ENEA Eigennamen Extraction und Analyse

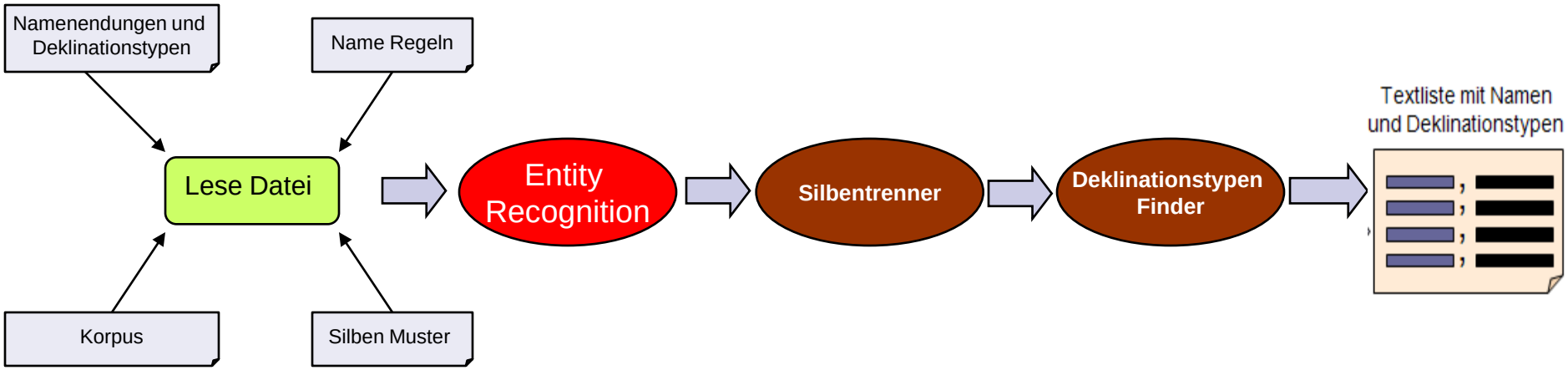
Studienprojekt 24.05.2007
Betreuer: Dr. Markus Demleitner
Matthias Hartung
Dr. Detlef Prescher
Referentin: Małgorzata Szczerbik

Das Ziel des Projekts war

Ein Programm zu entwickeln, das Personennamen aus polnischen Texten extrahiert und diese einem bestimmten Deklinationstyp zuweist

Namen im Polnischen werden dekliniert!

Überblick über das Programm



Zwischenschritte Herausforderungen

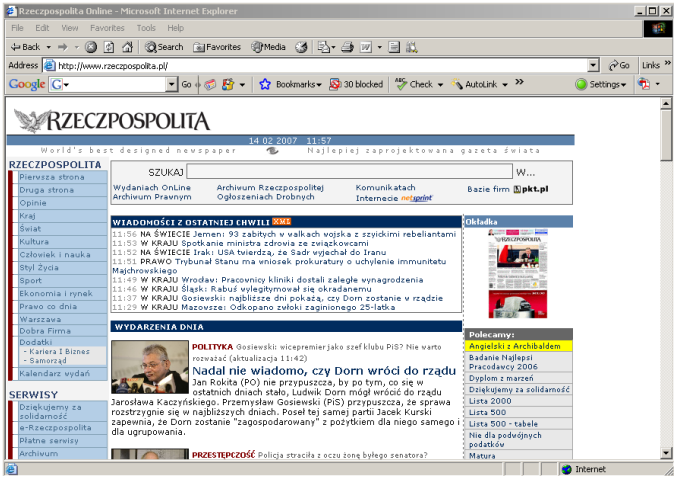
- Unicode in Java
  - Output auf der Konsole
  - Speichern von unicode-Text in Java
  - Einlesen und speichern unicode-Text in Java
- Reguläre Ausdrücke in Java

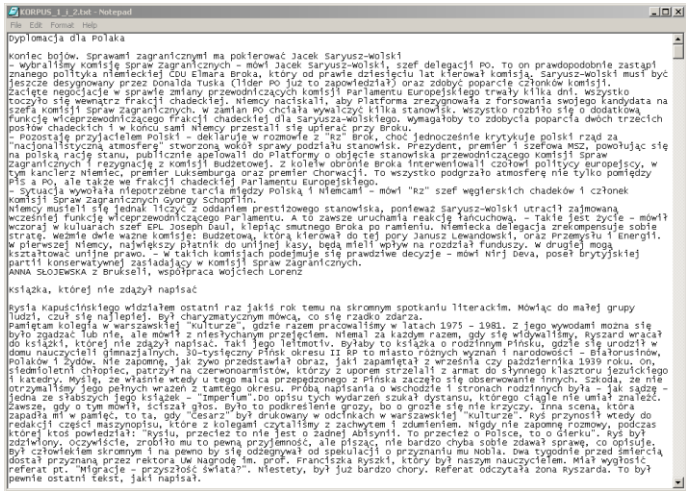
Kurz zu verwendeten Dateien

- Korpus
- Regeln
- Silbenmuster
- Namensendung mit dem Deklinationstyp

Korpus

- Artikel aus der polnischen Zeitung –online “Rzeczpospolita” (http://www.rzeczpospolita.pl/) in eine Text-Datei kopiert (manuell)





Korpus Bearbeitung

- Durchsuchung des Korpus per Handlese auf relevante Stellen und Markierung

- Erstellung der erwünschten Ergebnisse

  → 420 Namen

Erarbeitung den Regeln

- Überlegung wie können die Namen im Polnischen im Satz repräsentiert werden:

  zB: VN NN-NN oder VN NN oder NN

- Sammlung von den Schlüsselwörtern vor und nach den Namen aus verschiedenen Zeitschriften

- Entsprechende Erarbeitung den Regeln

Beispiel für die Schlüsselwörter

- Namen stehen nach Titeln

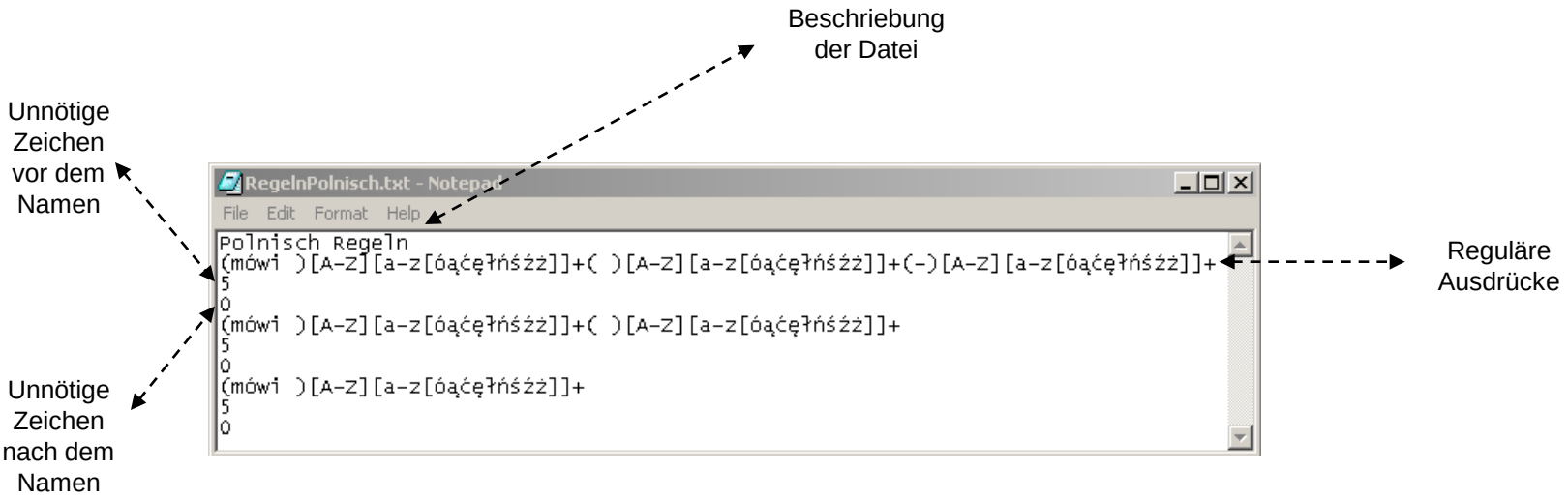
  Prof, gen. , Dr , Pan (Herr), Pani (Frau)
- Namen stehen vor Kommata, gefolgt von einer Berufs- oder Amtsbezeichnug

  Pani Pieczarek, przewodniczaca

  (Frau Pieczarek, Vorsitzende)
- Namen stehen nach Inquit-Formeln

  powiedział (sagte)

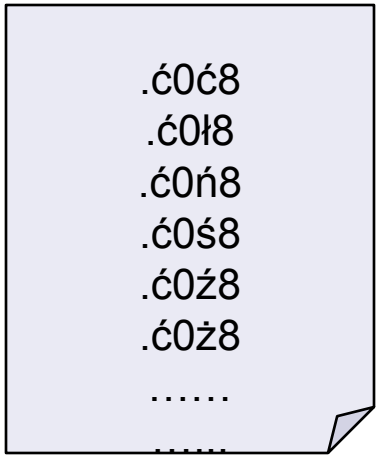
Struktur von Regelndatei



Regeln haben die Form: ein Schlüsselwort, wonach oder wovor ein Personenname erwartet wird wie oben

SILBENMUSTER BEISPIEL

Silbenmuster wurden übernommen von Hanna Kolodziejska



Namensendungen mit den Deklinationstypen

Paradigma mit der letzten Silbe und dem Deklinationstyp

Letze Silbe	**Deklinationstyp**
SKI	Sg. Nom. Vok. mask.
LEK	Sg. Nom. mask Sg. Nom. Gen. Dat. Akk. Ins. Lok. Vok. fem.
SKA	Sg. Nom. fem. Sg. Gen. Akk. mask.
REK	Sg. Nom. mask Sg. Nom. Gen. Dat. Akk. Ins. Lok. Vok. fem.

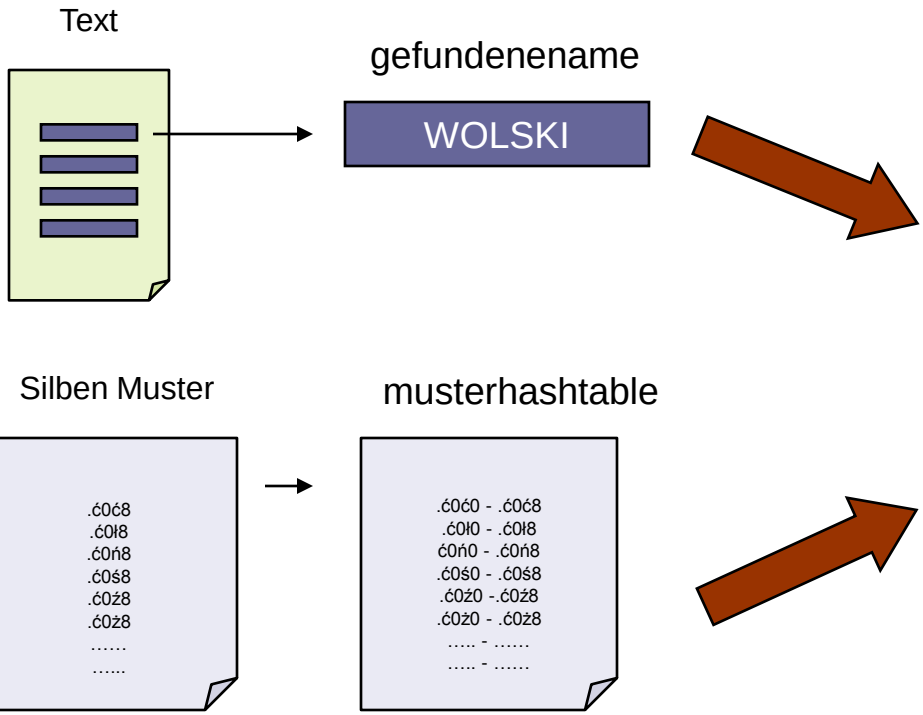
Das Programm Teil
Entity Recognition und Name-Regeln

- Programmiersprache Java
- Mehrere Zeilen werden analysiert, darauf die Regeln angewendet, um Namen zu finden
- Als output werden Namen ausgeben

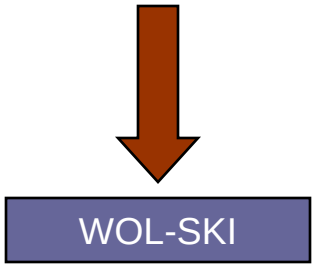
Das Programm Teil Silbentrenner

- Programmiersprache Java
- Basiert auf dem Algorithmus von Liang
- Der Algorithmus benutzt eine Liste von „hyphenation patterns“.
- Muster bestehen aus einer Buchstabenfolge mit Zahlenwerten zwischen 0 und 8 z.B.
  - “. 0 h 0 y 3 p 0 h 0“
- Die Zahlen geben die Trennbarkeit an:
  - gerade Zahl bedeutet: Trennung verboten,
  - ungerade Zahl bedeutet: Trennung erlaubt.

Silbentrenner



- m = .W0O0L0S0K0I0.
- Kurzepat (siehe nächste Seite) – es existieren viele
- Suche in Musterhashtable nach Kurzepat
- Wenn gefunden:
  - Wähle den original silben Muster
  - Vergleiche die Zahlen im m und orignal silben Muster.
  - Wenn die Zahl im orignal silben Muster größer ist als in m dann wechsle die Zahl in m
  - Lösche alle Punkte und gerade Zahlen
  - Tausche alle ungerade Zahlen gegen -



Silbentrenner
Kurzepat - .W0O0L0S0K0I0.

.											
.W	W										
.W0	W0	0									
.W0O	W0O	0O	O								
.W0O0	W0O0	0O0	O0	0							
.W0O0L	W0O0L	0O0L	O0L	0L	L						
.W0O0L0	W0O0L0	0O0L0	O0L0	0L0	L0	0					
.W0O0L0S	W0O0L0S	0O0L0S	O0L0S	0L0S	L0S	0S	S				
.W0O0L0S0	W0O0L0S0	0O0L0S0	O0L0S0	0L0S0	L0S0	0S0	S0	0			
.W0O0L0S0K	W0O0L0S0K	0O0L0S0K	O0L0S0K	0L0S0K	L0S0K	0S0K	S0K	0K	K		
.W0O0L0S0K0	W0O0L0S0K0	0O0L0S0K0	O0L0S0K0	0L0S0K0	L0S0K0	0S0K0	S0K0	0K0	K0	0	
.W0O0L0S0K0I	W0O0L0S0K0I	0O0L0S0K0I	O0L0S0K0I	0L0S0K0I	L0S0K0I	0S0K0I	S0K0I	0K0I	K0I	0I	I
.W0O0L0S0K0I.	W0O0L0S0K0I.	0O0L0S0K0I.	O0L0S0K0I.	0L0S0K0I.	L0S0K0I.	0S0K0I.	S0K0I.	0K0I.	K0I.	0I.	I. .

alle Buchstabenfolgen werden bis zu einer bestimmten Länge ermittelt.

Silbentrenner
Kurzepat - .W0O0L0S0K0I0.

.											
.W	W										
.W0	W0	0									
.W0O	W0O	0O	O								
.W0O0	W0O0	0O0	**O0**	0							
.W0O0L	W0O0L	0O0L	O0L	0L	L						
.W0O0L0	W0O0L0	0O0L0	O0L0	0L0	L0	0					
.W0O0L0S	W0O0L0S	0O0L0S	O0L0S	**0L0S**	L0S	0S	S				
.W0O0L0S0	W0O0L0S0	0O0L0S0	O0L0S0	0L0S0	L0S0	0S0	S0	0			
.W0O0L0S0K	W0O0L0S0K	0O0L0S0K	O0L0S0K	0L0S0K	L0S0K	0S0K	S0K	0K	K		
.W0O0L0S0K0	W0O0L0S0K0	0O0L0S0K0	O0L0S0K0	0L0S0K0	L0S0K0	0S0K0	S0K0	0K0	K0	0	
.W0O0L0S0K0I	W0O0L0S0K0I	0O0L0S0K0I	O0L0S0K0I	0L0S0K0I	L0S0K0I	0S0K0I	S0K0I	0K0I	K0I	0I	I
.W0O0L0S0K0I.	W0O0L0S0K0I.	0O0L0S0K0I.	O0L0S0K0I.	0L0S0K0I.	L0S0K0I.	0S0K0I.	S0K0I.	0K0I.	K0I.	0I.	I. .

.W8 **O1** **2L1S**

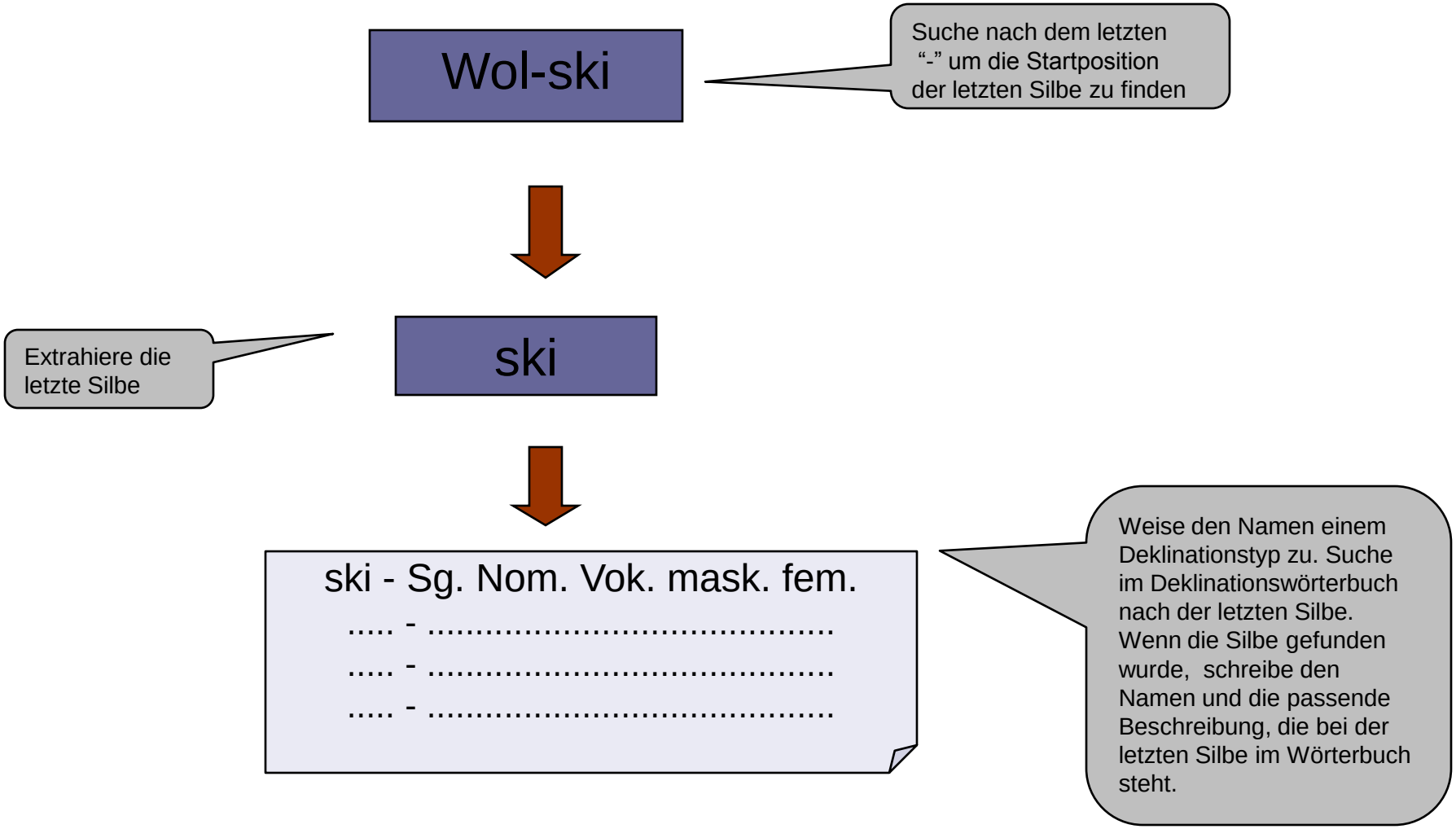
Kurzpat - .W0O0L0S0K0I0.

.	W	0	O	0	L	0	S	0	K	0	I	0	.
.	W	8											
			O	1									
				2	L	1	S						
.	W	8	O	2	L	1	S	O	K	0	I	0	.

Lösche alle Punkte und gerade Zahlen
Tausche alle ungerade Zahlen gegen “-”

WOL-SKI

Deklinationstypen Finder



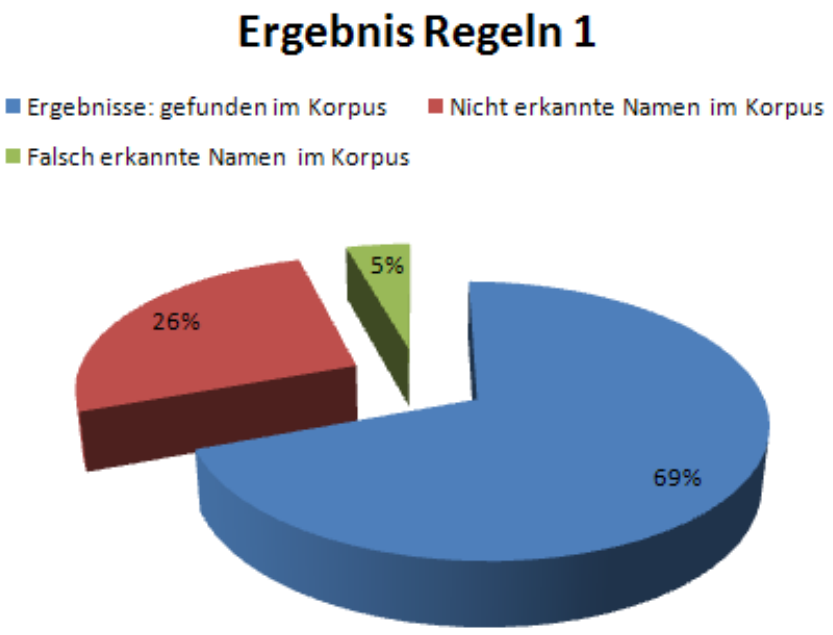
Ergebnis Beispiele

Input Namen
BOLEK, KOWALSKI, AGOREK

Namen	Deklinationstyp
AGOREK	Sg. Nom. mask. Sg. Nom. Gen. Dat. Akk. Ins. Lok. Vok. fem.
BOLEK	Sg. Nom. mask. Sg. Nom. Gen. Dat. Akk. Ins. Lok. Vok. fem.
KOWALSKI	Sg. Nom. Vok. mask.

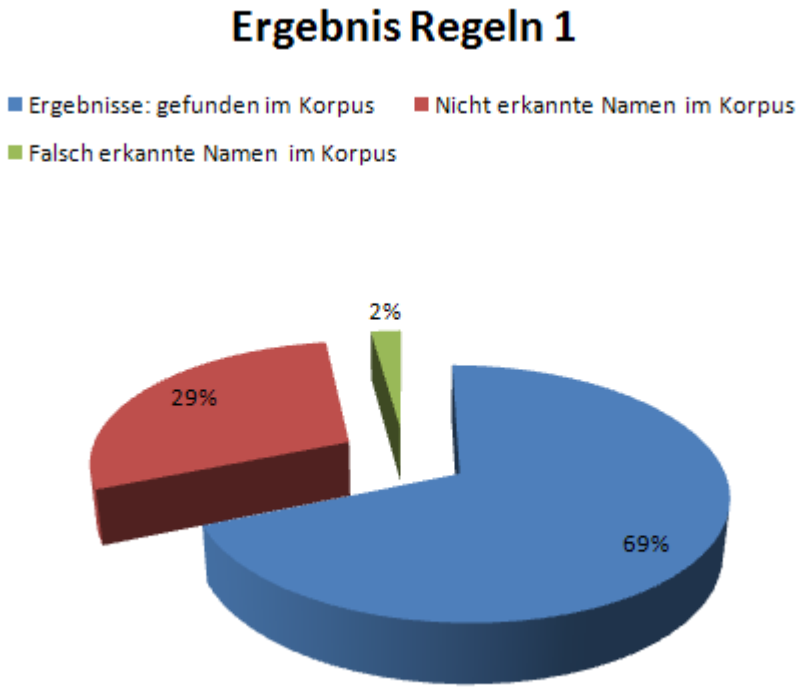
Ergebnisliste hat keine Doppeleinträge und ist sortiert

Ergebnis für Namenextraction



Korpus1				
Regeln 1	Alle Namen im Korpus	Ergebnisse: gefunden im Korpus	Nicht erkannte Namen im Korpus	Falsch erkannte Namen im Korpus
301	420	319	121	20

Ergebnis für Namenextraction

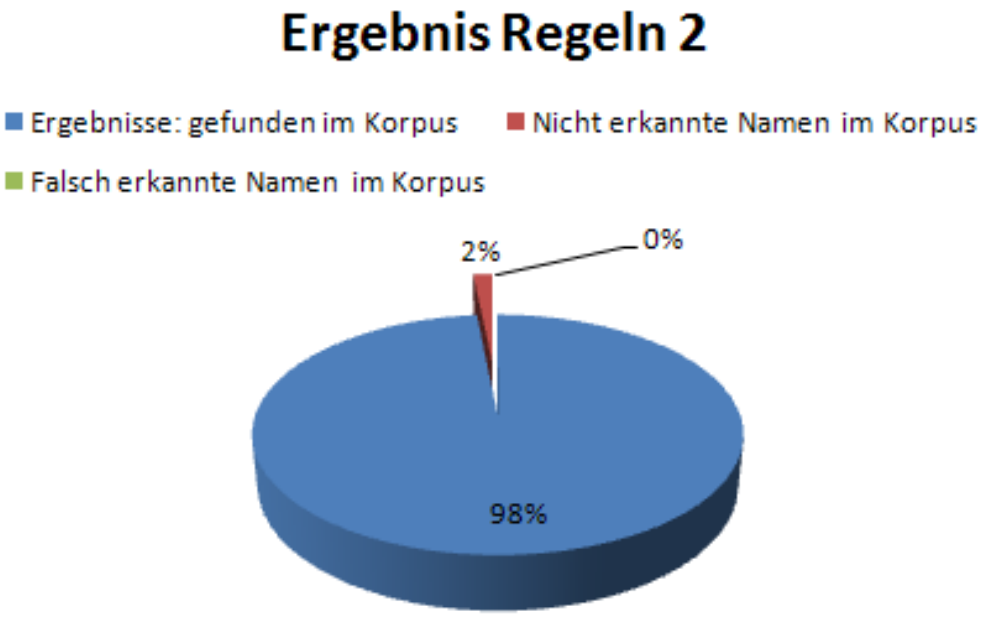


Korpus 2				
Regeln 1	Alle Namen im Korpus	Ergebnisse: gefunden im Korpus	Nicht erkannte Namen im Korpus	Falsch erkannte Namen im Korpus
301	178	125	53	4

Wichtige Beobachtung

- Regeln, die als Schlüsselwort ein Titel haben, zB. „Prof.“ liefern die besten Ergebnisse.

Ergebnis für Namenextraction



Korpus 2				
Regeln Titel	Alle Namen im Korpus , auf die die Regeln mit dem Schüsselwort Titel zutreffen	Ergebnisse: gefunden im Korpus	Nicht erkannte Namen im Korpus	Falsch erkannte Namen im Korpus
52	64	63	1	0

Gründe für nicht erkannte Namen

- Fehlende Regeln
- einfach Tipp Fehler

Gründe für falsch erkannte Namen

- Ambige Namen Personenname = Firmenname
- Im Polnischen ist es möglich, dass nach manchen Formen sowohl Personennamen als auch Land oder Firmen vorkommen, zB:
  - Prezydent Warszawy (Prezydent von Warschau)
  - Prezydent Kaczyński (Prezydent Kaczyński)
- In manchen Fällen trifft die Regel auch dann zu, wenn das Wort (eigentlich kein Name) am Anfang des Satzes (großgeschrieben) ist und danach folgt das Schlüsselwort, zB:
  - Profesor tłumaczy (Profesor erklärt)

Zusammenfassung

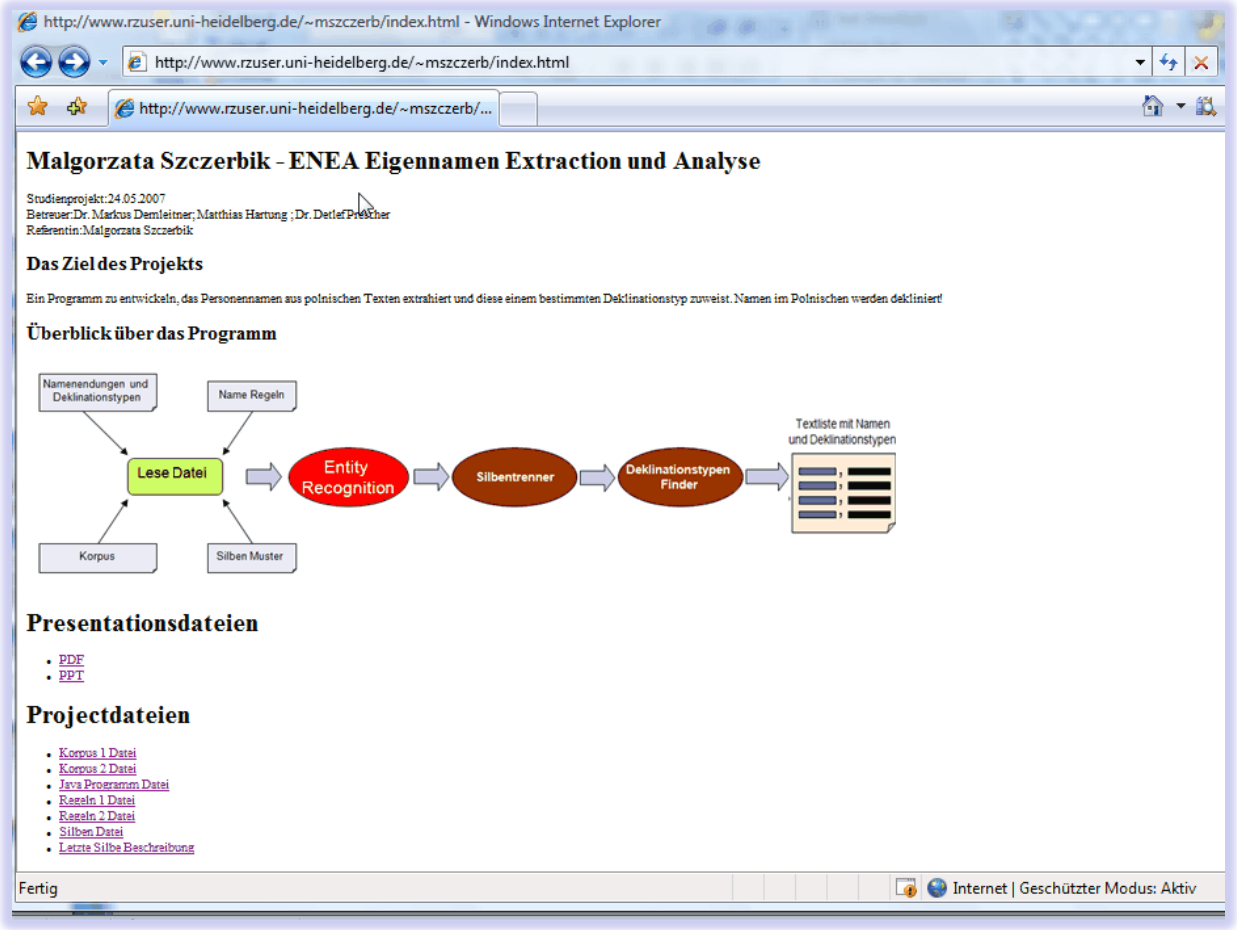
- Regeln aufwendig zu erstellen,
- Problem mit ambigen Namen die sowohl Firma- als auch Personenname repräsentieren
- Regeln die nur Titel beschreiben liefern die besten Ergebnisse
- Die Deklinationstypendatei muß um einsilbige Namen ergänzt werden
- Ambige Silben in der Deklinationstypendatei, wegen Konsonant-Alternationen

Lösungsvorschläge für Zweifelsfälle

- Für ambige Namen – Erstellung von Ignore Listen
- Für konsonanten Alternation – morphologische Analyse die zurück zum Stamm führt

Projekt Informationen

http://www.rzuser.uni-heidelberg.de/~mszczerb/index.html



- Danke für Eure Aufmerksamkeit
- Die Diskussion ist hiermit eröffnet