Faridis Alberteris Azar

Optimierung linguistischer Suche beim XML annotieren

Betreuer: Max Hadersbeck

Die Bachelorarbeit findet im Rahmen des Digital Humanities Projekts WiTTFind und der Finder App des CIS statt, die sich mit dem handschriftlichen Nachlass Ludwig Wittgensteins beschäftigt. Zur Verbesserung der Suche in der Finder App sollen XML-Annotationen besser genutzt werden und Verbesserungen an der derzeitigen Annotation vorgenommen werden. WiTTFind nutzt eine Reihe unterschiedlicher XML Dateien des Nachlasses, die sich u.A. in Silbentrennung, Wortgrenzen oder Groß/Kleinschreibung unterscheiden.

Mit Hilfe des TreeTagger von Helmut Schmid sollen die vom OCR gelesenen Sätze geparst werden, um dann zu besserer Named Entity Recognition zu kommen. Dabei gibt es seit Ende Februar aktualisierte Daten zum Nachlass mit verbesserter XML-Struktur und extra Eintragungen zu Personen als persName. Mit der Implementation der Tags werden inzwischen unter anderem flektierte Formen von Namen gefunden, allerdings bislang noch merkbar Lückenhaft.

Zur Verbesserung der Erkennung wurde eine Liste vorkommender Klarnamen extrahiert, die als Basislemmata für die flektierten Formen dienen. Pro Dokument wurde eine Liste erstellt. Zur Weiterverwertung besteht die Möglichkeit neue syntaktische Kategorien aus der XML-Annotation zusammen mit den Lexikoneinträgen zu kombinieren.

Als bisherige Resultate wurde die Zahl der gefundenen Namen merkbar erhöht. Von 168 Namen in 13 Dateien auf 833 Personen unter Verwendung der Heuristiken. Wodurch die merkbare Verbesserung auftritt, obwohl die neue Unterscheidung insbesondere Personennamen von Objektnamen unterscheidet, konnte bisher nicht erklärt werden. Zur genaueren Evaluation der Ergebnisse wurde nichts weiter angemerkt, deswegen bleibt unter anderem unklar, ob die gefundenen Namen korrekt gefunden nehmen, oder es eine große Reihe an false positives unter Verwendung der neuen Methoden gibt.

In weiteren Schritten soll untersucht werden wie stark sich Fehler in den Dokumenten auf die Suche auswirken. Dabei muss auch beachtet werden, dass zwischen Transkriptionsfehlern und Editionsproblemen differenziert wird, da diese unterschiedlich behandelt werden müssen. Alle weiteren Schritte sollen Verbesserungen durch bessere Tokenisierung, besseres Tagging oder einer genaueren Erkennung der korrekten Personen-XML Tags passieren.

Iuliia Khobotova

Comparing Representation Learning over word-level, character-level and their combination in NLP tasks.

Betreuer: Wenpeng Yin

In der Bachelorarbeit von Iuliia geht es um einen Vergleich unterschiedlicher Modellierungen von Embedding-Modellen von Wörtern durch Convolutional Neural Networks (CNNs). CNNs sind im Gegensatz zu Recursive Neural Networks (RNNs) hierarchische Netzwerke. Nachdem unter der Betreuung von Wenpeng Yin bereits vorheriges Jahr die Performance auf RNNs untersucht wurde, ist dies eine Folgearbeit und soll nach denselben Gesichtspunkten evaluieren wie für RNNs.

Die konkrete Arbeit soll die Netzwerkperformance unter dem Gesichtspunkt dreier Parameter untersuchen:

Embedding size – d.h. der Dimensionalität des Vektorraumes

Hidden Layer size – Zur Transformation von Input vor der Anwendung der Kostenfunktion

Batch size – Der Größe gleichzeitig zu Verarbeitender Trainingsbeispiele

In der Anpassung dieser drei Parameter gilt es die höchste Accuracy auf unterschiedlichen Tasks zu erzielen. Die Implementation eines Sentiment-Analyse-Tasks findet auf der Basis der Stanford Sent Treebank statt. Weitere Evaluationstasks sind vorgesehen.

Alexander Vordermaier

Comparison of Transfer Methods for Low Resource Morphology

Betreuer: Katharina Kann

Alexander bearbeitet einen nah verwandten Task zu Kristina Smirnovs Arbeit, welche vorherige Woche präsentierte. In seiner Arbeit geht es um dieselbe Anwendung von Transfermethoden für low resource morphology, also Sprachen zu denen wenig Ressourcen verfügbar sind, und damit die Verbesserung von Morphologieperformance durch nicht annotierte Daten und der Verwendung verwandter Sprachen als Input. Alexander arbeitet hierzu mit dem Mazedonischen und nimmt als Vergleichssprache das Bulgarische.

Im Task geht es konkret um die Zuordnung eines Lemmas zu seinen flektierten Formen und der Umgehung der Problematik von zu geringen Mengen von Trainingsdaten durch eine von zwei Möglichkeiten: Entweder durch die Adaption von annotierten Daten aus der verwandten Sprache (hier: Bulgarisch) oder durch das Autoencoder Verfahren, bei der von der Äquivalenz von Input und Output ausgegangen wird, die bei der Flexion häufiger Wörter auftritt. (??)

Der vorliegende Datensatz fasst die Trainingsbeispiele in die Unterteilungen LANG für die vorliegende Sprache, IN für das Inputlemma, und eine Anzahl an OUT-Parametern, welche die gesuchte flektierte Form beschreiben. Als Label dient das flektierte Wort in der betreffenden Sprache.

Bei einer bisherigen Fehleranalyse zeigt sich, dass es noch oft zu falschen Endungen bei Wörtern kommt, die vermutlich auf den Einsatz des Autoencoders zurückzuführen sind. Gleichzeitig hatte Alexander von den Problemen als nicht-muttersprachler in der Bewertung der konkreten morphologischen Fehler gesprochen. Zu den weiteren Schritten der Fehleranalyse gehört deswegen die Identifikation bestimmter Fehlerquellen und das weitere Einarbeiten in das bestehende Modell von Kann und Schütze, das als Basis für den Task dient. Zuletzt soll versucht werden mit bereits gängigen Verfahren aus der Literatur die Performance zu verbessern.

Repetitorium Einheit Datenvisualisierung

In dieser Repetitoriumssitzung wurden Möglichkeiten der Datenvisualisierung besprochen: Tortendiagramme zur Darstellung von Verhältnissen einer Gesamtmenge mit disjunkten Teilen, Balkendiagramme zum Vergleich unterschiedlich hoher nichtprozentualer Werte und Liniendiagramme für die Anzeige von stetigen Schwankungen über ein Zeitintervall.

Zur Darstellung von Plots in Python eignet sich die Bibliothek pyplot. Aufruf einer Darstellung mit plot(X,C), die default-Darstellung ist ein Liniendiagramm. Pyplot nimmt automatisch unterschiedliche Farben für unterschiedliche Plots im selben Diagramm. Diese lassen sich auch manuell konfigurieren, z.B. mit den Parametern color, linewidth, linestyle. Zusätzlich ist es möglich Achsen individuell zu beschriften und eine präzise Legende hinzuzufügen.

Zur Darstellung von Vektorräumen in einem 2-D Modell eignet sich ein sogenannter Scatterplot, bei dem im t-SNE Verfahren ein hochdimensionaler Vektorraum auf einen 2-Dimensionalen Raum gemappt wird.