Semantic Argument Classification

28. Januar 2015

Julian Baumann, Kevin Decker, Maximilian Müller-Eberstein

Institut für Computerlinguistik
Universität Heidelberg

UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

Gliederung

Problemstellung

Daten
Problemstellung

Umsetzung
Features
Featureextraktion
Schwierigkeiten

Experimente

Ausblick

Literatur

Referenzen

Problemstellung

Daten
- Problemstellung

Umsetzung
- Features
- Featureextraktion
- Schwierigkeiten

Experimente

Ausblick

Literatur

Semantic Argument Classification

Was ist Semantic Argument Classifcation?

- Zuweisung bestimmter Rollen in einem Satz $\Rightarrow$ „Wer tut wem was an?“
- It operates stores mostly in Iowa and Nebraska
- [**Arg0** *It*][**Pred** *operates*][**Arg1** *stores*][**ArgLoc** *mostly in Iowa and Nebraska*]

Daten

Problemstellung

Daten

Problemstellung

Umsetzung

Features

Featureextraktion

Schwierigkeiten

Experimente

Ausblick

Literatur

Referenzen

- NLTK
- PropBank

PropBank

- versucht generalisierte Argumente zu verwenden → Parser
- Argumentrollen sind für jedes Verb in Frames organisiert → weniger spezifisch
- ARG0 = Proto-Agent
- ARG1 = Proto-Patient
- ARG2-ARG5 = Argumente mit steigender Intensität

PropBank

- [**ARG0** *He*][**Predicate** *wrote*][**ARG1** *a book*]
- [**ARG0** *He*][**Predicate** *wrote*][**ARG2** *about them*]
- [**ARG0** *He*][**Predicate** *wrote*][**ARG3** *a book for his children*]
- $\rightarrow$ verschieden Rollen

Penn Treebank

Problemstellung

Daten

Problemstellung

Umsetzung

Features

Featureextraktion

Schwierigkeiten

Experimente

Ausblick

Literatur

Referenzen

- Subkorpus aus WSJ und Brown Corpus, bestehend aus ungefähr 1.000.000 Wörtern
- 112.917 Prädikat-Argument Strukturen annotiert nach PropBank-Annotationsschema
- 292.975 Instanzen
- wsj/00/wsj_0001.mrg 1 10 gold publish.01 p—a 10:0-rel 11:0-ARG0

Penn Treebank

Problemstellung

Daten

Problemstellung

Umsetzung

Features

Featureextraktion

Schwierigkeiten

Experimente

Ausblick

Literatur

Referenzen

```
((S
    (NP-MNR-SBJ
        (NP (DT The) (NN way) )
        (SBAR
            (WHADVP-1 (-NONE- 0) )
            (S
                (NP-SBJ (NNP MacArthur) )
                (VP (VBD said)
                    (NP (PRP his) (NN line) )
                    (ADVP-MNR (-NONE- *T*-1) )))))
(: - -)
```

Features

Problemstellung

Daten

Problemstellung

Umsetzung

Features

Featureextraktion

Schwierigkeiten

Experimente

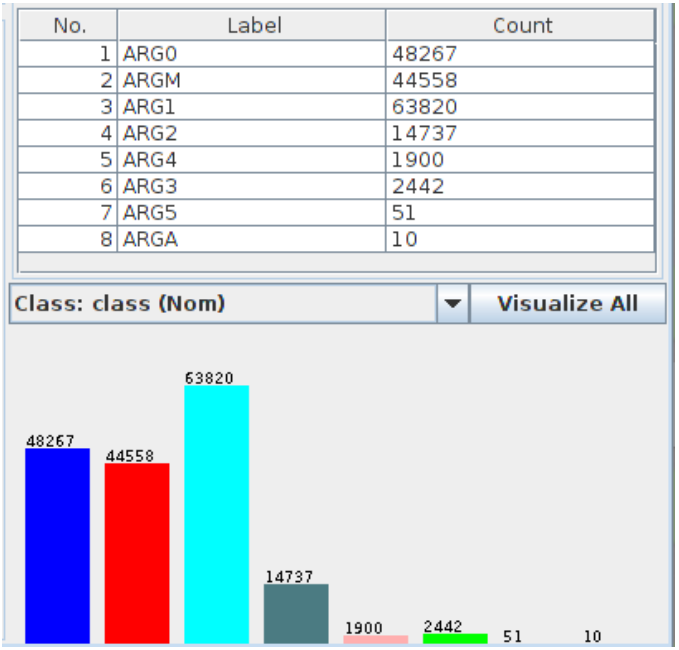
Ausblick

Literatur

Referenzen

- Predicate
- Path
- Phrase Type
- Position
- Voice

No.	Label	Count
1	ARG0	48267
2	ARGM	44558
3	ARG1	63820
4	ARG2	14737
5	ARG4	1900
6	ARG3	2442
7	ARG5	51
8	ARGA	10



Predicate

Problemstellung

Daten

Problemstellung

Umsetzung

Features

Featureextraktion

Schwierigkeiten

Experimente

Ausblick

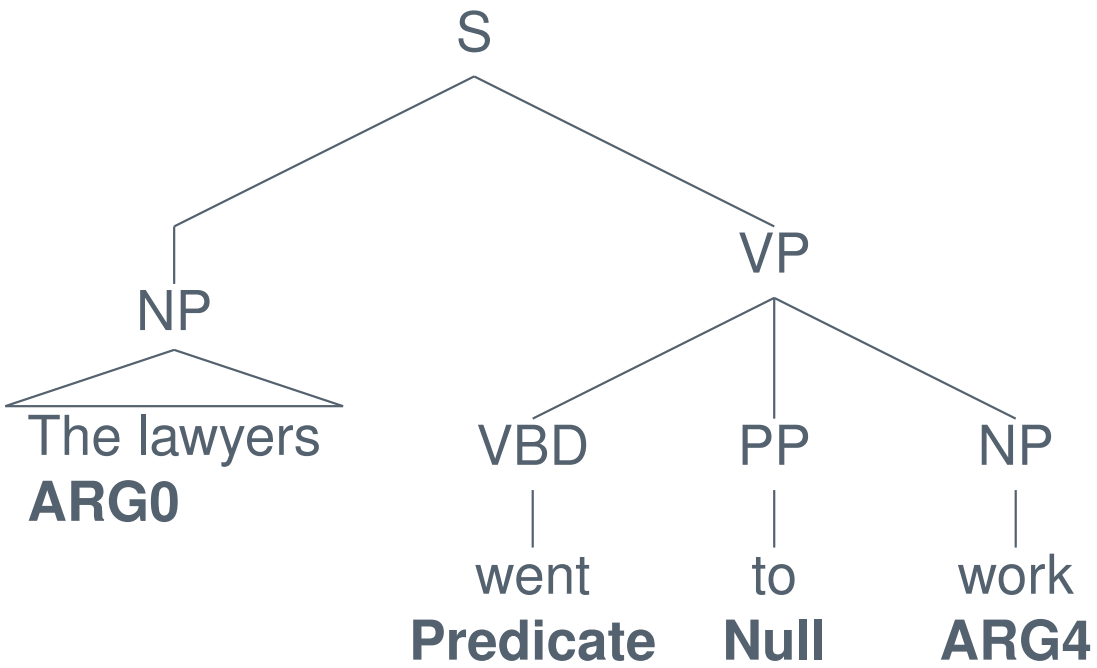
Literatur

Referenzen

- lemmatisierte Prädikat
- 3966 distinct

Path

- beschreibt Pfad zwischen ARG und Predicate
- vereinfacht z.B. NP-SBJ $\rightarrow$ NP
- extrahiert über Lowest Common Ancestor
- beispielsweise: NP↑S↓VP↓VBD
- 41737 distinct



Phrase Type

Problemstellung

Daten

Problemstellung

Umsetzung

Features

Featureextraktion

Schwierigkeiten

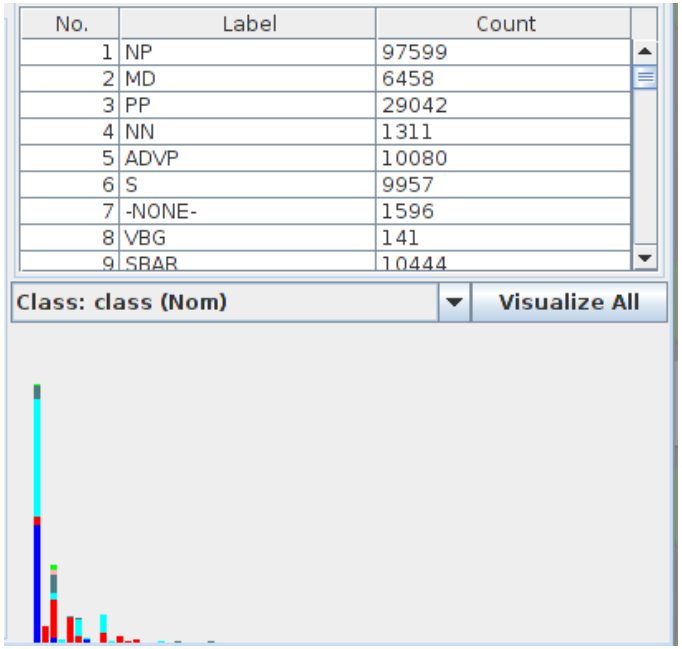
Experimente

Ausblick

Literatur

Referenzen

- beschreibt die Kategorie des Argument
- z.B: NP, MD, PP, SBAR
- 65 distinct feature values



No.	Label	Count
1	NP	97599
2	MD	6458
3	PP	29042
4	NN	1311
5	ADVP	10080
6	S	9957
7	-NONE-	1596
8	VBG	141
9	SBAR	10444

Class: class (Nom) Visualize All

Position

Problemstellung

Daten

Problemstellung

Umsetzung

Features

Featureextraktion

Schwierigkeiten

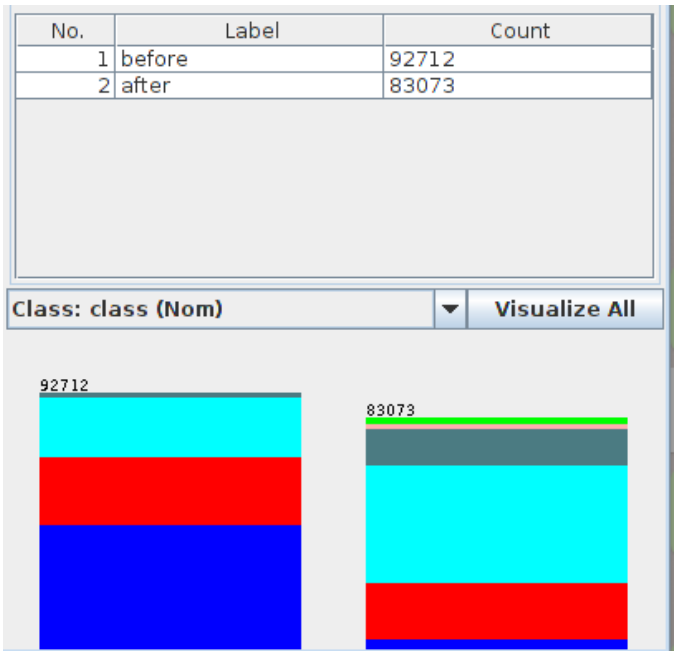
Experimente

Ausblick

Literatur

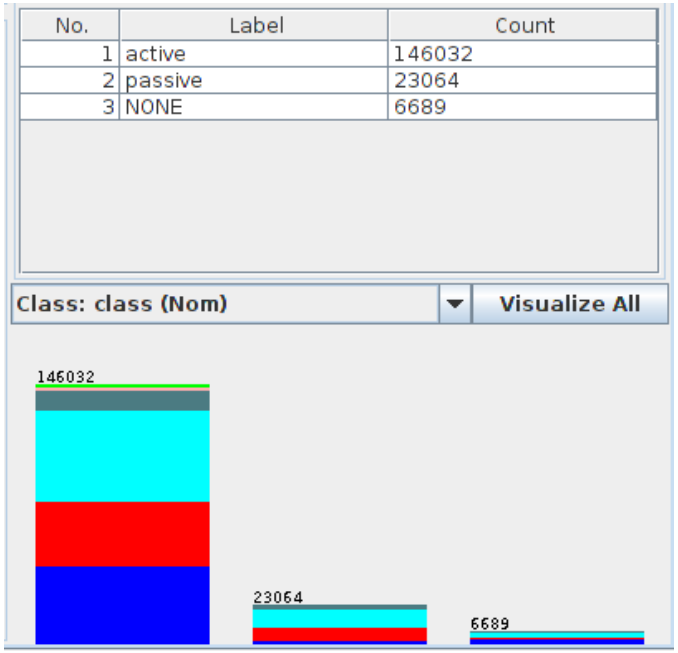
Referenzen

- Beschreibt, ob das Argument vor oder nach dem Prädikat steht
- Binäres Feature



Voice

- gibt an, ob das Prädikat aktiv oder passiv ist
- größtenteils annotiert
- 3 distinct feature values: active, passive, unknown



Featureextraktion


```
    featureList = [...] # zu extrahierende Features
    extArgList = []
    for pbInstance in pbInstances :
            for pbArg in pbInstance.arguments :
            for feature in featureList :
                    extArgList.append(extFeature(feature, pbArg,
pbInstance))
    # write features to file in ARFF
```

Featureextraktion

Problemstellung

Daten

Problemstellung

Umsetzung

Features

Featureextraktion

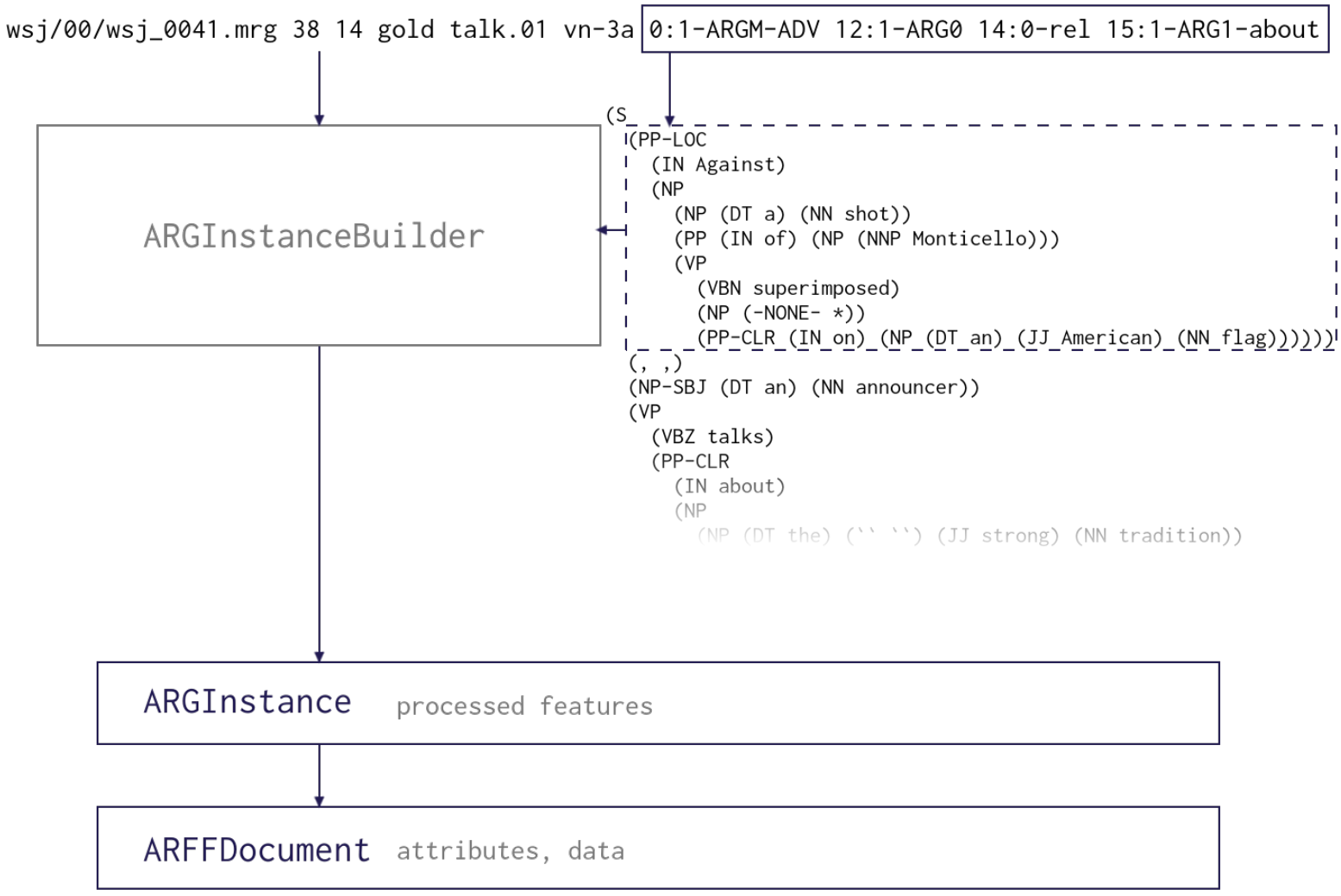
Schwierigkeiten

Experimente

Ausblick

Literatur

Referenzen



ARFF

@relation SAC_All

@attribute predicate {join,publish,name,use, make, cause, ...}
@attribute phraseType {NP, MD, PP, NN, ADVP, S, ...}
@attribute position {before, after}
@attribute path {NP^S!VP!VP, MD^VP^S!VP!VP,...}
@attribute voice {active, passive, NONE}
@attribute class {ARG0, ARGM, ARGA, ARG1, ...}

@data
join, NP, before, NP^S!VP!VP, active, ARG0
join, MD, before, MD^VP^S!VP!VP, active, ARGM
join, NP, after, NP^VP^VP^S!VP!VP, active, ARG1
join, PP, after, PP^VP^VP^S!VP!VP, active, ARGM
join, NP, after, VP^VP^S!VP!VP, active, ARGM

Schwierigkeiten

Problemstellung

Daten

Problemstellung

Umsetzung

Features

Featureextraktion

Schwierigkeiten 17

Experimente

Ausblick

Literatur

Referenzen

- PropBankChain- und PropBankSplitTreePointer
- Verwendung einer externen PennTreeBank
- einige Feature (bsp. path) nehmen sehr viele Werte an
- Speicherbedarf der Algorithmen J48 und SVM

Setup

Problemstellung

Daten

Problemstellung

Umsetzung

Features

Featureextraktion

Schwierigkeiten

Experimente

Ausblick

Literatur

Referenzen

- 60% train,20% dev, 20% test
- Baseline: ZeroR
- Naive Bayes, j48 tree, *(libSVM)*
- bisher: Training auf train, Evaluierung mit dev

Ergebnisse

	Precision	Recall	F-Measure
Baseline	*0.132*	*0.364*	*0.194*
Naive Bayes	0.771	0.778	0.770
j48 Tree	0.784	0.786	0.781

Feature Evaluation

a	b	c	d	e	f	g	h	<– classifie
14497	348	361	272	0	4	0	1	a = ARG0
291	11394	2143	1009	189	48	0	1	b = ARGM
3119	707	17064	375	31	27	0	0	c = ARG1
180	792	1854	2163	29	23	0	0	d = ARG2
2	217	23	141	379	3	0	0	e = ARG3
37	289	144	147	170	99	0	0	f = ARG4
0	13	0	1	0	0	3	0	g = ARG5
5	0	0	0	0	0	0	0	h = ARGA

Confusion Matrix (Naive Bayes)

	Precision	Recall	F-Measure	F-Measure Ch
All Features	*0.771*	*0.778*	*0.770*	*0*
- voice	0.748	0.754	0.745	-0.025
- path	0.778	0.783	0.776	**+0.006**
- phraseType	0.735	0.747	0.733	-0.037
- position	0.758	0.773	0.757	-0.013
-predicate	0.717	0.732	0.716	**-0.54**

toDo

Problemstellung

Daten

Problemstellung

Umsetzung

Features

Featureextraktion

Schwierigkeiten

Experimente

Ausblick

Literatur

Referenzen

- Path Feature überarbeiten
- HeadWord Feature implementieren
- genauere Evaluation
- *SVM?*
- Abschlussbericht schreiben

Quellen

[1] Omri Abend und Roi Reichart. *Unsupervised Argument Identification for Semantic Role Labeling*.

[2] Jean Carletta. "Assessing agreement on classification tasks: the kappa statistic". In: *Computational Linguistics* (1996), S. 249–254.

[3] Daniel Gildea. "Automatic labeling of semantic roles". In: *Computational Linguistics* 28 (2002), S. 245–288.

[4] Alessandro Moschitti und Cosmin Adrian Bejan. "A Semantic Kernel for Predicate Argument Classification". In: *IN CONLL 2004*. 2004, S. 17–24.

[5] Sameer Pradhan u. a. *Support Vector Learning for Semantic Argument Classification*. 2005.

Vielen Dank für Eure Aufmerksamkeit!
Noch Fragen?

UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386