22.05.17

Protokoll zur Sitzung vom 22.05.2017 – Computerlinguistisches Arbeiten

1. Referat: Faridis Alberteris Azar, Optimierung der linguistischen Suche beim XML-Annotierten Nachlass von Ludwig Wittgenstein (Dr. Maximilian Hadersbeck)

Die Arbeit wird im Rahmen des Digital-Humanities-Projekts „Wittgenstein in Co-Text“ erstellt und zwar in Zusammenarbeit mit dem Wittgenstein-Archiv der Universität in Bergen (WAB) in Norwegen. Der Wittgenstein-Nachlass ist ein XML-Annotiertes Manuskript. Das Ziel ist die Optimierung der linguistischen Suche beim XML-annotierten Nachlass. XML-annotierte Editionstexte sind XSLT-Dateien aus Bergen für die Konvertierung der originalen XML-Editionen, wobei es davon drei Typen gibt, zum Beispiel ORG für ‚alle Optionen‘.

Zur Einführung erklärt Faridis zunächst das Ziel der Arbeit, und erläutert anschließend ihre Herangehensweise um die einzelnen Aufgaben zu lösen. Sie beschreibt zuerst die einzelnen Arten von Dateien, die sie in ihrer Arbeit benutzt. Zum einen gibt es die Originaldatei, wo alle möglichen Versionen des Texts enthalten sind, die Wittgenstein hinzugefügt hat, zum anderen gibt es auch die normalisierten Dateien. Diese enthalten nur Versionen, wie die Wörter eigentlich geschrieben werden sollten. Sie erklärt, dass die sogenannten diplomatischen Dateien keine Veränderung der Originaldatei beinhalten, sondern die Version so übernimmt wie sie im Wittgenstein-Text zu finden ist.

Für die Umsetzung wird der probabilistischer POS-Tagger von Dr. Helmut Schmid benutzt, der auf Markovmodellen basiert. Im weiteren Verlauf des Experiments folgt die Eigennamen-Erkennung. Der Tagger erzeugt hierfür ein neues Dokument (der Tagger taggt die NORM-Dateien und generiert die NORM-tagged.xml Dateien.

Faridis erwähnt, dass die Editoren in Norwegen nach März die Dateien aktualisiert haben. In den neuen XML-Elementen steht der richtige Name nun als Lemma. Als nächster Schritt muss nun überprüft werden, wie der Tagger die neuen XML-Dateien verarbeitet. Das Resultat war, dass das Verhalten gegenüber negativen Beispielen sich nicht verändert hat. Das Ziel ist es nun, eine Methode zu finden um die XML-Informationen besser zu nutzen. Um dies zu erreichen sind zwei Schritte von Bedeutung:

1. Schritt: Eigennamen in den NORM.xml Dateien müssen lokalisiert werden
   Alle möglichen Fehler beim Tagging müssen dabei gesammelt werden.
   Eine nützliche Schnittstelle von Python hierfür ist TheElementTree XML API (etree bzw. ET)

2. Semantische Suche in WittFind verbessern: Eigennamen Finden
   Nach einem gewissen Muster werden die Eigennamen gesucht, allerdings hat dieses Muster einige Schwächen, die zu falschen Treffen in WittFind führen.

Der Vorschlag ist, dass eine neue syntaktische Kategorie ‚PersName' erzeugt wird, mittels eTree. Die neue Kategorie muss anschließend in Wittfind erzeugt und in das CIS-Lexikon bei EN eintragen werden.

Anschließend präsentiert Faridis das Ergebnis ihres Experiments:
WittFind findet derzeit in 13 Dateien insgesamt 168 Treffer. Das empfohlene System findet in allen Dateien über 800 Personennamen. Das Lexikon muss erweitert werden, wenn man die Wortliste bzw. Frequenzliste mit Hilfe von etree anstatt mit Regex erzeugt.

2. Referat: Iulija Khobotova, Comparing representation learning over word-level, character-level and combination of both in NLP tasks (Wenpeng Yin)

Iulija starts with presenting the objective of her dissertation with a few fundamental questions. Her work examines how differently the input styles can influence the accuracy of convolutional neural networks (CNN) and aims at finding the best set of parameters for CNN. In the next phase the goal is to measure how fast the accuracy can be calculated and how the combination of both word- and character-embeddings will change.

In her Experiment she changes the parameters for CNN in different dimensions. First she alters the embedding size, secondly she changes the hidden size and lastly the batch size. The goal is to find the best set of parameters that gives the maximum possible accuracy.

Iulija explains that CNN, which stands for Convolutional Neural Network, and RNN, which means Recurrent Neural Network) are the two main types of deep neural network that are widely explored to handle various NLP tasks and gives some examples for CNN's hierarchical and RNN's sequential architecture.

The different representations she used for her experiment are the commonly used word-embeddings, character-level NLP and the combination of both. Her task is to compare those representations in different NLP tasks, such as sentiment classification and POS tagging.

For the implementation of the experiment Iulija uses data from the Stanford Sentiment Treebank, which contains annotated data from movie reviews and about 215k unique phrases and a python framework called Theano (Theano Development Team 2016), which consists of the input layer, hidden layer and output layer. While the input and output layer stay the same, the hidden layer varies in neural networks.

Her current status of the dissertation is at the evaluation stage. She plans to evaluate her results statistically based on comparisons of accuracy and will use different graphical tools to present her results.

3. Referat: Alexander Vordermaier, Comparison of Transfer Methods for low Resource Morphology (Katharina Kann)

Zu Beginn erwähnt Alexander, dass er das gleiche Thema von letzter Woche vorstellt und zeigt anschließend die Gliederung seines Vortrags. Zuerst erläutert er die Motivation für die Arbeit und erklärt, dass es im Prinzip um die Paradigmen Komplettierung von Sprachen geht, also um die Zuordnung eines Lemmas zu seinen flektierten Formen.

Hier spielen High Ressource Sprachen und Low Ressource Sprachen eine bedeutende Rolle, denn vor allem für Low Ressource Sprachen ist die Aufgabe der Paradigmen Komplettierung schwierig, da nur begrenzt Ressourcen verfügbar sind.

Als nächstes beschreibt Alexanders sein Experiment und seine Vorgehensweise und Ergebnisse. Der Ansatz ist ähnliche Sprachen zu finden und zu Mischen um das Problem von Low Ressource Sprachen zu umgehen. In seinem Experiment verwendet er die Sprachen Bulgarisch (als High Ressource Sprache) und Mazedonisch (als Low Ressource Sprache).

In seiner Herangehensweise benutzt Alexander 3 Methoden für die Paradigmen Komplettierung: Die erste Methode ist die Sprachübergreifende Paradigmen Komplettierung, die zweite Methode entspricht einem Auto Encoding und die letzte Variante ist die Kombination aus den beiden ersten Methoden.

Er betont, dass bei der Suche einer High Ressource Sprache für eine Low Ressource Sprache die Ähnlichkeit der beiden Sprachen besonders wichtig ist. Hat man ein Sprachen-Paar gefunden, werden die Daten aus beiden Sprachen vermischt und zusammen trainiert. Er erhofft sich, dass darauf brauchbare Ergebnisse hervorgehen.

Anschließend wird der genaue Ablauf des Experiments beschrieben. Um das Modell zu trainieren werden dazu verschiedene Datenpakete mit verschieden Größen verwendet. Alle Kombinationen werden durchexperimentiert und auch isoliert ausgewertet. Auch nicht annotierte Daten werden in die Kombinationen hinzugefügt.

Alexander erklärt, dass Auto Encoding ein simples Verfahren ist, bei dem die Eingabe auch gleichzeitig die Ausgabe ist. Diese Methode wird benutzt um zu überprüfen ob viele Flektoren der Wörter gleich sind.

Die Form der Daten im Experiment wird ebenfalls erläutert. Um vom Modell verarbeitet werden zu können, müssen die Daten eine gewisse einheitliche Form haben, beispielsweise LANG für Sprache, IN für die Art des Wortes und OUT für den Tag. Am Ende des Übergabestrings wird das Lemma übergeben.

Aus den Ergebnissen lässt sich schließen, dass das Auto Encoding am schlechtesten funktioniert. Es treten vor allem viele Fehler aufgrund der Vorgehensweise auf. Außerdem erwähnt Alexander, dass die Analyse nicht einfach ist, wenn man die zwei Sprachen nicht versteht.

Seine weitere Aufgabe für die Arbeit ist es weitere Fehlerquellen zu identifizieren und bereits gängige Verfahren zu beleuchten.