

COMP90042 - Natural Language Processing

Workshop Week 3

Biaoyan Fang

16 March 2020

Recap Pre-processing

Pipeline

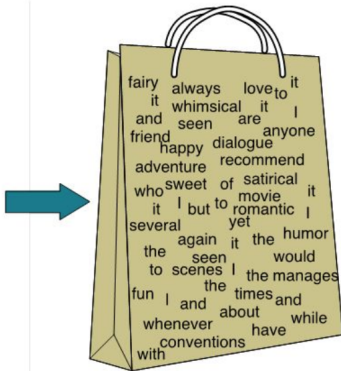
- Formatting
- Sentence Segmentation
- Tokenisation
- Normalisation
 - Lemmatisation
 - Stemming
- Remove Stopwords

1. Text-classification
2. Language Model

The Bag of Words Representation

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!

15



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
...	...

Abbildung 1: Bag of words

K-Nearest Neighbour

Euclidean distance:
Usually length is not a
distinguishing character

Cosine similarity:
Better;
Suffer from high-dimensionality
problem

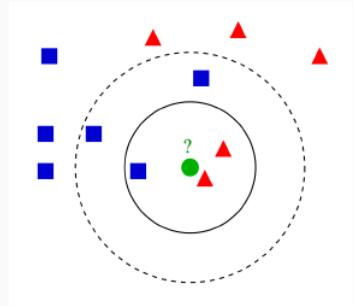


Abbildung 2: k-nearest neighbour

Decision Tree

Can be useful in finding
meaningful features

Spurious correlation;
Tend to rare features

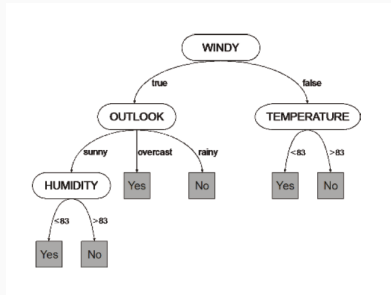


Abbildung 3: Decision Tree

Naive Bayes

Bayes law

Conditional independence of features

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Then

Surprisingly useful

$$p(c_n|f_1, \dots, f_m) = \prod_{i=1}^m p(f_i|c_n)p(c_n)$$

Logistic Regression

Useful. Relax the conditional independence requirement of Naive Bayes

Handle large numbers of features

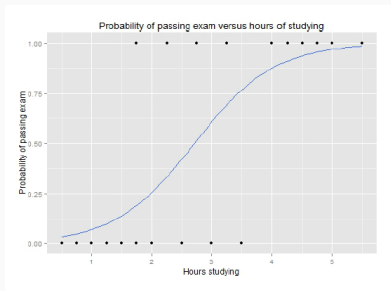


Abbildung 4: Logistic Regression

More powerful

Suffer from multiple classes problem

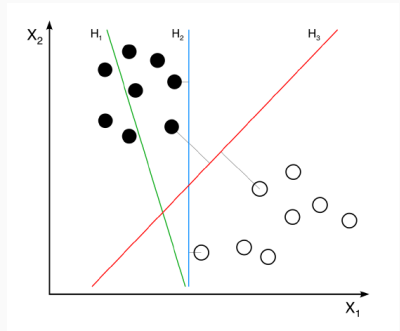


Abbildung 5: SVM

Probabilistic Language Model

A probability distribution over sequences of words. Given such a sequence, say of length m , it assigns a probability $P(w_1, \dots, w_m)$ to the whole sequence.

Goal: compute the probability of a sentence or sequence of words:

$$P(W) = P(w_1, w_2, \dots, w_m)$$

Chain rule:

$$P(w_1, w_2, \dots, w_m) = P(w_1)P(w_2|w_1) \dots P(w_m|w_1, \dots, w_{m-1})$$

Probabilistic Language Model

Markvo assumption:

$$P(w_i|w_1, \dots, w_{i-1}) \approx P(w_i|w_{i-n+1}, \dots, w_{i-1})$$

While,

$n = 1$, unigram language model:

$$P(w_1, w_2, \dots, w_m) = \prod_{i=1}^m P(w_i)$$

$n = 2$, bigram language model:

$$P(w_1, w_2, \dots, w_m) = \prod_{i=1}^m P(w_i|w_{i-1})$$

$n = 3$, trigram language model:

$$P(w_1, w_2, \dots, w_m) = \prod_{i=1}^m P(w_i|w_{i-2}w_{i-1})$$

Continuation counts:

$$P_{cont}(w_i) = \frac{|\{w_{i-1} : C(w_{i-1}, w_i) > 0\}|}{\sum_{w_i} |\{w_{i-1} : C(w_{i-1}, w_i) > 0\}|}$$

Questions