Einführung in die Computerlinguistik

Statistische Modellierung (+ Nachtrag Merkmalsstrukturen)

WS 2009/2010

Manfred Pinkal

Grammatische Merkmale

- Wie finden Sie die angehängten Bilder? Das sind Fotos, die im Rahmen des TALK-Projektes entstanden sind, uns gehören, und von BMW schon freigegeben waren. Außerdem vermitteln sie besser den Bezug zur Forschung.

Explizite Kodierung von Merkmalen

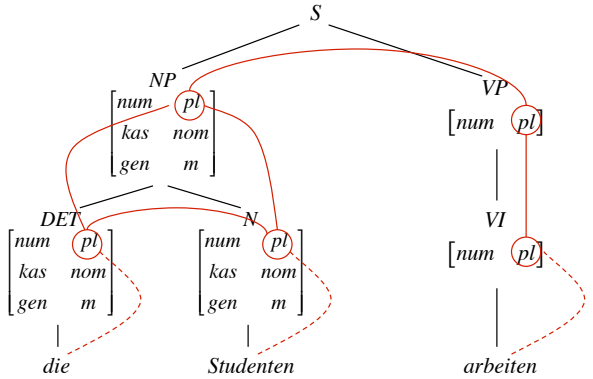
$$S \rightarrow NP\begin{bmatrix} num & sg \\ kas & nom \end{bmatrix} VP[num \quad sg] \qquad S \rightarrow NP\begin{bmatrix} num & pl \\ kas & nom \end{bmatrix} VP[num \quad pl]$$

$$VP[num \quad sg] \rightarrow VI[num \quad sg] \qquad VP[num \quad pl] \rightarrow VI[num \quad pl]$$

$$NP\begin{bmatrix} num & sg \\ kas & nom \\ gen & m \end{bmatrix} \rightarrow Det\begin{bmatrix} num & sg \\ kas & nom \\ gen & m \end{bmatrix} N\begin{bmatrix} num & sg \\ kas & nom \\ gen & m \end{bmatrix}$$

$$NP\begin{bmatrix} num & pl \\ kas & nom \\ gen & m \end{bmatrix} \rightarrow Det\begin{bmatrix} num & pl \\ kas & nom \\ gen & m \end{bmatrix} N\begin{bmatrix} num & pl \\ kas & nom \\ gen & m \end{bmatrix}$$

Explizite Kodierung von Merkmalen



Kontextfreie Grammatik mit Merkmalsstrukturen

- Konstituenten werden mit Paaren aus Kategoriensymbolen und Merkmalsstrukturen ausgezeichnet.
- Eine Merkmalsstruktur ist eine Menge von Merkmal-Wert-Paaren (auch „Attribut-Wert-Paaren“): Die Merkmalsstruktur des NP-Knotens im Beispiel hat drei Merkmale, das erste besteht aus dem Attribut *„num“* und dem atomaren Wert *„sg“.*
- Die explizite Kodierung von Merkmalen erlaubt die Formulierung von Bedingungen / Constraints, z.B. „Numerus von NP und Numerus von VP sind identisch“, oder „Subjekts-NP hat Kasus Nominativ“.
- Regeln der Grammatik sind zweiteilig: Sie bestehen aus einer Ersetzungsregel (wie üblich über Kategorien und lexikalische Ausdrücke formuliert) und einer Menge von Constraints über Merkmalsstrukturen.

CFG mit Merkmalsconstraints, Beispiel

S → NP VP
Numerus der NP = Numerus der VP
Kasus der NP = nom

VP → VI
Numerus der VP = Numerus von VI

VP → VT NP
Numerus der VP = Numerus von VT
Kasus der NP = akk

NP → DET N
Numerus von DET = Numerus von N
Genus von DET = Genus von N
Kasus von DET = Kasus von N
Numerus der NP = Numerus von N
Genus der NP = Genus von N
Kasus der NP = Kasus von N

VI → *arbeitet*
Numerus von VI = sg

VI → *arbeiten*
Numerus von VI = pl

N → Student
Numerus von N = sg
Genus von N = m
Kasus von N = nom

DET → *der*
Numerus von DET = sg
Genus von DET = m
Kasus von DET = nom

CFG mit Merkmalsconstraints, Beispiel

S → NP VP
<NP num> = <VP num>
<NP kas> = nom

VP → VI
<VP num> = <VI num>

VP → VT NP
<VP num> = <VT num>
<NP kas> = akk

NP → DET N
<DET num> = <N num>
<DET gen> = <N gen>
<DET kas> = <N kas>
<NP num> = <N num>
<NP gen> = <N gen>
<NP kas> = <N kas>

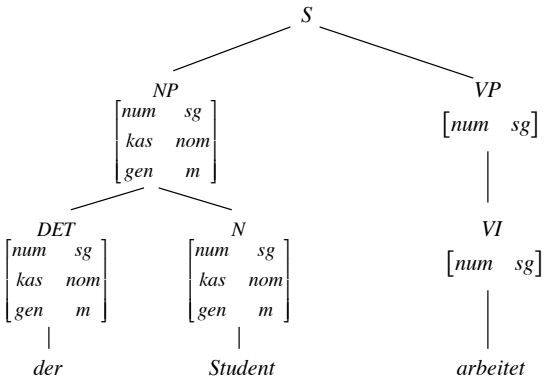
VI → arbeitet
<VI num> = sg

VI → arbeiten
<VI num> = pl

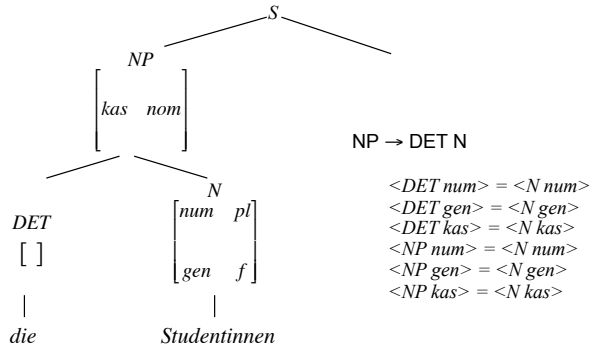
N → Student
<N num> = sg
<N gen> = m
<N kas> = nom

DET → der
<DET num> = sg
<DET gen> = m
<DET kas> = nom

Die Anwendung von Merkmals-Constraints



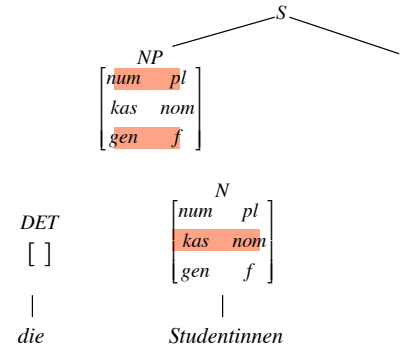
Die Anwendung von Merkmals-Constraints



NP → DET N

<DET num> = <N num>
<DET gen> = <N gen>
<DET kas> = <N kas>
<NP num> = <N num>
<NP gen> = <N gen>
<NP kas> = <N kas>

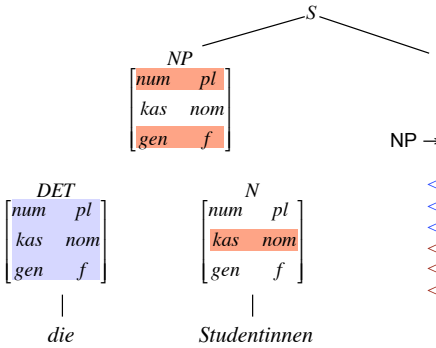
Die Anwendung von Merkmals-Constraints



NP → DET N

<DET num> = <N num>
<DET gen> = <N gen>
<DET kas> = <N kas>
<NP num> = <N num>
<NP gen> = <N gen>
<NP kas> = <N kas>

Die Anwendung von Merkmalsconstraints



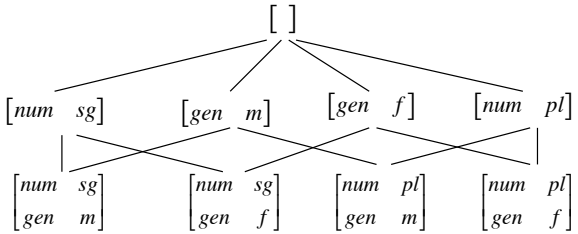
NP → DET N

<DET num> = <N num>
<DET gen> = <N gen>
<DET kas> = <N kas>
<NP num> = <N num>
<NP gen> = <N gen>
<NP kas> = <N kas>

Unifikation

- Bei der Anwendung von Pfadgleichungen wird nicht die Identität aktueller Werte abgeprüft, sondern
  - die Kompatibilität der vorhandenen Merkmalsinformation getestet, indem
  - die Information auf beiden Seiten der Gleichung zusammengeführt.
- Dies geschieht, indem die entsprechenden Merkmalsinformationen unifiziert werden. Wir schreiben A⊔B.
- Resultat der Unifikation ist die (allgemeinste) Merkmalsstruktur, die die Information aus beiden Merkmalsstrukturen umfasst, wenn es eine solche Struktur gibt. Ansonsten schlägt sie fehl.
- Wenn A allgemeiner ist als B ist, d.h., wenn Merkmalsstruktur B die Information aus Merkmalsstruktur A komplett enthält, sprechen wir von Subsumption: A subsumiert B, oder A ⊑ B.
- Das Resultat der Unifikation von A und B: A⊔B ist die allgemeinste Struktur C, sodass A ⊑ C und B ⊑ C, d.h., die Struktur, die genau die gemeinsame in A und B enthaltene Information enthält.

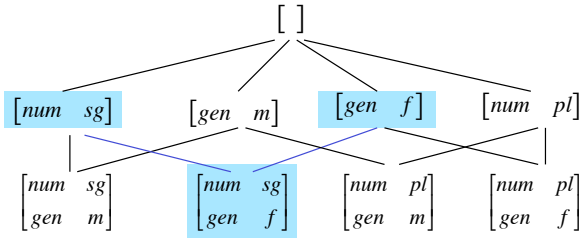
Subsumption



Die Graphik stellt die Halbordnung über Merkmalsstrukturen dar, die durch die Subsumptionsrelation etabliert wird: Die obere Struktur subsumiert jeweils die untere, zum Beispiel

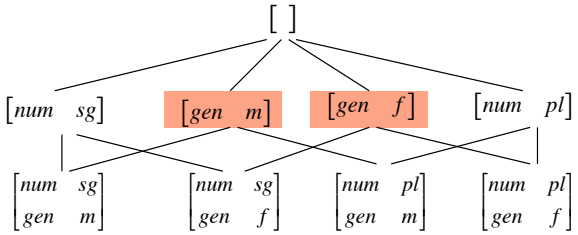
$$[\ ] \sqsubseteq \quad [gen \quad m] \quad \sqsubseteq \begin{bmatrix} num & sg \\ gen & m \end{bmatrix}$$

Subsumption und Unifikation



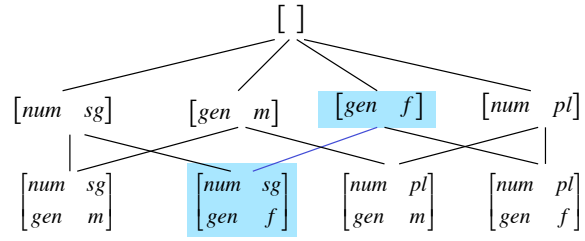
$$[num \quad sg] \sqcup [gen \quad f] = \begin{bmatrix} num & sg \\ gen & f \end{bmatrix}$$

Subsumption und Unifikation



$$[gen \quad m] \sqcup [gen \quad f] = \textit{fail}$$

Subsumption und Unifikation

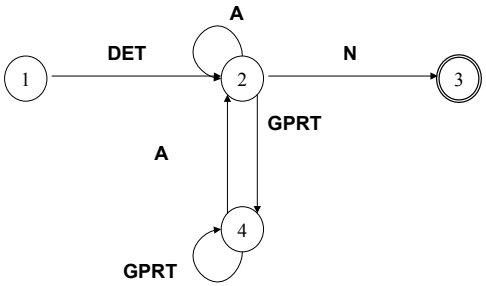


$$\begin{bmatrix} num & sg \\ gen & f \end{bmatrix} \sqcup [gen \quad f] = \begin{bmatrix} num & sg \\ gen & f \end{bmatrix}$$

allgemein: wenn A $\sqsubseteq$ B, so ist A$\sqcup$B = B

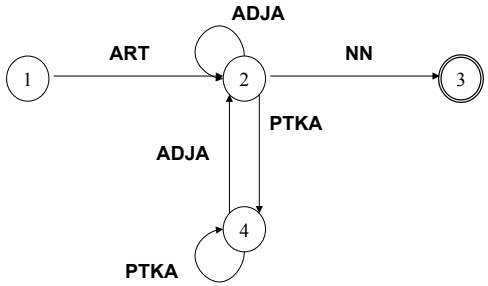
Der alte NP-Automat

die ziemlich interessante Vorlesung
das recht neue sehr sehr schöne rote Dach



Ein NP-Automat

... mit Wortart-Tags (Standardbezeichnungen für Wortarten)
aus dem Stuttgart-Tübinger Tagset (STTS)
`www.ims.uni-stuttgart.de/projekte/corplex/TagSets/stts-table.html`



Ein NP-Automat

- Erkennung von NPs mit Wortartmustern setzt als Vorverarbeitungsschritt die Annotation der Eingabe mit Wortart-Tags voraus:

die ziemlich interessante Vorlesung
↓ ↓ ↓ ↓
ART PTKA ADJA NN

- Aber:

einige zu laute Vorlesungen
↓ ↓ ↓ ↓
? ? ? NN

I made her duck
↓ ↓ ↓ ↓
PPER VVFIN ? ?

Ein NP-Automat

- Mögliche Wortarten der Wortform zu?

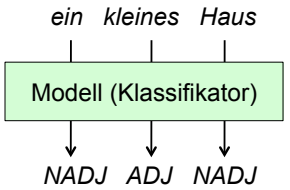
Die Tür ist zu.	*ADV*
Er ist nett zu mir.	*APPR*
Er ist mir zu nett.	*PTKA*
Er versucht zu schlafen.	*PTKZU*

Wortart-Tagger

- Eindeutige Wortartinformation ist für viele sprachtechnologische Anwendungen wichtig. Sie ist hilfreich bei Lemmatisierung, morphologischer und syntaktischer Analyse.
- Wortartinformation wird durch „Wortart-Tagger“ oder „POS-Tagger“ bereitgestellt (POS für „part of speech“, engl. „tag“ ist die Markierung).
- Wortart-Tagger sind heute Standardwerkzeuge der Sprachverarbeitung – wie Morphologie-Systeme. Anders als Morphologien arbeiten sie in der Regel mit statistischer Modellierung.

Beispielaufgabe: Adjektiverkennung

- Wortart-Tagger für das Deutsche müssen zwischen ca. 50 Kategorien wählen.
- Wir betrachten eine Teilaufgabe: Die Beantwortung der Frage, ob es sich bei einem Vorkommen eines Wortes in einem Text um ein Adjektiv handelt (also eine binäre Klassifikationsaufgabe).



Grundsätzliche Alternativen

- Lexikonbasiertes Modell:
  - Prüfe (mithilfe von Lemmatisierer/Morphologie), ob die Wortform zu einem Lemma im Lexikon gehört, entnehme die Wortartinformation dem Lexikon
  - Probleme: Mehrdeutigkeit und Abdeckung
- Regelbasiertes/ symbolisches Modell:
  - Formuliere ein System von wenn-dann-Regeln:

    Wenn `<Merkmal1>, ..., <Merkmaln>` vorliegen, dann weise `<Wortart>` zu.
- Datenbasiertes/ statistisches Modell:
  - Annotiere ein Korpus mit Wortart-Tag, lerne statistische Abhängigkeiten zwischen Merkmalen des Wortes (Merkmalsmustern) und seiner Wortart.

Datenanalyse: Informative Merkmalsmuster

- Woran erkenne ich, dass ein Wortvorkommen ein Adjektiv ist?

  eine laute Vorlesung

  das siebente Übungsblatt
- Spezifikation geeigneter Merkmale für die Modellierung:
  - Merkmale, die direkt abgelesen oder mit großer Sicherheit automatisch ermittelt werden können
  - Informativ in Bezug auf die Klassifikationsaufgabe sind
- Beispiele:
  - Groß-/Kleinschreibung
  - Groß-/Kleinschreibung des Folgewortes
  - Vorgängerwort ist Artikel/ kein Artikel

Symbolische Regeln

Beispiele:

- Nachfolger großgeschrieben → ADJ
  - *das Haus* – Korrektheitsproblem
- Nachfolger großgeschrieben, selbst kein Artikel → ADJ
  - *Ich sehe Peter* – Korrektheitsproblem
- Nachfolger großgeschrieben, selbst kein Artikel, Vorgänger Artikel → ADJ
  - große Bedenken – Vollständigkeitsproblem

Es ist schwer, Regeln zu schreiben, die die Abhängigkeit der Wortart von Merkmalsmustern korrekt und vollständig erfassen.

Alternative: Lernen des Zusammenhangs von Merkmalsmustern und Wortarten aus Korpora!

Statistisches Modell

- Wir spezifizieren eine Menge von geeigneten Merkmalen
  - Beispiel: selbst groß/klein, Folgewort groß/klein, Vorgänger Artikel/ kein Artikel
- Wir wählen ein Textkorpus aus („Trainingsdaten") und annotieren die Daten manuell mit Wortarttags
  - Beispiel: ADJ/ NADJ
- Wir extrahieren für jede Instanz (Textwort) das zugehörige Merkmalsmuster.

Informative Merkmalsmuster

Text:	*Vor*	*dem*	*kleinen*	*Haus*	*steht*	*ein*	*großer*	*Baum*
Annotation	NADJ	NADJ	ADJ	NADJ	NADJ	NADJ	ADJ	NADJ
Merkmale:								
n groß	+	-	-	+	-	-	-	+
n+1 groß	-	-	+	-	-	-	+	-
n-1 Art.	-	-	+	-	-	-	+	-

Statistisches Modell

- Wir spezifizieren eine Menge von geeigneten Merkmalen
  - Beispiel: selbst groß/klein, Folgewort groß/klein, Vorgänger Artikel/ kein Artikel
- Wir wählen ein Textkorpus aus („Trainingsdaten") und annotieren die Daten manuell mit Wortarttags.
  - Beispiel: ADJ/ NADJ
- Wir extrahieren für jede Instanz (Textwort) das zugehörige Merkmalsmuster.
- Wir trainieren den Klassifikator, sodass er für jede Instanz auf der Grundlage ihres Merkmalsmusters („Ereignis") eine Wahrscheinlichkeit für die Wortarttags (Klassen) ermittelt.
- Wir wenden den Klassifikator auf neue Daten an: Er gibt für jede Instanz die wahrscheinlichste Klasse zurück (und ggf. die zugehörige Wahrscheinlichkeit als „Konfidenzwert")

Wahrscheinlichkeit und Frequenz

- Das einfachste Verfahren für die Wahrscheinlichkeitsschätzung:
  - Auszählen der Frequenzen im Korpus
  - Relative Häufigkeit im Korpus als geschätzte Wahrscheinlichkeit
- Die unterschiedlichen Merkmalsmuster nennen wir „Ereignisse“. Die Merkmale mit ihren alternativ möglichen Werten spannen den „Ereignisraum“ auf.
- Der Klassifikator weist einer Instanz das für deren Merkmalsmuster häufigste Wortarttag zu.
- „Training“ des Klassifikators bedeutet in diesem einfachsten Fall im Grunde nur Auszählen des Korpus.

Wahrscheinlichkeit und Frequenz

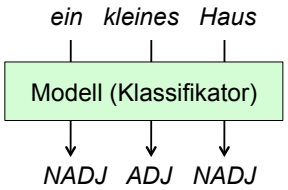
Frequenz im Korpus

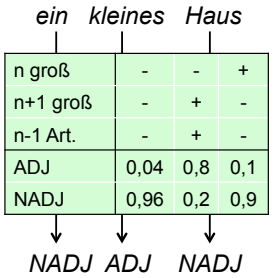
n groß	-	-	+
n+1 groß	-	+	-
n-1 Art.	-	+	-
ADJ	5	40	12
NADJ	120	10	108

Relative Frequenz/
Geschätzte Wahrscheinlichkeit

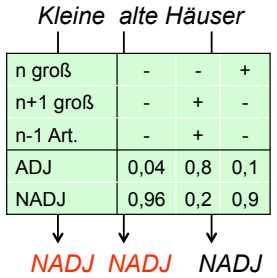
n groß	-	-	+
n+1 groß	-	+	-
n-1 Art.	-	+	-
ADJ	0,04	0,8	0,1
NADJ	0,96	0,2	0,9

Einfaches statistisches Modell





Klassifikationsfehler



Bessere Resultate durch mehr Merkmale?

- Welche weiteren Merkmale könnte man ausnutzen, um Adjektive zu identifizieren?
  - Wortendungen (morphologische Information)
    - Komparativ- und Superlativendungen
    - Suffix -ig, -lich, -isch, -sam
  - Gradpartikeln
    - sehr, besonders, ziemlich

    ...
- Wieso verwendet man nicht alle Merkmale, die irgendwie erfolgversprechend sind?

Größe des Merkmalsraums

- Produkt der Anzahl möglicher Werte aller Merkmale:
  - Wir haben im Beispiel 3 binäre Merkmale verwendet, es gibt also 2*2*2=8 Muster.
  - Wenn wir 10 binäre Merkmal verwenden, haben wir über 1000 Muster.
  - Wenn wir z.B. noch als zwei Merkmale das Vorgängerwort und das Nachfolgerwort selbst hinzunehmen, kommen wir auf Milliarden von Kombinationen.
- Die Instanzen im Trainingskorpus verteilen sich auf die Merkmalsmuster.
  - Das Trainingskorpus muss deutlich größer sein als das Testkorpus. Ansonsten treten viele Merkmalsmuster gar nicht auf („ungesehene Ereignisse"): Das Modell kann dafür keine Vorhersage machen.
  - „Sparse-Data"-Problem

Sparse-Data-Problem

- Je mehr Merkmale, umso besser ist grundsätzlich die Datenlage für die Entscheidung, aber:
- Je mehr Merkmale, auf desto mehr Ereignisse verteilen sich die Trainingsdaten. Die Wahrscheinlichkeitsschätzung wird ungenau oder sogar unmöglich.
- Faustregel für die Wahl einer geeigneten Merkmalsmenge:
  - Wenige gute (aussagekräftige) Merkmale sind besser als viele mittelmäßige
  - Merkmale mit weniger möglichen Werten sind grundsätzlich vorzuziehen.

Evaluation

- Annotation eines „Goldstandard": Testkorpus mit der relevanten Zielinformation (z.B. Wortart)
  - Um subjektive Varianz auszuschließen, wird durch mehrere Personen unabhängig annotiert und die Übereinstimmung („Inter-Annotator Agreement") gemessen.
  - Testkorpus und Trainingskorpus müssen disjunkt sein, um Effekte aus den Besonderheiten der Korpustexte auszuschließen („overfitting")
- Automatische Annotation des Testkorpus mit stastischem Modell/ Klassifikator
- Messung der Performanz durch Vergleich von automatischer Annotation mit Goldstandard

Akkuratheit

- Akkuratheit (accuracy) ist das einfachste Maß:

  Akkuratheit = korrekt klassifizierte Instanzen/alle Instanzen

- Fehlerrate (error rate) ist der Komplementärbegriff zu Akkuratheit:

  Fehlerrate = 1 – Akkuratheit

- Das Akkuratheitsmaß verdeckt oft die tatsächliche Performanz eines Verfahrens.
- Grundlage für eine feinere Evaluation des Klassifikators ist die Konfusionsmatrix.

Konfusionsmatrix

- Konfusionsmatrix (Verwechslungstabelle) für binäre Klassifikation:

	Echtes ADJ	Echtes NADJ
Klassifiziert als ADJ	**ok**	**falsch**
Klassifiziert als NADJ	**falsch**	**ok**

Konfusionsmatrix

- Konfusionsmatrix (Verwechslungstabelle) für binäre Klassifikation:

	Echtes ADJ	Echtes NADJ
Klassifiziert als ADJ	**ok**	Korrektheits-fehler
Klassifiziert als NADJ	Vollständigkeits-fehler	**ok**

Konfusionsmatrix

- Konfusionsmatrix (Verwechslungstabelle) für binäre Klassifikation:

	Echtes ADJ	Echtes NADJ
Klassifiziert als ADJ	true positive	false positive
Klassifiziert als NADJ	false negative	true negative

Konfusionsmatrix

- (Fiktives) Beispiel:

	Echtes ADJ	Echtes NADJ
Klassifiziert als ADJ	20	80
Klassifiziert als NADJ	20	880

- Von insgesamt 1000 Fällen sind 900 korrekt (Wahre Positive und wahre Negative): Akkuratheit ist also 90%, Fehlerrate 10%.
- Tatsächlich ist die Adjektiverkennung miserabel: von fünf als ADJ klassifizierten Instanzen ist nur eine korrekt.
- Recall und Precision als klassenspezifische Maße, die Vollständigkeits- und Korrektheitsfehler separat messen.

Recall

	Echtes ADJ	Echtes NADJ
Klassifiziert als ADJ	True positive	False positive
Klassifiziert als NADJ	False negative	True negative

- Welcher Anteil der echten X wurde tatsächlich gefunden (als X klassifiziert)?

Recall = True positives/(True positives + False negatives)

	Echtes ADJ	Echtes NADJ
Klassifiziert als ADJ	20	80
Klassifiziert als NADJ	20	880

Recall für ADJ = 20/(20+20) = 0,5

Precision

	Echtes ADJ	Echtes NADJ
Klassifiziert als ADJ	True positive	False positive
Klassifiziert als NADJ	False negative	True negative

- Welcher Anteil der als X klassifizierten Instanzen ist tatsächlich ein X?

Precision = True positives/(True positives + False positives)

	Echtes ADJ	Echtes NADJ
Klassifiziert als ADJ	20	80
Klassifiziert als NADJ	20	880

Precision für ADJ = 20/(20+80) = 0,2

Precision und Recall

- Precision und Recall sind im allgemeinen nur zusammen aussagekräftig
  - Hohe Präzision, hoher Recall: gutes Modell
  - Niedrige Präzision, niedriger Recall: schlechtes Modell
  - Hohe Präzision, niedriger Recall: „Vorsichtiges" Modell
    - Findet nicht alle Instanzen von X
    - Klassifiziert kaum keine Nicht-Xe als X
  - Niedrige Präzision, hoher Recall: „Mutiges" Modell
    - Findet fast alle Instanzen von X
    - Klassifiziert viele nicht-Xe fehlerhaft als X
  - Extremfälle
    - Modell klassifiziert alles als X: Recall 100%, Precision niedrig
    - Modell klassifziert nichts als X: Recall 0%, Precision nicht definiert

F-Score

- Der „F-Score“ ist ein Maß für die „Gesamtgüte“ der Klassifikation, in das Precision und Recall eingehen.

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

- F-Score für die Klasse ADJ im Beispiel:

$$F = \frac{2 * 0{,}2 * 0{,}5}{0{,}2 + 0{,}5} = 0{,}29$$