Datamining - Versuch 5

Lukas Spieß, Mathis Hoffmann

2. Dezember 2013

2 Durchführung

Hinweis zur Durchführung: Wir haben uns die Aufgabe so aufgeteilt, dass einer von uns die Datensammlung und das Erstellen der Artikel-Wort-Matrix implementiert und der andere die Implementierung der *NNMF* samt Auswertung übernimmt. Um besser die Übersicht zu wahren, haben wir in separaten Dateien gearbeitet. So ist die Datensammlung in der von Ihnen vorgegebenen Datei *newsfeatures.py* implementiert, die *NNMF* samt Auswertung hingegen finden Sie in der Datei *nnmf.py*.

Um besser auch zeitlich unabhängig voneinander arbeiten zu können, mussten wir uns auf eine Schnittstelle zwischen beiden Aufgabenteilen einigen, die auch das einfache Anlegen von Test-Datensätzen zur Implementierung der *NNMF* unabhängig vom Fertigstellungsstatus der Datensammlung erlaubt. Wir haben uns darauf geeinigt, die Daten (also die Artikel-Wort-Matrix) im .csv-Format zu übergeben. Dabei stehen in der ersten Zeile bzw. Spalte die einzelnen Wörter bzw. Artikeltitel. Diese .csv-Datei finden Sie unter dem Namen *artikel-wort-matrix.csv*.

2.1 RSS Nachrichten Feeds einbinden und parsen

1. *Was für eine Datenstruktur liefert die Funktion zurück?*

   Die Parsefunktion liefert ein verschachteltes Python Dictionary zurück, in dem sowohl die eigentlichen Feedinhalte, als auch Metadaten enthalten sind.

2. *Wie kann auf den Titel und die Beschreibung des RSS-Feeds zugegriffen werden?*

   Im von der Parsefunktion zurückgegebenen Dictionary befindet sich ein Dictionary *feed*, das die Keys *title* und *subtitle* enthält. Hier sind als Values die gewünschten Informationen hinterlegt.

2.2 Sammeln und speichern aller Worte der aktuellen Artikel aller eingebundenen Feeds

2.2.1 Worte sammeln

1. *Erklären Sie den Ablauf und die Rückgabewerte der Funktionen stripHtml(h) und separatewords(text) und nehmen Sie diese in das File newsfeatures.py auf.*

   Die *stripHtml* Funktion überprüft eigegebene Strings und entfernt gegebenenfalls HTML Markup aus diesen Strings. Dabei „merkt" sich die Funktion, sobald sie auf den Beginn eines HTML Tag trifft und ignoriert danach alle Zeichen, bis sie auf die schließende spitze Klammer des Tags trifft. Alle anderen Zeichen werden dem Rückgabestring hinzugefügt.

   Die *seperatewords* Funktion entfernt zuerst mit Hilfe des *regular expression* Moduls alle Zahlen aus dem Text, teilt diesen Text dann mit der Pythonfunktion *split()* in einzelne Wörter auf und gibt davon alle in lowercase zurück, die mehr als x Zeichen enthalten und nicht in den Stopwords des Natural Language Tool Kits vorkommen.

2.3 Anzeige der Merkmale und der Gewichte

2.3.3 Implementierung

1. *Geben Sie mindestens 3 Merkmale an und zu jedem Merkmal mindestens 3 Artikel, die das jeweilige Merkmal behandeln.*

   Merkmal: china, defense, country, airlines, united, japan

   - U.S. airlines give China flight plans for new defense zone
   - China, India spar over disputed border
   - Britain's Cameron 'turns page' on Dalai Lama row with China visit

   Merkmal: thanksgiving, friday, black, holiday, shopping, stores

   - U.S. Thanksgiving shopping binge brings Black Friday hangover
   - Nasdaq ends brief post-holiday session at 13-year high
   - Stores open early on Thanksgiving but shoppers in no rush

   Merkmal: twitter, money, tweets, broncos, company, friday

   - Monte Paschi outsources back-office ops, 1,100 jobs
   - Why the Euro Zone Could Unravel Shockingly Fast
   - A Turkish War of Religion: Kurdish Activists Sense a Conspiracy

   Anmerkung: So ganz überzeugend fallen die Ergebnisse (gerade in der letzten Kategorie) noch nicht aus. Sicherlich ließen diese sich durch Finetuning der Parameter noch optimieren. Darüber hinaus würde das Verfahren wahrscheinlich noch deutlich bessere Ergebnisse liefern, wenn nur Substantive für die *NNMP* verwendet würden.