Projektplan: Text Mining - Sentiment Analyse

Fragestellung: Ist es möglich, mithilfe der Sentiment Analyse festzustellen, ob Hauptcharaktere in Büchern gut oder böse sind?

Datenquelle: Harry Potter Romane in Englisch

Tools: Python, Nltk, Calibre, Javascript, D3, SentiStrength

Vorbereitung:

- Download der Bücher und Bereinigen der Texte mit Calibre (Umwandlung in txt-Files, Vorwort/Danksagung etc. entfernen)
- Download SentiStrength
- Lernen, wie man mit SentiStrength arbeitet
- Download D3
- Lernen, wie man mit D3 arbeitet

Programm:

- Text aufbereiten: erst Sätze, dann einzelne Wörter trennen, PoS-Tagging, Named Entity Recognition = Extrahieren der wichtigen Charaktere (Quelle: http://ilias.uni-duesseldorf.de/ilias_0500xy/ilias/goto.php?target=file_446596_download&client_id=UniRZ

- installiere nltk package, lade tagger für nltk package, lade tokenizer für nltk, preprocesossor funktionen um Text zu bereinigen:

installing nltk package
downlaoding taggers

nltk tokenizers/punkt/english.pickle for sentence tokenizing

ersetze '
ersetze \xe2\x80\x99 with '

entferne Seitennummer
entferne \xc2\x91 1 \xc2\x91

ersetze ""
\xe2\x80\x9c mit "
\xe2\x80\x9d mit ""

ersetze . . .
. . . -> ...

ersetze Kapitel
btext = re.sub('CHAPTER
(ONE|TWO|THREE|FIVE|SEVEN|EIGHT|NINE|TEN|ELEVEN|TWELVE|THIRTEEN|FOURTEEN|FIFTEEN
|SIXTEEN|SEVENTEEN|EIGHTEEN|FOUR|SIX)', '', btext)

ersetze Kapitelname
btext = re.sub('(THE WORST BIRTHDAY|DOBBY\'S WARNING|THE BURROW|AT FLOURISH AND
BLOTTS|THE WHOMPING WILLOW|GILDEROY LOCKHART|MUDBLOODS AND MURMURS|THE
DEATHDAY PARTY|THE WRITING ON THE WALL|THE ROGUE BLUDGER|THE DUELING CLUB|THE
POLYJUICE POTION|THE VERY SECRET DIARY|CORNELIUS FUDGE|ARAGOG|THE CHAMBER OF
SECRETS|THE HEIR OF SLYTHERIN|DOBBY\'S REWARD)', '', btext)

ersetze -
ersetze \xe2\x80\x94

- Häufigkeit der Charaktere zählen und nur die wichtigsten aufnehmen (also eine untere Grenze einführen)

- es kommt manchmal "Harry Potter" vor, meistens nur "Harry" und ab und zu nur Potter, das verfälscht das Ergebnis etwas. Man müsste hergehen und alle Synonyme ersetzen. Bsp: "Harry", "Potter", "Harry Potter" -> "Harry"

- NNP die keine Namen sind müssten entfernt werden (Mr., Mrs., Professor, Hogwarts)

- SentiStrength .jar nirgendwo zum Download gefunden (auf der dev seite gibt es nur für .net framework (windows))
https://www.dropbox.com/s/st8eg4dy98lxn2y/SentiStrength.zip broken link

- als Ersatz wird das Ergebnis direkt von der Website geholt und geparsed - das dauert leider sehr lange (ca. 60 min pro Buch)

- zur Visualisierung wird D3 benutzt. Dazu erstellt das Pythonscript eine Website, die lokal abgespeichert wird und die man sich im Browser ansehen kann

- Sentiment Analyse der Wörter in deren Umgebung: SentiStrength nimmt einen txt-File entgegen, wobei jede Zeile einer Sentiment Analyse unterzogen wird und am Ende der Zeile das Ergebnis

geschrieben wird. Man kann auch Keywords angeben, in deren Umgebung die Analyse gemacht wird, was ich mit den Charakternamen testen werde.

- Jedem Charakter einen Ranking-Wert zuordnen (negativ – „böse“, positiv – „gut“)

- Umwandlung in Datenaustauschformat (JSon) als Schnittstelle für JS

- Visualisierung der Ergebnisse mit

Visualisierung (ursprüngliche Version):

- Charaktere als Kugeln – je größer, desto wichtiger der Charakter
- Auf der linken Seite stehen die Charaktere, welche ein negatives Ranking haben, in der Mitte neutrale und rechts diejenigen, die ein positives Ranking haben
- Hintergrund in den Farben rot und grün (böse – gut)
- optional: Der Position auf der y-Achse eine Bedeutung geben
- optional: Beziehungen zwischen Charakteren mit Linien verdeutlichen

JETZT:

- Charaktere als Kugeln – je größer, desto wichtiger der Charakter
- Auf der linken Seite stehen die Charaktere, welche ein negatives Ranking haben, in der Mitte neutrale und rechts diejenigen, die ein positives Ranking haben
- die Kugeln in den den Farben rot und grün (böse – gut)