

WALU — Eine Annotations- und Lern-Umgebung für semantisches Markup in Texten

Andreas Wagner und Marc Rössler

Universität Duisburg-Essen

Computerlinguistik

Lotharstraße 65

D-47048 Duisburg

{andreas.wagner,marc.roessler}@uni-due.de

Abstract

WALU (WIKINGER Annotations- und Lern-Umgebung) ist eine Software zur Annotation und (semi-)automatischen Erkennung von Eigennamen sowie Instanzen anderer semantischer Kategorien in Texten. Ziel der Entwicklung von WALU ist die Realisierung eines komfortablen Werkzeugs, das von Experten unterschiedlichster Domänen ohne computerlinguistische und informatische Vorkenntnisse eingesetzt werden kann. Dies unterscheidet WALU von existierenden Annotations- und Lern-Umgebungen im Bereich Informationsextraktion, die auf andere Tasks zugeschnitten oder multifunktional ausgelegt sind, was einen erheblichen Konfigurationsaufwand erfordert. WALU ist Teil der kollaborativen Wissens-Infrastruktur, die im eScience-Projekt WIKINGER entwickelt wird. Darüber hinaus ist es als Stand-Alone-Tool einsetzbar. Dieser Beitrag spezifiziert die Design-Prinzipien und aktuell implementierten Funktionalitäten von WALU, gibt einen Überblick über die Pilotdomäne und skizziert laufende Experimente zum semantischen Markup mit maschinellen Lernverfahren.

1 Einleitung

WALU (WIKINGER Annotations- und Lern-Umgebung) ist eine Software zur Annotation und (semi-)automatischen Erkennung von Eigennamen sowie Instanzen anderer semantischer Kategorien in Texten. WALU wird im Rahmen des BMBF-Projekts WIKINGER entwickelt.

Dieser Beitrag gliedert sich wie folgt: Abschnitt 2 skizziert das Projekt WIKINGER. Abschnitt 3 befasst sich Besonderheiten der WIKINGER-Pilotdomäne. Abschnitt 4 diskutiert die sich aus diesem Kontext ergebenden Anforderungen an Eigennamenerkennungsverfahren im Allgemeinen und an das Design des Werkzeugs im Besonderen. Abschnitt 5 beschreibt den aktuellen Entwicklungsstand von WALU und zeigt weitere Perspektiven auf. Abschnitt 6 enthält einen generellen Vergleich von WALU mit anderen einschlägigen Annotations-Tools. Abschnitt 7 schließt mit Zusammenfassung und Ausblick.

2 WIKINGER

Das Projekt WIKINGER (WIKI Next Generation Enhanced Repository), vgl. [Hoeppner *et al.*, 2006; Bröcker *et al.*, 2007], wird im Rahmen der eScience-Initiative des BMBF gefördert (Förderungs-Zeitraum: 10/05–09/08). Ziel des

Projekts ist die Entwicklung einer intelligenten Infrastruktur (Plattform) für einen effizienten Austausch wissenschaftlicher Ergebnisse. Diese Infrastruktur wird als semantisches Wiki organisiert, d.h. als Repository von Dokumenten und Informationen, die mit Techniken des Semantic Web kodiert und vernetzt sind. Dieses Repository wird von den Mitgliedern einer Forschungscommunity online erstellt und modifiziert. Dies ermöglicht den verteilten Auf- und Ausbau eines Informationsnetzes, welches Fachdomänen-Wissen effizient zugänglich macht. In WIKINGER wird zunächst die Pilotdomäne *Katholische Zeitgeschichte* behandelt. Im Sinne der Nachhaltigkeit der entwickelten Infrastruktur ist ihre Wiederverwendbarkeit, d.h. ihre Portierbarkeit auf andere wissenschaftliche Domänen und industrielle Anwendungen, jedoch ein wichtiges desiderat.

Ein entscheidendes Projektziel ist die Entwicklung von Verfahren zum (semi-)automatischen Aufbau eines solchen semantischen Wissensnetzes aus einschlägigen Dokumenten der jeweiligen Domäne. Hierfür werden in einem ersten Schritt Named Entities (z.B. Personen, Orte, Organisationen) sowie die Vorkommen anderer für die Domäne wesentlicher semantischer Kategorien (z.B. bedeutende Ereignisse) in den Texten erkannt und klassifiziert (d.h. semantisch getaggt). In einem nächsten Schritt werden diese Entitäten mit semi-automatischen Methoden zu semantischen Netzen verknüpft, indem ihre Kookkurenzen und die zugehörigen Kontexte in den Texten analysiert werden. Sowohl die Ergebnisse des semantischen Taggings als auch die daraus resultierenden semantischen Netze unterliegen dem Feedback der Community-Mitglieder, die über das Wiki Zugriff darauf haben und ggf. Korrekturen und Ergänzungen vornehmen können.¹ Das Feedback der Domänenexperten wird zur Verfeinerung der automatischen Extraktionsverfahren eingesetzt.

An WIKINGER sind das Fraunhofer Institut für Intelligente Analyse- und Informationssysteme (IAIS), Sankt Augustin, die Computerlinguistik der Universität Duisburg-Essen, Duisburg, sowie die Kommission für Zeitgeschichte (KfZG), Bonn, beteiligt. Die Duisburger Computerlinguistik befasst sich mit der Eigennamen-Erkennung (Named Entity Recognition, NER) bzw. dem semantischen Markup. Ein Kernstück der diesbezüglichen Aktivitäten ist die Entwicklung des Tools WALU.

3 Besonderheiten der Pilotdomäne

Die Fachexperten der KfZG stellen einen Großteil der zu erschließenden Wissensquellen bereit: Die von der

¹Außerdem besteht die Möglichkeit, neue Daten in das Wiki aufzunehmen.

KfZG herausgegebene „Blaue Reihe“ umfasst mehr als 150 Bände zur zeitgeschichtlichen Katholizismusforschung. Alle diese Bände wurden im Rahmen des Projekts digitalisiert. Das domänenspezifische Inventar semantischer Kategorien wurde ausgehend von einem Standard-Inventar für NER [Chinchor, 1998] erarbeitet. Bei der manuellen Annotation exemplarischer Werke kristallisierte sich ein adäquates Inventar heraus, das wie folgt festgelegt wurde:

- GKPE (Geographische Kirchliche/Politische Einheit)
- Ort
- Einrichtung
- Organisation
- Person
- Abgetrennter Namensbestandteil (z.B. in Registern)
- Namenszusatz
- Rolle/Funktion
- Biographisches Ereignis
- Bedeutendes Ereignis
- Datum/Zeit

Da die Zeithistoriker der KfZG im Rahmen der WIKINGER-Aktivitäten insbesondere an biographisch-bibliographischen Informationen über katholische Persönlichkeiten im 19. und 20. Jahrhundert interessiert sind, konzentriert sich die Annotation zunächst auf die sog. Biogramme, d.h. Kurzbiographien, die i.d.R. in Fußnoten präsentiert werden.

Hervorzuheben ist, dass eingebettete Annotationen ausdrücklich vorgesehen sind, wie folgendes Beispiel illustriert:

```
<Rolle>Regens des  
<ORG>Regionalseminars  
<GKPE>Erfurt</GKPE></ORG></Rolle>
```

Hier wird ein Teil einer Instanz der Kategorie Rolle als Organisation markiert, davon ist wiederum ein Teil als GKPE annotiert.

4 Anforderungen an Annotations-Strategien

Aus dem Ziel der Wiederverwendbarkeit der in WIKINGER entwickelten Infrastruktur, d.h. ihrer Adaptierbarkeit an neue Domänen, ergeben sich wesentliche Anforderungen an die eingesetzten Strategien zur semantischen Annotation. Dies betrifft sowohl die prinzipiellen Verfahren als auch die konkrete Ausgestaltung der verwendeten Werkzeuge.

4.1 Annotationsverfahren

Die Fülle der zu erschließenden Daten, die für eine bestimmte Domäne gewöhnlich vorliegen, macht den Einsatz automatischer und semi-automatischer Annotationsverfahren unerlässlich. Für NER existieren sowohl regelbasierte Verfahren als auch statistische (maschinelle) Lernverfahren, vgl. z.B. entsprechende Beiträge zur Message Understanding Conference (MUC) 7 [Chinchor, 1998]. Im Hinblick auf das Desiderat der Adaptierbarkeit an beliebige Domänen haben maschinelle Lernverfahren signifikante Vorteile. Bei der Anpassung von regelbasierten Verfahren an eine neue Domäne und/oder Sprache ist es i.d.R. erforderlich, den verwendeten Satz von Erkennungsregeln substanziell zu überarbeiten und schlimmstenfalls neu aufzusetzen. Da ein adäquates Regelwerk für praktische Anwendungen sehr komplex ist (sowohl hinsichtlich der Zahl der

Regeln als auch hinsichtlich ihrer Interaktion), setzt diese Aufgabe profunde Kenntnisse über die neue Domäne einerseits und die Wirkungsweise und das Zusammenspiel der Erkennungsregeln andererseits voraus. Dies erfordert eine zeitintensive und aufwändige Zusammenarbeit zwischen den jeweiligen Domänenexperten und Computerlinguisten. Dagegen ist für die Anpassung von maschinellen / statistischen Lernverfahren im Wesentlichen die Erstellung neuer Trainingsdaten erforderlich. Diese können durch die Annotierung einschlägiger Texte durch die Domänenexperten gewonnen werden; der hierfür benötigte fachübergreifende Kooperationsaufwand mit Computerlinguisten ist vergleichsweise gering.

Dieser beispielbasierte Ansatz – die für das Lernverfahren benötigten Informationen werden durch Beispiele übermittelt, nicht durch explizite Regeln und Definitionen – erhöht die Domänen-Adaptivität auch bei dem Schritt, der dem eigentlichen Lernen vorausgeht: der Definition der zu lernenden semantischen Kategorien. Durch die manuelle Annotation verschiedener Instanzen einer Kategorie grenzen die Domänenexperten die Verwendung dieser Kategorie in ihrer Domäne ein (z.B.: Gilt eine Kirchengemeinde als geographische Entität?) und liefern so eine implizite Definition. Dies erspart die aufwändige Erarbeitung einer expliziten Definition. Ebenso ermöglicht das empirische Arbeiten mit den relevanten Texten im Zuge des Annotationsprozesses eine adäquatere Festlegung des Kategorien-Inventars: Neue Kategorien können eingeführt, praktisch irrelevante Kategorien eliminiert und gleichartige Kategorien zusammengefasst werden, wenn während der Annotation ein entsprechender Bedarf festgestellt wird.

Im Sinne der Adaptivität sind also lernbasierte Ansätze zu bevorzugen. Hierbei ist es entscheidend, einerseits domänen-unabhängige Merkmale zu verwenden (z.B. Wortformen und -affixe) sowie andererseits einfache Anbindungsmöglichkeiten domänen-spezifischer Ressourcen (z.B. Listen, s.u.) zu ermöglichen. Besonders interessant sind Verfahren, die versuchen, den initialen Annotationsaufwand möglichst zu minimieren, z.B. durch Active Learning. Jedoch ist in begrenztem Umfang auch der Einsatz regelbasierter Verfahren sinnvoll, wenn sie mit geringem Aufwand gute Erkennungsleistungen erzielen und/oder domänen-übergreifend einsetzbar sind. Beispielsweise bieten sich für die Erkennung von Datums- und Zeitangaben reguläre Ausdrücke an; diese sind vergleichsweise einfach definierbar und domänen- (wenn auch nicht sprach-) unabhängig. Unter das regelbasierte Paradigma fällt insbesondere die Verwendung von allgemeinen oder domänenspezifischen Listen von Kategorie-Instanzen (z.B. Personen oder Bistümern). Diese können entweder aus den annotierten Texten oder aus externen Quellen gewonnen werden. Wir halten die Anwendung von Listen für eine zufriedenstellende Erkennungsrate sowie zur Unterstützung bei der manuellen Annotation für unverzichtbar.

4.2 Werkzeug

Damit die Anpassung des Systems, d.h. die Erstellung von Trainingsdaten und das Trainieren und Anwenden von Lernverfahren, weitestgehend selbstständig von den jeweiligen Domänenexperten durchgeführt werden kann, ist es unabdingbar, hierfür ein geeignetes Werkzeug zur Verfügung zu stellen. Entscheidend ist, dass dieses Tool von Experten unterschiedlichster Domänen ohne computerlinguistische und informatische Vorkenntnisse intuitiv bedienbar ist. Es sollte eine komfortable Oberfläche für

den gesamten Annotations- und Lernzyklus bereit stellen. Dazu gehört der einfache Zugriff auf das Repository der zu annotierenden Texte, benutzerfreundliche Mechanismen zur manuellen Annotation, eine flexible Verwaltung des Kategorien-Inventars, vielfältige, konfigurierbare Anzeigemöglichkeiten, die nahtlose Integration listenbasierter Annotation sowie die Anbindung von automatischen Annotationsverfahren. Wie in Abschnitt 5 erläutert, ist WALU auf diese Anforderungen zugeschnitten. Im Sinne der Wiederverwendbarkeit ist WALU in Java implementiert und damit Betriebssystem-unabhängig einsetzbar.

5 WALU – aktueller Entwicklungsstand

Die Prioritäten bei der Entwicklung von WALU spiegeln die Erfordernisse in WIKINGER wider. In der bisherigen Phase stand die manuelle Annotation durch die Domänenexperten im Vordergrund. Mit den so gewonnenen Daten werden Listen erstellt und maschinelle Lernverfahren trainiert, die in den kommenden Phasen sukzessive für die automatische Annotation eingesetzt werden. Dementsprechend waren bisher für WALU vorrangig komfortable Mechanismen zur manuellen Annotation, einschließlich der Anbindung listenbasierter Annotatoren, zu realisieren. Jedoch sind bereits erste Experimente mit maschinellen Lernverfahren, die an WALU angebunden wurden, durchgeführt worden.

5.1 Manuelle Annotation

WALU bietet eine komfortable Annotationsoberfläche (vgl. Abbildung 1). Ein Default-Inventar semantischer Kategorien ist vorgegeben²; darüber hinaus können, auch text-spezifisch, neue Kategorien definiert werden. Eine Annotation erfolgt durch Markieren einer Instanz im Text und der Auswahl der entsprechenden Kategorie (über Kontextmenü, Buttons oder Shortcuts). Die Annotationen werden im Text farblich markiert (jeder Kategorie entspricht eine bestimmte Farbe) sowie in einer separaten Liste neben dem Textfeld angezeigt. Jede Kategorie lässt sich im Text und/oder in der Liste ein- und ausblenden. Eine Annotation kann mit einem Kommentar versehen werden. Neben der Annotation ermöglicht WALU die manuelle Editierung von Dokumenten.

5.2 Einfache (semi-)automatische Annotationsmechanismen

Aus den annotierten Texten extrahiert WALU Listen von Kategorie-Instanzen, die für die automatische Annotation weiterer Vorkommen dieser Instanzen eingesetzt werden können. Eine automatische Annotation wird zunächst als "unchecked" erfasst und dargestellt; eine manuelle Bestätigung führt zum Status "checked", der zu einer manuellen Annotation äquivalent ist. Die Listen werden bei der Durchsicht der Annotationen interaktiv angepasst, indem beim Löschen einer falschen Annotation auch der entsprechende Listen-Eintrag entfernt werden kann (jedoch nicht muss). So wird die Qualität der Listen sukzessive erhöht.

Ein weiterer Mechanismus zur automatischen Annotation sind reguläre Ausdrücke. Z.Zt. ist ein regulärer Annotator für Datums- und Jahresangaben integriert. Weitere vordefinierte und konfigurierbare reguläre Annotatoren werden hinzukommen.

²Es besteht die Möglichkeit, unabhängige Projekte mit jeweils eigenem Default-Inventar zu definieren.

5.3 Annotation durch maschinelle Lernverfahren

Momentan führen wir vielfältige Experimente zum Einsatz maschineller Lernverfahren in der Pilotdomäne durch. Zum jetzigen Zeitpunkt sind Implementationen zweier einschlägiger Verfahren in WALU integriert und können auf WIKINGER-Daten angewandt werden: MaxEnt (openNLP³) und SVM (SVMstruct⁴). Unser Ziel ist es, eine Reihe von Lernverfahren einzubinden, die unabhängig oder in Kombination anwendbar sein sollen, um maximale Performanz zu erreichen. Diese Methoden müssen so eingesetzt werden, dass sie die Akquisition eingebetteter Annotationen erlauben. Dies läuft letztlich darauf hinaus, den zu annotierenden Instanzen (hier: Token) multiple Klassen zuordnen zu können (z.B. erhält "Erfurt" im Beispiel in Abschnitt 3 die Klassen Rolle, Organisation und GKPE). "Klassische" ML-Verfahren weisen jeder Instanz nur eine Klasse zu. Aus diesem Grund wenden wir mehrere Klassifizierer an, die jeweils unterschiedliche semantische Kategorien zuweisen, und unifizieren die Ergebnisse. Bei ML-Verfahren, die auf binäre Klassifizierer beschränkt sind (z.B. SVM), wird für jede Kategorie ein separater Klassifizierer benötigt. Verfahren ohne diese Einschränkung (z.B. MaxEnt) ermöglichen flexiblere Konfigurationen. Unsere bisherigen Experimente mit MaxEnt-Modellen haben ergeben, dass mit einer Kombination von Klassifizierern, die jeweils eine unterschiedliche Kategorie "ignorieren", d.h. die, außer der jeweils ignorierten Klasse, alle Kategorien zuweisen, insgesamt bessere Ergebnisse erzielt werden als mit einer Kombination binärer Klassifizierer. Auf Token-Ebene erzielten diese vorläufigen Experimente F-Measures von bis zu 84,6% für Personen, 87,1% für Organisationen, 94,8% für GKPEs und 92,8% für Rollen.

Ein zentrales Kriterium zur Beurteilung eines NER-Systems ist die Anpassungsfähigkeit an eine neue Aufgabe, d.h. an eine neue Domäne und/oder eine neue Sprache. Um diese Eigenschaft zu überprüfen, wurde WALU in einem Experiment für die Erkennung von Named Entities in italienischen Zeitungen trainiert. Dies geschah im Rahmen des NER-Shared Task von EVALITA 2007 (Evaluation of NLP Tools for Italian, [Rössler *et al.*, 2007]). Im Vergleich mit den insgesamt sechs partizipierenden Systemen erreichte WALU die zweitbesten Erkennungsergebnisse, und das, obwohl keiner der Systementwickler Italienisch spricht.

5.4 Qualitätskontrolle

WALU ist mit einer einfachen Zeichenketten-basierten Suchfunktion ausgestattet. Darüber hinaus ist ein spezieller Such- und Sortier-Modus für Annotationen implementiert. In diesem Modus werden die annotierten Entitäten (Types) mit den entsprechenden Kategorien in einer Liste angezeigt (sortiert nach Häufigkeit oder Alphabet). Klickt man auf ein Entitäts-Kategorie-Paar, werden die entsprechenden Vorkommen mit ihren Kontexten im KWIC-Format angezeigt und können direkt bearbeitet werden. So können die einzelnen Annotationen bestätigt, gelöscht oder die Kategorie geändert werden. Dies ermöglicht eine effiziente Kontrolle und Korrektur automatischer Annotationen, insbesondere im Hinblick auf Ambiguitäten. Werden beispielsweise durch listenbasierte Annotation alle Vorkommen von „Singen“ als Ort markiert, können diese Annotationen im Sortier-Modus zusammen aufgelistet werden

³<http://maxent.sourceforge.net/>

⁴http://svmlight.joachims.org/svm_struct.html

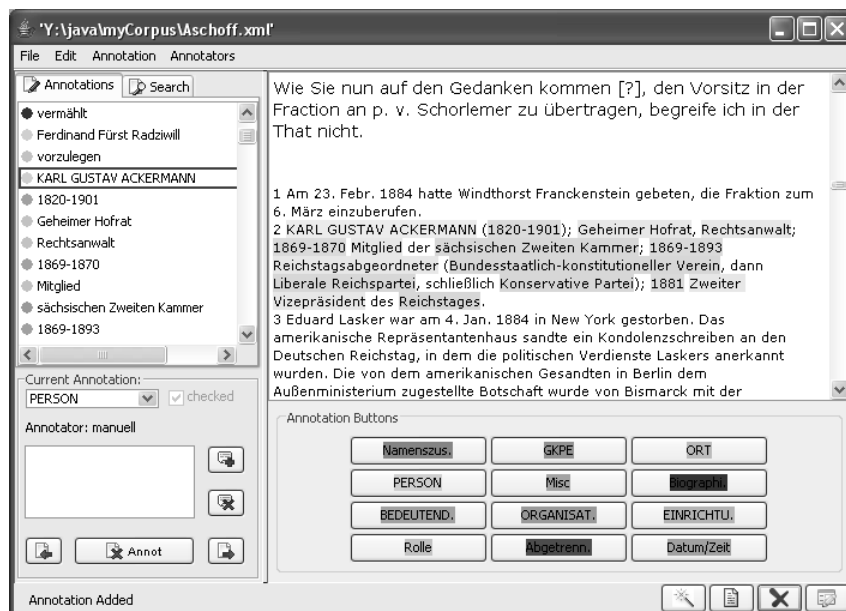


Abbildung 1: WALU-Annotationsoberfläche

und eventuelle Fehler (z.B. Fälle, in denen „Singen“ einem Personennamen oder der – nicht zu annotierenden – Gesangstätigkeit entspricht) korrigiert werden. Zudem ermöglicht diese Überblicks-Darstellung ein verlässlicheres Urteil darüber, inwieweit der Listen-Eintrag „Singen“ als Ort überhaupt hilfreich ist.

5.5 Import und Export

Beliebige Dokumente im Textformat können direkt in WALU importiert und annotiert werden. Die Importfunktion beinhaltet interaktive Möglichkeiten zur Auflösung von Mehrspalten-Text. Heuristisch wird zwischen Fließtext, Überschriften, Fußnoten sowie Kopf- und Fußzeilen unterschieden, die jeweils in unterschiedlichen Schriftgrößen dargestellt werden.

Annotierte Dokumente werden in einem XML-Standoff-Format gespeichert. Damit ist es möglich, diese Dokumente auch außerhalb von WALU zu bearbeiten. Details zu diesem Format sind in [Wagner and Rössler, 2007] beschrieben.

Die verschiedenen Datenformate werden in eine interne Repräsentation, den sog. *WARP (WALU Rich Paragraph) Stream*, überführt. Dies ist ein paragraphen-basierter Datenstrom, auf den auch die automatischen Annotatoren zugreifen.

5.6 Kommunikation mit der WIKINGER-Infrastruktur

WALU ist als Stand-Alone-Tool verwendbar, das Daten lokal liest und schreibt. Wie jedoch in Abschnitt 2 dargestellt, bildet WALU zudem einen Teil der in WIKINGER entwickelten verteilten Infrastruktur. Die Einbettung in diese Infrastruktur wird durch spezielle Kommunikations-Mechanismen realisiert. Die in WIKINGER verwendeten Dokumente werden in einem Dokumenten-Repository verwaltet, die zugehörigen Annotationen sowie weitere Informationen im sog. Metadata Repository. Diese Repositories werden auf einem entfernten Server betrieben und sind als gekapselte relationale Datenbanken mit Web-Service-Schnittstellen realisiert. WALU nutzt diese Web-Services zum Laden und Speichern von Daten.

6 Vergleich mit existierenden Tools

WALU ist speziell auf die Annotation semantischer Kategorien ausgerichtet. Wie in Abschnitt 4.2 angeführt, ist es ein entscheidendes Ziel, ein komfortables Werkzeug zu implementieren, das von Experten unterschiedlichster Domänen ohne computerlinguistische und informatische Kenntnisse verwendet werden kann. Dies unterscheidet WALU von existierenden Annotations- und Lern-Umgebungen, die im Bereich Informationsextraktion eingesetzt werden, z.B. GATE [Cunningham *et al.*, 2002], WordFreak [Morton and LaCivita, 2003], MMAX [Müller and Strube, 2001] oder PALinkA [Orasan, 2003]. Diese sind i.d.R. für Benutzer mit (computer-)linguistischem Hintergrund (oder zumindest mit entsprechendem nachhaltigen Support) konzipiert. Dies hat zur Folge, dass sie entweder auf andere, komplexere Tasks zugeschnitten sind (z.B. PALinkA für Diskurs-Annotation) oder in hohem Maße multifunktionell ausgelegt sind (z.B. GATE, WordFreak oder MMAX). Diese Multifunktionalität erlaubt einerseits einen flexiblen, auf komplexe Bedürfnisse zugeschnittenen Einsatz, ist jedoch andererseits mit einem erheblichen Konfigurationsaufwand und einer für „Laien“ mitunter unintuitiven Benutzerführung verbunden.⁵

Zudem ist WALU sowohl als Stand-Alone-Werkzeug als auch als integraler Bestandteil der WIKINGER-Infrastruktur konzipiert. In letzterer Eigenschaft sind spezifische Web-Kommunikationsmodule implementiert.

7 Zusammenfassung und Ausblick

WALU ist ein in der Entwicklung befindliches Werkzeug zum manuellen und (semi-)automatischen semantischen Tagging von Eigennamen und anderen Kategorien. Es wird im Kontext des Projekts WIKINGER entwickelt und bildet einen Teil der dort aufgebauten Infrastruktur. Ebenso ist WALU als Stand-Alone-Tool einsetzbar. Ein primäres Entwicklungsziel ist die nachhaltige Wiederverwendbarkeit, innerhalb und außerhalb der WIKINGER-Plattform.

⁵Z.B. muss in GATE beim Laden einer XML-Datei zusätzlich ein Dokument-Name angegeben werden, oder das System weist einen krytischen Bezeichner zu.

Dies wirkt sich sowohl auf die Wahl der automatischen Erkennungsverfahren (Beispielbasiertheit) als auch auf die konkrete Gestaltung des Tools (Komfortabilität und intuitive Bedienbarkeit geht vor Multifunktionalität) aus. WALU wird erfolgreich in der WIKINGER-Pilotdomäne eingesetzt.

Die nächsten Entwicklungsschritte konzentrieren sich auf die Anbindung weiterer maschineller Lernverfahren an WALU sowie der Untersuchung unterschiedlicher Kombinationen dieser Verfahren. Denkbar ist sowohl die sequentielle Anwendung verschiedener Methoden, sodass aus der Ausgabe eines Klassifizierers Merkmale für den folgenden Klassifizierer generiert werden, als auch die parallele Anwendung unterschiedlicher Klassifizierer, deren Ergebnisse mit Hilfe eines Voting-Mechanismus zusammengeführt werden. Konkret planen wir die Realisierung einer Schnittstelle zur Weka-Bibliothek [Witten and Eibe, 2005], die eine Reihe interessanter und einschlägiger Verfahren zur Verfügung stellt.

Literatur

- [Bröcker *et al.*, 2007] Lars Bröcker, Marc Rössler, Andreas Wagner, et al. WIKINGER - Wiki Next Generation Enhanced Repositories. In *Online Proceedings of the German E-Science Conference*, Baden-Baden, 2007.
- [Chinchor, 1998] Nancy A. Chinchor, editor. *Proceedings of the Seventh Message Understanding Conference*, Fairfax, VA, 1998.
- [Cunningham *et al.*, 2002] Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*, 2002.
- [Hoepfner *et al.*, 2006] Wolfgang Hoepfner, Marc Rössler, Heinz Ulrich Hoppe, and Nils Malzahn. Globale Forschungsgemeinde. IT-Werkzeuge unterstützen die Vernetzung von Wissen. In *Forum Forschung*, pages 20–23. Universität Duisburg-Essen, 2006.
- [Morton and LaCivita, 2003] Thomas Morton and Jeremy LaCivita. WordFreak: an open tool for linguistic annotation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, Edmonton, Canada, 2003.
- [Müller and Strube, 2001] Christoph Müller and Michael Strube. MMAX: A tool for the annotation of multimodal corpora. In *Proceedings of the 2nd IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, Seattle, WA, 2001.
- [Orasan, 2003] Constantin Orasan. PALinkA: A highly customisable tool for discourse annotation. In *Proceedings of the Fourth SIGdial Workshop on Discourse and Dialogue*, Sapporo, Japan, 2003.
- [Rössler *et al.*, 2007] Marc Rössler, Andreas Wagner, Felix Jungermann, and Wolfgang Hoepfner. Applying WALU to annotate named entities in Italian texts. In *Proceedings of the EVALITA 2007 (Evaluation of NLP Tools for Italian)*, Rome, September 2007.
- [Wagner and Rössler, 2007] Andreas Wagner and Marc Rössler. WALU — Eine Annotations- und Lernumgebung für semantisches Tagging. In Georg Rehm, Andreas Witt, and Lothar Lemnitzer, editors, *Data Structures for Linguistic Resources and Applications*, pages 263–271. Gunter Narr Verlag, Tübingen, 2007.
- [Witten and Eibe, 2005] Ian H. Witten and Frank Eibe. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2nd edition, 2005.