Recap

Inferential statistics

- Hypothesis testing
  - We have made some observations
  - How probably are the observations we have seen under different assumptions?
  - If the result is very unlikely under one assumption, the other must be true
- Not an idiot-proof tool though – think when interpreting results

Language Models

Sprachverarbeitung (VL + Ü)

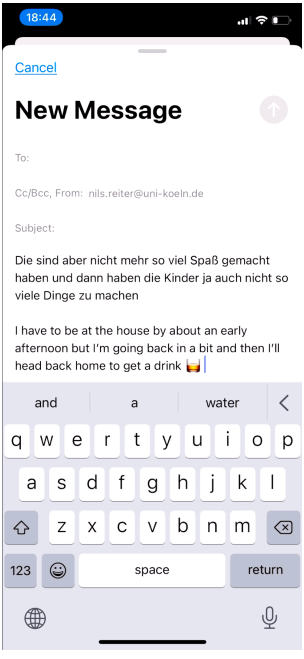
Nils Reiter

May 4, 2023

INSTITUT FÜR
DIGITAL HUMANITIES
UNIVERSITÄT ZU KÖLN

Section 1

Language Modeling


18:44
Cancel
New Message
To:
Cc/Bcc, From: nils.reiter@uni-koeln.de
Subject:
Die sind aber nicht mehr so viel Spaß gemacht haben und dann haben die Kinder ja auch nicht so viele Dinge zu machen
I have to be at the house by about an early afternoon but I'm going back in a bit and then I'll head back home to get a drink 🍻
and
a
water
q w e r t y u i o p
a s d f g h j k l
z x c v b n m
123
space
return

Introduction

- One of the oldest NLP tasks
  - Long before predictive typing on smart phones became a thing
  - Long before »large language models« became a thing
- Language model (LM) predicts the next word, given previous words (history)
- Formally: $p(\text{word}|\text{history})$

Introduction

- One of the oldest NLP tasks
  - Long before predictive typing on smart phones became a thing
  - Long before »large language models« became a thing
- Language model (LM) predicts the next word, given previous words (history)
- Formally: $p(\text{word}|\text{history})$

Example

Sue swallowed the large green ___

Introduction

- One of the oldest NLP tasks
  - Long before predictive typing on smart phones became a thing
  - Long before »large language models« became a thing
- Language model (LM) predicts the next word, given previous words (history)
- Formally: $p(\text{word}|\text{history})$

Example

Sue swallowed the large green ___

Reading

Christopher D. Manning/Hinrich Schütze (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, Massachusetts and London, England: MIT Press, Ch. 6.1–6.2. Ilias

History

- Not all textual histories can be treated individually
  - We couldn't predict anything on completely new histories
  - Chance of a text re-appearing is astronomically slim
- Predicting the next word on unseen sentences requires *generalization*

History

- Not all textual histories can be treated individually
  - We couldn't predict anything on completely new histories
  - Chance of a text re-appearing is astronomically slim
- Predicting the next word on unseen sentences requires *generalization*
- Instances of textual histories need to be grouped together
  - Manning/Schütze (1999, 192): »Equivalence Classes«

Forming Equivalence Classes

Different strategies

- Stemming/lemmatization: Don't look at word forms, look at lemmas or stems
  - E.e.: $p(\text{bark}|\text{the dog})$ instead of $p(\text{barks}|\text{The dog})$

Forming Equivalence Classes

Different strategies

- Stemming/lemmatization: Don't look at word forms, look at lemmas or stems
  - E.e.: $p(\text{bark}|\text{the dog})$ instead of $p(\text{barks}|\text{The dog})$
- Selected history: Only look at selected word classes
  - Content words like nouns, verbs, adjectives and adverbs
  - E.g., $p(\text{barks}|\text{dog})$ instead of $p(\text{barks}|\text{The dog})$

Forming Equivalence Classes

Different strategies

- Stemming/lemmatization: Don't look at word forms, look at lemmas or stems
  - E.e.: $p(\text{bark}|\text{the dog})$ instead of $p(\text{barks}|\text{The dog})$
- Selected history: Only look at selected word classes
  - Content words like nouns, verbs, adjectives and adverbs
  - E.g., $p(\text{barks}|\text{dog})$ instead of $p(\text{barks}|\text{The dog})$
- Both require linguistic pre-analysis of the text
  - Time-consuming and error-prone (on a large scale)

Forming Equivalence Classes

Different strategies

- Stemming/lemmatization: Don't look at word forms, look at lemmas or stems
  - E.e.: $p(\text{bark}|\text{the dog})$ instead of $p(\text{barks}|\text{The dog})$
- Selected history: Only look at selected word classes
  - Content words like nouns, verbs, adjectives and adverbs
  - E.g., $p(\text{barks}|\text{dog})$ instead of $p(\text{barks}|\text{The dog})$
- Both require linguistic pre-analysis of the text
  - Time-consuming and error-prone (on a large scale)
- Limit history: Only look at the last n words

Markov Assumption

- Assumption: Only the local context influences the next word

Markov property

Markov Assumption

- Assumption: Only the local context influences the next word — WMarkov property
- n-gram model: Only the last $n-1$ words are looked at to predict the nth word
  - Bigram model: $p(w_2|\langle w_1\rangle)$
  - Trigram model: $p(w_3|\langle w_1, w_2\rangle)$
  - 4-gram model: $p(w_4|\langle w_1, w_2, w_3\rangle)$

Markov Assumption

- Assumption: Only the local context influences the next word — WMarkov property
- n-gram model: Only the last $n-1$ words are looked at to predict the nth word
  - Bigram model: $p(w_2|\langle w_1\rangle)$
  - Trigram model: $p(w_3|\langle w_1, w_2\rangle)$
  - 4-gram model: $p(w_4|\langle w_1, w_2, w_3\rangle)$

Example

Bigram model: »green ___«

Markov Assumption

- Assumption: Only the local context influences the next word — WMarkov property
- n-gram model: Only the last $n-1$ words are looked at to predict the nth word
  - Bigram model: $p(w_2|\langle w_1\rangle)$
  - Trigram model: $p(w_3|\langle w_1, w_2\rangle)$
  - 4-gram model: $p(w_4|\langle w_1, w_2, w_3\rangle)$

Example

Trigram model: »large green ___«

Markov Assumption

- Assumption: Only the local context influences the next word — WMarkov property
- n-gram model: Only the last $n-1$ words are looked at to predict the nth word
  - Bigram model: $p(w_2|\langle w_1\rangle)$
  - Trigram model: $p(w_3|\langle w_1, w_2\rangle)$
  - 4-gram model: $p(w_4|\langle w_1, w_2, w_3\rangle)$

Example

4-gram model: »the large green ___«

Markov Assumption

- Assumption: Only the local context influences the next word — WMarkov property
- n-gram model: Only the last $n-1$ words are looked at to predict the nth word
  - Bigram model: $p(w_2|\langle w_1\rangle)$
  - Trigram model: $p(w_3|\langle w_1, w_2\rangle)$
  - 4-gram model: $p(w_4|\langle w_1, w_2, w_3\rangle)$

Example

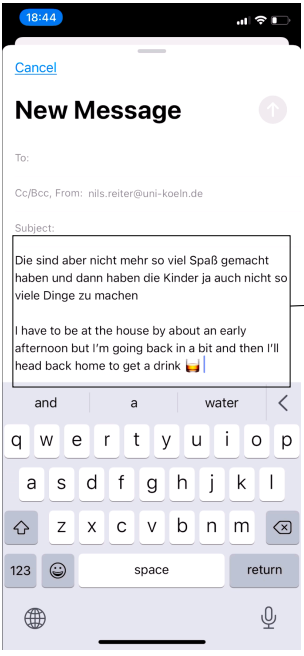
5-gram model: »swallowed the large green ___«

Markov Assumption

- Assumption: Only the local context influences the next word — WMarkov property
- n-gram model: Only the last $n-1$ words are looked at to predict the nth word
  - Bigram model: $p(w_2|\langle w_1\rangle)$
  - Trigram model: $p(w_3|\langle w_1, w_2\rangle)$
  - 4-gram model: $p(w_4|\langle w_1, w_2, w_3\rangle)$

Example

6-gram model: »Sue swallowed the large green ___«



Die sind aber nicht mehr so viel Spaß gemacht haben und dann haben die Kinder ja auch nicht so viele Dinge zu machen

I have to be at the house by about an early afternoon but I'm going back in a bit and then I'll head back home to get a drink

Increasing n

- The higher n, the better?
- Storage and training time increases
  - Number of parameters: Number of numbers (frequencies/probabilities) we need to store separately

Increasing n

- The higher n, the better?
- Storage and training time increases
  - Number of parameters: Number of numbers (frequencies/probabilities) we need to store separately
- Assuming a vocabulary of $20\,000$ words (= types)
  - Bigram model: $20\,000^2 = 400\,000\,000$ parameters
  - Trigram model: $20\,000^3 = 8\,000\,000\,000\,000 = 8 \times 10^{12}$ parameters
  - 4-gram model: $20\,000^4 = 1.6 \times 10^{17}$ parameters

Rechtschreibduden: $140\,000$

Increasing n

- The higher n, the better?
- Storage and training time increases
  - Number of parameters: Number of numbers (frequencies/probabilities) we need to store separately
- Assuming a vocabulary of $20\,000$ words (= types)
  - Bigram model: $20\,000^2 = 400\,000\,000$ parameters (= ca. 50 MB)
  - Trigram model: $20\,000^3 = 8\,000\,000\,000\,000 = 8 \times 10^{12}$ parameters (= ca. 8 GB)
  - 4-gram model: $20\,000^4 = 1.6 \times 10^{17}$ parameters (= ca. 20 PB)

Rechtschreibduden: $140\,000$

Again, a Compromise

- Longer n-grams would give better predictions
- Shorter n-grams would be easier/faster to train and use

Again, a Compromise

- Longer n-grams would give better predictions
- Shorter n-grams would be easier/faster to train and use
- Common: $n = 2$ or $n = 3$
  - Trigrams are surprisingly good at predicting the next word!

- Where do we actually get these probabilities from?
  - Corpora.

Training

- Where do we actually get these probabilities from?
  - Corpora.
- Training
  - Count frequences of features from data
  - Convert them into probabilities, maybe apply mathematical transformations

Training

- Where do we actually get these probabilities from?
  - Corpora.
- Training
  - Count frequences of features from data
  - Convert them into probabilities, maybe apply mathematical transformations
- Definition of conditional probabilities:

$$p(w_n|\langle w_1, \ldots, w_{n-1}\rangle) = \frac{p(\langle w_1, \ldots, w_n\rangle)}{p(\langle w_1, \ldots, w_{n-1}\rangle)}$$

Maximum Likelihood Estimation (MLE)

- Parameters that maximize probability of the training corpus
  = use the relative frequency from the training corpus as probability

$$p(\langle w_1, \ldots, w_n \rangle) = \frac{c(\langle w_1, \ldots, w_n \rangle)}{N}$$

Maximum Likelihood Estimation (MLE)

- Parameters that maximize probability of the training corpus
  = use the relative frequency from the training corpus as probability

$$p(\langle w_1, \ldots, w_n \rangle) = \frac{c(\langle w_1, \ldots, w_n \rangle)}{N}$$

$$p(w_n | \langle w_1, \ldots w_{n-1}) = \frac{p(\langle w_1, \ldots, w_n \rangle)}{p(\langle w_1, \ldots, w_{n-1} \rangle)}$$

Maximum Likelihood Estimation (MLE)

Example

History	w_n	Count
Bier und	Wein	4
Bier und	Schnaps	3
Bier und	Bratwürsten	1
Bier und	Männerschweiß	1
Bier und	nichtalkoholischen	1
...	...	1
Bier	und	29

Maximum Likelihood Estimation (MLE)

Example

History	w_n	Count
Bier und	Wein	4
Bier und	Schnaps	3
Bier und	Bratwürsten	1
Bier und	Männerschweiß	1
Bier und	nichtalkoholischen	1
...	...	1
Bier	und	29

$$
\begin{aligned}
p(\text{Bier und}) &= \frac{22}{1880232} \\
p(\text{Wein}|\text{Bier und}) &= \frac{p(\text{Bier und Wein})}{p(\text{Bier und})} \\
&= \frac{\frac{4}{1880232}}{\frac{22}{1880232}} \\
&= \frac{4}{1880232} \times \frac{1880232}{22} \\
&= \frac{4}{22} = 0.1818
\end{aligned}
$$

Application

- Training corpus used for estimating probability
- Test/application corpus used for using probability
- Never use the same corpus for training and testing

Application

- Training corpus used for estimating probability
- Test/application corpus used for using probability
- Never use the same corpus for training and testing
- After having trained, we can check how probable a new document/corpus is (= test/application)

Example

$$
\begin{aligned}
p(\text{Ich trinke gerne Bier und Wein}) &= p(\text{Ich}|\text{SYM SYM}) \times p(\text{trinke}|\text{Ich SYM}) \\
&\times p(\text{gerne}|\text{Ich trinke}) \times p(\text{Bier}|\text{trinke gerne}) \\
&\times p(\text{und}|\text{gerne Bier}) \times p(\text{Wein}|\text{Bier und})
\end{aligned}
$$

Maximum Likelihood Estimation (MLE)

Drawbacks

- What happens with words not in the training corpus? Zero probability
  - ›out of vocabulary‹ (OOV)
- Because of multiplication, everything will be zero

Maximum Likelihood Estimation (MLE)

Drawbacks

- What happens with words not in the training corpus? Zero probability
  - ›out of vocabulary‹ (OOV)
- Because of multiplication, everything will be zero
- There will be OOV words – because Zipf
- MLE conceptually important, but rarely used in NLP

$\Rightarrow$ We need another estimator for the probability

Lidstone's Law

- Core problem: All probability mass is used on words in vocabulary
- Nothing left for OOV words in test/application
- OOV words need to receive a probability > 0

Lidstone's Law

- Core problem: All probability mass is used on words in vocabulary
- Nothing left for OOV words in test/application
- OOV words need to receive a probability > 0

$$p(\langle w_1, \ldots w_n \rangle) = \frac{c(\langle w_1, \ldots w_n \rangle) + \lambda}{N + B\lambda}$$

Lidstone's Law

- Core problem: All probability mass is used on words in vocabulary
- Nothing left for OOV words in test/application
- OOV words need to receive a probability > 0

$$p(\langle w_1, \ldots w_n \rangle) = \frac{c(\langle w_1, \ldots w_n \rangle) + \lambda}{N + B\lambda}$$

- B: Number of different n-grams (i.e., n-gram types)
- λ: Parameter set to control how much mass remains for OOV words
  - Typical setting: $\lambda = \frac{1}{2}$ (for reasons see Manning/Schütze, 1999, 204)

Smoothing

- Lidstone's law is a ›smoothing‹ technique
- Goal
  - Prevent zero probabilities
  - Reserve some amount of probability mass for OOV words

Smoothing

- Lidstone's law is a ›smoothing‹ technique
- Goal
  - Prevent zero probabilities
  - Reserve some amount of probability mass for OOV words
- Different strategies
  - Often need for fine-tuning (e.g., what value to we use for λ?)

Section 2

Evaluation of Machine Learning Systems

Introduction

- So far: Descriptive methods
- Next weeks: Different machine learning strategies
  - Predictive methods: Given a text, predict some properties of it
- Today: Evaluation
- Goal, in general: Predict (linguistic) categories of text
  - Examples: Parts of speech, syntactic relations, semantic roles, word senses, ...

Introduction

- So far: Descriptive methods
- Next weeks: Different machine learning strategies
  - Predictive methods: Given a text, predict some properties of it
- Today: Evaluation
- Goal, in general: Predict (linguistic) categories of text
  - Examples: Parts of speech, syntactic relations, semantic roles, word senses, ...
- Why machine learning?
  - Development in NLP/CL over last 30 years
  - Language phenomena in the wild are complex and context-dependent
  - Rule-based systems difficult to develop and maintain

Evaluation

- For today, we consider the actual ML stuff as a black box
- How exactly do we evaluate? How do we measure how good predictions are?

Evaluation

- For today, we consider the actual ML stuff as a black box
- How exactly do we evaluate? How do we measure how good predictions are?

Example (Sentiment Analysis)

- Task: Assign a polarity (positive/neutral/negative) to a linguistic expression
- Linguistic expression: sentences, phrases, documents
  - In this example: Documents

Evaluation

- For today, we consider the actual ML stuff as a black box
- How exactly do we evaluate? How do we measure how good predictions are?

Example (Sentiment Analysis)

- Task: Assign a polarity (positive/neutral/negative) to a linguistic expression
- Linguistic expression: sentences, phrases, documents
  - In this example: Documents
- Classification task: Instances are sorted into previously known categories

Evaluation

- For today, we consider the actual ML stuff as a black box
- How exactly do we evaluate? How do we measure how good predictions are?

Example (Sentiment Analysis)

- Task: Assign a polarity (positive/neutral/negative) to a linguistic expression
- Linguistic expression: sentences, phrases, documents
  - In this example: Documents
- Classification task: Instances are sorted into previously known categories
- Data set: 100 documents that have labels
  - I.e., we know the result to expect

Evaluation Strategies

- Manual inspection by the developer: Run the tool, look at the results and decide
  - ⚠ Difficult to reproduce, prone to biases, implicit standards
  - Fast

Evaluation Strategies

- Manual inspection by the developer: Run the tool, look at the results and decide
  - ⊖ ⚠ Difficult to reproduce, prone to biases, implicit standards
  - ⊕ Fast
- Manual inspection by an expert: Run the tool, hand it over to an expert and let them decide
  - ⊖ Difficult to reproduce, expensive
  - ⊕ More reliable

Evaluation Strategies

- Manual inspection by the developer: Run the tool, look at the results and decide
  - ⊖ ⚠ Difficult to reproduce, prone to biases, implicit standards
  - ⊕ Fast
- Manual inspection by an expert: Run the tool, hand it over to an expert and let them decide
  - ⊖ Difficult to reproduce, expensive
  - ⊕ More reliable
- Plug into an application that benefits from a component: Extrinsic evaluation
  - ⊖ Need evaluation for the application, impact of component not always clear
  - ⊕ Realistic evaluation (if it's a realistic application)

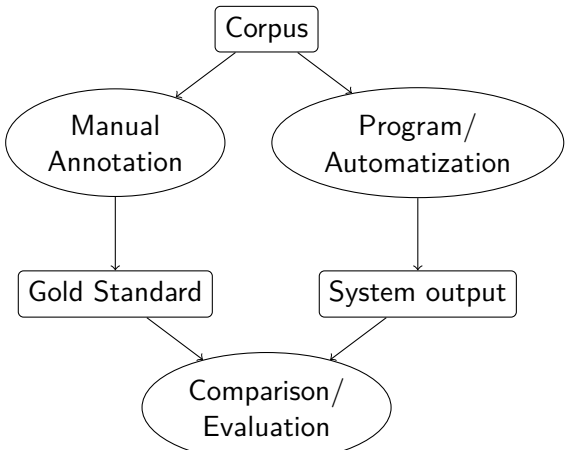
Evaluation Strategies

- Manual inspection by the developer: Run the tool, look at the results and decide
  - ⊖ ⚠ Difficult to reproduce, prone to biases, implicit standards
  - ⊕ Fast
- Manual inspection by an expert: Run the tool, hand it over to an expert and let them decide
  - ⊖ Difficult to reproduce, expensive
  - ⊕ More reliable
- Plug into an application that benefits from a component: Extrinsic evaluation
  - ⊖ Need evaluation for the application, impact of component not always clear
  - ⊕ Realistic evaluation (if it's a realistic application)
- Pre-defined reference data set
  - ⊖ Not always available, expensive, time-consuming
  - ⊕ Most reliable, easiest to reproduce
  - ML systems need annotated data anyway

Annotation Time!

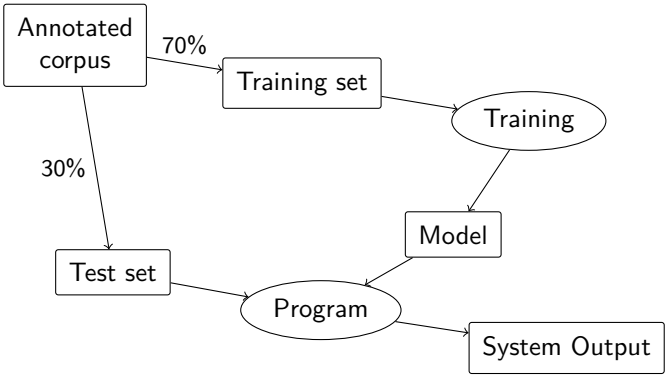
1. »Gefühlt ist die Lage wieder wie kurz nach der Einführung der Kontaktbeschränkungen: die eine Hälfte denkt, jetzt kann man wieder lustig bummeln gehen, die andere Hälfte ist total panisch und zählt Menschen im Park.«
2. »Besonders die Senioren werden von den Kontaktbeschränkungen schwer und hart getroffen, obgleich es zu ihrem eigenen Schutz dient.
   Wir dürfen in dieser schweren Zeit die Seniorinnen und Senioren nicht aus dem Blick verlieren.«
3. »Gute Regelung. Kontaktbeschränkungen max. 2 Personen.
   (Bemerkung: das sind immer die gleichen 2 Personen, sonst macht das keinen Sinn, das bitte noch klarstellen)
   1,5 bis 2 m Abstand
   Wenn immer es geht:
   #BleibtZuhause
   Eigener Hausstand OK.
   https://t.co/zuNpf0pjYr«

Experiments



Evaluation

- Goal: Predict the quality on new data
- The program cannot have seen the data, so that it's a realistic test



Evaluation

- Comparison of system output with gold standard
  - »Intrinsic evaluation«
- Two sets of predictions for the items
  - One set from the gold standard
  - One set from the system

Evaluation

- Comparison of system output with gold standard
  - »Intrinsic evaluation«
- Two sets of predictions for the items
  - One set from the gold standard
  - One set from the system

Example (Sentiment Analysis)

- Gold standard: `[1, 0, -1, -1]`
- System output: `[1, -1, 1, 0]`
- (positive: 1, neutral: 0, negative: -1)

Extrinsic Evaluation

- In some cases, GS data for a task doesn't exist or can't be created
- Extrinsic evaluation: Evaluate a downstream application
- Compare performance of downstream application
  - Without your component
  - With your component
- Assumptions
  - Your component helps performance of the downstream application
  - We know how to evaluate the downstream task

Extrinsic Evaluation

- In some cases, GS data for a task doesn't exist or can't be created
- Extrinsic evaluation: Evaluate a downstream application
- Compare performance of downstream application
  - Without your component
  - With your component
- Assumptions
  - Your component helps performance of the downstream application
  - We know how to evaluate the downstream task

Evaluation

Accuracy and Error Rate

- Accuracy
  - Percentage of correctly classified instances
  - Example above
    - $A = \frac{1}{4} = 0.25 = 25\%$
  - "the higher the better"

Evaluation

Accuracy and Error Rate

- Accuracy
  - Percentage of correctly classified instances
  - Example above
    - $A = \frac{1}{4} = 0.25 = 25\%$
  - "the higher the better"
- Error Rate
  - Percentage of *incorrectly* classified instances
  - Example above
    - $E = \frac{3}{4} = 0.75 = 75\%$
  - "the lower the better"

Evaluation

Accuracy and Error Rate

- Accuracy
  - Percentage of correctly classified instances
  - Example above
    - $A = \frac{1}{4} = 0.25 = 25\%$
  - "the higher the better"
- Error Rate
  - Percentage of *incorrectly* classified instances
  - Example above
    - $E = \frac{3}{4} = 0.75 = 75\%$
  - "the lower the better"
- $A + E = 1$, $E = 1 - A$ and $A = 1 - E$

Accuracy and Error Rate

Examples

- `G = [1, 0, 1], S = [0, 0, 1]`
  - $A = ?$

Accuracy and Error Rate

Examples

- `G = [1, 0, 1], S = [0, 0, 1]`
  - $A = ?$
- `G = ["f", "m", "u", "m", "f"], S = ["m", "f", "u", "m", "f"]`
  - $E = ?$

Accuracy and Error Rate

Examples

- `G = [1, 0, 1], S = [0, 0, 1]`
  - $A = ?$
- `G = ["f", "m", "u", "m", "f"], S = ["m", "f", "u", "m", "f"]`
  - $E = ?$

(We don't need the original data for evaluation, we are just comparing gold standard classes with system output.)

Baseline

A simple solution to the problem

- How well can the task be solved without investing (a lot of) time and work?
- What is a simple solution, and how well does it solve the problem?

Baseline

A simple solution to the problem

- How well can the task be solved without investing (a lot of) time and work?
- What is a simple solution, and how well does it solve the problem?
- Baselines are used for comparison in experiments
- ›Real‹ algorithms should be able to beat the baseline, i.e., achieve higher accuracy
- Baselines have obvious shortcomings, are not expected to work every time
  - Although, sometimes they work surprisingly well

Baseline

Group Exercises

What are reasonable baselines for these tasks?

- Detecting nouns in German texts
- Detecting sentence boundaries
- Detecting fake news
- Detecting the gender of dramatic characters (18-19th century)
- Predict the pos tag of the word after a determiner
- Given a corpus consisting of 'the Universal Declaration of Human Rights', 'Lord of the Rings' and the minutes of the European Parliament. Predict the origin of a random sentence.

Majority Baseline

- Select the most frequent category
- Works well in un-even data distributions
- Can be hard to beat
  - E.g. word sense disambiguation

Per Class Evaluation

- Accuracy gives us an overall score
- But we want to know more details:
  - Some classes are more important for applications
  - Error analysis!
- We want to evaluate per class (i.e., per polarity)

Sentiment Analysis

Different Kinds of Errors

Polarity	Document
positive	Awesome movie!
neutral	Great start, boring afterwards. Very good acting.
negative	Boring as hell
...	...

Table: Gold Standard

Sentiment Analysis

Different Kinds of Errors

Polarity	Document
positive	Awesome movie!
neutral	Great start, boring afterwards. Very good acting.
negative	Boring as hell
...	...

Table: Gold Standard

Variant	Output
GS	1, 0, -1, 1, 1, 0, -1, 1
Program 1	1, 0, -1, 1, 1, 0, 1, 1
Program 2	1, 0, -1, 1, -1, 0, -1, 1

Sentiment Analysis

Different Kinds of Errors

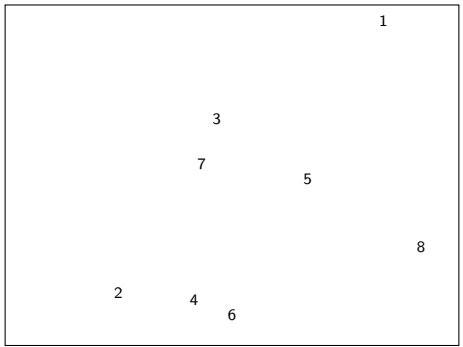


Figure: Visual representation of errors, focussing on -1 class

Sentiment Analysis

Different Kinds of Errors

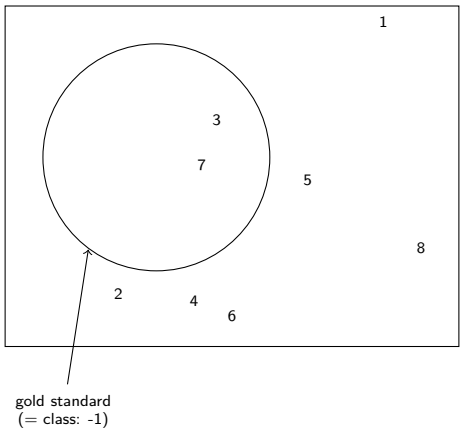


Figure: Visual representation of errors, focussing on -1 class

Sentiment Analysis

Different Kinds of Errors

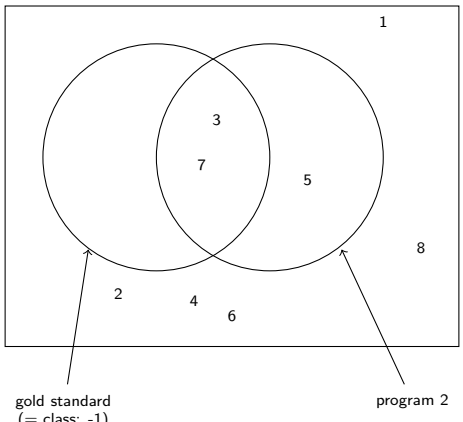
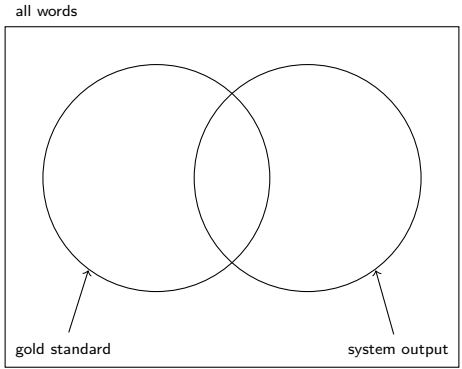
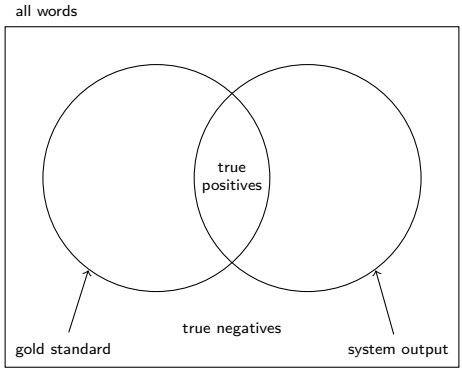


Figure: Visual representation of errors, focussing on -1 class

Different Kinds of Errors



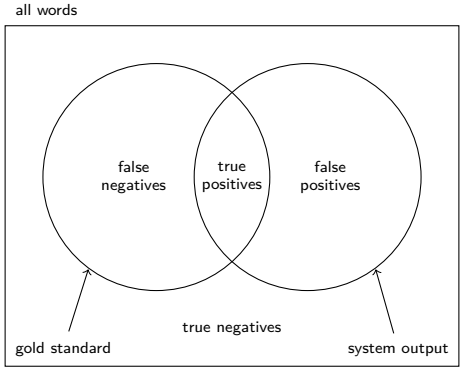
Different Kinds of Errors



true positive (tp) Correctly classified as target category

true negative (tn) Correctly classified as not target category

Different Kinds of Errors



true positive (tp) Correctly classified as target category

true negative (tn) Correctly classified as not target category

false positive (fp) Incorrectly classified as target category

false negative (fn) Incorrectly classified as not target category

Accuracy, revisited

Accuracy: Percentage of correctly classified instances

$$A = \frac{tp + tn}{tp + tn + fp + fn}$$

Accuracy, revisited

Accuracy: Percentage of correctly classified instances

$$A = \frac{tp + tn}{tp + tn + fp + fn}$$

Error rate: Percentage of incorrectly classified instances

$$E = \frac{fp + fn}{tp + tn + fp + fn}$$

Precision and Recall

Given the documents that the system marked as -1, how many of those are really -1?

Precision and Recall

Given the documents that the system marked as -1, how many of those are really -1?

$$\text{Precision} \quad P = \frac{tp}{tp + fp}$$

Precision and Recall

Given the documents that the system marked as -1, how many of those are really -1?

$$\text{Precision} \quad P = \frac{tp}{tp + fp}$$

How many of the -1 documents did the system find?

Precision and Recall

Given the documents that the system marked as -1, how many of those are really -1?

$$\text{Precision} \quad P = \frac{tp}{tp + fp}$$

How many of the -1 documents did the system find?

$$\text{Recall} \quad R = \frac{tp}{tp + fn}$$

Precision and Recall

- Enumerator: tp

Precision and Recall

- Enumerator: tp
- Precision
  - Denominator: $tp + fp$
  - Number of things that the system labelled as target category (correct and incorrect)
- Recall
  - Denominator: $tp + fn$
  - Number of things that the gold standard contained as target category (what the system should have found)

Precision and Recall

Importance/Weighting

- Weighting between P and R is application-dependent (and difficult to decide!)
- Guiding question: Which kind of error is more severe?

Precision and Recall

Importance/Weighting

- Weighting between P and R is application-dependent (and difficult to decide!)
- Guiding question: Which kind of error is more severe?
- If findings are inspected by humans
  - Precision errors are easy to spot, but recall errors cannot be detected
  - But: humans tend to trust computers

Precision and Recall

Importance/Weighting

- Weighting between P and R is application-dependent (and difficult to decide!)
- Guiding question: Which kind of error is more severe?
- If findings are inspected by humans
  - Precision errors are easy to spot, but recall errors cannot be detected
  - But: humans tend to trust computers
- Severity of consequences

Precision and Recall

Importance/Weighting

- Weighting between P and R is application-dependent (and difficult to decide!)
- Guiding question: Which kind of error is more severe?
- If findings are inspected by humans
  - Precision errors are easy to spot, but recall errors cannot be detected
  - But: humans tend to trust computers
- Severity of consequences

Example (Test performance in a pandemic)

- Individual health: Mistakenly being in quarantine is a severe limitation, and might have economic consequences
- Public health: Find more infections, even if it means a few people are mistakenly put in quarantine

Precision and Recall

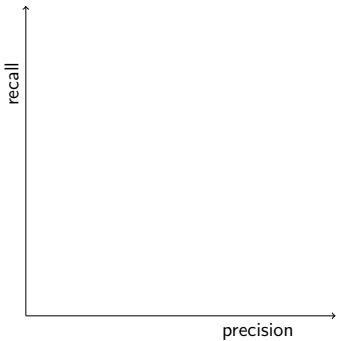
Thresholds

- Sometimes, we have a single parameter that directly controls P and R
  E.g., a threshold for document similarity
  - Lower threshold: More documents are included ⇒ Higher recall, at the cost of precision
  - Higher threshold: Less documents are included ⇒ Higher precision, at the cost of recall

Precision and Recall

Thresholds

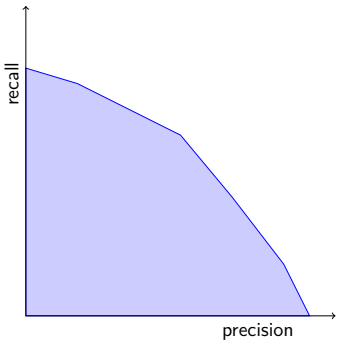
- Sometimes, we have a single parameter that directly controls P and R
  E.g., a threshold for document similarity
  - Lower threshold: More documents are included ⇒ Higher recall, at the cost of precision
  - Higher threshold: Less documents are included ⇒ Higher precision, at the cost of recall
- AUC: Area under curve



Precision and Recall

Thresholds

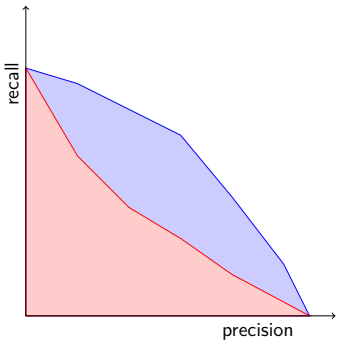
- Sometimes, we have a single parameter that directly controls P and R
  E.g., a threshold for document similarity
  - Lower threshold: More documents are included ⇒ Higher recall, at the cost of precision
  - Higher threshold: Less documents are included ⇒ Higher precision, at the cost of recall
- AUC: Area under curve



Precision and Recall

Thresholds

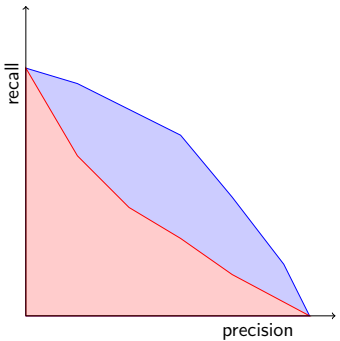
- Sometimes, we have a single parameter that directly controls P and R
  E.g., a threshold for document similarity
  - Lower threshold: More documents are included ⇒ Higher recall, at the cost of precision
  - Higher threshold: Less documents are included ⇒ Higher precision, at the cost of recall
- AUC: Area under curve



Precision and Recall

Thresholds

- Sometimes, we have a single parameter that directly controls P and R
  E.g., a threshold for document similarity
  - Lower threshold: More documents are included ⇒ Higher recall, at the cost of precision
  - Higher threshold: Less documents are included ⇒ Higher precision, at the cost of recall
- AUC: Area under curve



- AUC(blue) $>$ AUC(red):
  Blue system better

F-Score

- Sometimes, it is convenient to combine precision and recall into a single number
- F-Score is common way to do that
  (it's a fancy way of averaging)
  - β can be used to weight precision and recall differently
  - $\beta = 1$ means equal weighting
- F-Measure corresponds to the harmonic mean

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

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

Data Sets for Different Purposes

- Training data set: Count words, estimate probabilities
- Test data set: Simulate application to see how well it works
- Application data set: Do the actual application
  - Usually skipped in research

Data Sets for Different Purposes

- Training data set: Count words, estimate probabilities
- Test data set: Simulate application to see how well it works
- Application data set: Do the actual application
  - Usually skipped in research
- Development data set: Write code, test implementation on dummy examples, fix bugs
- Validation data set: Sometimes used for smoothing or hyperparameter tuning

Data Sets for Different Purposes

- Training data set: Count words, estimate probabilities
- Test data set: Simulate application to see how well it works
- Application data set: Do the actual application
  - Usually skipped in research
- Development data set: Write code, test implementation on dummy examples, fix bugs
- Validation data set: Sometimes used for smoothing or hyperparameter tuning

Generating Purpose-Specific Data Sets

- Annotated data is expensive and often the bottleneck

Generating Purpose-Specific Data Sets

- Annotated data is expensive and often the bottleneck
- Different ways to use an existing annotated data set

Generating Purpose-Specific Data Sets

- Annotated data is expensive and often the bottleneck
- Different ways to use an existing annotated data set

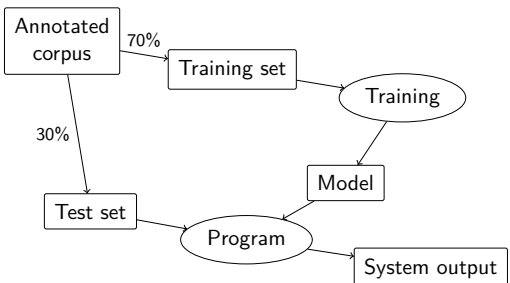


Figure: Percentage split

Generating Purpose-Specific Data Sets

- Annotated data is expensive and often the bottleneck
- Different ways to use an existing annotated data set

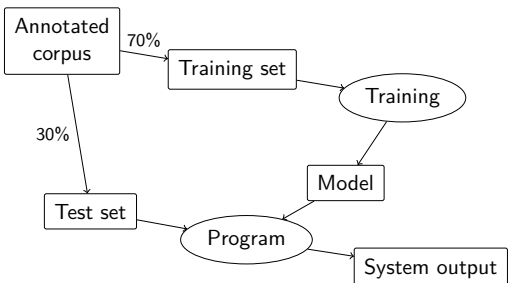
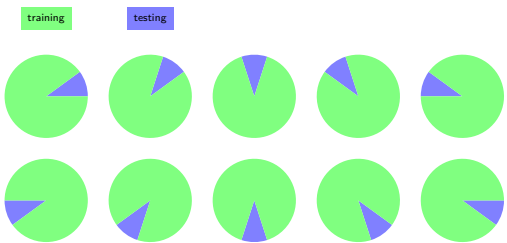


Figure: Percentage split



Calculate P/R/F individually, then average

Figure: Cross Validation

Randomness

- Some test options or algorithms involve random numbers
  - E.g., cross validation
- Results could be unrealistically good, by chance

Randomness

- Some test options or algorithms involve random numbers
  - E.g., cross validation
- Results could be unrealistically good, by chance
- Simple solution: Run the experiments repeatedly (e.g., 1000 times)

Summary

- Language modeling
  - Given some history, predict the next word
  - Use cases: Smart phone, ...
  - Maximum Likelihood estimation: Easy, but problematic
  - Lidstone's Law: Smoothing
    - Other smoothing techniques exist
  - Cross validation
- Machine Learning Evaluation
  - Classification

References I


Manning, Christopher D./Hinrich Schütze (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, Massachusetts and London, England: MIT Press.