Seminar „Medizinische Bildverarbeitung“

Word Assignment to Image Parts

Armin Fritsche

Human Language Technology and Pattern Recognition
Lehrstuhl für Informatik VI, Computer Science Department
RWTH Aachen - University of Technology
D-52056 Aachen, Germany

Übersicht

1. Einleitung
2. Objekterkennung und Wortvorhersage
   - Der EM-Algorithmus
   - Kontrolle des Vokabulars
   - Hinzunahme von Supervised Data
3. Einfluss der Merkmale
   - Wahl der Merkmale
4. Einfluss des Segmenters
   - Normalized Cuts
   - Segmenter und Wortvorhersage
5. Weiterführende Ideen

Literatur

Duygulu P, Barnard K, de Freitas JFG, Forsyth DA
Object Recognition as Machine Translation: Learning a Lexicon for a Fixed Image Vocabulary.
Proc. Seventh European Conference on Computer Vision, Copenhagen, Denmark, pp. IV: 97-112, 2002

Barnard K, Duygulu P, Guru R, Forsyth DA
The Effects of Segmentation and Feature Choice in a Translation Model Of Object Recognition.
Proc. Computer Vision and Pattern Recognition, pp. II: 675-682, 2003

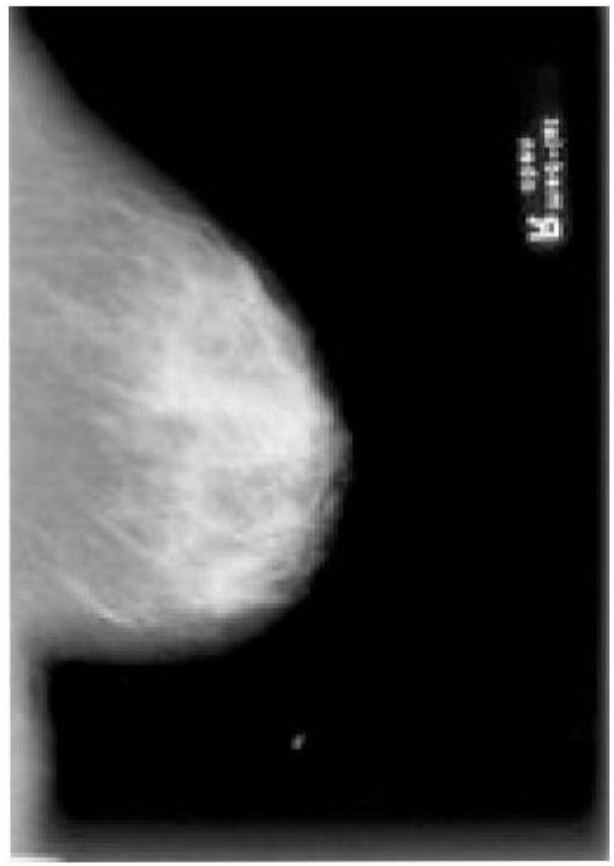
Dempster AP, Larid NM, Rubin DB
Maximum Likelihood from Incomplete Data via the EM-Algorithm.
Journal of the Royal Statistical Society, Series B vol. 39, pp. 1-38, 1977

Shi J, Malik J
Normalized Cuts and Image Segmentation.
IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 8, 2000

Forsyth DA, Ponce J
Computer Vision: A modern approach.
Upper Saddle River, NJ: Prentice Hall, 2003

Ziel

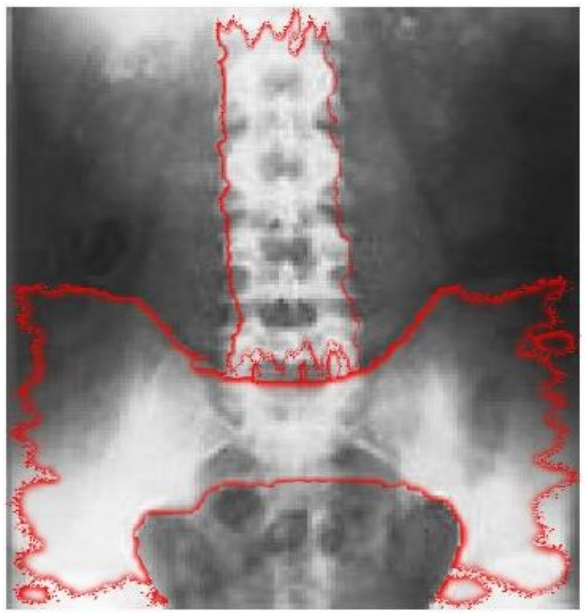
Beschriftung von Bildteilen:



Vokabular={Schädel, Wirbelsäule, Zahn, Niere, Brust, ...}

Ziel

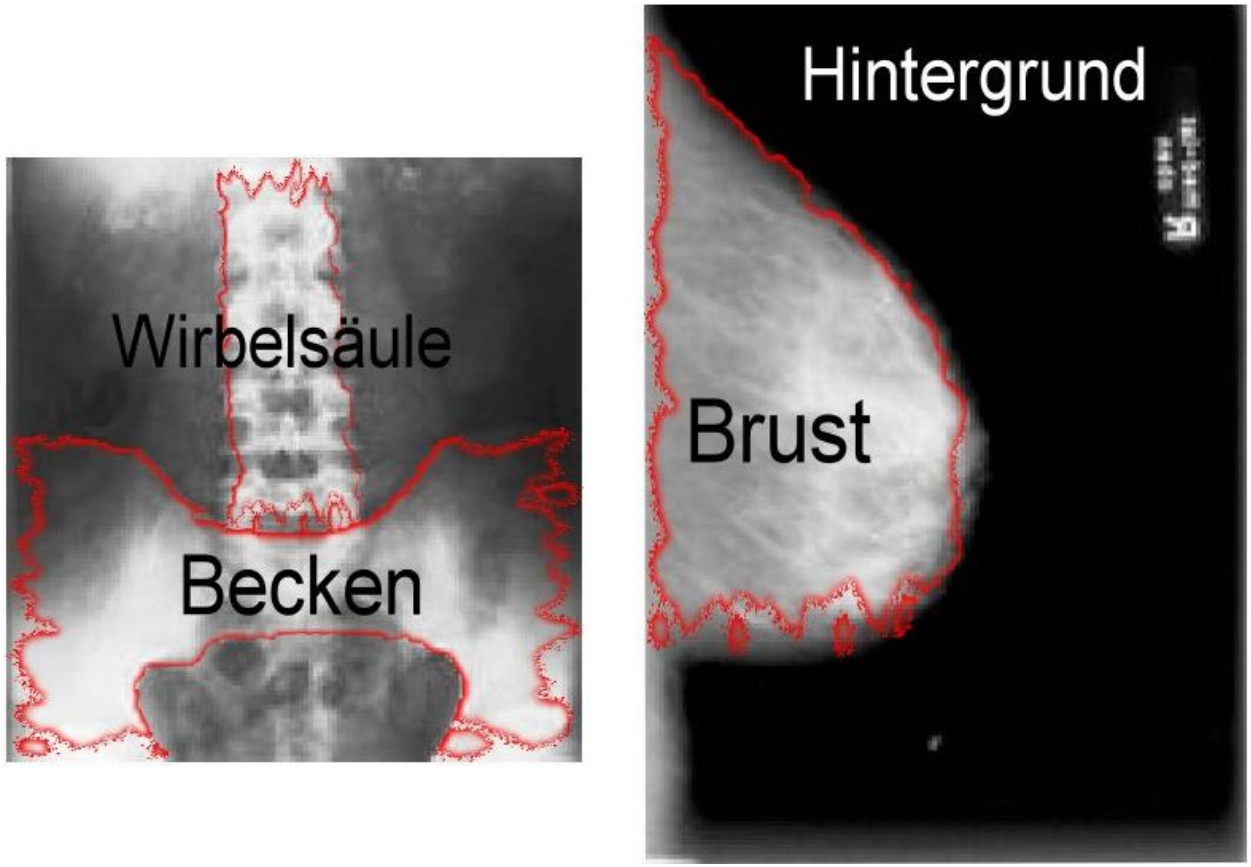
Beschriftung von Bildteilen:



Vokabular={Schädel, Wirbelsäule, Zahn, Niere, Brust, ...}

Ziel

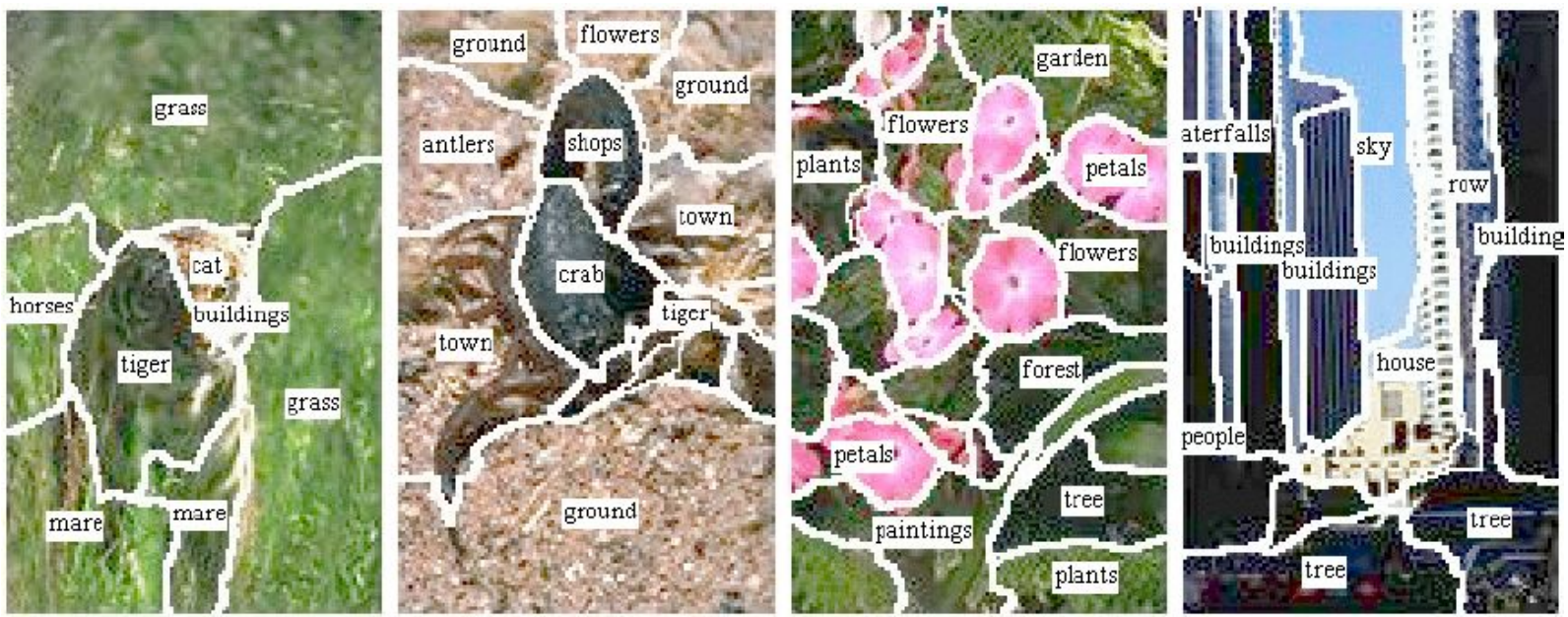
Beschriftung von Bildteilen:



Vokabular={Schädel, Wirbelsäule, Zahn, Niere, Brust, ...}

Für diesen Vortrag

- Nicht medizinische Bilder (Corel Data Set)
- Trainingsmenge: 4500 bzw. 6000 Bilder
- Testmenge: 2000 Bilder
- Vokabular bekannt
- Beschriftung im Bild
- Beispiel:

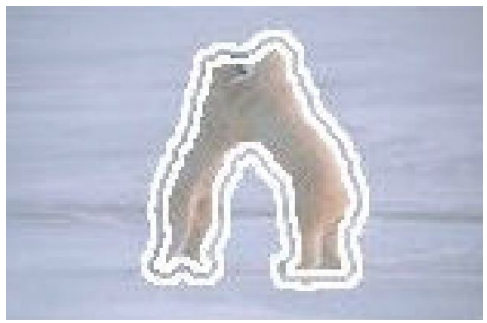


Übersicht

1. Einleitung
2. Objekterkennung und Wortvorhersage
   - Der EM-Algorithmus
   - Kontrolle des Vokabulars
   - Hinzunahme von Supervised Data
3. Einfluss der Merkmale
   - Wahl der Merkmale
4. Einfluss des Segmenters
   - Normalized Cuts
   - Segmenter und Wortvorhersage
5. Weiterführende Ideen

Bild $\Rightarrow$ Blobs

Bilder auf low-level Ebene $\Rightarrow$ Darstellung der Bildregionen

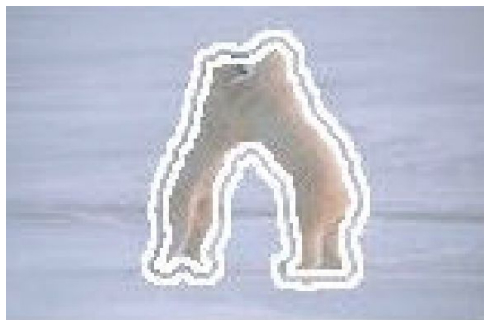


Segment $\longrightarrow$

Blob $\longrightarrow$ $\begin{pmatrix} \text{Farbe} \\ \text{Größe} \\ \vdots \end{pmatrix}$

Bild $\Rightarrow$ Blobs

Bilder auf low-level Ebene $\Rightarrow$ Darstellung der Bildregionen

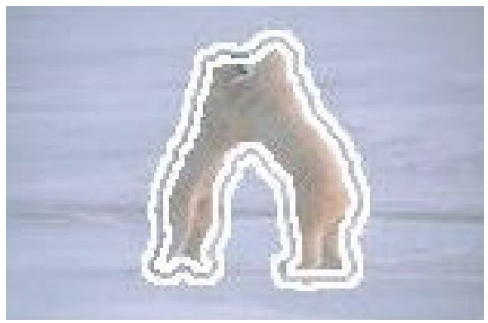


$\xrightarrow{Segment}$

$$\xrightarrow{Blob} \begin{pmatrix} \text{Farbe} \\ \text{Größe} \\ \vdots \end{pmatrix}$$

Bild $\Rightarrow$ Blobs

Bilder auf low-level Ebene $\Rightarrow$ Darstellung der Bildregionen



$\xrightarrow{Segment}$

$$\xrightarrow{Blob} \begin{pmatrix} \text{Farbe} \\ \text{Größe} \\ \vdots \end{pmatrix}$$

Notation

Bilder der Trainingsmenge und Wörter und Blobs für jeweils ein Bild werden durchnummeriert

Dann:

w_{nj}: j-tes Wort im n-ten Bild

b_{ni}: i-ter Blob im n-ten Bild

$a_{nj} = i \Leftrightarrow$ Das Wort w_{nj} wird dem Blob b_{ni} zugeordnet

Gesucht

Für gegebenen Blob b, das Wort w_{max} mit größter Wahrscheinlichkeit $p(w_{max}|b)$

$$\Rightarrow \text{Tabelle über alle } p(w|b) \text{ und } w_{max} = \arg\max_{w} p(w|b)$$

Dazu:

Diskretisierung der Regionen in 500 Blob-Cluster mit k-means

Dann:

$$p(w|b) = \prod_{n=1}^{N} \prod_{j=1}^{J_n} \sum_{i=1}^{I_n} p(a_{nj} = i) t(w_{nj}|b_{ni})$$

N = Anzahl Bilder

J_n = Anzahl Wörter im n-ten Bild

I_n = Anzahl Blobs im n-ten Bild

Gesucht

Für gegebenen Blob b, das Wort w_{max} mit größter Wahrscheinlichkeit $p(w_{max}|b)$

$$\Rightarrow \text{Tabelle über alle } p(w|b) \text{ und } w_{max} = \arg\max_{w} p(w|b)$$

Dazu:

Diskretisierung der Regionen in 500 Blob-Cluster mit k-means

Dann:

$$p(w|b) = \prod_{n=1}^{N} \prod_{j=1}^{J_n} \sum_{i=1}^{I_n} p(a_{nj} = i) t(w_{nj}|b_{ni})$$

N = Anzahl Bilder

J_n = Anzahl Wörter im n-ten Bild

I_n = Anzahl Blobs im n-ten Bild

Problem: Zum Berechnen von $p(a_{nj} = i)$ wird $t(w_{nj}|b_{ni})$ gebraucht und umgekehrt.

$\Rightarrow$ Missing Data Problem

Gesucht

Für gegebenen Blob b, das Wort w_{max} mit größter Wahrscheinlichkeit $p(w_{max}|b)$

$$\Rightarrow \text{Tabelle über alle } p(w|b) \text{ und } w_{max} = \arg\max_{w} p(w|b)$$

Dazu:

Diskretisierung der Regionen in 500 Blob-Cluster mit k-means

Dann:

$$p(w|b) = \prod_{n=1}^{N} \prod_{j=1}^{J_n} \sum_{i=1}^{I_n} p(a_{nj} = i) t(w_{nj}|b_{ni})$$

N = Anzahl Bilder
J_n = Anzahl Wörter im n-ten Bild
I_n = Anzahl Blobs im n-ten Bild

Problem : Zum Berechnen von $p(a_{nj} = i)$ wird $t(w_{nj}|b_{ni})$ gebraucht und umgekehrt.

$\Rightarrow$ Missing Data Problem

Gesucht

Für gegebenen Blob b, das Wort w_{max} mit größter Wahrscheinlichkeit $p(w_{max}|b)$

$$\Rightarrow \text{Tabelle über alle } p(w|b) \text{ und } w_{max} = \arg\max_{w} p(w|b)$$

Dazu:

Diskretisierung der Regionen in 500 Blob-Cluster mit k-means

Dann:

$$p(w|b) = \prod_{n=1}^{N} \prod_{j=1}^{J_n} \sum_{i=1}^{I_n} p(a_{nj} = i) t(w_{nj}|b_{ni})$$

N = Anzahl Bilder
J_n = Anzahl Wörter im n-ten Bild
I_n = Anzahl Blobs im n-ten Bild

Problem : Zum Berechnen von $p(a_{nj} = i)$ wird $t(w_{nj}|b_{ni})$ gebraucht und umgekehrt.

$\Rightarrow$ Missing Data Problem

Gesucht

Für gegebenen Blob b, das Wort w_{max} mit größter Wahrscheinlichkeit $p(w_{max}|b)$

$$\Rightarrow \text{Tabelle über alle } p(w|b) \text{ und } w_{max} = \arg\max_{w} p(w|b)$$

Dazu:

Diskretisierung der Regionen in 500 Blob-Cluster mit k-means

Dann:

$$p(w|b) = \prod_{n=1}^{N} \prod_{j=1}^{J_n} \sum_{i=1}^{I_n} p(a_{nj} = i) t(w_{nj}|b_{ni})$$

N = Anzahl Bilder
J_n = Anzahl Wörter im n-ten Bild
I_n = Anzahl Blobs im n-ten Bild

Problem : Zum Berechnen von $p(a_{nj} = i)$ wird $t(w_{nj}|b_{ni})$ gebraucht und umgekehrt.

$\Rightarrow$ Missing Data Problem

EM-Algorithmus (Dempster 1977)

2 Schritte: Expectation Schritt und Maximization Schritt

E-Step: Berechne Schätzer für deine Zuweisung $p(a_{nj} = i)$

M-Step: Berechne damit Näherungen für $t(w_{nj}|b_{ni})$

Iteriere E-Step und M-Step (Konvergenz!)

EM-Algorithmus (Dempster 1977)

2 Schritte: Expectation Schritt und Maximization Schritt

E-Step: Berechne Schätzer für deine Zuweisung $p(a_{nj} = i)$

M-Step: Berechne damit Näherungen für $t(w_{nj}|b_{ni})$

Iteriere E-Step und M-Step (Konvergenz!)

EM-Algorithmus (Dempster 1977)

2 Schritte: Expectation Schritt und Maximization Schritt

E-Step: Berechne Schätzer für deine Zuweisung $p(a_{nj} = i)$

M-Step: Berechne damit Näherungen für $t(w_{nj}|b_{ni})$

Iteriere E-Step und M-Step (Konvergenz!)

EM-Algorithmus (Dempster 1977)

2 Schritte: Expectation Schritt und Maximization Schritt

E-Step: Berechne Schätzer für deine Zuweisung $p(a_{nj} = i)$

M-Step: Berechne damit Näherungen für $t(w_{nj}|b_{ni})$

Iteriere E-Step und M-Step (Konvergenz!)

Güte des EM-Algorithmus

Wir definieren die Merkmale

- Präzision eines Wortes w =

$$\frac{\#\text{ korrekte Beschriftungen eines Blobs mit } w}{\#\text{ Beschriftungen eines Blobs mit } w}$$

- Recall eines Wortes w =

$$\frac{\#\text{ korrekte Beschriftungen eines Blobs mit } w}{\#\text{ tatsächliche Auftreten des Wortes } w}$$

Beobachtungen

- Sehr wenige gute Wörter (ca. 2-4%): Sehr hohe Präzision (0.6-1.0), aber auch niedriger Recall ($\approx$0.33)
- Hier: Recall $\uparrow \Rightarrow$ Präzision $\uparrow$
- Ziel: Erhöhung des Recalls und der Präzision
  $\Rightarrow$ Verkleinerung des Vokabulars

Beobachtungen

- Sehr wenige gute Wörter (ca. 2-4%): Sehr hohe Präzision (0.6-1.0), aber auch niedriger Recall ($\approx$0.33)
- Hier: Recall $\uparrow \Rightarrow$ Präzision $\uparrow$
- Ziel: Erhöhung des Recalls und der Präzision
  $\Rightarrow$ Verkleinerung des Vokabulars

Beobachtungen

- Sehr wenige gute Wörter (ca. 2-4%): Sehr hohe Präzision (0.6-1.0), aber auch niedriger Recall ($\approx$0.33)
- Hier: Recall $\uparrow \Rightarrow$ Präzision $\uparrow$
- Ziel: Erhöhung des Recalls und der Präzision
  $\Rightarrow$ Verkleinerung des Vokabulars

Beobachtungen

- Sehr wenige gute Wörter (ca. 2-4%): Sehr hohe Präzision (0.6-1.0), aber auch niedriger Recall ($\approx$0.33)
- Hier: Recall $\uparrow \Rightarrow$ Präzision $\uparrow$
- Ziel: Erhöhung des Recalls und der Präzision
  $\Rightarrow$ Verkleinerung des Vokabulars

Streichung nicht verwendeter Wörter

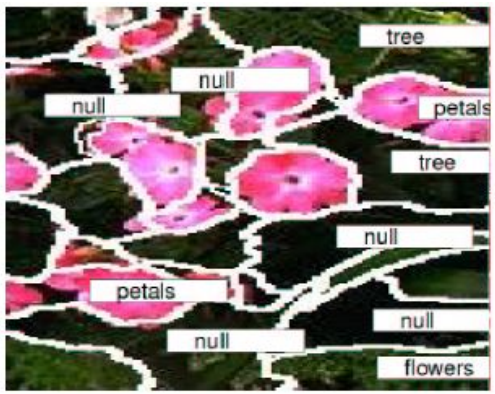
- Einige Wörter werden nie benutzt
- Streichung dieser Wörter
- Verkleinerung des Vokabulars um ca. 80%
- $\Rightarrow$ Leichte Steigerung des Recalls und der Präzision

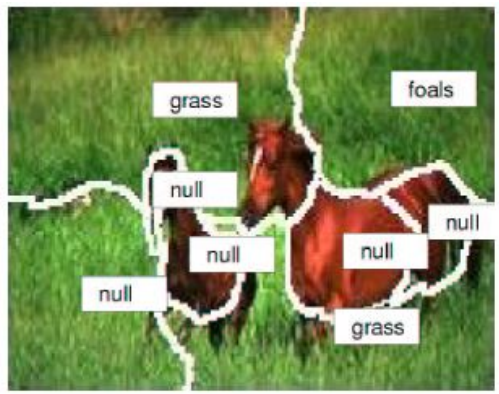
null-Schwellwert

- Definition von schlechten Vorhersagen
- Einführung eines Schwellwertes
- Schlechte Vorhersage: $p(w_{max}|b) < \mathit{Schwellwert}$
- Keine Beschriftungen mit Wörtern aus schlechten Vorhersagen
- Entsprechende Regionen werden statt dessen mit „null“ beschriftet

Wahl des Schwellwertes

- Schwellwert ist anwendungsabhängig
- In Corel Data Set: $0.1 <$ Schwellwert < 0.2
- Zwar gilt Schwellwert$\uparrow \Rightarrow$ Recall$\downarrow$, Präzision $\uparrow$





Clusterbildung I

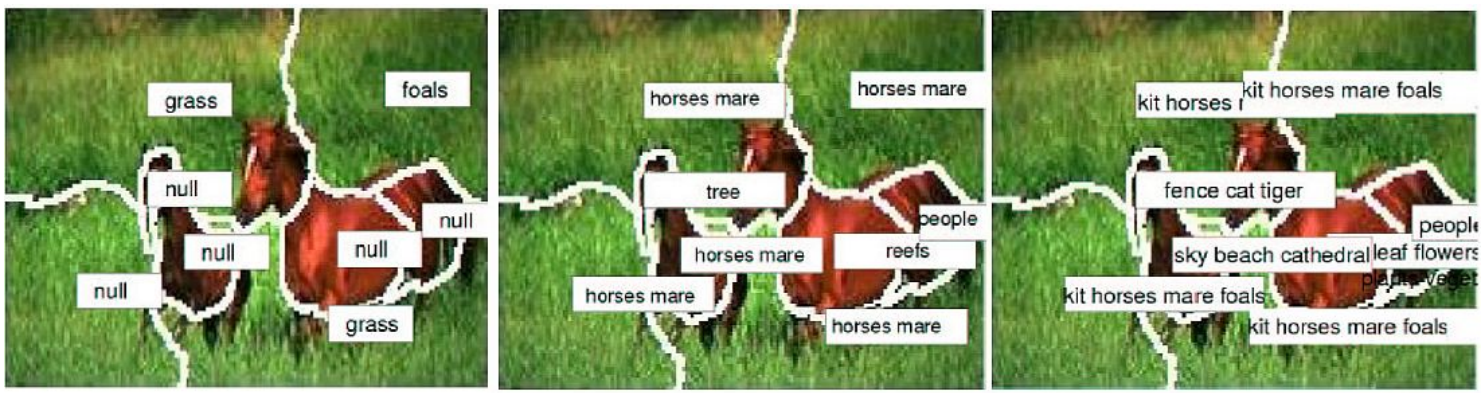
- Im Vokabular sind „ähnliche“ Wörter
- z.B. „Pferd“, „Stute“, „Hengst“ und „Fohlen“
  „Zug“, „Lokomotive“, „Triebwagen“ und „Bahn“
  aber auch „Jet“ und „Adler“
- Solche Wörter werden zu einem Cluster zusammengefasst.

Clusterbildung II

- Aufstellen einer Ähnlichkeitsmatrix
- Ähnlichkeitsmaß: Kullback-Leibler-Divergenz

$$d_{KL}(w_1, w_2) = \sum_b p(w_1|b) \log \frac{p(w_1|b)}{p(w_2|b)}$$

- Auf Ähnlichkeitsmatrix: Anwendung von „Normalized Cuts“
- $\Rightarrow$ Vokabular$\downarrow$(um ca. 25%), Recall$\uparrow$, Präzision $\uparrow$

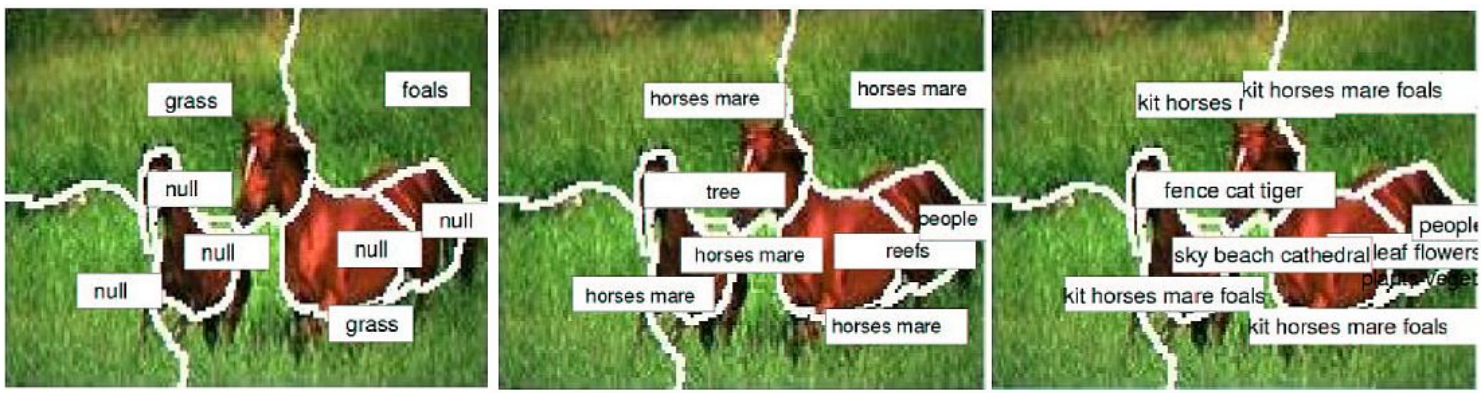


Clusterbildung II

- Aufstellen einer Ähnlichkeitsmatrix
- Ähnlichkeitsmaß: Kullback-Leibler-Divergenz

$$d_{KL}(w_1, w_2) = \sum_b p(w_1|b) \log \frac{p(w_1|b)}{p(w_2|b)}$$

- Auf Ähnlichkeitsmatrix: Anwendung von „Normalized Cuts“
- $\Rightarrow$ Vokabular$\downarrow$(um ca. 25%), Recall$\uparrow$, Präzision $\uparrow$



Hinzunahme von Supervised Data

- Bis jetzt: Performancesteigerung durch Kontrolle des Vokabulars
- Problem: Kein semantischer Bezug zwischen Wörtern und Blobs
- Lösung: Hinzunahme von Supervised Data

EM und Supervised Data

- Jedes Wort wird als ein Cluster definiert
- Initialisierung durch nearest-neigbour
- Danach: Retraining mit EM-Algorithmus

$\Rightarrow$ Lernen aus Supervised und Unsupervised Data

Zeige Verbesserung: Dafür neues Performancemaß

EM und Supervised Data

- Jedes Wort wird als ein Cluster definiert
- Initialisierung durch nearest-neigbour
- Danach: Retraining mit EM-Algorithmus

$\Rightarrow$ Lernen aus Supervised und Unsupervised Data

Zeige Verbesserung: Dafür neues Performancemaß

EM und Supervised Data

- Jedes Wort wird als ein Cluster definiert
- Initialisierung durch nearest-neigbour
- Danach: Retraining mit EM-Algorithmus

$\Rightarrow$ Lernen aus Supervised und Unsupervised Data

Zeige Verbesserung: Dafür neues Performancemaß

Messung der Performance

Für jedes Bild i ist die Performance:

$$E^{(Algorithmus)}(i) := r_i / J_i$$

$r_i = \#$ Wörter die für das Bild i richtig bestimmt wurden
$J_i = \#$ Wörter für das Bild i $(1 \leq J_i \leq 5)$

Gesamtperformance für n Bilder:

$$E^{(Algorithmus)} := \frac{1}{n} \sum_{i=1}^{n} E^{(Algorithmus)}(i)$$

Bezug auf empirische Verteilung der Wörter:
Performance:

$$E = E^{(Algorithmus)} _ E^{(Wortverteilung)}$$

Messung der Performance

Für jedes Bild i ist die Performance:

$$E^{(Algorithmus)}(i) := r_i / J_i$$

$r_i = \#$ Wörter die für das Bild i richtig bestimmt wurden
$J_i = \#$ Wörter für das Bild i $(1 \leq J_i \leq 5)$

Gesamtperformance für n Bilder:

$$E^{(Algorithmus)} := \frac{1}{n} \sum_{i=1}^{n} E^{(Algorithmus)}(i)$$

Bezug auf empirische Verteilung der Wörter:
Performance:

$$E = E^{(Algorithmus)} _ E^{(Wortverteilung)}$$

Auswertung Supervised Data

EM-Algorithmus ohne Supervised Data: E(Testmenge) = 0.0597

EM-Algorithmus mit Supervised Data: E(Testmenge) = 0.0988

$\Rightarrow$ Verbesserung durch Initialisierung mit Supervised Data

Auswertung Supervised Data

EM-Algorithmus ohne Supervised Data: E(Testmenge) = 0.0597

EM-Algorithmus mit Supervised Data: E(Testmenge) = 0.0988

$\Rightarrow$ Verbesserung durch Initialisierung mit Supervised Data

Übersicht

1. Einleitung

2. Objekterkennung und Wortvorhersage
   - Der EM-Algorithmus
   - Kontrolle des Vokabulars
   - Hinzunahme von Supervised Data

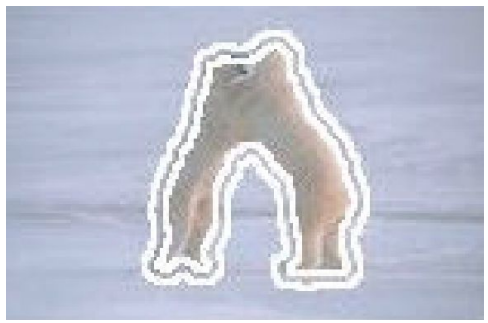
3. Einfluss der Merkmale
   - Wahl der Merkmale

4. Einfluss des Segmenters
   - Normalized Cuts
   - Segmenter und Wortvorhersage

5. Weiterführende Ideen

Bild $\Rightarrow$ Blobs

Bilder auf low-level Ebene $\Rightarrow$ Darstellung der Bildregionen



$\xrightarrow{Segment}$

$$\xrightarrow{Blob} \begin{pmatrix} \text{Farbe} \\ \text{Größe} \\ \vdots \end{pmatrix}$$

Basismerkmale

- Größe der Region
- Lage der Region im Bild
- 2 Formfunktionen:
  - Lage des Schwerpunktes in der Region
  - $\frac{\text{Fläche}}{(\text{Länge der Grenze})^2}$

Farbe

Speicherung von Mittelwert und Varianz der Farbe der Region:

- (R, G, B)
- $L * a * b$
  - Entspricht Farbwahrnehmung beim Menschen
  - L = Luminanz = Helligkeit
  - a = Chrominanz 1 = Farbe 1 (rot- und grün-Anteile)
  - b = Chrominanz 2 = Farbe 2 (blau- und gelb-Anteile)
- (r, g, S)
  - $S = R + G + B$
  - $r = R/S$
  - $g = G/S$

Auswertung Farbe

- *rgS* besser als $L * a * b$
- *rgS* und $L * a * b$ sogar signifikant besser als (R, G, B)

Merkmalmenge	E(Trainingsmenge)	E(Testmenge)
Basismerkmale	0.019	0.020
Basismerkmale, *RGB*	0.076	0.057
Basismerkmale, $L * a * b$	0.097	0.085
Basismerkmale, *rgS*	0.109	0.092

Textur

- Mittleres Ergebniss aus Anwendung von 12 Filtern mit verschiedener Orientierung und
- Mittleres Ergebniss der Differenz von 4 verschiedenen Kombinationen von 2 Gaussfiltern
- Filter sind hier nicht weiter bestimmt

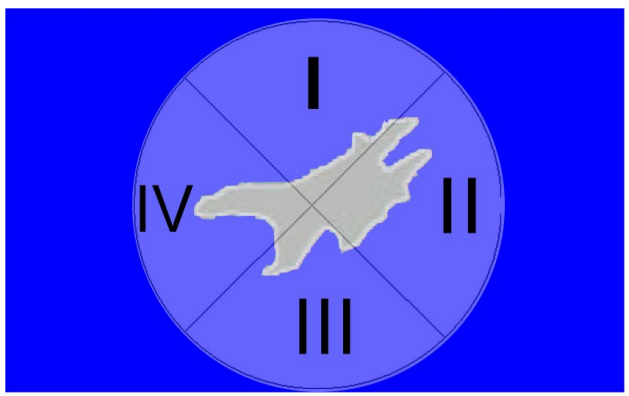
Auswertung Textur

- Nur Basismerkmale und Textur $\Rightarrow$ (signifikante) Verbesserung
- Mit rgS-Farbmerkmal: Keine Performancesteigerung

Merkmalmenge	E(Trainingsmenge)	E(Testmenge)
Basismerkmale	0.019	0.020
Basismerkmale, *rgS*	0.109	0.092
Basismerkmale, Textur	0.092	0.048
Basismerkmale, *rgS*, Textur	0.109	0.072

Farbkontext

- Kreis um den Schwerpunkt
- Einteilung in 4 Quadranten
- Farbkontext: Mittlere Farbe der Pixel im Quadranten aber nicht in der Region



Auswertung Farbkontext

- Farbkontext $\Rightarrow$ Performancesteigerung
- Allerdings: Nur geringe Verbesserung
- Möglicher Grund: Farbkontext nur in RGB gespeichert

Merkmalmenge	E(Trainingsmenge)	E(Testmenge)
Basismerkmale	0.019	0.020
Basismerkmale, *rgS*	0.109	0.092
Basismerkmale, *rgS*, Farbkontext	0.134	0.094

Form

- Parametrisierung der Grenzlinie durch deren Abstand zum Schwerpunkt
- Abtastung an 30 Punkten (1. Punkt oben links)

Auswertung Form

- Gut in Trainingsmenge, aber nicht so gut in Testmenge
- Mögliche Verbesserung durch Verbesserung des Segmenters

Merkmalmenge	E(Trainingsmenge)	E(Testmenge)
Basismerkmale	0.019	0.020
Basismerkmale, *rgS*	0.109	0.092
Basismerkmale, Form	0.053	0.016
Basismerkmale, *rgS*, Form	0.065	0.029

Übersicht

1. Einleitung
2. Objekterkennung und Wortvorhersage
   - Der EM-Algorithmus
   - Kontrolle des Vokabulars
   - Hinzunahme von Supervised Data
3. Einfluss der Merkmale
   - Wahl der Merkmale
4. Einfluss des Segmenters
   - Normalized Cuts
   - Segmenter und Wortvorhersage
5. Weiterführende Ideen

Segmenter

Vergleich von 3 Segmentern:

EM for Blobword: Stochastisches Segmentieren

Mean Shift : Segmentierung im Merkmalraum

Normalized Cuts: Graphtheoretisches Segmentieren

Segmenter

Vergleich von 3 Segmentern:

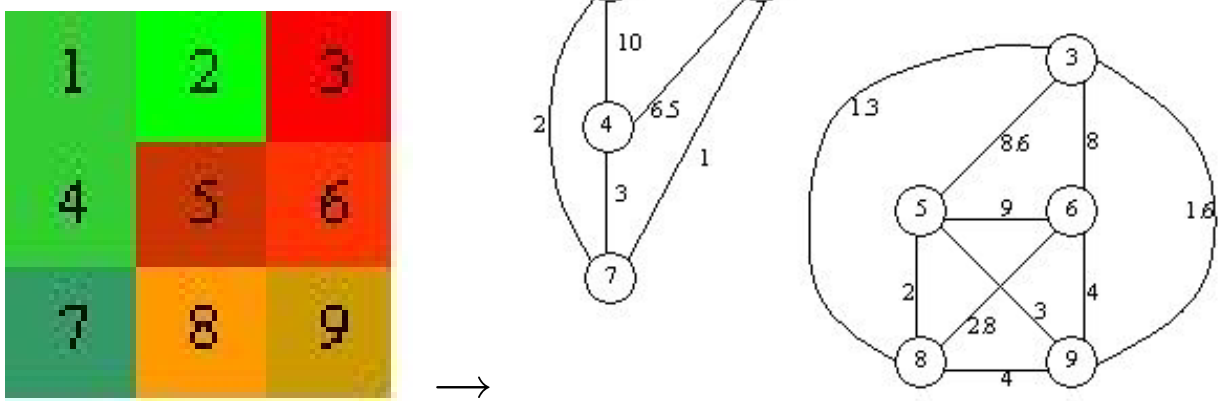
EM for Blobword: Stochastisches Segmentieren

Mean Shift : Segmentierung im Merkmalraum

Normalized Cuts: Graphtheoretisches Segmentieren

Bild $\Rightarrow$ Graph $G = (V, E)$

- Menge der Knoten = Menge der Pixel im Bild
- Kanten zwischen allen Knoten
- Kantengewichte $w_{a,b} \cong$ „Ähnlichkeit“ zwischen Pixeln a und b
- Lösche „kleine“ Kanten

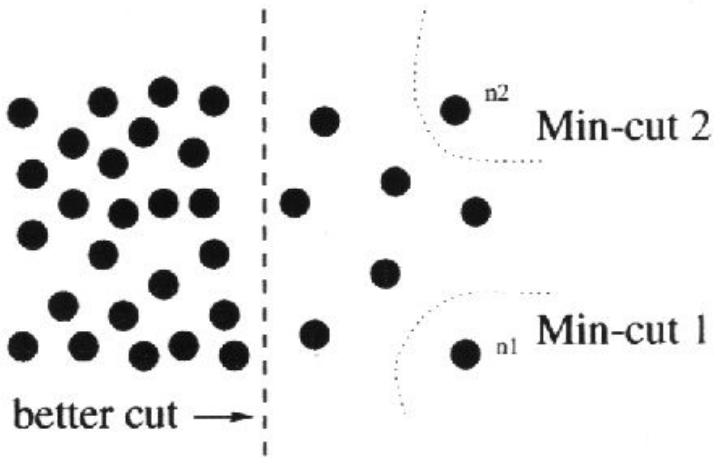


Idee

- Def.:

$$cut(A, B) = \sum_{a \in A, b \in B} w_{a,b}$$

- Schneide in Komponenten A und B, so dass $cut(A, B)$ minimal ist.
- Aber: Nicht immer optimal

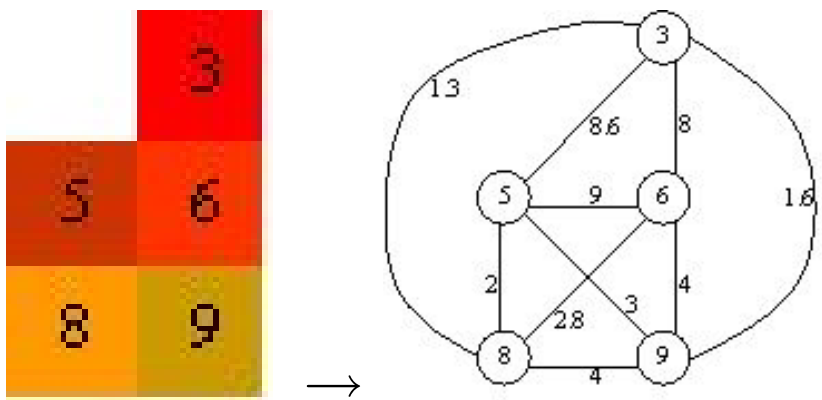


Normalized Cut

- Def.:

$$Ncut(A,B) = \frac{cut(A,B)}{cut(A,V)} + \frac{cut(A,B)}{cut(B,V)}$$

- Schneide in Komponenten A und B, so dass $Ncut(A,B)$ minimal ist.



$cut(\{3,5,6,9\},\{8\}) = 10.1$, $cut(\{3,5,6\},\{8,9\}) = 14.7$

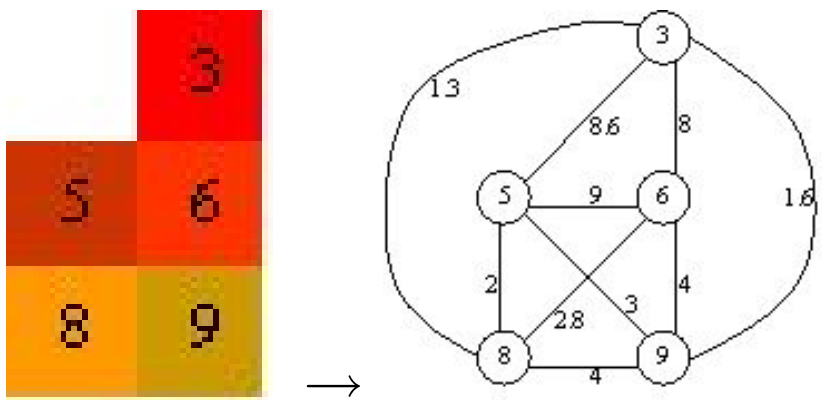
$Ncut(\{3,5,6,9\},\{8\}) \approx 1.22$, $Ncut(\{3,5,6\},\{8,9\}) \approx 1.15$

Normalized Cut

- Def.:

$$Ncut(A,B) = \frac{cut(A,B)}{cut(A,V)} + \frac{cut(A,B)}{cut(B,V)}$$

- Schneide in Komponenten A und B, so dass $Ncut(A,B)$ minimal ist.



$cut(\{3,5,6,9\},\{8\}) = 10.1$, $cut(\{3,5,6\},\{8,9\}) = 14.7$
$Ncut(\{3,5,6,9\},\{8\}) \approx 1.22$, $Ncut(\{3,5,6\},\{8,9\}) \approx 1.15$

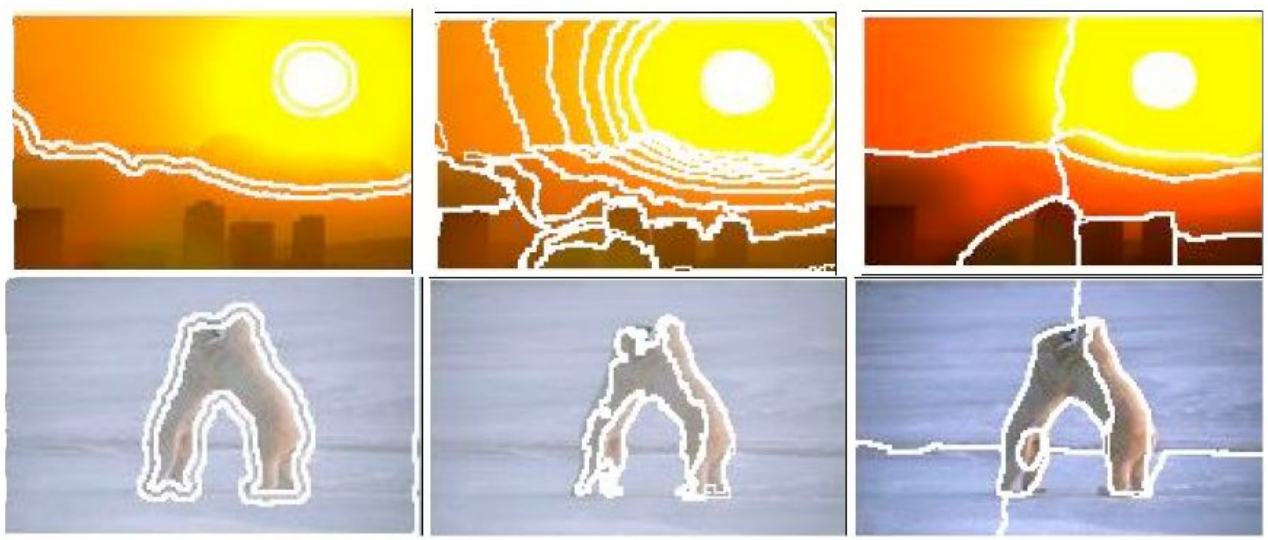
Segmenter

Vergleich von 3 Segmentern:

EM for Blobword: Stochastisches segmentieren

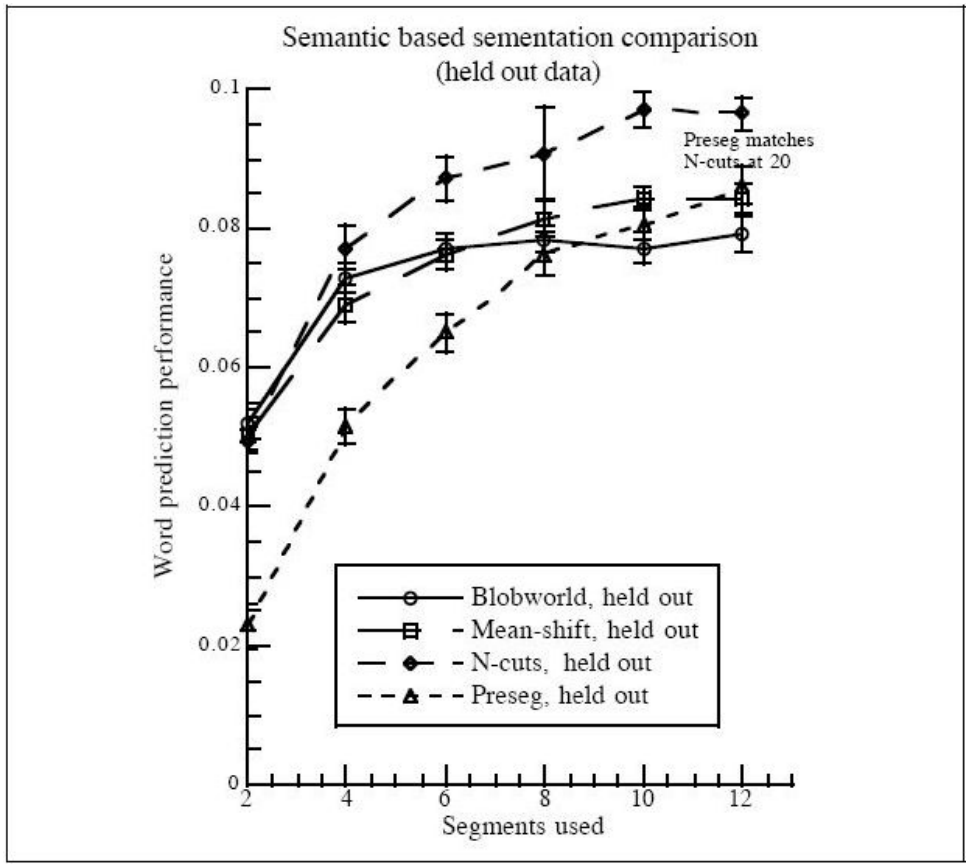
Mean Shift : Segmentierung im Merkmalraum

Normalized Cuts: Graphtheoretisches segmentieren



Ergebnis

- Normalized Cuts besser als Mean-Shift
- Mean-Shift besser als EM for Blobworld



Übersicht

1. Einleitung

2. Objekterkennung und Wortvorhersage
   - Der EM-Algorithmus
   - Kontrolle des Vokabulars
   - Hinzunahme von Supervised Data

3. Einfluss der Merkmale
   - Wahl der Merkmale

4. Einfluss des Segmenters
   - Normalized Cuts
   - Segmenter und Wortvorhersage

5. Weiterführende Ideen

Weiterführende Ideen

- Wortvorhersage als high-level-Konstrukt für Segmentierung
- Verfeinerung der Merkmale
- Anpassung für medizinische Datenmengen

Weiterführende Ideen

- Wortvorhersage als high-level-Konstrukt für Segmentierung
- Verfeinerung der Merkmale
- Anpassung für medizinische Datenmengen

Weiterführende Ideen

- Wortvorhersage als high-level-Konstrukt für Segmentierung
- Verfeinerung der Merkmale
- Anpassung für medizinische Datenmengen