Protokoll zur Sitzung vom 22.05.2017

Kurzvorstellung von der Bachelorarbeit von Faridis Alberteris Azar

BA-Betreuer: Hr. Dr. Maximilian Hadersbeck

Thema: Optimierung der linguistischen Suche beim XML-annotierten Nachlass von Ludwig Wittgenstein

Faridis hat in der heutigen Sitzung ihre Bachelorarbeit vorgestellt, in der es um Optimierung der linguistischen Suche beim XML-annotierten Nachlass von Ludwig Wittgenstein geht. Faridis schreibt ihre Bachelorarbeit im Rahmen des Digital-Humanities-Projekts „Wittgenstein in Co-Text“, in Zusammenarbeit mit Wittgensteins Archiv der Universität Bergen (WAB) in Norwegen. Das Ziel des Projekts ist eine FinderApp WittFind zu entwickeln, die nach Wörtern in Manu- und Typoskripten sucht. Das Ziel der Arbeit ist die Optimierung der linguistischen Suche durch die optimale Ausnutzung der XML-Annotation und die Verbesserung der XML-Edition. Den Schwerpunkt setzt man dabei auf Personennamen.

Die originalen XML-Editionen wurden zu drei Typen XSLT-Dateien in der Universität Bergen konvertiert: NORM (normalisierter Inhalt, wie es korrekt geschrieben werden soll), DIPLO (wie es L. W. im Original geschrieben hat) und ORG (beide Möglichkeiten). Faridis hat in der Sitzung ein Bespiel dazu angeführt: zuerst hat L. W. das Wort „Schmerzen“ in seinem Nachlass geschrieben, später hat er das Wort „Zahn“ dazugeschrieben, also wurde es zu „ZahnSchmerzen“ (DIPLO). Richtigerweise sollte das Wort so geschrieben werden: „Zahnschmerzen“ (NORM). Die beiden Varianten stehen in ORG. Für die Aufgabe verwendet man einen probabilistischen POS-Tagger: TreeTagger, der von Herrn Dr. Helmut Schmid in Stuttgart entwickelt wurde und sich dadurch unterscheidet, dass er Entscheidungsbäume für das Messen für Übergangswahrscheinlichkeiten verwendet. Der TreeTagger taggt die NORM-Dateien und erzeugt neue NORM-tagged.xml-Dateien. Das Problem dabei ist, dass nicht alle Personennamen werden erkannt, so wie „Tolstois“ wird wegen der Endung „-is“ im Genetiv nicht erkannt, d.h. der Tagger kann mit XML-Informationen nicht umgehen.

Faridis versucht in ihrer Bachelorarbeit Vorschläge zu geben, wie man den Tagger besser benutzen kann. Das Erste, was sie unternehmen würde, ist die Eigennamen in NORM.xml-Dateien zu lokalisieren, also alle möglichen Fehler beim Tagging zu sammeln. Dabei kann so eine nützliche Schnittstelle von Python helfen wie The Element Tree XML API. Schritt zwei ist, die semantische Suche in WittFind zu verbessern. Die Suchmaschine in der App sucht momentan die Eigennamen nach dieser RegEx: (([ADJA]|[ADJD])|[NN])&<EN>. Es gibt aber sehr viele falsche Treffer in WittFind („Venus“ ist ein Planet und „hellenisch“ ist eine Sprache, nicht ein Eigenname). Faridis schlägt vor, eine neue syntaktische Kategorie „persName“ zu erzeugen, wobei man in getaggten Dateien bei allen gesammelten Beispielen „persName“ hinzufügt, die neue Kategorie in WittFind erzeugt und diese Kategorie in CIS-Lexikon bei <EN> einträgt. Faridis hat ihre Ergebnisse präsentiert: im Vergleich zu der App findet das von der Studentin empfohlene System fast fünf Mal so viel Personennamen (168 Treffer von der App zu 833 Treffen von dem System). Eine offene Frage bleibt, die die Studentin nicht beantworten konnte: wie kann ein System mit einem zusätzlichen Filter mehr Treffer finden, als das System, das diesen Filter überhaupt nicht verwendet. Logischerweise kann ein zusätzlicher Filter Precision verbessern, aber nicht Recall. Weitere Schritte, mit denen sich Faridis noch beschäftigen möchte, sind Transkriptionsfehler und

Editionsprobleme zu erkennen, also Tokenisierung, Tagging und Personennamenerkennung zu verbessern.