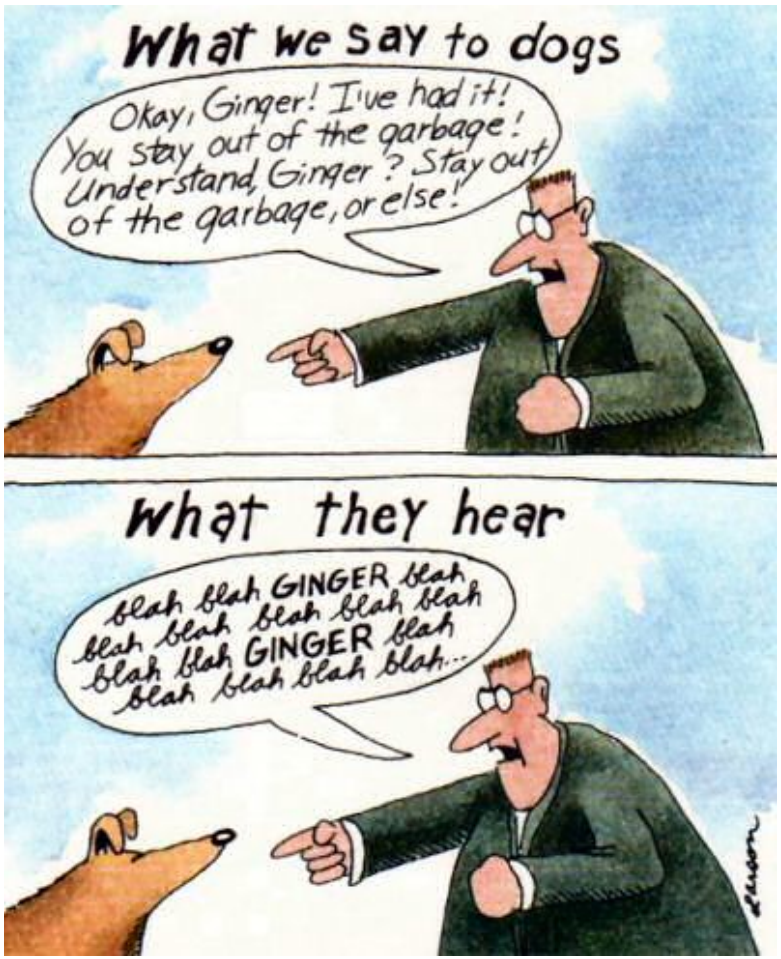
Natural Language Processing

Nov 19, 2019



Who wrote the Federalist Papers?

- 1787-8: anonymous essays try to convince New York to ratify U.S Constitution: Jay, Madison, Hamilton.
- Authorship of 12 of the letters in dispute.
- 1963: solved by Mosteller and Wallace using Bayesian methods.

Who wrote the Federalist Papers?

- 1787-8: anonymous essays try to convince New York to ratify U.S Constitution: Jay, Madison, Hamilton.
- Authorship of 12 of the letters in dispute.
- 1963: solved by Mosteller and Wallace using Bayesian methods.

By the end of this lecture we will see how to do that.

What makes it hard?

- Formal languages are:
  - *unambiguous*
- Natural languages are
  - *ambiguous*:
    - "He saw her duck".
    - "Time flies like an arrow. Fruit flies like a banana"

By the end of the class

By the end of the class we will see how to do:

1. Text Classification. E.g. Spam detection, Authorship identification.
2. Spell Correction. E.g. Auto-correct.
3. Word suggestion.

Regular Expressions

A formal language for specifying text strings.

Notations

- Disjunctions [] :

Pattern	Matches
[*Ww*]*oodchuck*	woodchuck, Woodchuck
[0123456789]	Any single digit

- Disjunctions |:

Pattern	Matches
abc|*def*	Find 'abc' or 'def'.
a|*b*|*ab*	Find 'a' or 'b' or 'ab'. Example: 'abc'

Notations

- Ranges:

Pattern	Matches
$[A-Z]$	An uppercase letter.
$[a-z]$	A lowercase letter.
$[0-9]$	A single digit.

- Negation ^. (Note: Carat means negation only when its first in [])

Pattern	Matches
$[\hat{}A-Z]$	Not upper case
$[\hat{}Ss]$	Not 'S' nor 's'
$[\hat{}e\hat{}]$	Not 'e' nor '^'
$a\hat{}b$	Search for the pattern'$a\hat{}b$'

Notations (? * . + ^ $)

?	0 or 1 of previous character
*	0 or more of previous character
+	1 or more of previous character
.	Any character
^	Start anchor
$	End anchor
\	Escape character

Examples

- Pattern: $\hat{}[A - Z]$.

Which of them are matches? “Class”, “cSCE”, “420”.

Examples

- Pattern: ^[$A - Z$].

Which of them are matches? “Class”, “cSCE”, “420”.

Class.

Examples

- Pattern: ^[A – Z].

Which of them are matches? “Class”, “cSCE”, “420”.

Class.

- Pattern: ^[^A – Z].

Which of them are matches? “Class”, “cSCE”, “420”.

Examples

- Pattern: ^[$A - Z$].

Which of them are matches? "Class", "cSCE", "420".

Class.

- Pattern: ^[^$A - Z$].

Which of them are matches? "Class", "cSCE", "420".

cSCE, 420.

Examples

- Pattern: ^[$A - Z$].

Which of them are matches? “Class”, “cSCE”, “420”.

Class.

- Pattern: ^[^$A - Z$].

Which of them are matches? “Class”, “cSCE”, “420”.

cSCE, 420.

- Pattern: .$

Which of them are matches? “end”, “end?”, “end!”, “end.”.

Examples

- Pattern: ^[$A - Z$].

Which of them are matches? “Class”, “cSCE”, “420”.

Class.

- Pattern: ^[^$A - Z$].

Which of them are matches? “Class”, “cSCE”, “420”.

cSCE, 420.

- Pattern: .$

Which of them are matches? “end”, “end?”, “end!”, “end.”.

end. end? end! end.

Examples

- Pattern: ^[$A-Z$].

Which of them are matches? "Class", "cSCE", "420".

Class.

- Pattern: ^[^$A-Z$].

Which of them are matches? "Class", "cSCE", "420".

cSCE, 420.

- Pattern: .$

Which of them are matches? "end", "end?", "end!", "end.".

end. end? end! end.

- Pattern: \.$

Which of them are matches? "end", "end?", "end!", "end.".

Examples

- Pattern: ^[$A-Z$].

Which of them are matches? "Class", "cSCE", "420".

Class.

- Pattern: ^[^$A-Z$].

Which of them are matches? "Class", "cSCE", "420".

cSCE, 420.

- Pattern: .$

Which of them are matches? "end", "end?", "end!", "end.".

end. end? end! end.

- Pattern: \.$

Which of them are matches? "end", "end?", "end!", "end.".

end..

Examples

- Pattern: *colou*?*r*.

Which of them are matches? “color”, “colour”, “colouur”.

Examples

- Pattern: *colou*?*r*.

Which of them are matches? “color”, “colour”, “colouur”.

color, colour.

Examples

- Pattern: $colou?r$.

Which of them are matches? "color", "colour", "colouur".

color, colour.

- Pattern: $colou + r$.

Which of them are matches? "color", "colour", "colouur".

Examples

- Pattern: $colou?r$.

Which of them are matches? "color", "colour", "colouur".

color, colour.

- Pattern: $colou + r$.

Which of them are matches? "color", "colour", "colouur".

colour, colouur.

Examples

- Pattern: $colou?r$.

Which of them are matches? "color", "colour", "colouur".

color, colour.

- Pattern: $colou + r$.

Which of them are matches? "color", "colour", "colouur".

colour, colouur.

- Pattern: $colou * r$.

Which of them are matches? "color", "colour", "colouur".

Examples

- Pattern: $colou?r$.

Which of them are matches? "color", "colour", "colouur".

color, colour.

- Pattern: $colou + r$.

Which of them are matches? "color", "colour", "colouur".

colour, colouur.

- Pattern: $colou * r$.

Which of them are matches? "color", "colour", "colouur".

color, colour, colouur.

Examples

- Pattern: *colou*?*r*.

Which of them are matches? "color", "colour", "colouur".

color, colour.

- Pattern: *colou* + *r*.

Which of them are matches? "color", "colour", "colouur".

colour, colouur.

- Pattern: *colou* ∗ *r*.

Which of them are matches? "color", "colour", "colouur".

color, colour, colouur.

- Pattern: *colou*.*r*.

Which of them are matches? "color", "colour", "colouur".

Examples

- Pattern: $colou?r$.

Which of them are matches? "color", "colour", "colouur".

color, colour.

- Pattern: $colou + r$.

Which of them are matches? "color", "colour", "colouur".

colour, colouur.

- Pattern: $colou * r$.

Which of them are matches? "color", "colour", "colouur".

color, colour, colouur.

- Pattern: $colou.r$.

Which of them are matches? "color", "colour", "colouur".

colouur.

Example

We need to find instances of “the” in a text.

Example

We need to find instances of “the” in a text.

the

Example

We need to find instances of “the” in a text.

the $\times$ ‘The’

Example

We need to find instances of “the” in a text.

the $\times$ ‘The’

$[Tt]he$

Example

We need to find instances of “the” in a text.

the $\times$ ‘The’

[*Tt*]*he* $\times$ ‘Theology’

Example

We need to find instances of “the” in a text.

$the \times$ ‘The’

$[Tt]he \times$ ‘Theology’

$[\hat{}A - Za - z][Tt]he[\hat{}A - Za - z]$

Text classification

- Assigning subject categories, topics, or genres.
- Spam detection.
- Authorship identification.
- Age/gender identification.
- Language Identification.
- ...

Text Classification

- Inputs:
  - Document d.
  - Fixed set of classes $C = \{c_1, c_2, \cdots, c_n\}$.
- Output:
  - A predicted class $c \in C$

Naive Bayes

Relies on simple representation of document – Bag of Words.

Naive Bayes

Relies on simple representation of document – Bag of Words.

For a document d and a class c

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)}$$

Naive Bayes Classifier

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(c|d)$$

MAP - Maximum a posteriori (most likely class).

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} \frac{P(d|c)P(c)}{P(d)}$$

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(d|c)P(c)$$

Naive Bayes Classifier

$$c_{MAP} = \underset{c \in C}{\text{argmax}}\, P(d|c)P(c)$$

Let's say that the document is represented by n features $x_1, x_2, \cdots x_n$

$$c_{MAP} = \underset{c \in C}{\text{argmax}}\, P(x_1, x_2, \cdots x_n|c)P(c)$$

Assumptions

Bag of words: Position of words does not matter.
Conditional Independence: The feature probabilities $P(x_i|c)$ are independent given the class c.

$$P(x_1, x_2, \cdots x_n|c) = \prod_{i=1}^{n} P(x_i|c)$$

Bag of word 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.

Bag of word representation

great	2
love	2
recommend	1
laugh	1
happy	1
...	...

Naive Bayes: Learning

What do we need?
Training set of m hand-labeled documents
$(d_1, c_1), \cdots, (d_m, c_m)$

Naive Bayes: Learning

Let N_D be the number of documents, and N_{c_j} be the number of documents present in class c_j.
Let V_{c_j} be the set of all words in the documents of class c_j
Now we find the maximum likelihood estimates:

$$\hat{P}(c_j) = \frac{N_{c_j}}{N_D}$$

$$\hat{P}(w_i|c_j) = \frac{count(w_i, c_j)}{\sum_{w \in V_{c_j}} count(w, c_j)}$$

Now we can classify a document d by:

$$c_d = \underset{c_j \in C}{\text{argmax}}\, \hat{P}(c_j) \prod_{w_i \in d} \hat{P}(w_i|c_j)$$

Naive Bayes: Learning

What if we come across an unknown word in the document d.
Let w_u be the unknown word $\hat{P}(w_u|c_j) = 0, \forall c_j$.

Laplace smoothing

Let V be the set of all words in the test documents, i.e, $V = \cup_{c_j} V_{c_j}$
Add one word for the unknown word in the vocabulary.

$$\hat{P}(w_i|c_j) = \frac{count(w_i, c_j) + 1}{\sum_{w \in V_{c_j}} count(w, c_j) + |V| + 1}$$

So, for all unknown words, we have:

$$\hat{P}(w_u|c_j) = \frac{1}{\sum_{w \in V_{c_j}} count(w, c_j) + |V| + 1}$$

Example

Training set:

#	Text	Class
1	Carla Betty Carla	a
2	Carla Carla Suzanne	a
3	Carla Matt	a
4	Taylor Jessica Carla	b

Example

Training set:

#	Text	Class
1	Carla Betty Carla	a
2	Carla Carla Suzanne	a
3	Carla Matt	a
4	Taylor Jessica Carla	b

$\hat{P}(a) =$

Example

Training set:

#	Text	Class
1	Carla Betty Carla	a
2	Carla Carla Suzanne	a
3	Carla Matt	a
4	Taylor Jessica Carla	b

$\hat{P}(a) = 3/4$
$\hat{P}(b) =$

Example

Training set:

#	Text	Class
1	Carla Betty Carla	a
2	Carla Carla Suzanne	a
3	Carla Matt	a
4	Taylor Jessica Carla	b

$\hat{P}(a) = 3/4$
$\hat{P}(b) = 1/4$

Example

Training set:

#	Text	Class
1	Carla Betty Carla	a
2	Carla Carla Suzanne	a
3	Carla Matt	a
4	Taylor Jessica Carla	b

Example

Training set:

#	Text	Class
1	Carla Betty Carla	a
2	Carla Carla Suzanne	a
3	Carla Matt	a
4	Taylor Jessica Carla	b

$\hat{P}(Carla|a) =$

Example

Training set:

#	Text	Class
1	Carla Betty Carla	a
2	Carla Carla Suzanne	a
3	Carla Matt	a
4	Taylor Jessica Carla	b

$\hat{P}(Carla|a) = (5+1)/(8+6+1) = 6/15$
$\hat{P}(Taylor|a) =$

Example

Training set:

#	Text	Class
1	Carla Betty Carla	a
2	Carla Carla Suzanne	a
3	Carla Matt	a
4	Taylor Jessica Carla	b

$\hat{P}(Carla|a) = (5+1)/(8+6+1) = 6/15$
$\hat{P}(Taylor|a) = (0+1)/(8+6+1) = 1/15$

Example

Document d_5: *Carla Carla Carla Taylor Jessica*

Example

Document d_5: *Carla Carla Carla Taylor Jessica*

$\hat{P}(a) = 3/4$ and $\hat{P}(b) = 1/4$
$\hat{P}(Carla|a) = 6/15, \hat{P}(Carla|b) = 2/10$
$\hat{P}(Taylor|a) = 1/15, \hat{P}(Taylor|b) = 2/10$
$\hat{P}(Jessica|a) = 1/15, \hat{P}(Jessica|b) = 2/10$

Example

Document d_5: *Carla Carla Carla Taylor Jessica*

$\hat{P}(a) = 3/4$ and $\hat{P}(b) = 1/4$
$\hat{P}(Carla|a) = 6/15, \hat{P}(Carla|b) = 2/10$
$\hat{P}(Taylor|a) = 1/15, \hat{P}(Taylor|b) = 2/10$
$\hat{P}(Jessica|a) = 1/15, \hat{P}(Jessica|b) = 2/10$

$P(a|d_5) =$

Example

Document d_5: *Carla Carla Carla Taylor Jessica*

$\hat{P}(a) = 3/4$ and $\hat{P}(b) = 1/4$
$\hat{P}(Carla|a) = 6/15, \hat{P}(Carla|b) = 2/10$
$\hat{P}(Taylor|a) = 1/15, \hat{P}(Taylor|b) = 2/10$
$\hat{P}(Jessica|a) = 1/15, \hat{P}(Jessica|b) = 2/10$

$P(a|d_5) = 3/4 \times (6/15)^3 \times 1/15 \times 1/15 \approx 0.0002$
$P(b|d_5) =$

Example

Document d_5: *Carla Carla Carla Taylor Jessica*

$\hat{P}(a) = 3/4$ and $\hat{P}(b) = 1/4$
$\hat{P}(Carla|a) = 6/15, \hat{P}(Carla|b) = 2/10$
$\hat{P}(Taylor|a) = 1/15, \hat{P}(Taylor|b) = 2/10$
$\hat{P}(Jessica|a) = 1/15, \hat{P}(Jessica|b) = 2/10$

$P(a|d_5) = 3/4 \times (6/15)^3 \times 1/15 \times 1/15 \approx 0.0002$
$P(b|d_5) = 1/4 \times (2/10)^3 \times 2/10 \times 2/10 \approx 0.00008$

Naive Bayes

Naive Bayes is not so naive!!

- Robust to Irrelevant Features.
- Optimal if the independence assumptions hold.
- A good dependable baseline for text classification. - There exists other classifiers that give better accuracy

Federalist Papers

Discussion: Federalist papers.
E.g. What training set do we need?

Language Modelling

Goal: Assign probability to a sentence.

Language Modelling

Goal: Assign probability to a sentence.
Why?

Language Modelling

Goal: Assign probability to a sentence.

Why?

- **Machine Translation.** P(high winds tonight) $>$ P(large winds tonight)

Language Modelling

Goal: Assign probability to a sentence.
Why?

- **Machine Translation.** P(high winds tonight) $>$ P(large winds tonight)
- **Spell Correction.** P(about fifteen minutes from) $>$ P(about fifteen minuets from)

Language Modelling

Goal: Assign probability to a sentence.
Why?

- **Machine Translation.** P(high winds tonight) $>$ P(large winds tonight)
- **Spell Correction.** P(about fifteen minutes from) $>$ P(about fifteen minuets from)
- **Speech Recognition.** P(I saw a van) $>$ P(eyes awe of an)

Language Modelling

- **Goal:** compute the probability of a sentence or sequence of words. $P(W) = P(w_1, w_2, \cdots, w_n)$
- **Related task:** probability of an upcoming word. $P(w_i | w_1, w_2, \cdots, w_{i-1})$

Language Modelling

- **Goal:** compute the probability of a sentence or sequence of words. $P(W) = P(w_1, w_2, \cdots, w_n)$
- **Related task:** probability of an upcoming word. $P(w_i|w_1, w_2, \cdots, w_{i-1})$

Chain rule:

$$P(x_1, x_2, \cdots, x_n)$$

$$= P(x_1)P(x_2|x_1)P(x_3|x_1, x_2)\cdots P(x_n|x_1, x_2 \cdots, x_{n-1})$$

$$= \prod_i P(x_i|x_1, x_2, \cdots x_{i-1})$$

Example

P(its water is so transparent that) $= P(its) \times P(water|its) \times P(so|its, water) \times P(transparent|its, water, is, so) \times P(that|its, water, is, so, transparent)$

Can we count?

$$P(that|its, water, is, so, transparent)$$

$$= \frac{P(its, water, is, so, transparent, that)}{P(its, water, is, so, transparent)}$$

Example

P(its water is so transparent that) $= P(its) \times P(water|its) \times P(so|its, water) \times P(transparent|its, water, is, so) \times P(that|its, water, is, so, transparent)$

Can we count?

$$P(that|its, water, is, so, transparent)$$

$$= \frac{P(its, water, is, so, transparent, that)}{P(its, water, is, so, transparent)}$$

No. Too many possibilities.

Markov Assumption

Take only the k words preceding it.

$$P(w_i|w_1, w_2, \cdots, w_{i-1}) \approx P(w_i|w_{i-k} \cdots, w_{i-1})$$

Markov Assumption

Take only the k words preceding it.

$$P(w_i|w_1, w_2, \cdots, w_{i-1}) \approx P(w_i|w_{i-k} \cdots, w_{i-1})$$

$$P(that|its, water, is, so, transparent) = P(that|transparent)$$

or,

$$P(that|its, water, is, so, transparent) = P(that|so, transparent)$$

Unigram, Bigram and N-gram

Unigram model:

$$P(w_1, w_2, \cdots, w_{n-1}, w_n) \approx \prod_i P(w_i)$$

Bigram model:

$$P(w_i|w_1, w_2, \cdots, w_{i-1}) \approx P(w_i|w_{i-1})$$

N-gram model:

Extension to trigram, 4-gram, 5-gram, etc.

Discussion

1. Spell correction

Discussion

1. Spell correction
2. Word suggestion.