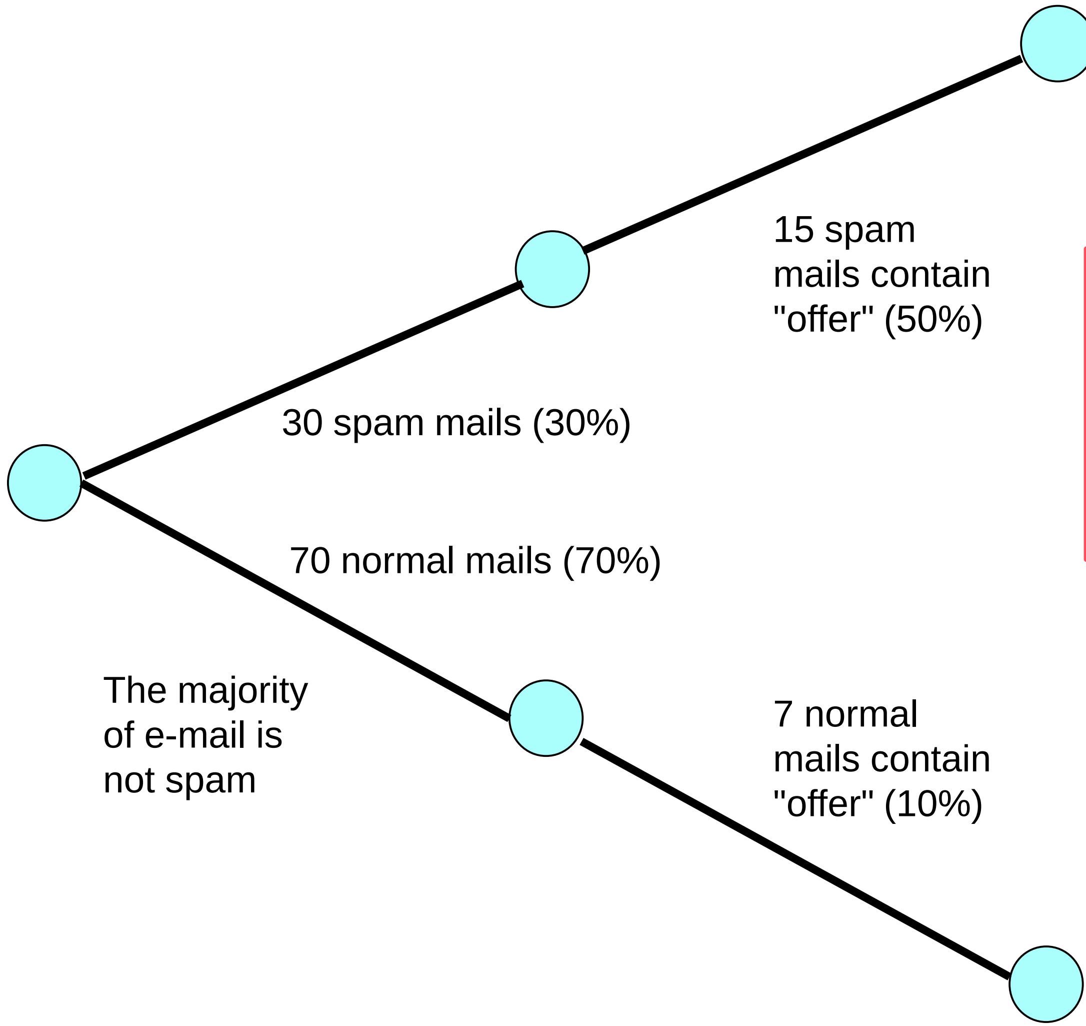
Lemmatization and stemming

- Lemmatization: means reducing a word to its grammatical stem
- Removing pre- and suffixes for things like gender, number, tense, aspect, etc.
- Going, goes, gone, go → go
- Falo, falas, fala, falamos, falam, falava, falavas, falávamos, falavam, falei, falaste, falou, falámos, falaram, falarei, falarás, falará, falaremos, falarão, falaria, falarias, falaria, falaríamos, falariam,… → falar
- Not included in *sklearn.* Some text mining libraries include *nltk* and others

Topics

- Natural Language Processing (NLP)
- Modeling text in English and similar languages
- Classification with Naïve Bayes

Bayes' theorem



If we encounter the word "offer", the e-mail is about twice as likely (15/7) to be spam than not

Bayes' theorem in text mining

- We can use Bayes' theorem to calculate a probability that a text belongs to a certain category (e.g., spam)
- The frequency of each word determines the probability
- But how do we combine the probabilities of the different words?

Naïve Bayes

- If I flip two coins, the probability of one coin being heads does not influence the other: they are independent. We can multiply the probabilities.

- The probabilities of two words being in a text (e.g., ‘police’ and ‘crime’) are definitely **not** independent

- Yet this is exactly what is assumed in Naïve Bayes (hence: ‘naïve’), and it works well in practice

Exercise 2: Naïve Bayes

For this exercise, we will not use example code. Instead, you will write the code yourself based on the kind of code that you wrote before. See the *Cookbook* for reference.

See the documentation of sklearn.naive_bayes_MultinomialNB for the Naive Bayes model object.

Steps:

1. Import the relevant libraries and/or objects
2. Create X and y variables – y is the colummn with the character names, X is the document-feature matrix
3. Split the data into a training and a test set
4. Fit a NB model on the training set
5. Predict the classes (Lisa or Bart) of the test set and store the result in a variable
6. Calculate the accuracy of your model (there is a method in MultinomialNB for this)

Classification

- Under the hood (like Random Forest) the output of the Naive Bayes algorithm is not actually a class

- Instead, it gives probabilities for each class C_i:

$P(Y = C_i \mid X)$: the probability of class C

- The classification is based on the class with the highest probability

Evaluation of classification

Confusion matrix:

	Predicted: Not spam	Predicted: Spam	Total
Actual: Not spam	20	10	**30**
Actual: Spam	40	60	**100**
Total	**60**	**70**	**130**

What proportion is correctly predicted?

$$accuracy = \frac{correctly\ pred.}{total\ cases} = \frac{20+60}{20+40+10+60} = \frac{80}{130} = 0.62$$

How much of the predicted 'spam' is actually spam?

$$precision\ (spam) = \frac{correctly\ pred.\ (spam)}{total\ pred.\ (spam)} = \frac{60}{60+10} = \frac{60}{70} = 0.86$$

How much of the real spam is predicted as spam?

$$recall\ (spam) = \frac{correctly\ pred.\ (spam)}{total\ actual\ (spam)} = \frac{60}{40+60} = \frac{60}{100} = 0.60$$

Exercise 3: evaluation

In this exercise we will evaluate the model and delve further into the data. Remember that you can use the *Cookbook* for examples of code.

1. Start where you left off. Create a confusion matrix.

2. Calculate the accuracy and the recall and precision for both Bart and Lisa.

3. Check out the documentation on how to get the *probabilities* for a certain text belonging to a class (tip: it's a *method*), instead of a categorization. Try it on a line of dialogue.

4. Create a loop that prints out a few lines of dialogue and the associated probabilities for Bart and Lisa. Tip: the array with the probabilities is 2-dimensional.

5. Check out the output. Do you see patterns (based on the data and your knowledge of the Simpsons)?

Image credit

- Simpons by 20th century Fox, fair use claimed
- Sloth meme by author unknown, fair use claimed
- Bag by wixin_56: public domain
- Languages of the world by Alumnum: CC-ASA 4.0