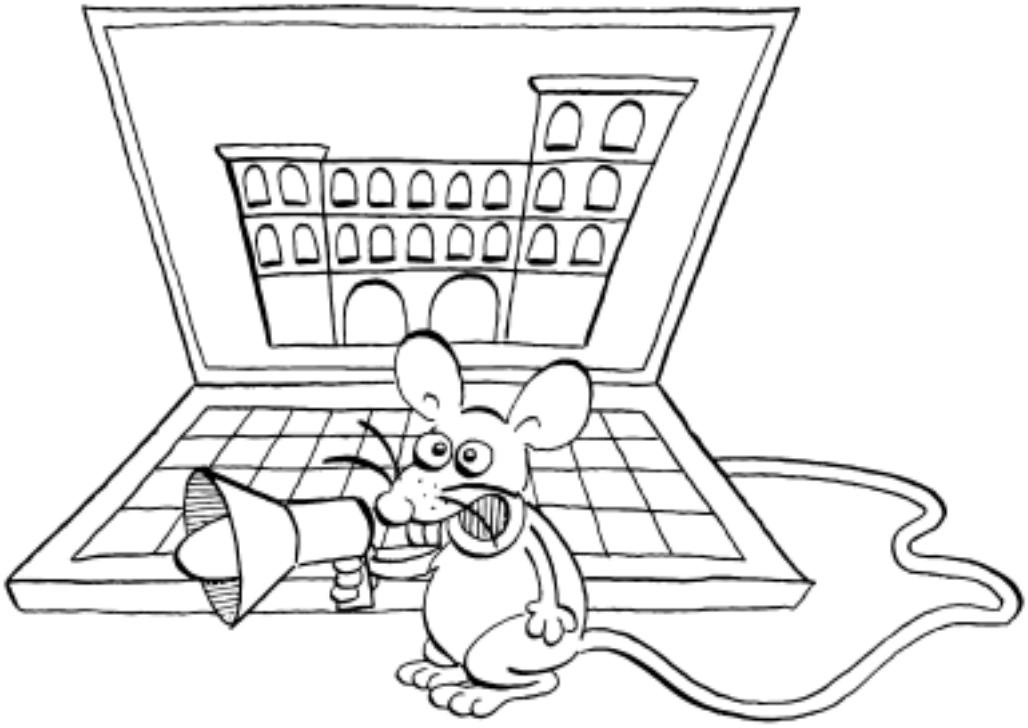
Tagung der Computerlinguistik-Studierenden

Universität Trier

1. - 3. Juni 2012

bacoss

ABSTRACTS



Inhaltsverzeichnis

Andrei Beliankou

Einführung in die Textverarbeitung mit Ruby

Ruby [1] ist eine moderne objektorientierte Skriptsprache. In den letzten Jahren hat sie durch die Verbreitung der Webapplikationen, insbesondere auf der Grundlage von Rails [2], eine sehr große Anwendung gefunden. Langsam findet Ruby auch den Weg zum bisher verborgenen Gebiet der Systemprogrammierung, wo Shell, AWK, Perl, Python sehr lange vorherrschten.

Ruby eignet sich sowol für schnelle Prototypisierung (dafür wird diese Sprache von NASA, CERN und anderen Großunternehmen eingesetzt), als auch für große Web-, Desktop- und Serverapplikationen (wie z.B. Twitter). Für die Verarbeitung natürlicher Sprache entstehen auch zahlreiche Bibliotheken [3]. Die Spannweite der Einsatzmöglichkeiten reicht von morphologischen Aufgaben bis in den semantischen Bereich.

Ruby verbindet eine unvergleichbare Eleganz und die Stärke von Lisp, Perl, Smalltalk, AWK bei der Arbeit mit den Textdaten.

Die angebotene Übung richtet sich in erster Linie an die Teilnehmer, die mit Ruby keine oder nur wenig Erfahrung haben. Im Workshop werden folgende Schwerpunkte gesetzt:

- Entwicklung von einzeiligen Kommandozeilenprogrammen für Datenextraktion und Formattierung;
- Verarbeitung von XML-Dokumenten;
- regelbasierte Datenextraktion mit Hilfe von regulären Ausdrücken;
- Tokenisierungs- und Segmentierungsaufgaben am Beispiel von einem naiven Tokenizer.

Teilnahmevoraussetzungen: jeder Teilnehmer braucht einen eigenen WLAN-fähigen Rechner (Laptop, Tablet) mit der Software für die Erstellung einer SSH-Verbindung (z.B. Putty [4]). Jeder Teilnehmer bekommt für die Zeit der Übung eine vorkonfigurierte virtuelle Linux-Maschine (CentOS) zugewiesen.

[1] http://www.ruby-lang.org/
[2] http://rubyonrails.com/
[3] http://mendicantbug.com/2009/09/13/nlp-resources-for-ruby/
[4] http://www.chiark.greenend.org.uk/~sgtatham/putty/

André Beyer, Daniel Leidisch, Lee Mills

Maschinelle Textkategorisierung anhand von syntaktischen Motiven

Die Textkategorisierung mittels rein semantischer Informationen, z.B. die Kontextuierung mit Schlagwörtern o.Ä., mangelt an einer sprachlichen Informationsquelle, nämlich dem syntaktischen Textfluss. Zwar können viele Informationen über das verbreitete "bag-of-words"-Modell entnommen werden, doch dieses ist für die Betrachtung realisierter Sprachformen i.d.R. wohl nicht adäquat. Das Menzerathsche Gesetz spricht für diese Annahme, da die syntaktische Komplexität und die damit verbundene Anstrengung je nach Textintension verschieden lange Konstituenten bilden kann. Verschiedene pragmatische Absichten und konventionelle Stile in der Texterschaffung legen nahe, dass sich syntaktische Unterschiede in verschiedenen Textkategorien finden lassen könnten. Ein Gesetzestext wird beispielsweise erwartungsgemäß eine höhere syntaktische Komplexität aufweisen als ein Großteil gesprochener Reden.

In Anlehnung an die Musiktheorie wurden für Texte sequenzielle Einheiten mit quantitativer Ausprägung, die sog. Motive, gebildet. Zu unterscheiden sind L-, F- und T-Motive, welche für Längen-, Frequenz- und thematische Informationen stehen. Ein Motiv ist definiert als eine Sequenz von monoton steigenden Zahlen, welche gewisse Eigenschaftswerte repräsentieren. Für ein Längenmotiv können beispielsweise Wort oder Silbenlängen dienen, für die Frequenzmotive Worthäufigkeit pro Text oder Korpus und für T-Motive Werte wie die

Polysemie eines Wortes. Innerhalb einer Motiveinteilung eines Satzes oder Abschnittes können daher auf- oder absteigende Gruppen gebildet werden. Wenn, zur Veranschaulichung, ein Text in L-Motive eingeteilt wurde, bilden benachbarte, aufsteigende Wortlängen (hier gemessen in Buchstaben je Wort) eine Gruppe. Die nächste Gruppe beginnt, sobald eine Länge geringer ist als die Vorherige. Z.B.:

My identity, too, is bewilderingly cloudy.

(2 8) (3) (2 13) (6)

Über diese Motivstruktur können weitere Strukturen gebildet werden, z.B. eine weitere Schicht L-Motive – in diesem Falle sog. LL-Motive usw.:

(2) (1 2) (1)

Mit dem Gedanken, dass sich in verschiedenen Textsorten entsprechend unterschiedliche Ausprägungen von Motiven finden lassen, wurden zwei Kategorisierungsalgorithmen implementiert (Rocchio und K-Nearest). Dabei werden Vektoren gebildet, die aus verschiedenen Motivkombinationen bestehen. Diese Vektoren – einer pro Text – werden zunächst für Texte bereits bekannter Textsorten ermittelt und diesen zugeordnet (überwachtes Lernverfahren). Die Auswahl an Motivkombinationen wird dabei vor Beginn der Lernphase festgelegt und über den gesamten Lern- und Kategorisierungsprozess hinweg beibehalten. Zur Kategorisierung von Texten unbekannter Textsorte wird dann, nach Abschluss der Lernphase, der für den zu kategorisierenden Text ermittelte Vektor im Vergleich mit den jeweiligen Vektoren der einzelnen Textsorten oder aus diesen erstellten paradigmatischen Vektoren (Zentroiden) verglichen und so gemäß des jeweiligen Kategorisierungsverfahrens eingeordnet.

Quellen

Boroda, Moisei, 1982, "Häufigkeitsstrukturen musikalischer Texte" In: Orlov, Jurij K.; Boroda, Moisei G.; Nadarejŝvili, Isabela S. (eds.),"Sprache, Text, Kunst. Quantitative Analysen." Bochum: Brockmeyer, 231-262

Köhler, Reinhard; 1984, "Zur Interpretation des Menzerathschen Gesetzes." In: Boy, Joachim; Köhler, Reinhard (eds.), Glottometrika 6, Bochum, 177-183

Köhler, Reinhard; Naumann, Sven, 2008, "Quantitative text analysis using L-, F- and T-segments." In: Preisach, Burkhardt, Schmidt-Thieme, Decker (eds.) Data Analysis, Machine Learning and Applications. Berlin, Heidelberg: Springer, 637-646

Köhler, Reinhard; Naumann, Sven, 2010, "A syntagmatic approach to automatic text classification. Statistical properties of F- and L-motifs as text characteristics." In: Grzybek, Peter; Kelih, Emmerich; Maĉutec, Ján (eds.), Text and Language. Structures, Functions, Interrelations. Wien: Praesens Verlag, 81-90

Köhler, Reinhard, 2012, "Quantitative Syntax analysis", Berlin: Walter de Gruyter, 114-126

Joachim Bingel

Semantische Klassifikation von Adverbien

Im Rahmen einer Bachelorarbeit wurde eine automatisierte typen- und tokenbasierte semantische Klassifizierung von Adverbien in acht semantische Klassen mit Hilfe von Vektorraummodellen und maschinellem Lernen vorgenommen.

Keine dem Autor bekannte semantische NLP-Ressource bietet eine Kategorisierung von Adverbien hinsichtlich ihrer temporalen, räumlichen oder anderen semantischen Beiträge. Dabei liegt der Nutzen solcher semantischer Informationen gerade für Aufgaben innerhalb der Informationsextraktion, aber auch jeglicher anderer komplexen NLP-Systeme, auf der Hand. Diese Arbeit ist ein Beitrag dazu, dieses Defizit zu überwinden.

Zunächst wird ein Datensatz von 13,960 Adverbtokens (224 Typen) im Korpus manuell nach Klassenzugehörigkeit annotiert, was als Grundlage für das datengetriebene Lernen und die Evaluation dient.

Das Klassifikationsproblem wird auf zwei verschiedene Weisen angegangen. Der erste Ansatz fußt auf der distributionellen Hypothese, welche besagt, dass semantisch ähnliche Wörter in ähnlichen Kontexten auftreten. Daraus abgeleitet ist die Annahme, dass Adverbien, die derselben semantischen Klasse angehören, ebenfalls in ähnlichen Kontexten auftreten. Aufgrund dieser Annahme wird das Klassifikationsproblem mit Hilfe eines Vektorraummodells (VSM) gelöst.

Der zweite Ansatz ist auf der Beobachtung begründet, dass die verschiedenen semantischen Klassen unterschiedliches syntaktisches Verhalten aufweisen. Aus einer Baumbank werden so für jedes Adverbvorkommen acht syntaktische Features extrahiert und als Merkmale in einem maschinellen Lernverfahren verwendet, welches aus diesen syntaktischen Eigenschaften eines Adverbs die jeweilige semantische Klasse ableitet.

Die beiden Ansätze erreichen eine korrekte Klassifikation von 69,4% bzw. 68,2% der Testdaten.

Referenzen

Conlon, S. P.-N. and Evens, M. (1992). Can computers handle adverbs? In Proceedings of the 14th International Conference on Computational Linguistics (COLING), Nantes, France.

Jacobson, S. (1964). Adverbial positions in English. PhD thesis, Uppsala Universitet.

Hartung, M. and Frank, A. (2010). A Structured Vector Space Model for Hidden Attribute Meaning in Adjective-Noun Phrases. In Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC), Valletta, Malta.

Padó, S. and Lapata, M. (2007). Dependency-based construction of semantic space models. In Computational Linguistics, 33(2):161-199.

Turney, P. D. and Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. In Journal of Artificial Intelligence Research, 37:141-188.

Jan Burse

Bedeutungsextraktion als Deduktion

Dieses Projekt ist ebenfalls ein Anschluss an den Piloten des KTI Projekt 6621.2. Haben wir bis jetzt versucht den Oberbau zu stärken, geht es in diesem Projekt darum, den Ober- und den Unterbau auf eine neue Grundlage zu stellen. Die Arbeiten begannen im Jahr 2009 und sind noch nicht abgeschlossen.

Im Pilot wurde eine Anwendung zur Extraktion von Produktprofilen aus Textkatalogen erstellt. Die Textkataloge beschreiben Normpositionen für die Leistungsbeschreibung im Bauwesen. Der Oberbau dient der Anwendung von Grammatik, Lexikon und Thesaurus, und extrahiert eine Interlingua. Der Unterbau transformiert die Interlingua in die Produktprofile. Während des Piloten haben Lexikographen, Grammatiker und Fachexperten einige Textkataloge zur Extraktion gebracht.

Der ursprüngliche Pilot wurde hauptsächlich in Java geschrieben und bei der Abbildung der Algorithmen wurde direkt der imperative Programmierstil von Java benutzt. Das Projektziel besteht darin, möglichst viele Algorithmen in Prolog zu überführen, um später von verschiedenen neueren Technologien im Bereich der Logik Programmierung profitieren zu können.

Eine direkte Überführung in Prolog würde nicht viel Sinn machen, weil dadurch nur die imperative Programmierweise imitiert werden würde. In der Form einer Komponente für die Vorwärtsverkettung haben wir eine zusätzliche Schicht gefunden die uns sowohl eine deklarative Abbildung des Ober-, als auch des Unterbau erlauben wird. Die bisherigen Erfahrungen sind vielversprechend.

Der Abschluss der Arbeit soll zeigen, dass wir eine höhere Flexibilität in der Definition der einzelnen Schritte sowie einfachere Möglichkeiten zur Integration der Schritte erreichen werden.

Referenzen

[Ingebrigt 1936] Ingebrigt, J: Der Minimalkalkül, ein reduzierter intuitionistischer Formalismus, Compositio Mathematica 4, 119–136.
[Pereira & Warren 1983] Pereira, F.C.N., and Warren, D.H.D.: Parsing as Deduction, Artificial Intelligence Center, SRI International, 1983
[Bonner 1988] Bonner, A.: Hypothetical Datalog: Complexity and Expressibility, Theoretical Computer Science, 1988
[Holzbaur 1990] Holzbauer, C.: Specification of Constraint Based Inference Mechanisms through Extended Unification. PhD thesis, Technisch-Naturwissenschaftliche Fakultät der Technischen Universität Wien, 1990
[Hüsler 2006] Hüsler, R.: Expertensystem BAU für die bauorientierte Ausschreibung, Hochschule für Technik und Architektur, Luzern, 2006

Kilian Evang

Kollaborativ und tiefensemantisch annotieren: die Groningen Meaning Bank

Die Groningen Meaning Bank ist ein im Aufbau begriffenes, freies Textkorpus des Englischen mit formaler tiefensemantischer Annotation gemäß der Discourse Representation Theory. Ihr Ziel ist es, die Computersemantik für den Einsatz statistischer Methoden zu öffnen, die große annotierte Korpora benötigen, ähnlich wie Baumbanken für statistische Parser. Da rein manuelle Annotation sehr aufwändig wäre, versuchen wir, die Stärken von Menschen (Linguist/inn/en und Laien) und Maschinen (Tokenizer, POS-Tagger, NE-Tagger, Parser, "Boxer") möglichst geschickt miteinander zu kombinieren. Wir lassen Annotationen von Maschinen produzieren und von Menschen korrigieren, und zwar öffentlich über ein web-basiertes, wiki-artiges Interface für Linguist/inn/en sowie über ein "Game with a Purpose" ähnlich Phrase Detectives. Korrekturen werden nicht nur einmal vorgenommen, sondern in Form von "Bits of Wisdom" gespeichert, die im Gegensatz zu schlichten Änderungen zukünftige Änderungen auf tieferen Annotations-

ebenen überleben. In diesem Vortrag will ich erklären, wie das funktioniert, und in die Benutzung des wiki-artigen Interface einführen - in der Hoffnung, dass einige der Zuhörer/innen selbst anfangen mitzu-annotieren!

Johannes Hellrich

Semantische Suche und Visualisierung von biomedizinischen Relationsdaten

Die ständig steigende Zahl der wissenschaftlichen Veröffentlichungen macht es Forschern schon lange unmöglich, alle neuen Publikationen zu lesen. Ein typisches Gebiet, in dem diese Informationsflut immer mehr zum Problem wird, ist die Biomedizin. Um die in diesem Bereich tätigen Forscher bei der Suche nach für sie relevanten Publikationen zu unterstützen, werden spezialisierte (semantische) Suchmaschinen entwickelt [3, 2].

Ein Schwerpunkt der biomedizinischen Forschung sind Protein-Protein-Interaktionen (PPIs), die zur besseren Übersicht meist als Graph visualisiert werden. Aus computerlinguistischer Sicht handelt es sich bei PPIs um Relationen zwischen Named Entities, die aus Fachtexten extrahiert werden [1]. In meiner Masterarbeit habe ich, auf Basis der semantischen Suchmaschine SEMEDICO[1], einen Prototypen entwickelt, der PPIs speichern, durchsuchen und visualisieren kann. Speicherung und Suche basieren auf RDF[2], einem W3C Standard für das Semantic Web. Die interaktiven Graphen werden mit CYTOSCAPE WEB[3] erzeugt, einer Onlineversion der derzeit populärsten biomedizinischen Visualisierungssoftware. Durch diese Kombination aus Suchmaschine und Visualisierung ergibt sich ein neuartiges Hilfsmittel für die bio-medizinische Forschung, da die klassische termbasierte Suche in Richtung einer Relationensuche erweitert wird.

Literatur

[1] Buyko, E. ; Faessler, E. ; Wermter, J. ; Hahn, U. : Syntactic Simplification and Semantic Enrichment - Trimming Dependency Graphs for Event Extraction. In: Computational Intelligence 27 (2011), Nr. 4, S. 610–644, doi:10.1111/j.1467–8640.2011.00402.x

[2] Lu, Z. : PubMed and beyond: a survey of web tools for searching biomedical literature. In: Database 2011 (2011), article ID baq036, doi:10.1093/database/baq036

[3] Spree, U. ; Feißt, N. ; Lühr, A. ; Piesztal, B. ; Schroeder, N. ; Wollschläger, P. : Semantic search: State-of-the-art-Uberblick zu semantischen Suchlösungen im WWW. In: Lewandowski, D. (Hrsg.): Handbuch Internet-Suchmachinen. 2. Heidelberg: AKA Verlag, 2011, S. 77–109

[1]www.semedico.org/app

[2]www.w3.org/RDF/

[3]cytoscapeweb.cytoscape.org/

Jürgen Hermes, Stephan Schwiebert

Tesla – Ein Labor für Computerlinguisten

Das Text Engineering Software Laboratory (Tesla) ist eine Software, die einen virtuellen Laborarbeitsplatz simuliert, an dem Computerlinguisten und andere Wissenschaftler, die sich mit Texten auseinandersetzen, experimentell arbeiten können.

Laboratorien sind bislang eher aus der naturwissenschaftlichen oder medizinischen Forschung bekannt. Sie werden genutzt, um Rohmaterialien zu lagern, Versuchsaufbauten zusammenzustellen und Experimente durchzuführen. Analog sollte ein computerlinguistisches Labor Zugriff auf Rohmaterialien (hier: Texte) anbieten und die Möglichkeit eröffnen, diese Daten in selbst zusammengestellten Versuchsanordnungen (in Form annotierender und analysierender Werkzeuge) zu prozessieren, um im Anschluss die Ergebnisse dieser Prozessierung (d.h. annotierte und analysierte Daten) auswerten zu können.

Tesla realisiert ein solches computerlinguistisches Labor durch ein Client-Server-System, bei dem Clients die Interaktion mit Anwendern übernehmen und die genannten Funktionalitäten anbieten, während der Server für die u.U. ressourcenintensive Prozessierung und die Datenhaltung zuständig ist.

Der Client basiert auf Basis der Plattform Eclipse und bietet zwei verschiedene Perspektiven, die zwei unterschiedliche Anwenderkreise anspricht:

1. In der Entwickler-Perspektive können neue Werkzeuge (Komponenten) für die Prozessierung von Texten entwickelt werden.
2. In der Linguisten-Perspektive können diese Werkzeuge über einen graphischen Workflow-Editor (siehe Abbildung) zu Versuchsanordnungen (Experimenten) kombiniert und konfiguriert werden. Die Experimente werden vollständig dokumentiert; die gewonnenen Erkenntnisse sind dadurch jederzeit reproduzierbar.

Tesla steht unter der Eclipse Public Licence (Open-Source) und kann unter http://tesla.spinfo.uni-koeln.de heruntergeladen werden. Dort finden sich auch weitere Informationen (Tutorials, Dokumentation, Veröffentlichungen) zum System.

Sebastian Lohmeier

Indirect Anaphors in a Programming Language: Anchoring, Coreference and Referential Ambiguity

Even though programming languages are called "languages", (computational) linguists and computer scientists do seldom look at programming languages from a linguistic perspective. The concept of Naturalistic Programming [1,2] invites such joint effort, and this talk seeks to contribute to it.

The talk introduces work on transferring a cognitive model of the resolution of indirect anaphors [3] to a compiler of an extension [4] of the Java programming language. An indirect anaphor refers to a referent that is not mentioned in the text but is instead related to an explicitly mentioned referent via world knowledge. Indirect anaphors are suited for integration into object-oriented programming languages, because the latter are used to model partial world knowledge.

It is explained how indirect anaphors are anchored in source code and how coreference is handled. Occurrences of referential ambiguity in the current implementation are shown and proposals are made for reducing ambiguity in future versions of the implementation.

[1] C. V. Lopes, P. Dourish, D. H. Lorenz, and K. Lieberherr. Beyond AOP: toward naturalistic programming. SIGPLAN Not., 38(12):34–43, 2003.

[2] R. Knöll and M. Mezini. Pegasus: first steps toward a naturalistic programming language. In OOPSLA ’06: Companion to the 21st ACM SIGPLAN symposium on Object-oriented programming systems,

languages, and applications, pages 542–559, New York, NY, USA, 2006. ACM.

[3] M. Schwarz. Indirekte Anaphern in Texten. Niemeyer, Tübingen, 2000.

[4] S. Lohmeier. Continuing to Shape Statically-Resolved Indirect Anaphora for Naturalistic Programming: A transfer from cognitive linguistics to the Java programming language. 2011.

Victor Persien

Pure Data als Werkzeug phonetischer Analyse

Bei Pure Data (Pd) handelt es sich um eine grafische Echtzeit-Multimedia-Programmiersprache, die Anfang der 1990er Jahre von Miller S. Puckette entwickelt wurde. Grafisch bedeutet in diesem Zusammenhang, dass der Programmierer nicht, wie in herkömmlichen Programmiersprachen, Code in Form von Text schreibt, sondern Operationen, Kontrollstrukturen, Routinen, etc. als Objekte auf der Arbeitsfläche platziert und miteinander verbindet. Die Sprache war ursprünglich dazu gedacht, selber mehr oder weniger intuitiv Software-Synthesizer programmieren zu können. Mittlerweile sind aber viele weitere Bibliotheken (genannt: Externals) aus der Community beigesteuert worden, worunter sich neben reinen Abstraktionen etwa zur Array-Manipulation auch "echte" Erweiterungen zur Verarbeitung von Videosignalen, zur Visualisierung oder zum physikalischen Modellieren befinden. Das Haupteinsatzgebiet von Pd liegt weiterhin im künstlerischen Bereich. Doch auch zur produktiven Manipulation von (akustischen) Daten benötigt man analytische Mittel. Und diese unterscheiden sich glücklicherweise nicht wesentlich von denen, die auch in der akustisch-phonetischen Analyse Anwendung finden. Es wäre also fahrlässig, Pd im Hinblick auf seine Anwendbarkeit im phonetischen Bereich ungeprüft zu lassen.

Soweit mir bekannt, finden zur Zeit weder Pd noch die populärere (unfreie und kommerzielle) Schwester Max/MSP Verbreitung unter

Phonetikern. Der Vortrag soll also dazu dienen, die Software überhaupt erst einmal einem linguistischen Publikum vorzustellen. Dazu werde ich die ein oder andere phonetisch relevante Funktionsweise vorstellen, sowie auf mögliche Anwendungen, Stärken und Schwächen von Pd eingehen. Anschließend hoffe ich auf eine fruchtbringende Diskussions- und Fragerunde. Neben grundlegenden Vorkenntnissen in Phonetik, die man aus Basisseminaren kennt, sind keine tiefergreifenden Kenntnisse akustischer Zusammenhänge nötig, um dem Vortrag folgen zu können.

Pure Data ist freie Software (BSD-Lizenz), Open Source und in der verbreiteteren Variante Pd-extended unter www.puredata.info erhältlich.

Dirk Reimers

Nutzung der Amazon-Wolke für verteiltes Zusammenarbeiten an linguistischen Daten

Die Computerlinguistik ist auf digitale – und oft auch vernetzte – Ressourcen angewiesen. Die verfügbare Rechenleistung setzt der Forschung, will sie Hypothesen belegen und neue Daten schafften, die Obergrenzen. Mit den technischen Fortschritten im Bereich der Cloud bietet sich daher auch der Computerlinguistik neue Chancen.

In den vergangenen Jahren hat sich das Schlagwort Cloud in konkrete Geschäftsmodelle gewandelt, was wiederum der technischen Ausgestaltung dieses Bereichs neuen Raum gab. In der AWS-Cloud von Amazon lassen sich ohne hart gesetzte Einschränkung beliebig viele Rechnerinstanzen nutzen, um komplexe Berechnungen durchzuführen.

In diesem Vortrag wird dargestellt, wie sich die AWS-Cloud unter Steuerung durch das fat solutionlab für diese Zwecke einsetzen lässt. Dabei soll vorgestellt werden:

1. AWS und EC2 - Funktionsweise, Leistungsfähigkeit, Skalierbarkeit

2. fat solutionlab - Wie die Benutzeroberfläche hilft, die Ressourcen optimal und zielgerichtet einzusetzen

1. AWS bietet eine Vielzahl verschiedener Cloud Ressourcen an. Die Amazon Elastic Compute Cloud (EC2) ist dabei für Forschungszwecke am mächtigsten. Über Erstellen eines Maschinen-Images ist es möglich, eine Vielzahl gleichartiger Rechner zu erstellen, die in Kombination komplexe Probleme lösen können.

2. Ressourcen allein erlauben noch kein zielgerichtetes Arbeiten. Ein Interface muss zumindest die Rahmenbedingungen vorhalten, um die Ressourcen so einsetzen zu können, wie für das Forschungsziel zweckdienlich ist. Um dies zu demonstrieren, ist eine Betrachtung der Benutzeroberflächen von AWS und fat solutionlab hilfreich. AWS bietet in seiner Konsole eine Vielzahl von Einstellungsmöglichkeiten und erlaubt es, Instanzen präzise zu verwalten und anzupassen. Bei der Verwaltung einer großen Menge von Maschinen oder verschiedenen Benutzern in einem Account offenbart diese allerdings Schwächen. Das fat solutionlab stößt genau in diese Lücke und bietet unter anderem für die Anwendung im akademischen Bereich Werkzeuge, um in der Cloud effektiv zu arbeiten.

Weiterführende Links

Amazon Web Services, EC2: http://aws.amazon.com/de/ec2/
Das fat solutionlab: http://static.solutionlab.fat.de

Peter Stahl

Der „Event Finder“ – Eine Suchmaschine für Veranstaltungen aller Art

Moderne Suchmaschinen wie Google, Bing, oder Yahoo machen es heutzutage für ihre Nutzer mit der Angabe von nur wenigen Stichwörtern einfach, relevante Webseiten innerhalb kürzester Zeit zu finden. Allerdings liegt genau hier der Schwachpunkt: Sie präsentieren nur Verweise auf möglicherweise relevante Webseiten, aber sie liefern nicht selbst die eigentlichen Informationen, nach denen gesucht wurde. Eine derjenigen Domänen, in denen dieser Umstand von besonderem Nachteil ist, ist die Suche nach Veranstaltungen, wie z.B. Konzerte, Theateraufführungen, Straßenfeste und Flohmärkte.

Veranstaltungsinformationen findet man auf Webseiten jeder Größenordnung: Privatpersonen, Städte und Gemeinden, und auf Events spezialisierte Portale konkurrieren miteinander. Diese Fülle unterschiedlicher Quellen ist gleichzeitig ihre größte Schwäche: Keines der Angebote ist vollständig. Kleinere Events wie Straßenfeste werden auf vielen großen Portalen nicht einmal erwähnt. Stattdessen verstecken sich solche Informationen auf kleinen und oftmals nicht professionell erstellten Webseiten, die häufig nicht suchmaschinenfreundlich aufbereitet worden sind und sehr weit hinten in den Ergebnislisten von Suchmaschinen auftreten. Große Event-Portale werden leichter gefunden, müssen aber auch von Veranstaltern manuell mit Informationen gefüttert werden. Das Eintragen von Veranstaltungen ist aber oftmals langwierig und bedenklich, weil man zunächst ein Benutzerkonto anlegen und dabei private Daten offenlegen muss. Zusätzlich müssen Veranstaltungen in eine Vielzahl von verschiedenen Kategorien einsortiert werden, damit Nutzer solcher Portale sie später finden können. Doch selbst wenn sich Veranstalter diese Mühe machen, ist es mit nur einem Portal nicht getan, weil deren Zielgruppe sich vielleicht bevorzugt woanders informiert. All das macht es für

Veranstalter kompliziert und für potentielle Besucher zeitraubend, Events zu bewerben bzw. zu finden.

Im Rahmen meiner Master-Abschlussarbeit möchte ich untersuchen, inwiefern das Suchen und Finden von Veranstaltungen im Web vereinfacht und verbessert werden kann. Das Ziel ist die Konzeption und prototypische Umsetzung eines „Event Finder“, einer spezialisierten Suchmaschine für Veranstaltungen aller Art. Eine Suchanfrage wie *Flohmärkte Trier 01.06.2012* soll idealerweise direkt eine Liste von allen Flohmärkten liefern, die zu den gegebenen Orts- und Zeitangaben stattfinden, mit möglichst präziser Angabe von z.B. Stadtteil, Straße und Uhrzeit. Mein Vortrag wird verschiedene relevante Teilaspekte behandeln: Vom fokussierten Crawling von entsprechenden Seiten, über die Extraktion der relevanten Informationen, bis zur Evaluation und Präsentation der Ergebnisse. Nicht zuletzt soll ein Überblick über bestehende Angebote gegeben und untersucht werden, inwiefern sie als Hilfsmittel für diesen Zweck verwendet werden können.

Philipp Vanscheidt, Michael Bender

TextGrid für Sprachwissenschaftler

TextGrid entwickelt eine virtuelle Forschungsumgebung mit einer modularen Software, um über ein Nutzer- und Rechtemanagement die weltweite Zusammenarbeit an Projekten zu erleichtern, und einer Infrastruktur, um eine dauerhafte Archivierung von Forschungsdaten zu gewährleisten und wissenschaftlich zitierbare Publikation permanent vorzuhalten [1]. Zudem stellt TextGrid eine Reihe von linguistischen Werkzeugen zur Verfügung:

1. Die Wörterbuchsuche erlaubt die Arbeit mit dem Wörterbuchnetz [2].

2. Der Lemmatisierer ermöglicht, das Lemma und die Flexion einer gegebenen deutschen Wortform zu bestimmen [3].
3. LEXUS stellt eine Umgebung zur Erstellung von Online-Lexika dar [4].
4. Cosmas 2 dient Abfragen in den Korpora des Instituts für Deutsche Sprache [5].
5. ANNEX gestattet es, Videoressourcen in Verbindung mit Anmerkungen abzuspielen [6].

In der virtuellen Forschungsumgebung bestehen diese Werkzeuge nicht mehr für sich allein, sondern lassen sich miteinander und mit weiteren Instrumenten verknüpfen. Über das neue Workflow Tool können außerdem weitere sprachwissenschaftliche Dienste eingebunden und über den "Marketplace" zusätzliche Werkzeuge angeboten werden.

[1] http://www.textgrid.de
[2] http://www.woerterbuchnetz.de
[3] Siehe http://www.ims.uni-stuttgart.de/projekte/gramotron/PAPERS/LREC04/smor.pdf und http://www1.ids-mannheim.de/lexik/TextGrid/morphisto.
[4] http://www.lat-mpi.eu/tools/lexus und http://www.owid.de/wb/elexiko/start.html.
[5] http://www.ids-mannheim.de/cosmas2/. Zurzeit ist der Zugang über TextGrid aus urheberrechtlichen Gründen allerdings begrenzt.
[6] http://www.lat-mpi.eu/tools/elan/

Ekaterina Volkova

ePETaLS: Online annotation tool for emotional text labeling

Text annotation is one of the most popular methods of linguistic data collection. The quality of the resulting corpus is one of the major concerns for the researchers, especially when the annotation process is performed by participants who have not received any specific task-related training. One important factor that can help to ensure high resulting quality is a user-friendly annotation environment.

In this talk we present a new annotation system ePETaLS [1] that can help researchers to collect texts annotated for various emotions. Pre-formatted texts can be uploaded onto the system and their annotation can be assigned to a participant, whose task is to mark each phrase in the text with a specific emotion or leave it neutral. For each phrase, the annotator is also asked to assign the emotional forse and mark the word on which the emotional emphasis falls. Before submission the annotation is checked and the user is informed of any missing values. This step helps to ensure higher quality of the resulting texts. The time spent on each annotation is also logged which helps to detect outliers who spend extremely little or too much time on their annotation tasks. The resulting annotation is saved in a XML format and is ready for data extraction.

Before an annotation procedure can begin, each text is automatically split into small annotation units. These units correspond to short phrases that people would usually pronounce without pausing when they read the text out loud. Each sentence in the text can contain one and more of such units, a typical unit length is three to seven word tokens. This component of ePETaLS is based on supervised machine learning system TiMBL [2] and uses WebLicht [3] for linguistic data extraction, e.g. lemmas, POS, dependency relation, etc. The machine learning algorithm uses a small corpus of texts that were split into phrases by naíve participants.

The annotation system is at present used for collecting a corpus of fairy tales in English written down by Andrew Lang [4]. Each text is annotated for ten to thirteen emotions. The final goal of the project is to create an automatic sentiment analysis system for emotional virtual character animation.

[1] e-petals.kyb.tyebingen.de/uploads
[2] http://ilk.uvt.nl/timbl/
[3] http://weblicht.sfs.uni-tuebingen.de/weblicht.shtml
[4] http://www.gutenberg.org/browse/authors/l#a79

Veranstaltungsplan

	Freitag, 1.6.2012	Samstag, 2.6.2012	Sonntag, 3.6.2012
9:00	**Anmeldung** Tagungsräume P-Gebäude	**Frühstück** Tagungsräume	**Frühstück** Tagungsräume
10:30		*Johannes Hellrich* Semantische Suche und Visualisierung von biomedizinischen Relationsdaten	*André Beyer, Daniel Leidisch, Lee Mills* Maschinelle Textkategorisierung anhand von syntaktischen Motiven
11:00	**Begrüßung** *Prof. Dr. Michael Jäckel (Präsident der Universität Trier) , Prof. Dr. Reinhard Köhler (Geschäftsführer LDV), TaCoS22-Team*		
11:15		**Kaffeepause**	**Kaffeepause**
11:30	**Keynote** *Prof. Dr. Reinhard Köhler*		
11:45		*Kilian Evang* Kollaborativ und tiefensemantisch annotieren: die Groningen Meaning Bank	*Victor Persien* Pure Data als Werkzeug phonetischer Analyse
12:15	*Philipp Vanscheidt, Michael Bender* TextGrid für Sprachwissenschaftler		
12:30		**Mittagspause** zur freien Gestaltung	*Ekaterina Volkova* ePETaLS: Online Annotation Tool for Emotional Text Labeling
13:00	**Mittagspause** Mensa Tarforst		
13:15			*TaCoS22-Team* **Verabschiedung**

	Freitag, 1.6.2012	Samstag, 2.6.2012	Sonntag, 3.6.2012
14:00	*Jürgen Hermes, Stephan Schwiebert* Tesla – Ein Labor für Computerlinguisten	*Peter Stahl* Der „Event Finder“ – Eine Suchmaschine für Veranstaltungen aller Art	
14:45	*Jan Burse* Bedeutungsextraktion als Deduktion	*Joachim Bingel* Semantische Klassifikation von Adverbien	
15:30	**Kaffeepause**	**Kaffeepause**	
16:00	*Dirk Reimers* Möglichkeiten und Mehrwerte der Cloud für die Computerlinguistik	**Workshop** *Andrei Beliankou* Einführung in die Textverarbeitung mit Ruby	
16:45	*Sebastian Lohmeier* Indirect Anaphors in a Programming Language: Anchoring, Coreference and Referential Ambiguity		
17:30	**Organisatorische Hinweise**	**Organisatorische Hinweise**	
17:45	**Pause** (zur freien Gestaltung)	**Pause** (zur freien Gestaltung)	
19:00	**Stadtführung** Innenstadt Trier	**Grillabend** im Studierendenhaus	