Katja Bertholdt
CIS, LMU München
22.05.17

Protokoll zur Sitzung vom 22.05.17 – Computerlinguistik Kolloquium

1. Vortrag: Faridis Alberteris Azar, „Optimierung der linguistischen Suche beim XML-annotierten Nachlass von Ludwig Wittgenstein“

Den ersten Vortrag des Tages hält Faridis Alberteris Azar. Ihre Bachelorarbeit behandelt das Thema „Optimierung der linguistischen Suche beim XML-annotierten Nachlass von Ludwig Wittgenstein“ und sie wird von Dr. Hadersbeck betreut. Ihre Arbeit schreibt Alberteris Azar im Rahmen des digital humanities Projekts „Wittgenstein in co-Text“ in Zusammenarbeit mit dem Wittgensteinarchiv der Universität Bergen. Im Rahmen dieses Projekts wurde eine Suchmaschine namens „WittFind“ entwickelt, die nach Wörtern oder Begriffen im Nachlass von Wittgenstein sucht. Dieser Nachlass besteht aus unveröffentlichten Transkripten und Manuskripten, die Wittgenstein mit einer Schreibmaschine oder händisch verfasst hat, noch bevor sein Werk „Tractatus“(1906-1916) erschienen ist. Dem CIS München stehen 5.000 Seiten des Nachlasses zur Verfügung, die von der Universität Bergen in XML-Dateien transkribiert wurden.
Das Ziel der Bachelorarbeit ist eine Optimierung der linguistischen Suche beim XML annotierten Nachlass von Ludwig Wittgenstein durch die optimale Ausnutzung der XML Annotation und die Verbesserung der XML Editionen. Das Hauptaugenmerk liegt auf einer Verbesserung der Erkennung von Personennamen und eine damit verbundene Erweiterung des Lexikons.
Für das CIS München werden drei verschiedene XML Dateien erzeugt: Eine Originaldatei (ORG), eine normalisierte Datei (NORM) und eine diplomatische Datei (DIPLO). Anhand eines Beispiels erklärt Alberteris Azar, was die jeweiligen Dateien enthalten. In ihrem Beispiel geht es um die nachträgliche Ergänzung des Wortes „Zahn“ vor das Wort „Schmerzen“. In der ORG Datei sind alle Ergänzungen, Änderungen und Streichungen von Wittgenstein enthalten, sowie die eigentlich korrekte (erwartete) Form *(hier: <Zahn> <s> <S><chmerzen>)*. Die NORM Datei beinhaltet nur die Version, die nach den getätigten Änderungen erwartet wird, auch wenn Wittgenstein sie nicht explizit vorgenommen hat *(hier: <Zahnschmerzen>).* Die DIPLO Version enthält lediglich die von Wittgenstein getätigten Änderungen, ohne Rücksicht auf deren Korrektheit *(hier: <ZahnSchmerzen>).* In ihrer Arbeit verwendet Alberteris Azar die NORM Dateien.
Die Suchmaschine „WittFind“ verwendet einen probabilistischen POS-Tagger, der auf einem Markov Modell basiert und von Helmut Schmid entwickelt wurde. Mithilfe dieses Tree Taggers werden in den NORM Dateien nicht alle Personennamen erkannt, was bspw. an einem Fehlen des Namens im Lexikon liegen kann.
Im März 2017 wurde von der Universität Bergen ein vorher noch nicht existierendes neues XML Element in die Dateien eingefügt, das auf Personennamen hinweist. Der Tagger benutzt diese XML Information jedoch nicht, sodass sie zu keiner Verbesserung der Ergebnisse des Taggers führen.
In ihrer Arbeit möchte Alberteris Azar Vorschläge erarbeiten, wie der Tagger die XML Elemente verwenden kann, um die Personennamen zu erkennen. Dazu werden in einem ersten Schritt alle nicht erkannten Personennamen lokalisiert. Außerdem werden die NORM Dateien zusätzlich mit einem Python Parser geparst, um Personennamen zu erkennen. In einem zweiten Schritt wird eine neue syntaktische Kategorie „<+persName>“ für die semantische Suche in „WittFind“ erzeugt und dem bisherigen Suchmuster hinzugefügt, da dieses im Hinblick auf Personennamen noch nicht spezifisch genug war. Die neu identifizierten Personennamen sollen dann auch in das CIS- Lexikon mit aufgenommen werden.
Nach Angaben von Alberteris Azar liefert die Suche nach Personennamen in 20 Dateien mit dem aktuellen Suchmuster von „WittFind“ ein Ergebnis von 168 Treffern. Mit den vorgeschlagenen Verbesserungen führt eine Suche zu 833 Treffern. Hierbei ist erstaunlich, dass trotz eines

zusätzlichen Filters eine größere Ergebnismenge erzielt wird.
In ihrer Arbeit möchte sich Alberteris Azar des Weiteren Transkripitionsfehler und Editionsprobleme in XML Dateien identifizieren und verbessern. Außerdem soll das Lexikon erweitert werden, indem die Wortliste und die Frequenzliste nicht wie bisher mit regulären Ausdrücken extrahiert werden, sondern mittels eines Python Parsers, um dieses Ergebnis dann mit dem des Regex zu vergleichen.