

# Statusreport zur Bachelorarbeit 'Semantische Beziehungen'

Ruben Müller

2015

**Hinweis:** Referenzangaben zum Projektplan sind kursiv und in runden Klammern im Format  $(x,xx)$ .

## KW13

Am 25.03.2015 um 11 Uhr fand das Kick-Off Meeting $(1,01)$  im Raum 321 der Hochschule der Medien mit Prof. Dr. Johannes Maucher, M.Sc. Andreas Stiegler (via Skype) und Ruben Müller statt. Hier wurde der Umfang und Inhalt der Bachelorarbeit besprochen.

Am 27.03.2015 habe ich den Projektplan $(2,01)$ , anhand einer Vorlage von [www.meinevorlagen.com](http://www.meinevorlagen.com), angefangen und am 28.03.2015 fertig gestellt.

## KW14

Am 30. und 31. März 2015 habe ich den Abstract $(2,02)$  geschrieben. Um den Abstract zu schreiben, habe ich mich ein wenig in die Grundfunktionen von L<sup>A</sup>T<sub>E</sub>X eingearbeitet  $(3,03)$ . Am 02. April 2015 habe ich mit der Einarbeitung in Word2Vec $(3,01)$  und dem Wikipedia-Korpus $(3,02)$  begonnen. Nachdem ich in PyCharm dann eine Out-of-Memory-Exception hatte, habe ich die neueste Version des Programms heruntergeladen (Version 4.0.5), die auch eine 64-Bit Version bereitstellt. Hier konnte ich dann die Größe des Speichers ausreichend vergrößern.

Mit dem Perl-Skript von <http://mattmahoney.net/dc/textdata.html>, unter Appendix A, kann der Wikipedia-Dump preprocessed werden. Allerdings ist dann der komplette Text in einer Zeile und enthält keine Satzzeichen mehr, die aber für Word2Vec gebraucht werden, da der Input hier Sätze sind. Hier ist das Perl-Skript anzupassen, was nicht viel Aufwand war. Da in der jetzt wegfallenden Regel auch sämtliche Wiki-Formatierungszeichen entfernt werden, müssen diese jetzt wieder in das Skript eingebaut werden.

Nach einigem ausprobieren und mehrfachem Skript ausführen, kam ich dann zum gewünschten Ergebnis.

## KW15

Nach Ostern habe ich dann das Preprocessing der Wikipedia Korpora anhand des kleineren Korpus (die ersten 1 Milliarde Zeichen) weitergeführt. Um einen besseren Überblick zu erhalten, ob das Preprocessing das tut, was ich will, habe ich das große File in kleinere aufgesplittet um diese in Notepad++ überhaupt betrachten zu können. Als nächster Schritt kam die Aufsplittung in einzelne Sätze, da das Word2Vec Modell Sätze als Input verlangt. Die Tokenisierung habe ich mit dem NLTK Tokenizer ('tokenizers/punkt/english.pickle') gemacht.

Als ich die Sätze in ein Array einlesen wollte, um diese dann ins Modell zu geben, bekam ich immer wieder einen MemoryError. Ich habe dann versucht nur eine Teilmenge der Files einzulesen, was dann geklappt hat. Allerdings war mir unklar, was genau das Problem war, da die Fehlermeldung nicht aussagekräftig war.

Ich habe einiges ausprobiert, unter anderem die build\_vocab()-Funktion des Word2Vec Modells, ich vermute aber, dass die Verwendung nicht ganz korrekt war.

Durch kleinere Denkfehler kam ich eine Weile nicht voran, bzw. durch die längere Laufzeit der Verarbeitung verging viel Zeit.

## KW16

Am Anfang der Woche habe ich dann entdeckt, dass ich zwar PyCharm in der 64-bit Version nutze, allerdings die Python-Distribution nur in der 32-bit Version. So habe ich dann die 64-bit Version installiert und die benötigten Komponenten. Ich war sehr erfreut zu sehen, dass ich den kompletten kleinen Korpus ins Word2Vec-Modell laden konnte.

Dann habe ich mich an den kompletten Wikipediadump-Korpus getraut. Selbst das Preprocessing mittels Perl-Skript und meinen Skripts dauerte schon mehrere Stunden. Nachdem ich hier dann wieder einen MemoryError erhalten habe, habe ich das Preprocessing noch optimiert, dass nicht so viel im Arbeitsspeicher gehalten wird, sondern direkt in das separate File geschrieben wird.

Als ich dann die Sätze wieder in einem Array zwischenspeichern wollte, reichten mir meine 32 GB Arbeitsspeicher nicht aus und es gab wieder MemoryError. An einem Punkt, bevor das Programm abstürzte, sah es so aus, als ob einige Daten in die Auslagerungsdatei auf meiner Festplatte geschrieben wurde. Danach kam dann der Error auf. Da ich meine Auslagerungsdatei nur 4 GB groß hatte, da ich mir nie Gedanken gemacht habe, ob meine 32 GB RAM nicht ausreichen, war mein Plan die Auslagerungsdatei zu vergrößern und dann zu schauen ob dieses Mal der Speicher reicht. Aber auch mit 32 GB RAM und 24 GB Auslagerungsdatei kam wieder ein MemoryError. Da ich mein Betriebssystem nur auf einer 128 GB großen SSD habe, konnte ich die Auslagerungsdatei nicht vergrößern. Eine Überlegung war dann noch, auf einer größeren Festplatte Windows nochmals zu installieren und zu versuchen die Auslagerungsdatei so groß wie möglich zu machen. Allerdings wollte ich zuvor erst noch einmal überprüfen, ob es eine andere Möglichkeit gibt, die Sätze

in dem 18 GB großen File ins Word2Vec Modell zu bekommen.

## KW17

Den ersten Versuch, den kompletten Wikipediadump ins Modell zu bekommen, habe ich mit 'gensim.models.word2vec.LineSentence()' gemacht. Dieser Versuch war dann auch erfolgreich. So konnte ich dann zum ersten Mal das Modell auf dem gesamten Wikipediadump erstellen und trainieren. Der erste Durchlauf, mit folgenden Parametern, *size* = 200, *window* = 5, *min\_count* = 20, *workers* = 12, dauerte dann ca. 6 Stunden (32 GB RAM, i7-3770 Quadcore bei 3,4 GHz).

Darauffolgend habe ich mir Gedanken gemacht, wie ich die großen XML-Files in einzelne Wikipedia-Artikel splitten kann, da diese ja klassifiziert werden sollen. Mittels eines SAX-Parsers habe ich dann die einzelnen Artikel in einzelne Files zwischengespeichert. Diese Files habe ich dann mittels Perl-Skript und meinen Preprocessing Skripten wieder in die gewünschte Form gebracht.

Ausblick auf die nächste Berichtsperiode:

Der komplette Wikipediadump soll mit guten Parametern im Word2Vec-Modell gelernt werden (4,01). Des weiteren soll ein Domänenkorpus aus dem kompletten Wikipediadump extrahiert werden(4,02). Das Verhalten der beiden Modelle soll dann später verglichen werden.

## KW18

largetrainedModel400825: 10,5h

largetrainedModel300810: 8,5h

2015-04-29 09:33:51,921 : INFO : capital-common-countries: 81.8% (414/506)

2015-04-29 09:43:43,187 : INFO : capital-world: 79.6% (1745/2192)

2015-04-29 09:44:02,069 : INFO : currency: 0.0% (0/70)

2015-04-29 09:53:58,558 : INFO : city-in-state: 47.3% (1043/2203)

2015-04-29 09:55:42,911 : INFO : family: 84.5% (321/380)

2015-04-29 09:59:26,937 : INFO : gram1-adjective-to-adverb: 16.6% (135/812)

2015-04-29 10:01:22,808 : INFO : gram2-opposite: 37.9% (159/420)

2015-04-29 10:06:50,884 : INFO : gram3-comparative: 68.4% (814/1190)

2015-04-29 10:08:57,792 : INFO : gram4-superlative: 33.1% (153/462)

2015-04-29 10:13:10,012 : INFO : gram5-present-participle: 48.6% (452/930)

2015-04-29 10:19:17,184 : INFO : gram6-nationality-adjective: 85.6% (1174/1371)

2015-04-29 10:25:34,414 : INFO : gram7-past-tense: 51.1% (718/1406)

2015-04-29 10:30:00,645 : INFO : gram8-plural: 59.3% (588/992)

2015-04-29 10:32:35,388 : INFO : gram9-plural-verbs: 46.3% (278/600)

2015-04-29 10:32:35,388 : INFO : total: 59.1% (7994/13534)

largetrainedModel300105: 7,7h

2015-04-29 19:24:36,700 : INFO : capital-common-countries: 83.0% (420/506)

2015-04-29 19:45:12,723 : INFO : capital-world: 80.2% (1759/2192)

2015-04-29 19:45:50,961 : INFO : currency: 0.0% (0/70)

2015-04-29 20:05:06,782 : INFO : city-in-state: 50.4% (1111/2203)

2015-04-29 20:08:13,267 : INFO : family: 81.1% (308/380)

2015-04-29 20:14:57,813 : INFO : gram1-adjective-to-adverb: 16.5% (134/812)

2015-04-29 20:18:29,605 : INFO : gram2-opposite: 38.8% (163/420)

2015-04-29 20:28:27,904 : INFO : gram3-comparative: 68.9% (820/1190)

2015-04-29 20:32:19,788 : INFO : gram4-superlative: 28.4% (131/462)

2015-04-29 20:40:00,747 : INFO : gram5-present-participle: 49.9% (464/930)

2015-04-29 20:51:19,611 : INFO : gram6-nationality-adjective: 88.8% (1217/1371)

2015-04-29 21:02:59,292 : INFO : gram7-past-tense: 48.6% (683/1406)

2015-04-29 21:11:12,999 : INFO : gram8-plural: 60.3% (598/992)

2015-04-29 21:16:14,655 : INFO : gram9-plural-verbs: 44.7% (268/600)

2015-04-29 21:16:14,657 : INFO : total: 59.7% (8076/13534)

extend tokenized line: 319284725 Sätze im kompletten Korpus

Tagging von Wiki-Artikeln.

## KW19

Tagging von Wiki-Artikeln.

Anpassen Perl-Skript, nicht bekannte zeichen löschen, nicht mit " " ersetzen!

## KW20

Die einzelnen Artikel mit dem Perl-Skript reinigen.

Klassifizierung der Artikel und erstellen des Teilkorpus.

Modell aus Teilkorpus erstellen.