Automatic prediction of linguistic properties, Evaluation, Task types

VL Sprachliche Informationsverarbeitung

Nils Reiter

nils.reiter@uni-koeln.de

November 10, 2022

Winter term 2022/23

INSTITUT FÜR
DIGITAL HUMANITIES
UNIVERSITÄT ZU KÖLN

Section 1

Automatic prediction of linguistic properties

Introduction

- Focus of computational linguistics
- Linguistic understanding: Part of speech, lemma, syntactic structure, semantic representation, ...
- Detection of content-related aspects: Named entities, sentiment, speech acts, ...
- Applications: Machine translation, question answering, dialoge systems, ...

Introduction

- Focus of computational linguistics
- Linguistic understanding: Part of speech, lemma, syntactic structure, semantic representation, ...
- Detection of content-related aspects: Named entities, sentiment, speech acts, ...
- Applications: Machine translation, question answering, dialoge systems, ...
- How to do that? Machine learning, nowadays

From Rules to Neural Networks

Rule-based part of speech tagging

```
# list of German determiners
determiners = ["der","die","ein",...]

for token in tokens:
  if token[0].islower() and
    token.endswith("en"):
    return "VERB"
  elif token[0].isupper():
    return "NAME"
  else:
     if token in determiners:
       return "DET"
...
```

From Rules to Neural Networks

Rule-based part of speech tagging

```
# list of German determiners
determiners = ["der","die","ein",...]

for token in tokens:
  if token[0].islower() and
    token.endswith("en"):
    return "VERB"
  elif token[0].isupper():
    return "NAME"
  else:
     if token in determiners:
       return "DET"
...
```

Which token properties are used here?

From Rules to Neural Networks

Rule-based part of speech tagging

```
# list of German determiners
determiners = ["der","die","ein",...]

for token in tokens:
  if token[0].islower() and
    token.endswith("en"):
    return "VERB"
  elif token[0].isupper():
    return "NAME"
  else:
     if token in determiners:
       return "DET"
...
```

Which token properties are used here?

- Casing (upper/lower)
- Suffix (en)
- word list (Determiners)

From Rules to Neural Networks

Rule-based part of speech tagging

```
# list of German determiners
determiners = ["der","die","ein",...]

for token in tokens:
  if token[0].islower() and
    token.endswith("en"):
    return "VERB"
  elif token[0].isupper():
    return "NAME"
  else:
     if token in determiners:
       return "DET"
...
```

Which token properties are used here?

- Casing (upper/lower)
- Suffix (en)
- word list (Determiners)

Which properties are *not* used?

From Rules to Neural Networks

Rule-based part of speech tagging

```
# list of German determiners
determiners = ["der","die","ein",...]

for token in tokens:
  if token[0].islower() and
    token.endswith("en"):
    return "VERB"
  elif token[0].isupper():
    return "NAME"
  else:
     if token in determiners:
       return "DET"
...
```

Which token properties are used here?

- Casing (upper/lower)
- Suffix (en)
- word list (Determiners)

Which properties are *not* used?

- Prefixes
- Token length
- Sequence: Previous tag

From Rules to Neural Networks

'Classical' machine learning

```
tokens = ["the", "dog", "barks"]
tags = ["DET", "NN", "VBZ"]

table = extract_features(tokens)

model = train(table, tags)
```

- Token properties $\rightarrow$ features
- Feature extraction / feature engineering
  - Finding useful features based on domain knowledge (e.g., linguistic knowledge)
  - 'Playground': What works well can really only be known after experiments

From Rules to Neural Networks

'Classical' machine learning

```
tokens = ["the", "dog", "barks"]
tags = ["DET", "NN", "VBZ"]

table = extract_features(tokens)

model = train(table, tags)
```

- Token properties $\rightarrow$ features
- Feature extraction / feature engineering
  - Finding useful features based on domain knowledge (e.g., linguistic knowledge)
  - 'Playground': What works well can really only be known after experiments
- Training: Estimate which features in which order allow best decisions
  - A large collection of algorithms has been developed: Decision trees, support vector machines, naive Bayes, …
  - Training data needed!

From Rules to Neural Networks

'Classical' machine learning

- Annotated data
  - Used for training
  - Used for evaluation

From Rules to Neural Networks

‘Classical’ machine learning

- Annotated data
  - Used for training
  - Used for evaluation
- Three stages
  - Training (train a model with annotated data)
  - Testing (test an existing model on annotated data)
  - Application (use an existing model)

From Rules to Neural Networks

'Classical' machine learning

- Annotated data
  - Used for training
  - Used for evaluation
- Three stages
  - Training (train a model with annotated data)
  - Testing (test an existing model on annotated data)
  - Application (use an existing model)
- This still applies in the deep learning realm

From Rules to Neural Networks

Deep learning

- No more feature engineering
  - Let the computer figure out what it needs to know
- More computing (and more data)
- Black box
  - Intermediate states not interpretable for us humans
  - Only input and output can be understood

Machine Learning

- Collection of techniques for automatic
  - decision making
  - pattern detection
  - data analysis
- Machine learning vs. rule-based systems
  - Rule-based: Decision rules are hand-coded
    - if/then/else, ...
  - Machine learning: Decision rules are 'learned' from data
  - Data is used to estimate weights and criteria

Understanding Machine Learning

- Levels of understanding
  - Intuition
  - Formalization (math)
  - Implementation (code)

Understanding Machine Learning

- Levels of understanding
  - Intuition
  - Formalization (math)
  - Implementation (code)
- Areas to distinguish
  - Learning algorithm
  - Prediction model
  - Data preparation
    - Feature extraction (classical ML)
    - Shape of input data

Section 2

Types of Tasks

Task types

- Many ML/DL/NLP tasks are structurally similar
- Structurally similar: The same system can be used, all differences can be encoded in the training data

Task types

- Many ML/DL/NLP tasks are structurally similar
- Structurally similar: The same system can be used, all differences can be encoded in the training data

Example

- Part of speech tagging: Each token gets a label
  - Labels: NN, VBZ, DET, ADJA, ADJD, …
- Named entity recognition: Each token gets a label
  - O, B-PER, I-PER, B-LOC, I-LOC, …

Task types

- Many ML/DL/NLP tasks are structurally similar
- Structurally similar: The same system can be used, all differences can be encoded in the training data

Example

- Part of speech tagging: Each token gets a label
  - Labels: NN, VBZ, DET, ADJA, ADJD, …
- Named entity recognition: Each token gets a label
  - O, B-PER, I-PER, B-LOC, I-LOC, …

- Two important task types for NLP
  - Text classification: An entire text is classified (e.g., genre, sentiment, …)
  - Sequence labeling: Each individual word is classified (e.g., pos-tagging, …)

Task types

Text classification

- Texts belong to a class of texts

Examples

- Customer reviews $\rightarrow$ sentiment
- Novel $\rightarrow$ genre (fiction, non-fiction, …)
- Posting $\rightarrow$ $\pm$ hate speech
- E-mail $\rightarrow$ $\{$spam, not spam, really important$\}$

Task types

Sequence labeling

- Words (or sequences of words) belong to classes
  - Sequence labeling: Classification + sequential dependency between classes

Examples

- Words $\rightarrow$ part of speech (noun, verb, adjective, ...)
- Words $\rightarrow$ proper noun
- Paragraphs $\rightarrow$ $\pm$ narrative scene
- **?** Collected works by Shakespeare $\rightarrow$ {comedy, tragedy}

Task types

Sequence labeling

- Words (or sequences of words) belong to classes
  - Sequence labeling: Classification + sequential dependency between classes

Examples

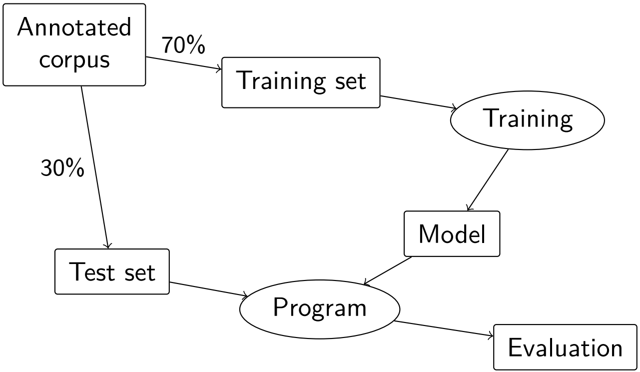
- Words $\rightarrow$ part of speech (noun, verb, adjective, …)
- Words $\rightarrow$ proper noun
- Paragraphs $\rightarrow$ $\pm$ narrative scene
- **?** Collected works by Shakespeare $\rightarrow$ {comedy, tragedy}
  - Sequence of works probably irrelevant

Section 3

Evaluation of Machine Learning Systems

Training and Testing

- Goal: Apply the model on new data (and estimate its performance then)
- The program cannot have seen the data, so that it is a realistic test



Evaluation

- We *always* want to know how well a model works
- Straightforward evaluation: Comparison with a gold standard

Evaluation

- We *always* want to know how well a model works
- Straightforward evaluation: Comparison with a gold standard
- Most simple metric: Accuracy
  - Percentage of correctly classified instances (the higher the better)
  - Inverse: Error rate (percentage of incorrectly classified instances)

Evaluation

- We *always* want to know how well a model works
- Straightforward evaluation: Comparison with a gold standard
- Most simple metric: Accuracy
  - Percentage of correctly classified instances (the higher the better)
  - Inverse: Error rate (percentage of incorrectly classified instances)
- What could be problems with this metric?

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

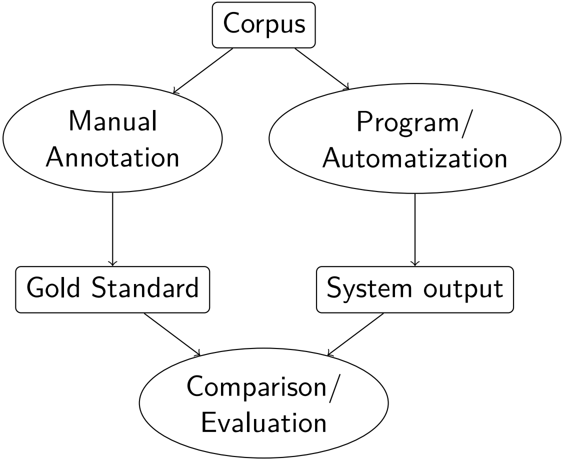
Evaluation Strategies

- 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

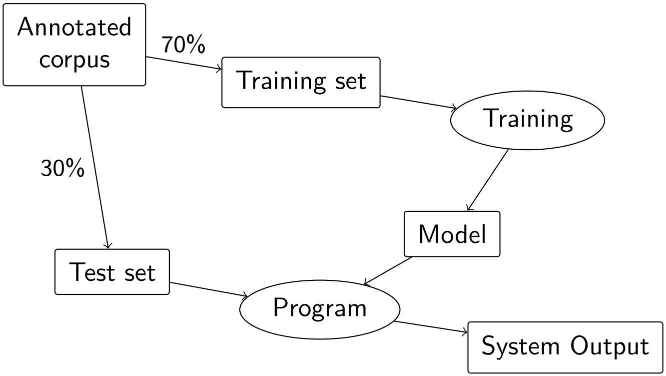
- Manual inspection by the developer: Run the tool, look at the results and decide
  - − ⚠ Difficult to reproduce, prone to biases, implicit standards
  - + Fast
- Manual inspection by an expert: Run the tool, hand it over to an expert and let them decide
  - − Difficult to reproduce, expensive
  - +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

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]`
  - $A = \frac{1}{3}$
- `G = ["f", "m", "u", "m", "f"], S = ["m", "f", "u", "m", "f"]`
  - $E = \frac{2}{5}$

(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

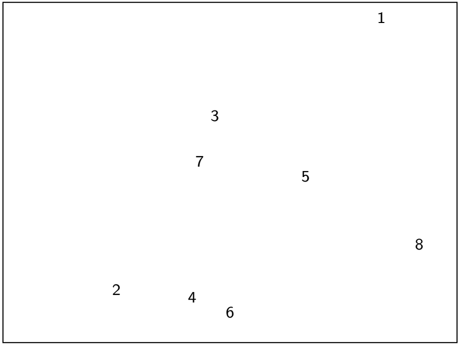


Figure: Visual representation of errors, focussing on -1 class

Sentiment Analysis

Different Kinds of Errors

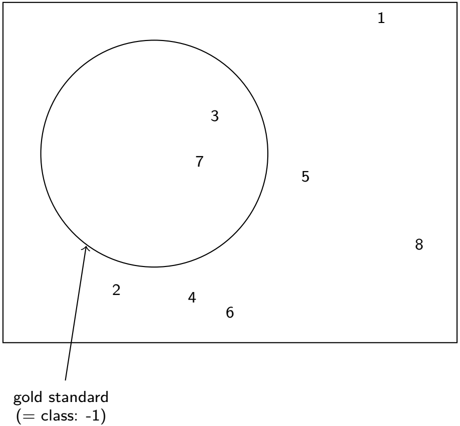


Figure: Visual representation of errors, focussing on -1 class

Sentiment Analysis

Different Kinds of Errors

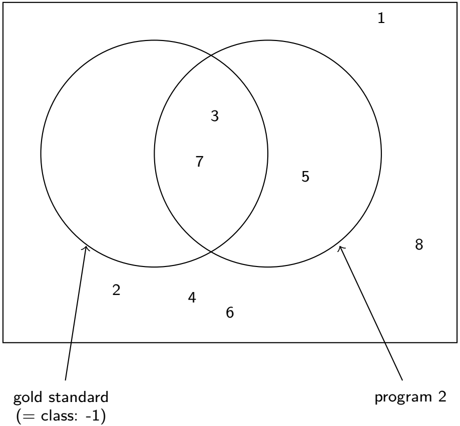
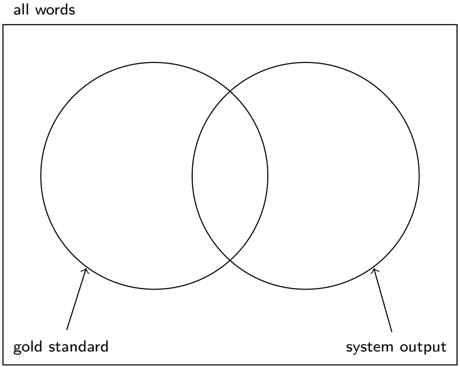
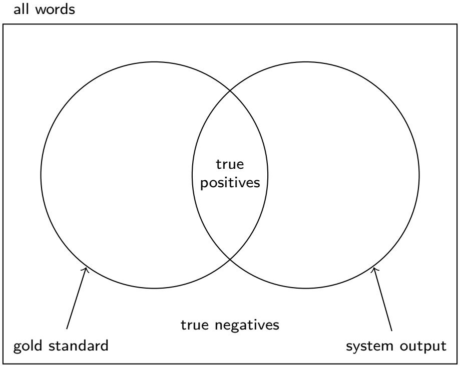


Figure: Visual representation of errors, focussing on -1 class

Different Kinds of Errors



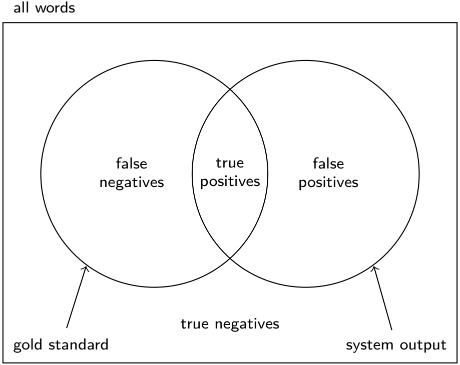
Different Kinds of Errors



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

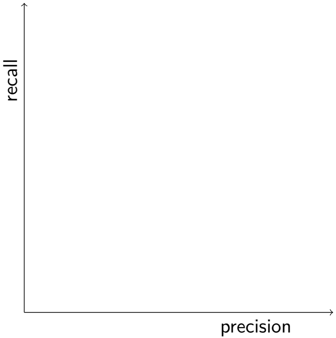
Thresholds

- 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 $\Rightarrow$ Higher recall, at the cost of precision
  - Higher threshold: Less documents are included $\Rightarrow$ Higher precision, at the cost of recall

Precision and Recall

Thresholds

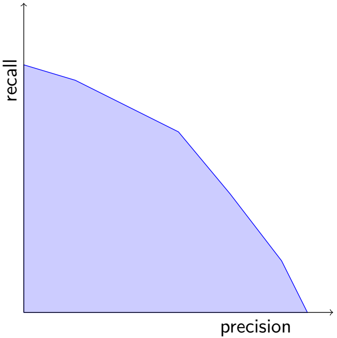
- Sometimes, we have a single parameter that directly controls P and R
  E.g., a threshold for document similarity
  - Lower threshold: More documents are included ⇒ Higher recall, at the cost of precision
  - Higher threshold: Less documents are included ⇒ Higher precision, at the cost of recall
- AUC: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



Precision and Recall

Thresholds

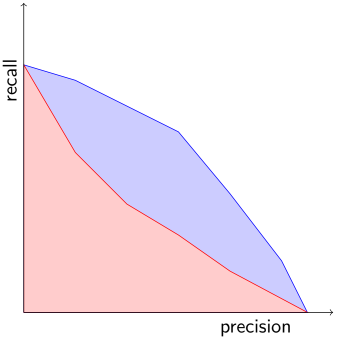
- Sometimes, we have a single parameter that directly controls P and R
  E.g., a threshold for document similarity
  - Lower threshold: More documents are included ⇒ Higher recall, at the cost of precision
  - Higher threshold: Less documents are included ⇒ Higher precision, at the cost of recall
- AUC: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



Precision and Recall

Thresholds

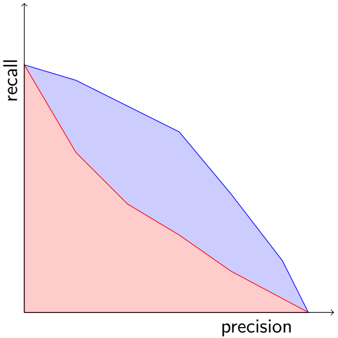
- Sometimes, we have a single parameter that directly controls P and R
  E.g., a threshold for document similarity
  - Lower threshold: More documents are included ⇒ Higher recall, at the cost of precision
  - Higher threshold: Less documents are included ⇒ Higher precision, at the cost of recall
- AUC: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

Section 4

Summary

Summary

- Task Types
  - Classification: One item belongs individually to one category
  - Sequence labeling: Each item in a sequence belongs to a category, and the items have dependencies
- Evaluation
  - Accuracy/error rate: Percentage of correctly/incorrectly classified instances
  - Precision/recall: Calculated over true positives, false positives and false negatives
  - Area under curve: Metric for systems with thresholds
  - Baseline: Comparison system(s)
  - Use different data sets for different purposes
- Next week: Bring your computer!