Natural Language Processing 2021

Tutorial 1 — NLP21

Basic NLP Pipeline

Ziel des ersten Tutorial ist es, eine NLP Pipeline zu erstellen und sich dabei mit den gängigen NLP Modulen vertraut zu machen. Sie werden die meisten der Aufgaben mit *Pandas* sowie dem *nltk* Modul bewältigen können, die die Grundlage für viele NLP Projekte darstellen.
Bearbeiten Sie die Aufgaben in einem *Jupyter Notebook*.

1. **Module importieren**.

   (a) **Import der wichtigsten NLP Module**.
   Stellen Sie sicher, dass die benötigten Module *pandas*, *numpy*, *nltk*, *sklearn* und *re* importiert sind und lassen Sie sich die Versionsnummern der Module ausgeben.

   (b) **Import von "Quality-of-Life Modulen"**.
   Oft sind Module nicht notwendig, erleichtern aber die Arbeit mit größeren Korpora. Installieren sie *pandarallel*. Importieren sie die Methode *pandarallel* (für Parallelization in Pandas) und initialisieren sie diese mit *pandarallel.initialize()*. Sie können nun bei parallelisierbaren Aufgaben *parallel_apply()* anstelle der Pandas Methode *apply()* verwenden.

2. **Daten importieren**
   In diesem Tutorial werden Sie mit einen Dataset aus Witzen arbeiten. Diese wurden mithilfe von Crawlern von den Plattformen "stupidstuff.org", "wocka.com" sowie "reddit.com" gesammelt.

   (a) Lesen Sie die beigefügten .json-Dateien ein und führen Sie diese in einem *Pandas-Dataframe* zusammen. Stellen Sie dabei sicher, dass die Quelle (reddit, wocka, stupidstuff) der Daten als Key oder Label erhalten bleibt.

Fabian Haak: fabian.haak@th-koeln.de

3. **Data Preprocessing.**
   Wie es sehr oft bei gecrawlten Daten der Fall ist, sind die Daten aus den unterschiedlichen Quellen sehr unterschiedlich beschaffen und getaggt.

   (a) **Text bereinigen**.
   Bereiten sie die Daten so auf, dass in der Spalte "body" jeweils der gesamte Text des Witzes enthalten ist und dass keine Format-Tokens (z.B. "\n") mehr enthalten sind.

   (b) **Tokenization**.
   Nutzen sie nltk um die Texte zu *Tokenisieren*. Nutzen sie dazu den *RegexpTokenizer* des nltk Moduls, um mit einem passenden *regulären Ausdruck* nur Tokens aus Wörtern und Zahlen zu übernehmen (keine Satzzeichen). Die Tokens sollen nur kleine Buchstaben enthalten. Speichern Sie die Ergebnisse in einer Spalte "tokens".

   (c) **Stopword Removal**.
   Entfernen sie alle englischen *Stopwords* aus den erzeugten Tokens.

   (d) **POS Tagging**.
   Bestimmen Sie *Part-of-Speech-Tags* für die tokenisierten Texte und speichern Sie diese in einer Spalte "pos".

   (e) **Lemmatisierung**.
   Lemmatisieren sie die Tokens der Texte und speichern sie die bestimmten Lemmata in einer Spalte "Lemmata".
   BONUS: Berücksichtigen sie bei der Lemmatisierung die Wortformen der Tokens.

   (f) **Frequencies**.
   Fügen sie in einer neuen Spalte "frequencies" die Häufigkeiten der lemmatisierten Tokens für jeden Text hinzu. Hinweis: Auch hier bietet das nltk Modul eine einfache Möglichkeit.

4. **Data Analysis**.

   (a) **Überblick über Themen**. Lassen Sie sich die gecrawlten Kategorien für die von Stupidstuff und Wocker gesammelten Witze als Liste ausgeben.

   (b) **Überblick über numerische Werte**. Lassen sie sich die durchschnittlichen Bewertungen für die Kategorien der Witze von Stupidstuff *aufsteigend sortiert* ausgeben. Lassen Sie sich anschließend mithilfe van Pandas einen Überblick über deskriptive Statistiken der Stupidstuff Witze ausgeben, erneut nach Kategorien gruppiert.

5. **Bonusaufgabe**.
   Analysieren oder bearbeiten Sie einen Aspekt ihrer Wahl des Datensatzes. Beispielsweise können Sie für jede der Plattformen den Anteil der Witze berechnen, die ein bestimmtes Wort enthalten, Gesamtworthäufigkeiten der verschiedenen Quellen oder Genres berechnen oder eine andere beliebige Fragestellung bearbeiten, die ohne weitere Module zu importieren umsetzbar ist.