Automatische Erkennung von Figuren in deutschsprachigen Romanen

Jannidis, Fotis
fotis.jannidis@uni-wuerzburg.de, Universität Würzburg

Krug, Markus
markus.krug@uni-wuerzburg.de, Universität Würzburg

Reger, Isabella
isabella.reger@uni-wuerzburg.de, Universität Würzburg

Toepfer, Martin
toepfer@informatik.uni-wuerzburg.de, Universität Würzburg

Weimer, Lukas
lukas.weimer@stud-mail.uni-wuerzburg.de, Universität Würzburg

Puppe, Frank
frank.puppe@uni-wuerzburg.de, Universität Würzburg

Eine wichtige Grundlage für die quantitative Analyse von Erzähltexten, etwa eine Netzwerkanalyse der Figurenkonstellation, ist die automatische Erkennung von Referenzen auf Figuren in Erzähltexten, ein Sonderfall des generischen NLP-Problems der Named Entity Recognition [Sharnagat 2014]. Mit dem Stanford Parser [Finkel 2005] unter Verwendung eines Modells für deutsche Sprache [Faruqui and Pado 2010] liegen inzwischen auch freie Werkzeuge für Texte in deutscher Sprache vor. Allerdings ist die Erkennungsrate des Modells, das an einem Korpus von Zeitungstexten trainiert wurde, für literarische Texte nur eingeschränkt brauchbar (Abb. 1). Eine Auswertung anhand unseres Testkorpus (265 000 Tokens) hat einen F1-Score von nur 31% ergeben, was vor allem am sehr niedrigen Recall lag. Dieser Befund deckt sich mit vergleichbaren Erfahrungen aus der Computerlinguistik: Viele NLP-Werkzeuge müssen erst für einen neuen Anwendungsbereich angepasst werden, um brauchbare Resultate zu erbringen. Im Fall des Romankorpus führt die Einbeziehung von Appellativen in die Named Entity-Definition und deren häufige Verwendung in Romantexten zu dem schlechten Ergebnis. Da die Figurenreferenzen allerdings für fast alle nachfolgenden Verarbeitungsschritte eine hohe Relevanz haben, sind wir *nicht* den Weg einer automatischen Domänenadaption [Qi Li 2012] gegangen, sondern haben ein umfangreiches Trainingskorpus aufgebaut, um auf diese Weise möglichst hohe Erkennungsraten zu erhalten. Im Folgenden berichten wir über unser Vorgehen, diese Aufgabe möglichst effizient zu gestalten. Zusammenfassend können wir feststellen, dass wir die Erstellung des notwendigen Trainingskorpus durch ein Werkzeug erheblich beschleunigen konnten, das den Annotatoren bereits gute Vorschläge machte. Außerdem konnten die Resultate des verwendeten Lernverfahrens dadurch deutlich verbessert werden, dass über die üblichen Standardfeatures hinaus word2vec-Informationen (s.u.) als Feature verwendet wurden.

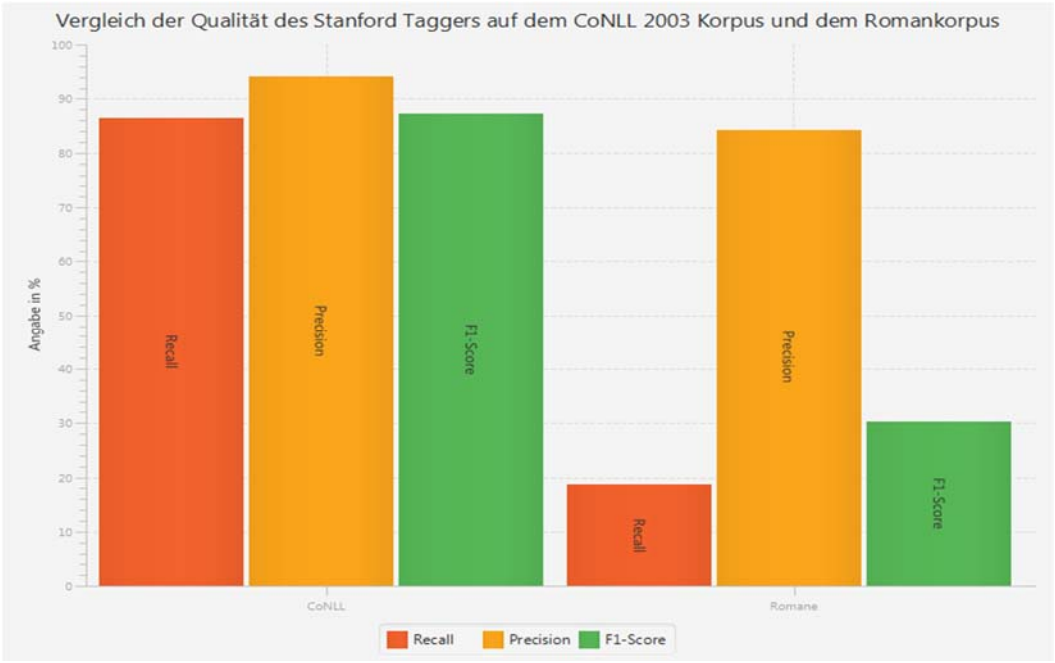

Abb. 1: Ergebnisse des Stanford-Parsers mit deutschem Modell (Faruqui and Pado 2010) angewandt auf ein Zeitungskorpus (CoNLL 2003) und ein Korpus deutschsprachiger Romane.

Material und Methoden

Als annotierte Trainings- und Testdaten dienten das Zeitungskorpus der CoNLL 2003 [Sang 2003] (ca. 220 000 Tokens) und ein von uns aufbereitetes Romankorpus mit je 130 zusammenhängenden Sätzen aus 50 Romanen mit 140 000 Tokens für das erste Experiment, und 85 Romanen mit 265 000 Tokens für das zweite Experiment. Die Annotation geschah mittels einem eigens für diesen Zweck entwickelten Werkzeug, das über eine komfortable grafische Benutzeroberfläche dem Annotator die mit einfachen Regeln ermittelten Vorschläge zur Bearbeitung anbietet, wodurch sich die Annotation erheblich beschleunigen ließ (die vorher direkt in XML-Dateien und dann in einem Annotationswerkzeug durchgeführt wurde, das nicht spezifisch für die Aufgabe angepasst wurde). Notiert wurden folgende Eigenschaften:

a) Handelt es sich um einen wirklichen Namen, z.B. „Effi Briest“, oder um einen Appellativ, z.B. der „Lehrer“.
b) Handelt es sich um eine einzelne Person oder um eine Personengruppe bzw. um mehr als eine Person, z.B. die „Gäste“.
c) Koreferenz per Identität (ID), d.h. alle Referenzen auf die gleiche Figur erhalten die gleiche grafisch angezeigte ID.

Für die Anwendung unüberwachter Lernverfahren verwendeten wir Texte aus der FAZ (ca. 15 Millionen Tokens) und unser Erweiterungskorpus deutschsprachiger Romane (ca. 60 Millionen Tokens), beide Textsammlungen nicht annotiert.
In der ersten Serie von Experimenten wurde die Frage untersucht, mit welchen Features das maschinelle Lernverfahren Conditional Random Fields (CRF), das auch im Stanford Parser eingesetzt wird, die besten Ergebnisse erbringt. Folgende sechs Features, die vom Stanford-Tagger [Finkel 2005] verwendet werden, wurden als Basis betrachtet:

1) Current Word: das Wort an Position i
2) Previous Word: das Wort an Position i-1
3) Next Word: das Wort an Position i+1
4) Word Shape: für Groß/Kleinschreibung oder Zahlen
5) Part-Of-Speech Tags (POS-Tags) an den Positionen i, i-1 und i+1, die mit Hilfe des TreeTaggers [Schmid 1995] bestimmt wurden.
6) Präfix bzw. Suffix, das aus den ersten oder letzten 2 Zeichen besteht.

Außerdem getestete Features:

7) Gazeteers: Listen bestehend aus rd. 5200 männlichen, 3400 weiblichen Vornamen, 160 Adelstiteln, Anreden und 8700 Berufen.
8) Semantische Felder, je nach Wortart 15-23, auf der Grundlage von GermaNet
9) Satzsubjekt ermittelt mit dem Mate-Dependecy Parsers [Bohnet 2010].
10) Compound-Words: alle von SFST [Fitschen 2004] erkannten Teilworte des Eingabewortes inkl. Prä- und Suffixe.
11) Head-Lemma: Grundform des zum Subjekt gehörenden Verbes.
12) LDA-Cluster: Es wird die Zugehörigkeit aller Nicht-Stop-Wörter zu dem wahrscheinlichsten von 250 Clustern mit der Latent-Dirichlet-Allocation (LDA) [Blei 2003] in Anlehnung an [Chrupala 2011] auf der Basis der oben erwähnten nicht annotierten Korpora mit 15 Millionen bzw. 60 Millionen Token ermittelt. Das LDA wurde mit dem Framework MALLET [MALLET 2002] implementiert.
13) Word2Vec-Cluster: Es wird ebenfalls die Zugehörigkeit aller Nicht-Stop-Wörter zu einem semantischen Cluster ermittelt. Dabei wurde eine effiziente Implementierung des “Continuous Bag-of-Words” Modells nach [Mikolov 2013] genutzt und die resultierenden Vektoren mit einem k-means Verfahren geclustert.

Ergebnisse

Zum Testen der gelernten CRFs wurde eine 10-fache Kreuzvalidierung auf der Trainingsmenge des Romankorpus (120.000 Tokens) durchgeführt. Die Baseline mit den Features 1-6 erbrachte einen F1-Score von 86,66%. Die Kombination der besten Features (letzte Zeile) erzielte einen F1-Score von 89,98, d.h. eine Steigerung um 3,32 Prozentpunkte. Der mit Abstand größte Anteil an dieser Steigerung ging auf das semantische Feature "Word2Vec-Cluster" zurück. Dagegen erbrachte das semantische Clustering mit LDAs einen eher negativen Effekt. In [Tkachenko 2012] wird der gleiche Effekt berichtet und die Vermutung geäußert, dass die LDA-Cluster redundant zu den POS-Tagging-Features sind. Beim Trainingskorpus mit den Zeitungsartikeln war die Baseline mit 87,9% etwas besser, aber die Steigerung durch Hinzunahme des Word2Vec-Cluster mit 1,6 Prozentpunkten (auf 89,5%) etwas schlechter.

Verfahren	**Precision in %**	**Recall in %**	**F1-Score in %**	**Unterschied zur Baseline (F1-Score) in %**
Baseline (Features 1-6)	95.12	79.60	86.66	+0
Baseline + (Feature 7)	95.73	79.28	86.70	+0.04
Baseline +(8)	94.53	81.74	87.65	+0.99
Baseline + (9)	94.96	79.74	86.67	+0.01
Baseline + (10)	95.07	81.00	87.45	+0.79
Baseline + (11)	95.03	79.63	86.63	-0.03
Baseline + (12)	**96.47**	77.83	86.13	-0.53
Baseline + (13)	94.97	85.28	89.84	+3.18
Baseline + (7),(8),(10),(13)	94.86	**85.60**	**89.98**	**+3.32**

Tab. 1. Einfluss verschiedener Features auf die NER mit CRFs; Trainingsset ca. 120 000 Tokens.

Wir haben beim Feature 13 "Word2Vec-Cluster" untersucht, welchen Einfluss die Anzahl der vorgegeben Cluster im k-means Verfahren zwischen 100 und 1000 auf die Qualität der NER hat. Dabei stellte sich heraus, dass bei einer Clusteranzahl ab 250 (relativ konstant bis 1000) das beste Ergebnis erzielt wird, so dass in weiteren Experimenten die Clusteranzahl von 250 gewählt wurde.

In unserem zweiten Experiment beschäftigten wir uns mit den Fragen, wie groß unser annotiertes Korpus für das Training eines praktisch nutzbaren NER-Modells sein muss, bzw. ab welcher Größe eine Erweiterung des Trainingsmaterials keine nennenswerte Verbesserung der Erkennungsleistung mehr bringt. Als zweiten Aspekt gilt es das für unseren Task beste Lernverfahren zu ermitteln. Für diesen Zweck haben wir die Erkennungsgenauigkeit mit immer größeren Mengen von Trainingsdaten gemessen: Für beide Domänen wurde zunächst nur eine Trainingsmenge von 30 000 Tokens genutzt, die dann in Schritten von 10 000 Tokens auf die Maximalzahl von 230 000 Tokens bei den Romanen bzw. 170 000 Tokens bei den Zeitungsartikeln gesteigert wurde. Als Features haben wir die jeweils beste Feature-Menge für das CRF verwendet. Neben dem CRF-Klassifikator wurden auch Maximum-Entropy, Naive Bayes und Decision-Trees mit der gleichen Menge an Features getestet. Abb. 2 zeigt, dass die beiden besten, von uns getesteten Klassifikationsverfahren MaxEnt, sowie CRFs sind. Auf dem Zeitungskorpus sind CRFs ca. 3-5% besser als MaxEnt, die Evaluation auf dem Romankorpus zeigt genau entgegengesetzte Ergebnisse. Eine Ausnutzung der Zustandsübergangsinformation, die CRFs zusätzlich zu MaxEnt nutzen, scheint im Fall der Romane keine nützlichen Informationen zu liefern, sondern das Ergebnis zu verschlechtern. Dies könnte in einer deutlich höheren durchschnittlichen Satzlänge (24,2 Tokens vs. 16,3 Tokens) in unserer Domäne begründet liegen. Ab einer Trainingsmenge von etwa 150 000

Tokens zeigt sich keine signifikante Verbesserung der Ergebnisse mehr. Wenn statt dieser 10-Fold Cross-Validation eine Leave-One-Out-Evaluation verwendet wird, bei der der zu testende Roman nicht in der Trainingsmenge enthalten ist, verringert sich der durchschnittliche F1-Score um ca. fünf Prozentpunkte von 88% auf 83.4%. Entgegen unserer Erwartung führte die Hinzunahme von 35 Romanen in dem Trainingskorpus zu keiner Verbesserung der Erkennungsrate, sondern sogar zu einer Verschlechterung um ca. 2%. Eine genauere Analyse zeigte, dass unter diesen zufällig ausgewählten Romanen auch solche mit Dialekten und anderen Besonderheiten waren, was die Verschlechterung erklären könnte.[1]

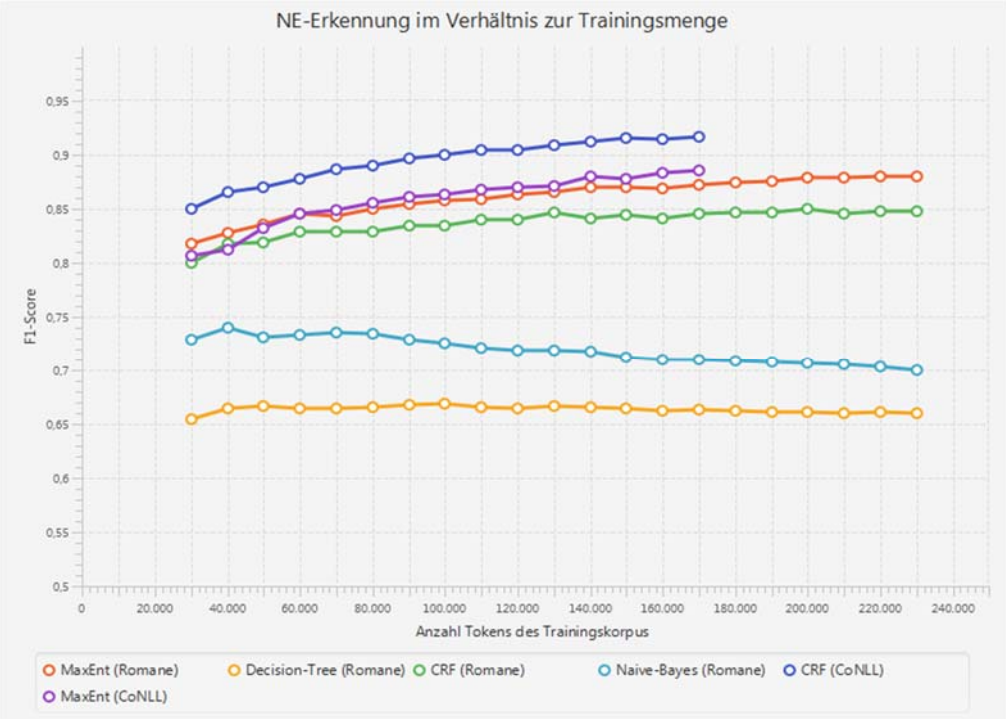

Abb. 2. Einfluss verschiedener Größen von Trainingsdaten von 30 000 bis 230 000 bzw. 170 000 Tokens auf den F1-Wert der NER mit CRFs in zwei verschiedenen Domänen (Romane und Zeitungsartikel) und verschiedenen maschinellen Lernverfahren

Ausblick

Es gibt eine Reihe von weiteren Optimierungsverfahren, die im Anschluss an die berichteten Experimente exploriert werden sollen. Wir haben bisher nur Lernverfahren für die NER in Romanen auf der Basis annotierter Textkorpora untersucht. Wir versprechen uns sowohl beim Erstellen eines Goldstandards, als auch bei dem erzielbaren F1-Wert der NER Verbesserungen durch die Integration von komplexeren regelbasierten Verfahren [Klügl et al. 2014] zur Information Extraction. Außerdem soll der Vermutung nachgegangen werden, dass die Erkennungsleistung durch Verwendung von Strategien der Domänenanpassung noch verbessert werden kann, wenn diese auf das vorhandene umfangreiche Korpus mit nicht-annotierten Daten angewandt werden [Qi Li 2012]. Außerdem sollen Alternativen zum

1 Unsere Implementierung des MaxEnt-Modells ist unter https://github.com/MarkusKrug/NERDetection/ zu finden. Sie ist so aufbereitet, dass sie mit dem DkPro-Framework kompatibel ist. Die Eingliederung dort soll demnächst folgen.

word2vec-Feature erprobt werden, die in NLP-Tasks gleichwertige Ergebnisse erbracht haben [Pennington 2014].

Literatur

Blei, D., Ng, A. and Jordan, M. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*. 3, 993–1022.

Bohnet, B. (2010). *Very High Accuracy and Fast Dependency Parsing is not a Contradiction*. The 23rd Int. Conference on Computational Linguistics (COLING 2010), Beijing, China.

Chrupala, G. (2011). Efficient induction of probabilistic word classes with LDA. *Proceedings of 5th International Joint Conference on Natural Language Processing*, 363-372.

Faruqui, M. and Pado. S. (2010) Training and Evaluating a German Named Entity Recognizer with Semantic Generalization. Proceedings of Konvens 2010, Saarbrücken, Germany.

Finkel, F., Grenager, T. and Manning, C. (2005). Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pp. 363-370. http://nlp.stanford.edu/~manning/papers/gibbscrf3.pdf

Fitschen, A., Schmid, H. and Heid, U. (2004) SMOR: A German computational morphology covering derivation, composition, and inflection. *Proceedings of the IVth International Conference on Language Resources and Evaluation* (LREC 2004), 1263–1266.

Klügl, P., Toepfer, M., Beck, P.D., Fette, G., Puppe, F. (2014) UIMA Ruta: Rapid development of rule-based information extraction applications. *Natural Language Engineering* First View, 1–40 (2014). DOI 10.1017/S1351324914000114.

McCallum, A. MALLET: A Machine Learning for Language Toolkit. 2002.

Mikolov, T., Chen, K., Corrado, G. and Dean, J. 2013. Efficient Estimation of Word Representations in Vector Space. In Proceedings of Workshop at ICLR, 2013.

Nadeau, D. and Sekine, S. (2007) A survey of named entity recognition and classification. Lingvisticae Investigationes 30 (1), 3-26

Pennington, J, Socher, R. and Manning, C. (2014) Glove: Global Vectors for Word Representation. *Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*.

Qi Li (2012): Literature Survey. *Domain Adaption Algorithms for Natural Language Processing*. nlp.cs.rpi.edu/paper/qisurvey.pdf

Sang E. and Meulder, F. (2003) Introduction to the CoNLL-2003 shared task: language-independent named entity recognition. *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 (*4), 142-147.

Schmid, H. (1995) Improvements in Part-of-Speech Tagging with an Application to German. *Proceedings of the ACL SIGDAT-Workshop*. Dublin, Ireland.

Sharnagat, R. (2014) *Named Entity Recognition: A Literature Survey*. Surveys of the Center for Indian Language Technology. http://www.cfilt.iitb.ac.in/resources/surveys/rahul-ner-survey.pdf

Tkachenko, M. and Simanovsky, A. (2012) Named entity recognition: Exploring features. *Proceedings of KONVENS 2012*, 118-127.