Abschlussbericht

Named Entity Classification

Madita Huvar, Phillip Richter-Pechański, Sanaz Safdel

28. Februar 2017

Inhaltsverzeichnis

1. Einführung **3**
1.1. Was sind Named Entities? 3

2. Unser Projekt **3**
2.1. Ziel 3
2.2. Organisation 3

3. Daten und Tools **4**
3.1. Korpus 4
3.2. Korpusreader 4
3.3. Korpusklassen 5

4. Klassifizierung **5**
4.1. Featureset 5
4.2. Klassifizierertyp 6

5. Evaluation **6**

6. Ausblick und Zusammenfassung **7**

Literatur **8**

A. Arbeitsplan **11**

B. Tabellen **12**

C. Abbildungen **16**

1. Einführung

Named Entity Recognition (NER) ist bereits seit den 1990er Jahren ein aktives Forschungsfeld.[1][13][6] NER bilder die Grundlage für weitere Forschung in vielen Bereichen der maschinellen Sprachverarbeitung, u.a. Information Retrieval, Semantic Annotation, Question Answering, Opinion Mining, usw.[6]

1.1. Was sind Named Entities?

Named Entities sind Phrasen, die Namen von Personen, Organisationen, Währungen, Orten usw. enthalten. Eine Named Entity kann abstrakter oder physischer Natur sein. Für weitere Informationen siehe (`https://en.wikipedia.org/wiki/Named_entity`).
Im Folgenden sind einige Beispiele für Named Entities aufgelistet:

- The Speaker of the [ORG U.N.] ...
- President [PER Obama] ...
- The price of the [MONEY Dollar] lost ...
- The city of [LOC Moscow] is the capital of Russia.

2. Unser Projekt

Typischerweise werden Named Entity Recognition und Named Entity Classification (NEC) zusammen betrachtet. Nur wenige Untersuchungen beschäftigen sich ausschließlich mit NEC.[11][4] Im Gegensatz zur NER beschäftigt sich die NEC nur mit positiven NE-Instanzen. Dies führt dazu, dass dieses Projekt als globales Evaluationsmaß lediglich Accuracy nutzt, da $Precision = Accuracy$. [1]

2.1. Ziel

Dieses Projekt konzentriert sich auf NEC und stellt die Frage, **welchen Einfluss Feature Selection auf die Klassifikationsergebnisse eines Named Entity Klassifizierers hat**. Dabei nutzen wir einfache syntaktische und lexikalische Features, die bereits in fast allen Forschungsarbeiten in ähnlicher Form genutzt wurden. [14][5][12]

2.2. Organisation

Für unser Projekt haben wir einige nützliche Tools zum kollaborativen Arbeiten genutzt. Für Aufgaben rund ums Thema Programmierung haben wir GitHub (`https://github.com/MaviccPRP/ml_ner`) eingesetzt. Zum Austausch organisatorischer Fragen, nutzten wir

[1] siehe `http://www.cl.uni-heidelberg.de/courses/ss16/annotierteKorpora/material/NLPW_SS16_F04_EvalPRFA.pdf`

eine eigene Whatsapp Gruppe. Zwischenergebnisse und Protokolle gemeinsamer Treffen haben wir in einer eigens angelegten Wikiseite festgehalten. [2]

3. Daten und Tools

Unser Projekt baute im Wesentlichen auf folgende Tools auf:

- Python 3.4+
- Scikit Learn als Klassifizierer
- liac-arff
- matplotlib
- Weka zur Korpusanalyse

Python 3 kommt zum Einsatz, weil diese Version eine problemlose Verarbeitung von Unicode ermöglicht. Der Scikit Learn Klassifizierer bietet gute Evaluationstools, die in diesem Projekt zum Einsatz gekommen sind. Das Modul liac-arff wandelt Python Dictionaries in .arff-Dateien um, wodurch wir mittels WEKA Korpusanalysen durchführen konnten.

3.1. Korpus

Für die Named Entity Klassifikation wurde der OntoNotes Korpus 2012 genutzt.[15] Wir konzentrierten uns dabei lediglich auf die englischen Nachrichtentexte des 'The Wall Street Journal'. Für unsere Entwicklungsphase stellte der Korpus ein bereits vorgefertigtes Developmentset zu Verfügung. Für die eigentliche Klassifikation der Named Entities kamen dann die Trainings- und Testdatensets des Korpus zum Einsatz. Während das Trainingsset den manuell annotierten Goldstandard enthielt, griffen wir beim Testset auf die automatisch annotierten Datensets des OntoNotes Korpus zurück, da kein Goldstandard zur Verfügung stand. Die Anzahl der Named Entity Instanzen in den verschiedenen Datensets sind in Tabelle 1 zusammengefasst.

3.2. Korpusreader

Für die Extraktion der Named Entities wurde ein Korpusreader erstellt. Der Reader extrahiert alle Named Entities und weist ihnen folgende Informationen zu.

- POS-Tags der einzelnen Token
- Phrasenart der Instanz
- Kontextwörter (ne-1, ne+1) der Instanz

[2] `https://wiki.cl.uni-heidelberg.de/foswiki/bin/view/Studenten/MLGruppeNERRec`

- Instanzklasse

Im Folgenden eine Beispielextraktion zu dem Satz:
*Says **Peter Mokaba**, President of the South African Youth Congress:*

{'PERSON':[['Peter', 'NNP'],['Mokaba', 'NNP'],'NP', ('Says', ',')]}

3.3. Korpusklassen

Die verwendeten Korpusklassen im originalen OntoNotes Korpus sind in Tabelle 2 aufgelistet.
Zehn Klassen enthalten dabei nur relativ wenige Named Entity Instanzen. Deshalb haben wir die Klassen für unser Projekt teilweise zusammengefasst. Selten vorkommende Klassen wurden aus dem balancierten Korpus entfernt. Neben semantisch ähnliche Klassen wie NORP und GPE wurden zudem numerische Klassen wie MONEY, PERCENT und CARDINAL zusammengefasst. Eine Beschreibung der Klassen findet sich in Tabelle 3. Die Anzahl der Named Entity Instanzen sind in Tabelle 4 aufgelistet.

4. Klassifizierung

4.1. Featureset

Um eine Art untere Grenze für die Evaluationsergebnisse unseres Klassifizierers zu setzen, wurde eine Baseline definiert. Diese enthielt lediglich Unigramme als Features. Die Unigramme haben wir zuvor nach folgenden Kriterien aus dem Korpus extrahiert:

- Die Unigramme sind Teil einer Named Entity Instanz
- Die Unigramme kamen mindestens fünfmal im gesamten Trainingskorpus vor.

Weitere Details zur Unigrambaseline finden sich in Tabelle 5. Insgesamt produzierte die Baseline 1317 verschiedene Unigramme. Zur besseren Unterscheidung nutzen wir im Folgenden den Begriff Featurekategorie, wenn wir von einem Featuretyp sprechen, wie etwa 'Unigram' und 'POS Tags'. Features sind die atomaren Teile der Featurekategorien. Während die Featurekategorie 'Unigram' über 1300 Features enthält, steht die Featurekategorie 'isInWiki' für lediglich ein Feature.

Unser komplettes Featureset, inklusive Beschreibung und Beispielextraktionen, ist in Tabelle 6 zusammengefasst. Dabei haben wir drei mehrdimensionale Featurekategorien genutzt. Neben den Unigrammen nutzten wir den Instanzcontext, der jeweils das Token vor und nach der Named Entity Instanz enhält. Darüber hinaus definierten wir noch die Anzahl der verschiedenen POS Tags in den Instanzen.
Insgesamt bestand das komplette Featureset aus 1716 atomaren Features.

4.2. Klassifizierertyp

Zur Klassifizierung der Named Entities wurde eine Support Vector Maschine mit linearem Kernel verwendet. Nach einer ausführlichen Evaluation[3] aller möglichen Hyperparameter haben sich folgende Einstellungen als optimal erwiesen.

- Klassifiziererklasse='sklearn.svm.LinearSVC'
- loss='squared hinge'
- penalty='l2'

Unsere Featurevektoren haben sehr viele atomare Features, weshalb der linearer Kernel die besten Ergebnisse lieferte. Bei nichtlinearen Kerneln scheint es zum Overfitting zu kommen. Zudem hat Chih-Wei[3] festgestellt, dass das Mapping in einen höheren Featurespace eines nicht-linearen Kernels kaum Klassifizierungsverbesserungen gibt.

Alternativ wurde ein Decisiontree getestet, dieser hatte allerdings mit allen Featurekombinationen tendenziell schlechtere Evaluationsergebnisse. Zudem trainierte die SVM deutlich schneller.

5. Evaluation

Insgesamt wurden elf Featurekategorien eingesetzt. Um die Performance der einzelnen Featurekategorien zu testen, wurde zuerst die Potenzmenge des Featuresets gebildet.

Für die Evaluation entscheidend waren alle Teilmengen, die die Featurekategorien 'Unigram' und 'Context' enthielten und insgesamt mind. drei Featurekategorien besitzen. Diese Einschränkungen wurden zum einen aufgrund von Zeitersparnissen gewählt, außerdem ergaben Tests, dass Klassifizierer ohne diese Mindestanforderungen erheblich schlechtere Ergebnisse lieferten:

- Accuracy aller Teilmengen ohne die Featurekategorien 'Unigram' und 'Context': $<69\%$.
- Accuracy nur mit 'Unigram' und 'Context': 87.42%

Schließlich wurde der Klassifizierer auf allen restlichen 1013 Teilmengen durchgeführt. Die Accuracy der einzelnen Klassifizierungen sind in Abbildung 1 dargestellt.
Die höchste Accuracy erreichten wir ab sieben Featurekategorien. Ab vier Featurekategorien gab es kaum mehr Verbesserungen der Accuracywerte. Featurekategorien, die zur Erhöhung der Accuracy beitragen waren insbesondere:

- 'POS'

[3]Zum Tuning der Hyperparameter kam das Scikit Learn Tool GridSearch zum Einsatz: `http://scikit-learn.org/stable/modules/grid_search.html`

- 'is all caps'
- 'is in wiki'
- 'is np'
- 'contains digit'

Das Featureset bestehend aus den vier Featurekategorien: **'Unigram', 'Context', 'POS', 'is all caps'** erreichte die beste Accuracy bei möglichts kleinem Featureset.
Die ROC Kurve 2 und die Confusionmatrix 7 geben weitere Auskunft über die Qualität der Klassifizierung. Auffallend ist hier das schlechte Abschneiden der PERSONEN Instanzen. Insbesondere die Precision fällt hier deutlich ab. Mögliche Ursache könnten sein, das wir auch nach der Klassenzusammenfassung, relativ wenige PERSONEN Instanzen im Trainingsset hatten.

Tabelle 8 fasst die Evaluationsergebnisse der Baseline im Vergleich zum optimalem Featureset zusammen. Hierbei wird deutlich, dass sich die Accuracy sowohl der balancierten als auch der unbalancierten Klassen deutlich verbessert hat. Das beste Ergebnis wurde mit dem optimalen Featureset und balancierten Klassen erreicht. Der Accuracywert lag bei 92.3%.

6. Ausblick und Zusammenfassung

Zusammenfassend lässt sich sagen, dass mehr Featurekategorien nicht unbedingt bessere Evaluationsergebnisse liefern. Sehr wohl scheint die Dimensionalität der Featurekategorien, einen entscheidenen Einfluss auf die Klassifikationsergebnisse zu haben. Hochdimensionale Featurekategorien, wie 'Unigram' und 'Context', trugen maßgeblich zu besseren Klassifikationsergebnissen bei. Darüber hinaus hat die Zusammenfassung der Named Entity Klassen ebenfalls zur Verbesserung der Ergebnisse geführt.

Als mögliche weitere Schritte könnte man den Context, der zur Zeit auch Satzzeichen einbezieht (oft ',' oder '.'), auf alphanumerische Zeichen beschränken. Damit würden eventuell häufig vorkommende Contextfenster, die lediglich aus Satzzeichen bestehen, durch eindeutigere Features bestehend aus Inhalts- und Funktionswörtern ersetzt.

Die schlechteren Ergebnisse bei der PERSON-Klassifizierung könnte durch die Generierung weiterer PERSON-Instanzen möglicherweise verbessert werden.

Literatur

[1] Borthwick, A. (1999): A Maximum Entropy Approach to Named Entity Recognition, Diss., New York.

[2] Chieu H. (2003): Named Entity Recognition with a Maximum Entropy Approach. In Proceedings of CoNLL-2003.

[3] Chih-Wei (2003): A Practical Guide to Support Vector Classification.

[4] He, Q.; Spangler, S. (2016): Semi-supervised data integration model for named entity classification. Google Patents.

[5] J. Kazama, K. Torisawa (2007): Inducing Gazetteers for Named Entity Recognition by Large-scale Clustering of Dependency Relations. In: Proceedings of ACL-08.

[6] Marrero, M.; Urbano, J. (2013): Named Entity Recognition. Fallacies, challenges and opportunities. In: Computer Standards & Interfaces 35 (5).

[7] Marrero, M.; Sánchez-Cuadrado, S. (2009): Evaluation of Named Entity Extraction Systems. In: Advances in Computational Linguistics. Research in Computing Science.

[8] Mayfield, J.; McNamee, P. (2003): Named entity recognition using hundreds of thousands of features. In: Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003.

[9] Munro, R.; Ler, D. (2003): Meta-Learning Orthographic and Contextual Models for Language Independent Named Entity Recognition. In Proceedings of the seventh conference on Natural language learning at HLT-NAACL.

[10] Nadeau, D.; Turney, P. (2006): Unsupervised Named-Entity Recognition: Generating Gazetteers and Resolving Ambiguity. In: Proceedings of the 19th international conference on Advances in Artificial Intelligence: Canadian Society for Computational Studies of Intelligence.

[11] Primadhanty, A.; Carreras, X. (2014): Low-Rank Regularization for Sparse Conjunctive Feature Spaces: An Application to Named Entity Classification. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics.

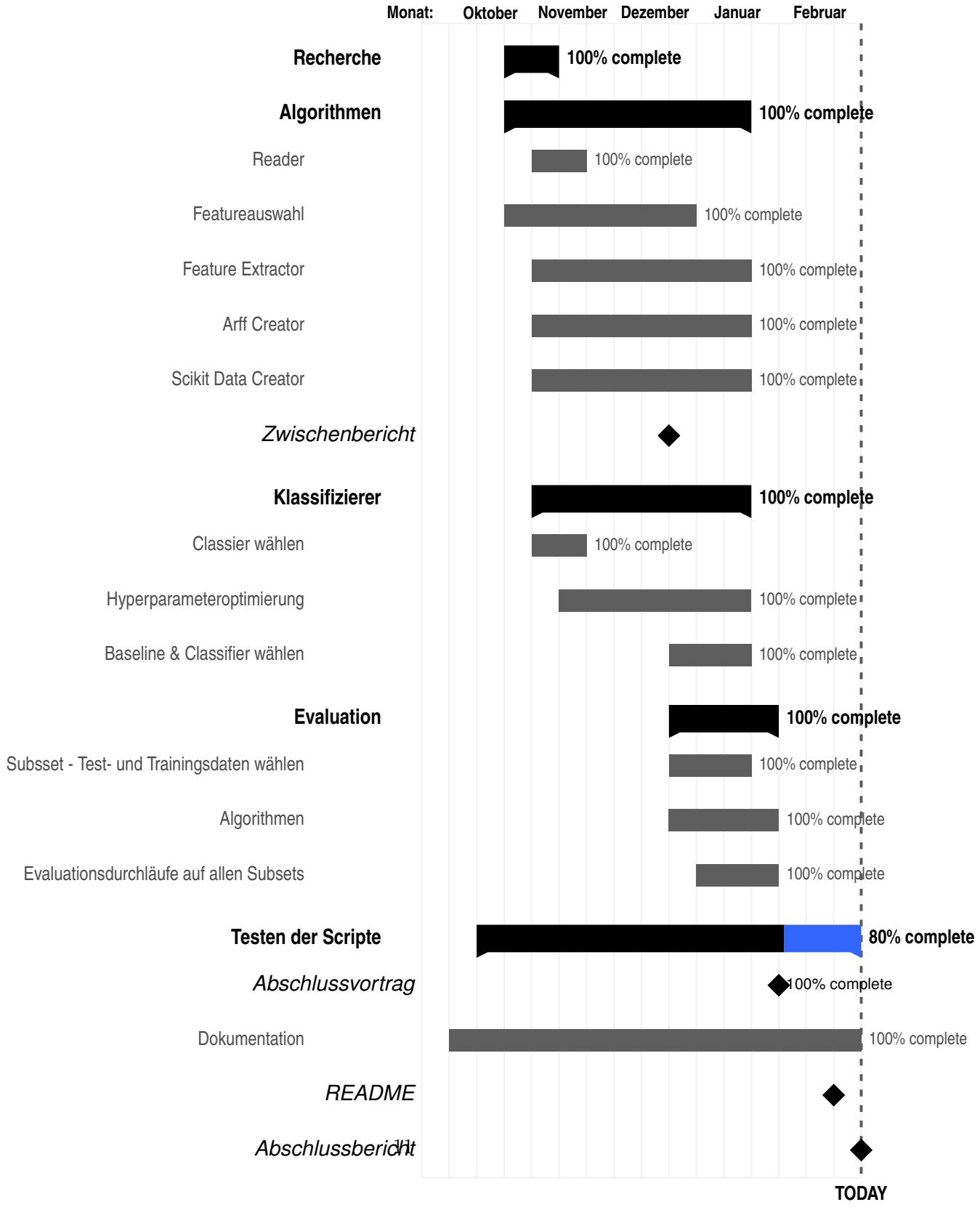
[12] Ratinov, L.; Roth, D. (2009): Design Challenges and Misconceptions in Named Entity Recognition. In Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL).

[13] Tjong Kim Sang,E.; De Meulder, F. (2003): Language-Independent Named Entity Recognition. In Proceedings of CoNLL-2003.

[14] Toral, A.; Munoz, R. (2006): A proposal to automatically build and maintain gazetteers for Named Entity Recognition by using Wikipedia.

[15] Weischedel, R. (2013): OntoNotes release 5.0. [Philadelphia, Pa.]: Linguistic Data Consortium.

A. Arbeitsplan



B. Tabellen

Tabelle 1: Anzahl an Named Entities

Developmentset	Trainingset	Testset
3325	23686	2996

Tabelle 2: Klassen im OntoNotes Korpus (OntoNotes Release 5.0 2012)

Klassen	Trainingset
ORG	5788
DATE	4080
PERSON	3756
GPE	3601
CARDINAL	1852
MONEY	1509
NORP	1484
PERCENT	1061
FAC, LOC, PRODUCT, EVENT, WORK_OF_ART, LAW, LANGUAGE, TIME, QUANTITY, ORDINAL	< 1800

Tabelle 3: Balancierte Klassen

Klassen	Beschreibung
PERSON	People, including fictional
NORP_GPE	Nationalities or religious or political groups; Countries, cities, states
ORGANIZATION	Companies, agencies, institutions, etc.
DATE	Absolute or relative dates or periods
PERCENT_MONEY_CARDINAL	Percentage (including "%"); Monetary values, including unit; Numerals that do not fall under another type

Tabelle 4: Verteilung der Klassen nach Balancierung

Klassen	Developmentset	Trainingset	Testset
ORG	930	5857	859
GPE_NORP	732	5134	588
PERCENT_CARDINAL_MONEY	564	4672	529
DATE	613	4254	601
PERSON	486	3759	413

Tabelle 5: Featurekategorie für den Baseline-Klassifizierer

Feature	Wert	Beschreibung	Beispiel
Unigram	numerisch	Vorkommenshäufigkeit der Unigramme (lemmatisiert) in der NE, die mindestens fünfmal im Trainingscorpus vorkommen. [8]	(america : 1, north : 1)

Tabelle 6: Featurekategorien für den Klassifizierer

Feature	Wert	Beschreibung	Beispiel
Unigram	numerisch	Häufigkeit der Unigramme (lemmatisiert), die mindestens fünfmal im Trainingscorpus vorkommen. [8]	(america : 1, north : 1)
POS	numerisch	Häufigkeit von 36 POS-Tags aus der Penn Treebank [2]	NNP: '2'
isAllCaps	boolean	Wörter nur in Großschreibung [10]	(0)
Context	numerisch	Häufigkeit der Kontexttokens. Beinhaltet Vorgänger- und Nachfolgetoken der NE. [9]	(,_and : 1)
containsDigit	boolean	Vorkommen von Nummern.	(0)
isInWiki	boolean	Vorkommen der NE in der Wikipedia. [14]	(1)
isTitle	boolean	Prüft, ob Titelbezeichnungen (z.B. Mr., MA) vorkommen. [12]	(0)
isNP	boolean	Ist NE eine Nominalphrase. [7]	(1)
isName	boolean	Prüft, ob Vornamen vorkommen.[12]	(0)
containsDash	boolean	Vorkommen von Viertelgeviertstrichen. [8]	(1)
isComName	boolean	Prüft auf kommerzielle Bezeichner (Corp., Inc.)	(0)

Tabelle 7: Confusion Matrix

361	28	22	2	0	PERSON
29	549	10	0	0	GPE_NORP
71	45	736	6	1	ORG
0	2	1	591	7	DATE
0	2	0	2	525	PERCENT_CARDINAL_MONEY

Tabelle 8: Final Evaluation

Featureset		Accuracy
Baseline	unbalanced	0.7867
'unigram'	balanced	0.8408
Optimales Featureset	unbalanced	0.8728
'pos', 'is_all_caps', 'is_in_wiki', 'is_np', 'contains_digit', 'unigram', 'context'	balanced	0.9237

C. Abbildungen

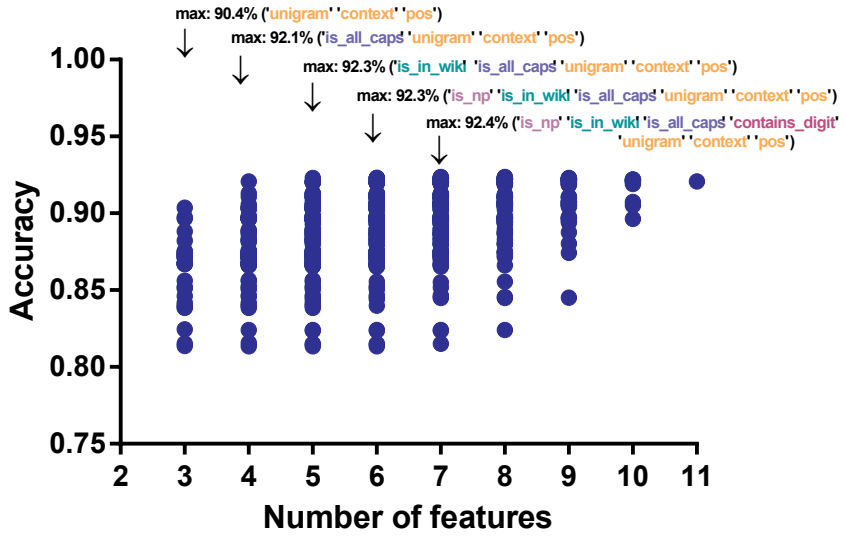


Abbildung 1: Accuracywerte der einzelnen Subsets. Jeder blaue Punkt stellt ein Subset dar.

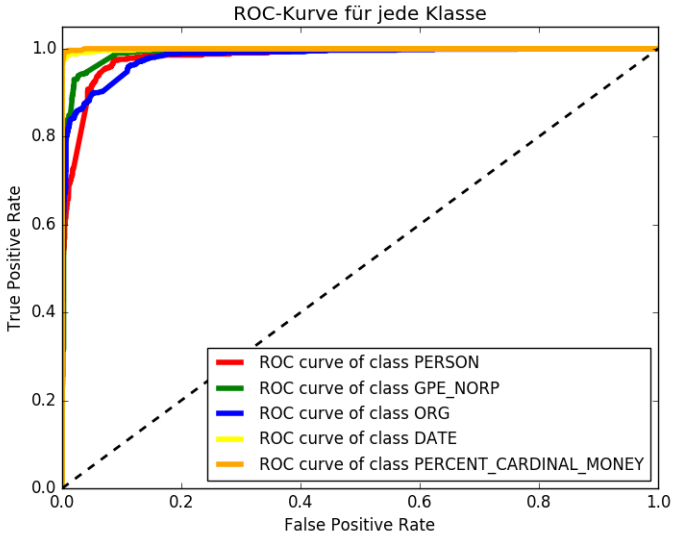


Abbildung 2: ROC Kurve für das optimale Featureset mit sieben Features.