

Projektseminar Angewandte Informationswissenschaft

Thorsten Brückner

Datenanalyse mit Hilfe von Twitter API, Python und Networkx zum Thema: Release von World of Warcraft – Legion am 30.08.16

Bereits fertig (Zusammenarbeit mit Raphael Katschke):

- Twitter-Crawler in Python auf Basis der Twitter-API programmieren.
 1. Twitter API-Key erstellen (Access Token)
 - a. Twitter Account benutzen und <https://dev.twitter.com/apps/new> besuchen
 - b. Application erstellen mit Name, Beschreibung und URL (optional)
 - c. TOS akzeptieren
 - d. Twitter application abschließen und API Key kopieren
 2. Twitter Access Token erstellen
 - a. Consumer Key, Consumer Secret, Access Token, Access Token Secret kopieren und bei Theme Options einfügen
 3. Crawler in Python programmieren
 4. Python OAuth2/Tweepy library benutzen
 5. Accessing Data

To Do:

- Tweets zum Thema WoW-Legion erkennen und extrahieren über den Zeitraum von XX.XX.XX bis zum Release am 30.08.2016.

Mit Hilfe von GEO Tags, die Herkunftsländer von den Tweets analysieren und mit Networkx in Python grafisch darstellen.

1. Key-attributes/fields wählen
 - a. Text
 - b. Geo-Tag
 - i. Coordinates
 - c. ID
2. Tweets fetchen
 - a. Letzten 3000 Tweets bis zum 29.08.16
 - b. Streaming der Tweets vom 30.08.16
3. Daten/Tweets in CSV speichern
4. Daten/Tweets in Python importieren
5. Visualisierung mit Networkx (&Gephi?) umsetzen
 - a. Länder anhand von GEO-Tags unterscheiden und anzeigen
 - i. Filtern von Geo-Tag/Coordinates
 - ii. Filtern von Tweets ohne Geo Tags
 1. Bekommt separate Nodes/Edges

- iii. Benutzen von „Complete Graph“ and „Bipartite Graph“
- iv. Mögliche Graph Layouts (Entscheidung fällt mit Ergebnissen):
 - 1. Open Ord für „besser unterscheidbare Cluster“
 - 2. Yifan Hu für „dynamisches Map Clustern“
 - 3. Force Atlas für „Galaktisches Clustern“
 - 4. Fruchterman Rheingold für „Mass-Partikel Clustern“
 - 5. Welches Layout verwendet wird, entscheidet sich anhand der Ergebnisse erst später (zwecks Übersicht)

- b. Allgemeinen und Twitter Graphen generieren
 - i. Hinzufügen von Notes und Edges anhand von Geo-Tags/Coordinates/ID

• Ebenfalls soll die Sprache der Tweets analysiert und über Networkx dargestellt werden.

- 1. Stoppwortliste
 - a. Englisch
 - b. Deutsch
 - c. Französisch
 - d. Spanisch
- 2. Langdetect Lib für Python
 - a. Wörterbücher erweitern mit nötigen „Game und Szenespezifischen Wörtern/Tags in erforderlichen Sprachen“
 - i. Z.B. Englisch → Deutsch → Spanisch → Französisch
 - a. Warrior → Krieger → Guerrero → Guerrier
 - b. Paladin → Paladin → Paladín → Paladin
 - c. Hunter → Jäger → Cazador → Chasseur
 - d. Rogue → Schurke → Pícaro → Voleur
 - e. Priest → Priester → Sacerdote → Prêtre
 - f. Death Knight → Todesritter → Caballero de la Muerte → Cevalier de la mort
 - g. Shaman → Schamane → Chamán → Chaman
 - h. Mage → Magier → Mago → Mage
 - i. Warlock → Hexenmeister → Brujo → Démoniste
 - j. Monk → Mönch → Monje → Moine
 - k. Druid → Druide → Druida → Druide
 - l. Demon Hunter → Dämonenjäger → Cazador de demonios → Chasseur de démons

- b. Text + ID (duplicates vermeiden)
 - i. Funktion schreiben, die „Text“ als String entgegen nimmt und mit Hilfe von der Langdetect Lib die Sprache ausgibt/identifiziert
 - ii. Anhand der Sprache wird danach zwischen verschiedenen „Dictionaries“ unterschieden und eingetragen
 - iii. Nicht erkannte Texte manuell überprüfen und Sprache markieren

3. Plotly

- a. Als Visualisierungs-Lib benutzen um Sprachen in Diagrammen anzuzeigen
 - i. Importieren von Plotly
 - 1. Simple Bar Charts vs Grouped Bar Charts vs Stacked Bar Charts
 - 2. Relative Barmode
 - 3. Dashboard – Geo Map?
 - ii. „Bar plot“ erstellen
 - 1. Anpassen vom Plot
 - 2. Labels den Bars/Achsen hinzufügen
 - 3. Customizing Colors
 - a. Color Mapping

- 4. Über PHP die Ergebnisse auf Homepage anzeigen lassen (Optional wenn die Zeit es zulässt)