INTRO TO NLP AND NAIVE BAYES

Natural Language Processing

Sergey Nikolenko

Harbour Space University, Barcelona, Spain
January 8, 2018

NLP PROBLEMS

- Three classes of problems.
- The first class — more "syntactic" problems:
    - they are more or less well-defined,
    - they can usually be posed as classification problems,
    - it is clear how to collect datasets (albeit it may require manual labor, of course).
- These problems can be solved reasonably well by classical techniques, but DL improves upon these results.
- But we will see how even "simple" problems require "full-scale understanding" in hard cases.

NLP PROBLEMS

- Part-of-speech tagging:
  - The panda eats shoots and leaves
    (ok, this one is about punctuation)

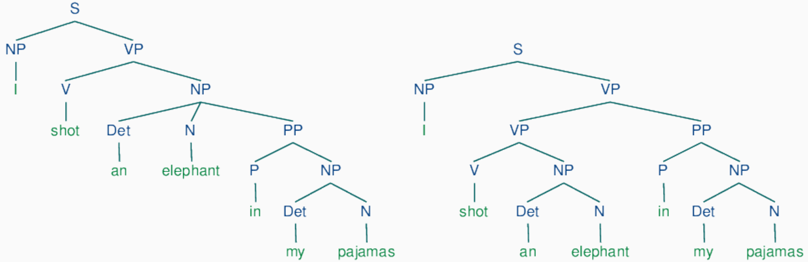
```
>>> text = word_tokenize("They refuse to permit us to obtain the refuse permit")
>>> nltk.pos_tag(text)
[('They', 'PRP'), ('refuse', 'VBP'), ('to', 'TO'), ('permit', 'VB'), ('us', 'PRP'),
('to', 'TO'), ('obtain', 'VB'), ('the', 'DT'), ('refuse', 'NN'), ('permit', 'NN')]
```

- Morphological segmentation
- Stemming or lemmatization
- Sentence boundary disambiguation:
  - Rolls-Royce Motor Cars Inc. said it expects its U.S. sales to remain steady at about 1,200 cars in 1990.
  - Later, he recalls the words of his Marxist mentor: "The people! Theft! The holy fire!"
  - About 40 Italian businesses, including Fiat S.p.A. and Ing. C. Olivetti & Co., have formed a consortium to lobby for holding the expo in Venice.

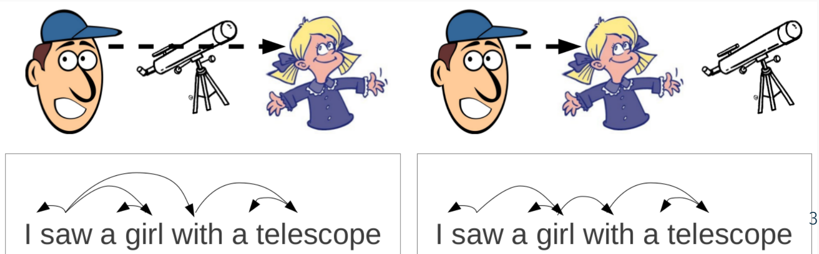
NLP PROBLEMS

- Word segmentation (Asian languages)
- Named entity recognition:
  - «In 2001, Michael Jordan retired from the editorial board of *Machine Learning*»
  - «In 2001, Michael Jordan returned from his second retirement to play for the *Washington Wizards*»
- Word sense disambiguation:
  - «I have a cold today» vs. «We've had a cold day»
  - «After listening to the great bass, Boris Christoff, we ate sea bass at the restaurant»
  - granularity is unclear (e.g., "knife" as a kitchen utensil vs. a weapon)

- Syntactic parsing:



- Dependency parsing:



- Coreference resolution, anaphora resolution:
    - «The laptop did not fit in the bag because it was too small»;
    - «The laptop did not fit in the bag because it was too big».
- Pragmatics:
    - «Alice and Betty are mothers»;
    - «Alice and Betty are sisters».
- Big problems with *common sense reasoning*:
  AI models don’t have it

- Second class – more complex problems that require understanding even more often, but we still know the right answers and can get quality metrics.
- Language modeling:
    - big breakthroughs from RNNs;
    - direct use for speech recognition and the like, but generally the underlying problem for all NLP applications.
- Sentiment analysis:
    - recursive neural networks;
    - requires syntactic parsing first;
    - can we solve sentiment analysis? yeah, right...

- Relationship extraction, fact extraction:
    - usually a CNN on vector representations of words + positional embeddings (how far each word is from each entity in the sentence).
- Question answering:
    - formally contains everything else;
    - in reality – only very simple questions:
        - Mary went to the bathroom.
        - John moved to the hallway.
        - Mary travelled to the office.
        - Where is Mary?
    - QA will probably encode "general text understanding".

- Problems where we not only understand text but try to generate new text:
  - text generation per se;
  - automatic summarization;
  - machine translation;
  - dialog and conversational models.
- There are machine learning models for all these problems, and currently state of the art models come from deep learning.
- But we will have to start at an earlier point.

TEXT CATEGORIZATION: NAIVE BAYES

TEXT CATEGORIZATION

- Classical NLP problem: text categorization (classification).
- Given a text, which category is it in?
- Bag-of-words model: forget about word order, construct a vocabulary.
- Now a document is a vector of word counts.

NAIVE BAYES

- Even this is a very big simplification.
- But still, we can't expect to get enough statistics for $p(a_1, a_2, \ldots, a_n | x = v)$.
- We need more simplifying assumptions.
- Naive Bayes classifier — the simplest model: assume that all words in a dictionary are conditionally independent given the category/

- In other words:

$$p(w_1, w_2, \ldots, w_n | x = v) = p(w_1 | x = v) p(w_2 | x = v) \ldots p(w_n | x = v).$$

- Naive Bayes classifier chooses v as

$$v_{NB}(w_1, w_2, \ldots, w_n) = \arg\max_{v \in V} p(x = v) \prod_{i=1}^{n} p(w_i | x = v).$$

- Despite indeed very naive assumptions, NB works pretty well in practice (and there are reasons for this).

- There are important details in NB implementation.
- Two basic approaches: multinomial and multivariate.

- In the multivariate model a document is a vector of binary attributes that show whether a word has occurred there.
- Let $V = \{w_t\}_{t=1}^{|V|}$ be the vocabulary.
- Then a document d_i is a vector of size $|V|$ consisting of bits B_{it}; $B_{it} = 1$ iff w_t occurs in d_i.

- Likelihood of a document is the product of probabilities for multivariate Bernoulli trials:

$$p(d_i \mid c_j) = \prod_{t=1}^{|V|} \left(B_{it} p(w_t \mid c_j) + (1 - B_{it})(1 - p(w_t \mid c_j)) \right).$$

- To train this classifier we need to train $p(w_t \mid c_j)$.
- How?

- Easy: consider the training set $D = \{d_i\}_{i=1}^{|D|}$, where words are already distributed among classes c_j (perhaps even probabilistically) with vocabulary $V = \{w_t\}_{t=1}^{|V|}$. We know B_{it}.
- Optimal probability estimates for Bernoulli trials with Bayesian (Laplace) smoothing:

$$p(w_t \mid c_j) = \frac{1 + \sum_{i=1}^{|D|} B_{it} p(c_j \mid d_i)}{2 + \sum_{i=1}^{|D|} p(c_j \mid d_i)}.$$

- Prior probabilities of classes are simply $p(c_j) = \frac{1}{|D|}\sum_{i=1}^{|D|} p(c_j \mid d_i)$.
- Now classification becomes

$$c = \arg\max_j p(c_j)p(d_i \mid c_j) =$$

$$= \arg\max_j \left(\frac{1}{|D|}\sum_{i=1}^{|D|} p(c_j \mid d_i)\right)\prod_{t=1}^{|V|}\left(B_{it}p(w_t \mid c_j) + (1 - B_{it})(1 - p(w_t \mid c_j))\right) =$$

$$= \arg\max_j \left(\log(\sum_{i=1}^{|D|} p(c_j \mid d_i)) + \sum_{t=1}^{|V|}\log\left(B_{it}p(w_t \mid c_j) + (1 - B_{it})(1 - p(w_t \mid c_j))\right)\right)$$

- In the multinomial model a document is a sequence of events; each event means taking a word out of the bag.
- The naive assumption is that we take words out of a bag independently of each other.
- We get a multiplicative generative model: a document d_i is a vector of length $|d_i|$ consisting of words each of which was taken with probability $p(w_t \mid c_j)$.

- Likelihood that d_i belongs to class c_j:

$$p(d_i \mid c_j) = p(|d_i|)|d_i|! \prod_{t=1}^{|V|} \frac{1}{N_{it}!} p(w_t \mid c_j)^{N_{it}},$$

  where N_{it} is the number of times w_t occurs in d_i.
- To train this classifier we need to train the probabilities $p(w_t \mid c_j)$.
- How?

- Easy: for a training set $D = \{d_i\}_{i=1}^{|D|}$ distributed among classes c_j (perhaps probabilistically) with vocabulary $V = \{w_t\}_{t=1}^{|V|}$ we know N_{it}.
- Again we compute posterior estimates with Bayesian smoothing:

$$p(w_t \mid c_j) = \frac{1 + \sum_{i=1}^{|D|} N_{it} p(c_j \mid d_i)}{|V| + \sum_{s=1}^{|V|} \sum_{i=1}^{|D|} N_{is} p(c_j \mid d_i)}.$$

MULTINOMIAL MODEL

- Prior probabilities of classes are again $p(c_j) = \frac{1}{|D|} \sum_{i=1}^{|D|} p(c_j \mid d_i)$.
- Now classification becomes

$$
\begin{aligned}
c = \arg\max{}_j p(c_j) p(d_i \mid c_j) = \\
= \arg\max{}_j \left(\frac{1}{|D|} \sum_{i=1}^{|D|} p(c_j \mid d_i) \right) p(|d_i|) |d_i|! \prod_{t=1}^{|V|} \frac{1}{N_{it}!} p(w_t \mid c_j)^{N_{it}} = \\
= \arg\max{}_j \left(\log \left(\sum_{i=1}^{|D|} p(c_j \mid d_i) \right) + \sum_{t=1}^{|V|} N_{it} \log p(w_t \mid c_j) \right).
\end{aligned}
$$

THANK YOU!

Thank you for your attention!