Vortrag von Tobias Ramoser, BA Betreuer Martin Schmitt
Thema: Phonologically-Enhanced Character Embeddings

Zunächst gibt der Vortragende einen Überblick über sein Thema.
Die Motivation der Arbeit ist: Statt dem Wort, die Beziehungen und Zusammenhänge zwischen Buchstaben im Vektorraum (phonologische Vektorraumpräsentation von Buchstaben) als Grundlage maschineller Sprachverarbeitung zu verwenden.
Ziel der Arbeit ist es verschiedene Vektorrepräsentationen von Buchstaben mit phonologischen Features zu erstellen. Diese Repräsentationen werden dann mit Zufallsvektoren bei der Transkription in SAMPA verglichen. SAMPA ist ein auf ASCII-basiertes, maschinen-lesbares, phonetisches Alphabet, mit welchem die Aussprache der Laute dargestellt wird. Zum weiteren thematischen Hintergrund gehören Phonetik, Phonologie und Word2Vec. Aus der Phonetik werden Artikulation und Stimmhaftigkeit näher erläutert. Bei der Artikulation wird unterschieden zwischen Artikulationsart (Wie wird ein Laut gebildet?) und Artikulationsort (Wo wird ein Laut gebildet?). Stimmhaft bedeutet, dass der Kehlkopf beim äußern eines Lautes vibriert, ansonsten spricht man von stimmlosen Lauten.
Phonologie beschreibt die Systematik der Laute innerhalb einer Sprache. Es wird verglichen zwischen Lauten, die wortunterscheidend wirken. Diese Klasse von Lauten nennt man Phoneme.
Word2Vec ist ein Programm zur automatischen Vektorerstellung von Wörtern. Zwei verschiedene Modelle stehen für Word2Vec zur Verfügung, das „Continuous Bag-of-Words“ (CBOW) Modell und das „Skip-Gram“ Modell. CBOW ist ein zweischichtiges neuronales Netz und sagt aus einem Kontext ein bestimmtes Wort voraus. „Skip-Gram“ ist auch ein zweischichtiges neuronales Netz welches aber den Kontext aus einem gegebenen Wort voraussagt.
Insgesamt wurden für die Arbeit vier Experimente durchgeführt:

- 2 Implementierungen für Vektorenerstellung + Zufallsvektoren
- Word2Vec Vektoren
- Transkription in SAMPA

Zunächst werden die Phonemvektoren berechnet. Wenn eine Eigenschaft (Phonologische Features) auf ein Phonem zutrifft, erhält es den entsprechenden Vektor. Die Buchstabenvektoren (Char-Vectors) werden mithilfe des Durchschnitts aller entsprechenden Phonemvektoren berechnet.
Im Unterschied zu den Char-Vectors, liegt bei den One-hot Vektoren eine binäre Klassifikation vor. Die Vektorgröße erhöht sich damit auf 22 Dimensionen.
Für die Zufallsvektoren wird das „random“-numpy-Modul verwendet. Die Zufallsvektoren generieren Buchstabenvektoren in 15 und 100 Dimensionen.
Die Word2Vec Vektoren erhalten als Trainingsdaten Buchstaben statt Wörter. Die Buchstaben werden auf ihre phonologische Ähnlichkeit trainiert. Die Parameter werden per Default festgelegt und auch für diese Vektoren werden die Dimensionen 15 und 100 verwendet.
Die quantitative Analyse der Experimente zeigt, dass ein Zufallsvektor mit der Dimension 100 (random100) die höchste Accuracy erreicht. Die weitere Reihenfolge von gut nach schlecht ist: One-hot Vektor, char-vectors, word2vec100, random15 und word2vec15. Die qualitative Analyse (Berechnung einer Fehlerquote mittels Levenshtein-Distanz) bestätigt die Ergebnisse der quantitativen Auswertung.

Verfasser: Thomas Ebert