UIMA als Plattform für die nachhaltige Software-Entwicklung in den Digital Humanities

Hellrich, Johannes
johannes.hellrich@uni-jena.de
Graduiertenkolleg „Modell Romantik“, Friedrich-Schiller-Universität Jena, Jena, Deutschland

Matthies, Franz
franz.matthies@uni-jena.de
Jena University Language & Information Engineering (JULIE) Lab, Friedrich-Schiller-Universität Jena, Jena, Deutschland

Hahn, Udo
udo.hahn@uni-jena.de
Jena University Language & Information Engineering (JULIE) Lab, Friedrich-Schiller-Universität Jena, Jena, Deutschland

Texte und ihre automatische Analyse stehen im Zentrum vieler Untersuchungen in den Digital Humanities, etwa zur Erforschung sprachlicher und kultureller Wandlungsprozesse (siehe etwa Michel u.a. (2011)) oder im Bereich der Stilometrie (siehe etwa Jannidis (2014)). Die automatische Analyse von Texten beinhaltet typischerweise eine Reihe zunehmend komplexer werdender Schritte, angefangen bei der Segmentierung von Sätzen und Wörtern (Leerzeichen sind kein hinreichendes Kriterium, vgl. „*New York*“) über die syntaktische und semantische Analyse bis hin zu diskursstrukturellen und pragmatischen Analysen. Die für diese einzelnen Schritte nötigen sprachtechnologischen Komponenten sind oft, zumindest innerhalb einer Anwendungsdomäne, wiederverwendbar. Folglich gibt es mittlerweile eine Fülle von Software-Repositorien, die entsprechende computerlinguistische Komponenten sammeln, und Frameworks, die ihre Integration in sogenannte Pipelines, also funktionsbezogene sequenzielle Kombinationen von einzelnen Komponenten, erleichtern. Die dadurch ermöglichte Wiederverwendung von Komponenten ist im Sinne nachhaltiger Forschung, da diese so nicht mehrfach entwickelt werden müssen und der Software-Austausch zwischen Gruppen unterstützt wird.

Uima *(Unstructured Information Management Architecture)* ist ein solches Framework, das sowohl im akademischen Kontext (in Deutschland u.a. DKPro (de Castilho & Gurevych, 2014) und JCoRe (Hahn u.a., 2016)) als auch in industriellen Anwendungen (etwa bei IBMs *Jeopardy* Champion Watson (Ferrucci u.a., 2010)) breite Verwendung findet (einen Vergleich unterschiedlicher Frameworks stellen Bank und Schierle (2012) an). Uima ist *open source* unter der Apache-Lizenz verfügbar und unterstützt mehrere Programmiersprachen, wobei Java in der Praxis eine dominierende Rolle zukommt.

Wir nutzen mit JCoRe seit fast einem Jahrzehnt Uima für computerlinguistische Problemstellungen in verschiedenen Domänen bzw. Sprachen und stellen die dabei entwickelten Komponenten öffentlich zur Verfügung. Aktuell arbeiten wir daran, unser ursprünglich für bio-medizinische Fragestellungen und englischsprachige Fachtexte entwickeltes Repositorium auf den DH-Bereich, primär für das Deutsche, zu erweitern. JCoRe stellt nicht nur sprachtechnologische Komponenten zur Verfügung, sondern auch die dafür nötigen Modelle für verschiedene Domänen — denn vor allem die Erstellung dieser Modelle ist ein enorm zeit- und rechenintensiver Prozess, der zudem ein hohes Maß an computerlinguistischer Expertise verlangt. Um die Einstiegshürden für die Benutzung solcher Ressourcen zu senken, bieten wir Anleitungen und Beispiele zur deklarativen Erstellung von Textanalyse-Pipelines mit Uima und haben zudem eine interaktive Anwendung entwickelt (Hahn u.a., 2016).

Eine Vielzahl von existierenden Sprachanalyse-Komponenten und Repositorien kann über Uima eingebunden werden, darunter auch einige, die nicht originär für das Framework entwickelt wurden, wie etwa das über DKPro verfügbare Stanford CoreNLP (Manning u.a., 2014) oder OpenNLP . Während Uima für den produktiven Einsatz entwickelt wurde, steht beim alternativen *Natural Language Toolkit* (NLTK) der Einsatz in der Lehre im Zentrum (Bird u.a., 2009). Uima ist eher mit dem *General Architecture for Text Engineering* (GATE) Framework (Cunningham u.a., 2011) vergleichbar, das aber ein „geschlossenes“ NLP-System repräsentiert, das exklusiv von den Entwicklern von Gate verwaltet wird. Generell sind integrierte Frameworks vorteilhaft gegenüber Pipelines aus einzelnen Werkzeugen, die mittels Textdateien/-strömen kommunizieren, da nicht bei jedem Schritt zwischen verschiedenen Formaten konvertiert werden muss. Insbesondere werden die bei selbstständigen Werkzeugen verbreiteten *in-line*-Annotationen (wie etwa *„das_Artikel Haus_Nomen“*) vermieden, die sich oft als unübersichtlich und fehleranfällig erweisen.

Uima und die anderen bisher genannten Frameworks sind primär für den Einsatz auf lokaler Rechner-Infrastruktur gedacht und somit nur bedingt mit Systemen wie WebLicht (Hinrichs u.a., 2010) vergleichbar, die als Webservice verschiedene dezentral verteilte Komponenten zusammenführen. Dadurch wird zwar der Einstieg in die Nutzung sprachtechnologischer Systeme erleichtert, jedoch sind derartige Systeme nicht für die Verarbeitung großer Datenmengen geeignet und es entsteht eine eher intransparente Abhängigkeit von fremder Infrastruktur. Uima ist somit kein Konkurrent für WebLicht, sondern ermöglicht es vielmehr, Komponenten zu entwickeln, die bei Bedarf auch (durch in DKPro enthaltene Konverter) in WebLicht eingebunden werden können.

Im Kern ist Uima für die sequentielle Anreicherung mit Metadaten ausgelegt. Die möglichen Annotationen werden frei über ein objektorientiertes Typensystem definiert (siehe etwa Hahn u.a., 2007). In Uima wird zwischen Komponenten unterschieden, die Annotationen vornehmen *(Analysis Engines)*, und solchen, die Texte in das interne CAS *(Common Analysis System)* Format konvertieren *(Collection Reader)*; letztere können dabei auch bereits im Ursprungstext kodierte Metadaten verarbeiten. Die ersten Komponenten, die im Rahmen der Erweiterung JCoRes um DH-Komponenten entstanden und öffentlich zugänglich gemacht wurden, sind ein solcher *Collection Reader*, der die neuerdings vom *Deutschen Textarchiv* (Geyken, 2013) zur Verfügung gestellten Dateien mit TCF- und *Dublin Core*-Annotationen verarbeiten kann, sowie eine entsprechende Erweiterung unseres Typensystems. In der unmittelbaren Zukunft geplante Erweiterungen betreffen *Analysis Engines* für Text- bzw. Wortsegmentierung und Wortartenerkennung (POS-Tagging) in historischen (literarischen) Texten.

Wir möchten durch unseren Beitrag insbesondere diejenigen, die primär computerlinguistische *Anwendungen* für Fragestellungen der Digital Humanities realisieren wollen (und damit meist keine computerlinguistischen *Entwicklung*sinteressen verfolgen), anregen, sich aus dem breiten Fundus existierender Komponenten zu bedienen und diese durch den Einsatz des Uima-Frameworks zu verbinden. Die dadurch implizit eingeführte Modularität erleichtert zudem die Durchführung von Funktionstests, die Anpassung an neue Domänen und darüber hinaus den Austausch mit anderen Forschenden — allesamt Anforderungen an eine nachhaltige Software-Infrastruktur.

Fußnoten

1. https://uima.apache.org
2. https:// DKPro .github.io
3. http://julielab.github.io
4. http://stanfordnlp.github.io/Core NLP
5. https://open NLP .apache.org
6. http://www.nltk.org
7. https://weblicht.sfs.uni-tuebingen.de
8. http://www.deutschestextarchiv.de/download
9. http://weblicht.sfs.uni-tuebingen.de/weblichtwiki/index.php/The_TCF_Format
10. http://dublincore.org

Bibliographie

Bank, Mathias / Schierle, Martin (2012): „A survey of text mining architectures and the Uima standard“, in: *Proceedings of LREC 2012* 3479–3486.

Bird, Steven / Klein, Ewan / Loper, Edward (2009): *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. Sebastopol, CA: O'Reilly.

de Castilho, Eckart R. / Gurevych, Iryna (2014): „A broad-coverage collection of portable NLP components for building shareable analysis pipelines“, in: *OIAF4HLT 2014 – Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT @ COLING 2014* 1–11.

Cunningham, Hamish / Maynard, Diana / Bontcheva, Kalina (2011): *Text Processing with GATE*. Murphys, CA: Gateway Press.

Ferrucci, David A. / Brown, Eric / Chu-Carroll, Jennifer / Fan, James / Gondek, David C. / Kalyanpur, Aditya A. / Lally, Adam / Murdock, J. William / Nyberg 3rd, Eric H. / Prager, John M. / Schlaefer, Nico / Welty, Christopher A. (2010): „Building Watson: An overview of the DeepQA project“, in: *AI Magazine* 31 (3): 59–79.

Geyken, Alexander (2013): „Wege zu einem historischen Referenzkorpus des Deutschen: das Projekt Deutsches Textarchiv“, in: *Perspektiven einer corpusbasierten historischen Linguistik und Philologie* 221–234.

Hahn, Udo / Buyko, Ekaterina / Tomanek, Katrin / Piao, Scott / McNaught, John / Tsuruoka, Yoshimasa / Ananiadou, Sophia (2007): „An annotation type system for a data-driven NLP pipeline“, in: *LAW 2007 – Proceedings of the Linguistic Annotation Workshop @ ACL 2007* 33–40.

Hahn, Udo / Matthies, Franz / Faessler, Erik / Hellrich, Johannes (2016): „Uima-based JCoRe 2.0 goes GitHub and Maven Central: State-of-the-art software resource engineering and distribution of NLP pipelines“, in: *LREC 2016 – Proceedings of the 10th International Conference on Language Resources and Evaluation* 2502–2509.

Hinrichs, Erhard W. / Hinrichs, Marie / Zastrow, Thomas (2010): „WebLicht: Web-based LRT services for German“, in: *Proceedings of ACL-2010: System Demonstrations* 25–29.

Jannidis, Fotis (2014): „Der Autor ganz nah: Autorstil in Stilistik und Stilometrie“, in: Schaffrick, Matthias / Willand, Marcus (eds.): *Theorien und Praktiken der Autorschaft*. Berlin: de Gruyter 169–195.

Manning, Christopher D. / Surdeanu, Mihai / Bauer, John / Finkel, Jenny Rose / Bethard, Steven J. / McClosky, David (2014): "The Stanford CoreNLP Natural Language Processing Toolkit", in: *Proceedings of ACL-2014: System Demonstrations* 55–60.

Michel, Jean-Baptiste / Shen, Yuan K. / Aiden, Aviva P. / Veres, Adrian / Gray, Matthew K. / The Google Books Team / Pickett, Joseph P. / Hoiberg, Dale / Clancy, Dan / Norvig, Peter / Orwant, Jon / Pinker, Steven / Nowak, Martin A. / Aiden, Erez L. (2011): „Quantitative analysis of culture using millions of digitized books“, in: *Science* 331 (6014): 176–182.