Raphael Katschke

1994872

BrainWasheD

**Einleitung**

Das Thema der Projektidee befasst sich mit der Datenanalyse von Tweets in Hinblick auf Häufigkeiten, Schlagwort-, Erwähnungs- und Hashtagverteilungen zu dem Release der World of Warcraft Erweiterung: Legion.

World of Warcraft (WoW) kam erstmals am 23. November 2004 in Neuseeland, Australien, Mexiko, Kanada und den USA auf den Markt und erschien europaweit am 11. Februar 2005. Das Spiel zählt zu den erfolgreichsten Computerspielen der Welt und erreichte Ende 2010 ein Hoch von 12 Millionen Abonnenten, welche jedoch bis 2015 auf 5.5 Millionen gesunken sind. Am 8. September 2016 wurde berichtet, dass die Anzahl der gleichzeitig aktiven Spieler in der Veröffentlichungswoche von Legion so hoch gewesen ist wie seit 2010 nicht mehr.

Das Spiel baut in seiner Geschichte auf den RTS (Real-time strategy) Spielen Warcraft I bis III auf und erweiterte sie in bis dato 22 Romanen und einem Kinofilm. Es ist öfter Teil von Diskussionen rund um Computerspielesucht und wurde in einer Folge der Sendung Southpark parodiert. Das Entwicklerstudio Blizzard Entertainment ist regelmäßig auf Computerspielemessen vertreten und organisiert selbst eine eigene Messe zu allen Blizzard Entertainment Spielen (BlizzCon). Die Ankündigung der 6. Erweiterung Legion erfolgte 2015 auf der Gamescom und wurde Live auf Twitch von ~250.000 Menschen verfolgt.

WoW gilt als König der MMORPGs und es ist bislang keinem Spiel gelungen, WoW zu „entthronen“. Es ist eine der letzten Bastionen der Abonnementmodelle im MMORPG Genre zusammen mit Final Fantasy 14: A Realm Reborn (5 Millionen registrierte Accounts).

Der mit dem Spiel verbundene Hype soll anhand von Häufigkeitsdistributionen und Wortzusammenhängen genauer analysiert werden. Im Fokus stehen hier die meist genutzten Begriffe und die mit denen zusammen vorkommenden Worte, sowie eine Stimmungsanalyse aller Tweets in Form von Positiv, Neutral und Negativ.

**Projektplan**

Der Plan ist in fünf Schritte nach Komplexität aufgeteilt.

**Schritt 1** sieht vor, dass der Twittercrawler geschrieben und angepasst wird. Da bereits ein Clientcrawler auf vorherigen Projekten vorhanden war, musste dieser lediglich in den Key-Attributes angepasst werden, sodass nach den richtigen Hashtags gesucht und die benötigten Daten gespeichert werden. Dieser Crawler zieht die aktuellsten 3000 Tweets zu einem gegebenen Thema raus.

Ein funktionstüchtiger Streamlistener war nicht gegeben, lediglich ein Grundgerüst. Dieser musste also neu geschrieben und angepasst werden. Durch Testläufe konnten Abstürze und falsch-gespeicherte Daten ausfindig gemacht und behoben werden. Der Streamlistener startet sich automatisch selbst neu, sollte er abstürzen, sodass ein reibungsloser Datenfang gegeben ist.

**Schritt 2** sieht vor, dass der vorhandene CSVReader auf die neuen Bedürfnisse angepasst wird. Er liest die angegebene CSV Datei ein und verteilt die Daten in verschiedene Listen und Dictionaries für den Text, die erkannte Sprache, die Tweetanzahl zu jeder Uhrzeit, die TweetID, Koordinaten und das Land. Da der Reader universal für Thorsten und mich funktionieren sollte, haben wir alle Daten für uns beide gespeichert.

**Schritt 3** geht in die Vorbereitung für die spätere Analyse. Hier soll eine Stoppwortliste auf die Tweettexte angewandt werden um später leichter die Worte trennen zu können. Benutzt wurden dann schließlich 10 Stoppwortlisten der 10 häufigsten Sprachen in WoW (Anhand von Serversprachen ausgewählt) um möglichst effektiv Stoppworte zu entfernen.

Daraufhin sollen Listen für Schlagworte, Erwähnungen und Hashtags erstellt werden, aber eine zusätzlich, die alle vorherigen vereint ist dazugekommen.

Der letzte Punkt deckt sich mit dem nächsten Schritt.

**Schritt 4** behandelt die Analyse der aufbereiteten Daten. Zuerst werden Toplisten für Keywords, Hashtags und Erwähnungen erstellt, dann zusätzlich eine Topliste wo alle drei vereint werden. Darauf aufbauend werden dann gewisse Tendenzen herausgearbeitet. So wird die Anzahl der Erwähnungen in Relation zur Anzahl aller Tweets gestellt, und die einzelnen Erwähnungen bis zu einer gewählten Grenze (Ab 25 Erwähnungen) in einer weiteren Kuchengrafik dargestellt. Aus den beiden Ergebnissen wird dann auf der Homepage der prozentuale Anteil an Kontaktaufnahmen zu den Entwicklern/ Support schriftlich festgehalten. Weiterhin wird eine Sentimentanalyse mit Hilfe von Listen positiver und negativer Worte durchgeführt. Hierfür wurden zusätzlich die emotional aussagekräftigsten Begriffe des WoW-Vokabulars hinzugefügt um möglichst flächendeckend Emotionen aufzugreifen. Zuletzt laufen die Top25 Begriffe – in diesem Falle die vereinten – einzeln durch alle Tweets und kriegen alle zusammen vorkommenden Worte zugewiesen. Aus den entstandenen Dictionaries wird eine Häufigkeitsverteilung angefertigt, aus welcher Wortcluster in Form von Wordclouds erstellt werden. On-Top lassen sich alle Resultate auch Sprachspezifisch erstellen, obwohl ursprünglich die einzelnen Sprachen nicht weiter berücksichtigt werden sollten.

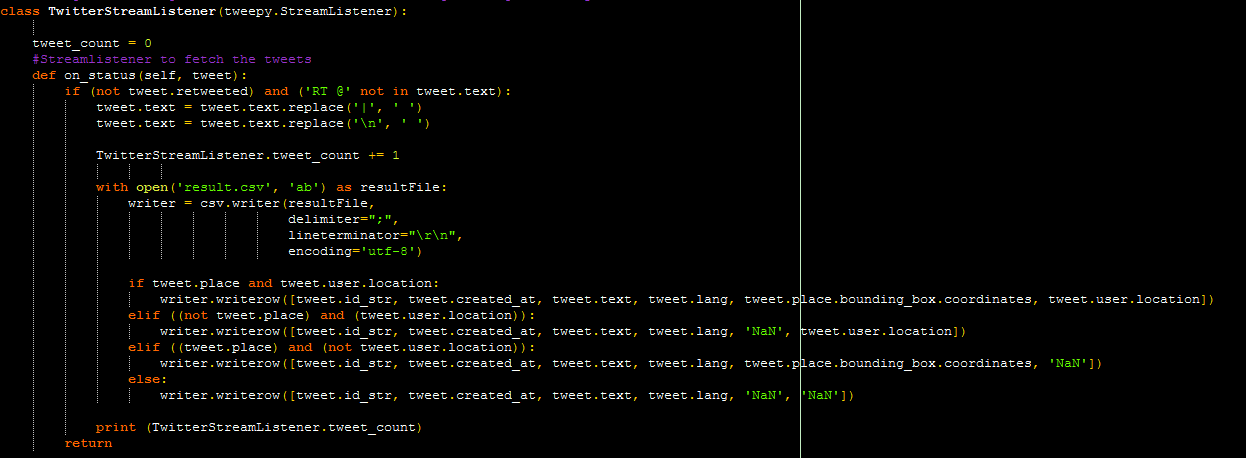
**Schritt 5** fasst die grafische Darstellung der Analyse noch einmal zusammen, so werden Balkendiagramme für die Häufigkeitsverteilungen der Hashtags, Keywords und Mentions erstellt, eine Multilinechart für die Timeline, Kuchengrafiken für Mentions und Sentimentergebnisse, sowie Wordclouds für die Clusterbildung der Wortzusammenhänge der einzelnen Topterme.

Die Ergebnisse werden auf einer Homepage veröffentlicht. Aufgrund der vielen Grafiken – allein 25 Wordclouds – werden nur die wichtigsten/ interessantesten und aussagekräftigsten Grafiken in den Artikel eingebunden. Die restlichen Grafiken werden im dort verlinkten Github Repository hochgeladen.

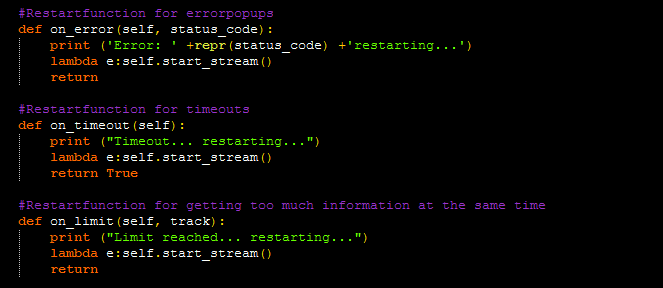
Zusätzlich gibt es einen Scatterplot, der alle Wordclouds zusammengefasst und interaktiv darstellt.

**Methoden**

Für das Sammeln der Daten wurde ein CustomStreamListener der Tweepy Python Library verwendet, welcher die ID, created\_at, text, lang, coordinates und user.location in einer .csv Datei abspeichert.



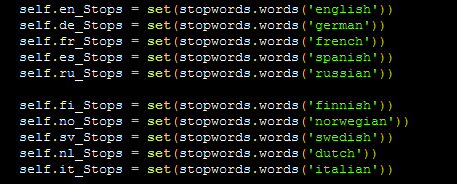
Für den Fall, dass der Stream unterbrochen wird, wurden die drei wichtigsten Fehlermeldungen in entsprechenden Funktionen abgefangen. So werden bei Abstürzen, Zeit- und Limitüberschreitungen Benachrichtigungen in der Konsole ausgegeben und der Stream sofort neugestartet.



Die enstandene .csv Datei wird von einem csvreader eingelesen und in ihre einzelnen Datenteile aufgespalten, welche dann in einem ersten Durchlauf für die weitere Bearbeitung aufbereitet und in Dictionaries abgespeichert werden.



Daraufhin werden die Textdaten an die stopwords.py übergeben, in welcher die Texte durch Stoppwortlisten von den 10 meist genutzten Sprachen in World of Warcraft durchgejagt werden um möglichst viele Stoppworte zu erfassen und entfernen.



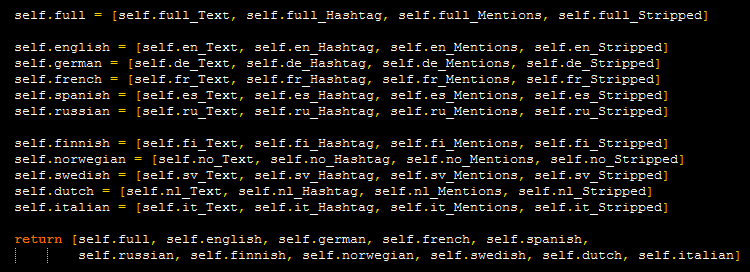
Die Sprachen wurden anhand der Serversprachen und zusätzlich verfügbare Stoppwortlisten der Pythonlibrary NLTK gewählt.



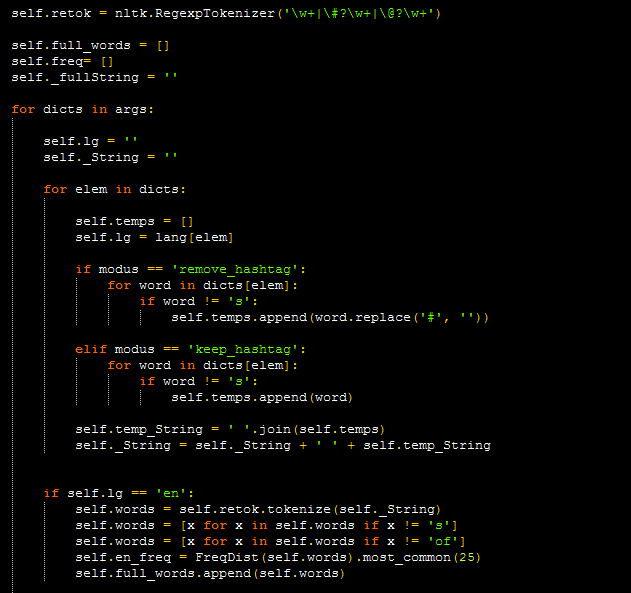
Desweiteren werden die häufigsten nicht von den Stoppwortlisten erfassten Affixe entfernt.



Zuletzt teilt stopwords.py die Daten noch in entsprechende Listen nach Sprache und Texte | Hashtags | Erwähnungen | Schlagworte ein.

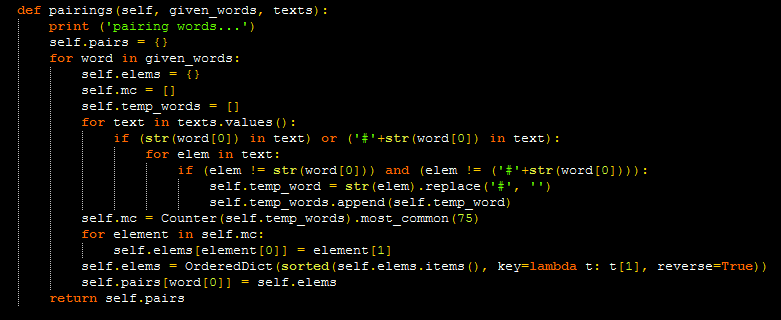


Nach der Stoppwortentfernung und Listenzuweisung werden die entsprechenden Daten in der toplists.py auf ihre Häufigkeitsverteilung analysiert. Hierfür wird sich wieder der NLTK Lib bedient, um Frequency Distributions und Tokenizer mit Hilfe von Regex zu verwenden. Das Programm erkennt zwei Modi, welche entscheiden ob zwischen Hashtags und Schlagworten unterschieden wird oder nicht. Wird der Modus „remove\_hashtag“ angegeben werden bspw. #DemonHunter und DemonHunter zusammengezählt – dieser Modus ist essenziell, bedenkt man die häufige Verwendung von Hashtags in Tweets – der semantische Wert ist oft der gleiche.

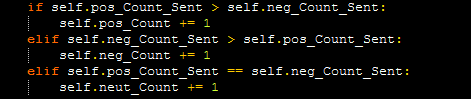


Auch hier lassen sich die Toplisten wieder On-Top bei Bedarf nach Sprachen erstellen. Ausgegeben werden am Ende Listen von Tupeln, die die FrequencyDistributions beinhalten.

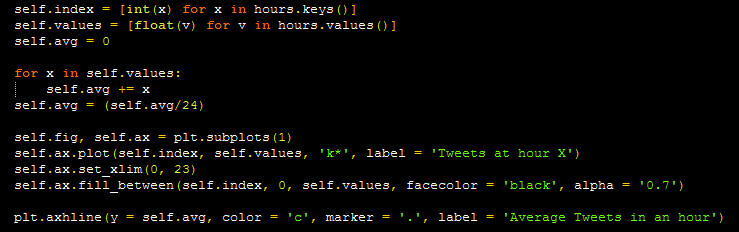
Um die Häufigkeitsverteilungen der mit den aus den Toplisten gewonnenen Begriffen zusammenhängenden Worte zu erhalten, wird die entsprechende Topliste an das Programm wordpairs.py weitergegeben. Um nun die Verteilung zu erarbeiten, werden in einer for-Schleife sämtliche Tweets nacheinander nach dem jeweiligen Begriff durchsucht. Enthält ein Tweet einen solchen Begriff, werden alle Worte anhand einer Liste dem Begriff zugeordnet. Nachdem alle Tweets einer Iterationsrunde durchlaufen sind, werden die gesammelten Worte gezählt und in einem OrderedDict in absteigender Reihenfolge sortiert gespeichert. Nachdem alle 25 Begriffe abgearbeitet wurden, werden die gesammelten Dictionaries in einem übergeordneten Dictionary ausgegeben.



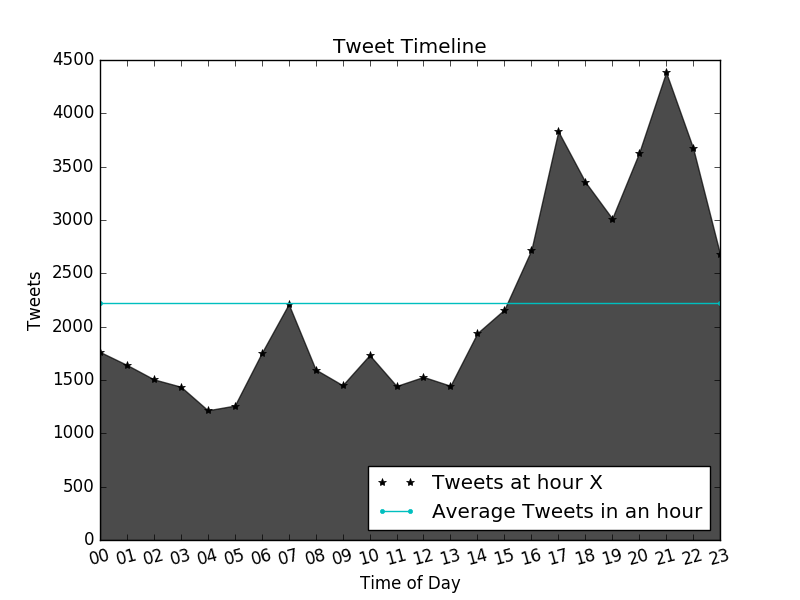
Für die Sentimentanalyse wurden positive und negative Wörterlisten hinzugezogen und nachträglich bearbeitet um die gängigsten emotional behafteten Begriffe des WoW-Jargon aufzufangen. Anhand der Listen werden alle Tweets nach den entsprechend emotional zugeordneten Begriffen durchsucht und diese gezählt. Sollten mehr positive als negative Begriffe vorkommen, wird der entsprechende Tweet als positiv gewertet und vice versa. Im Falle dass positive und negative Begriffe gleichermaßen häufig vorkommen, wird er als neutral betrachtet. Aufgrund der begrenzten Zeichenanzahl bei Twitter und der Durchschnittslänge eines englischen Wortes wurde auf eine genauere Unterscheidung/ Gewichtung verzichtet.



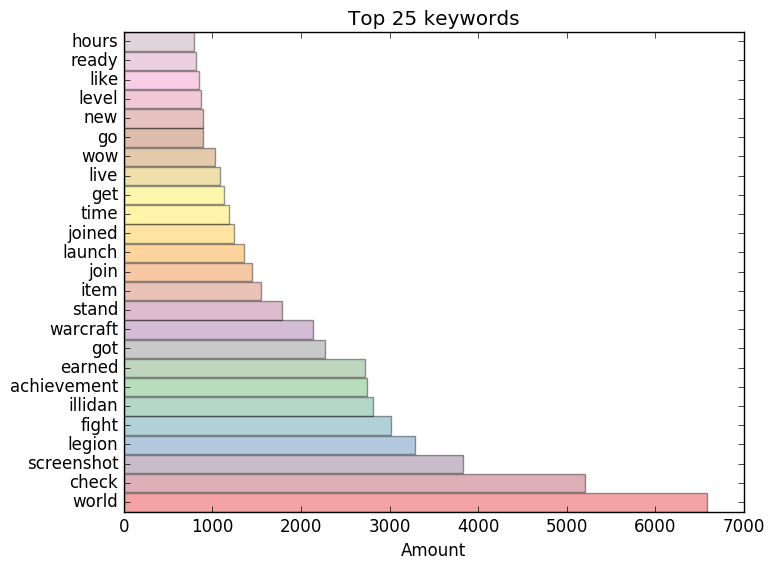
Für die Timeline wurde eine simple Multilinechart ausgewählt, welche die Anzahl der Tweets zur jeweiligen Uhrzeit darstellt. Hierfür wurden mehrere Tage zusammengezählt und auf 24 Stunden verteilt.



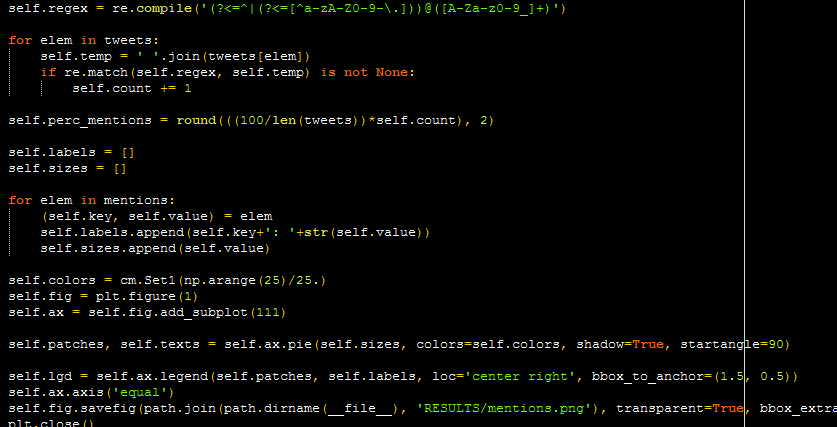
Eine zweite Linie zeigt den gesamten Durchschnitt der stündlichen Tweets.



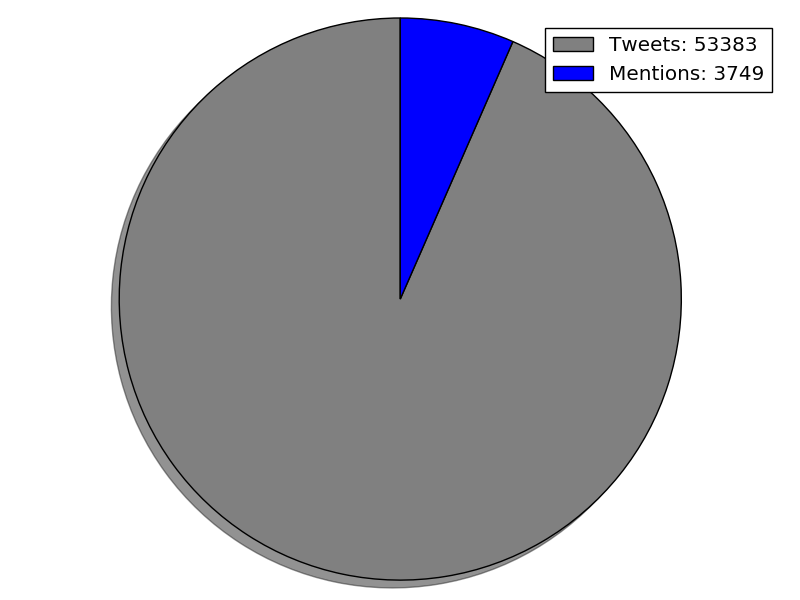
Die Toplisten werden in einer minimalistischen Barchart angezeigt und können Hashtags, Keywords, Mentions und die Zusammenführung derer für einzelne Sprachen übersichtlich darstellen.



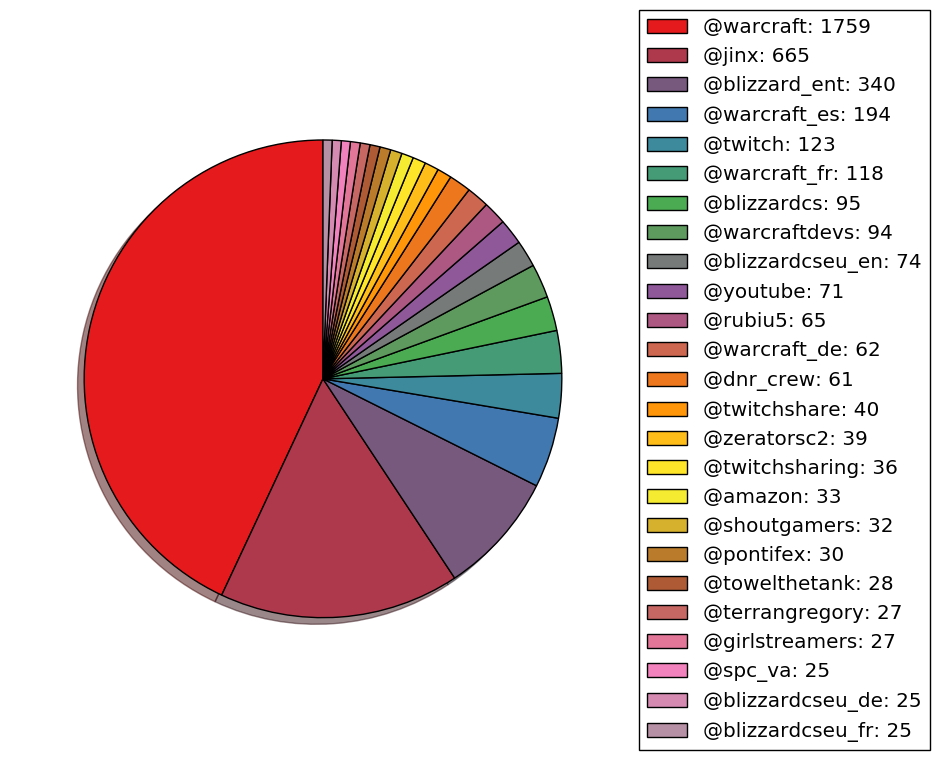
Die Ausnahme bilden die Erwähnungen, welche noch zusätzlich für eine intellektuelle Schlussfolgerung der Tendenzen bezüglich der Relation Kontaktaufnahme/ Tweetanzahl zwei Kuchengrafiken erhalten.



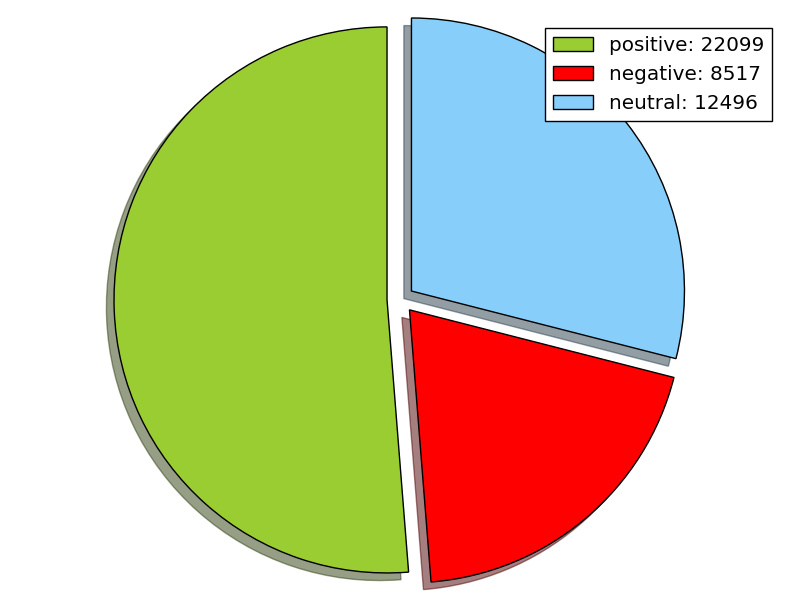
Die erste Kuchengrafik stellt die Anzahl der Tweets mit einer oder mehr Erwähnungen in Relation zu allen Tweets dar.



Die Zweite zeigt die 25 meist genutzten Erwähnungen auf, welche zusätzlich mindestens 25 mal gezählt wurden. Hieraus werden intellektuell die Entwickler- und Supportaccounts gelesen und gezählt um diese ebenfalls in Relation zu allen Tweets zu stellen.



Auch für die Sentimentanalyse stellte sich die Kuchengrafik als optimal dar, da hier lediglich jeweils 3 Ergebnisse in Relation zueinander gestellt werden mussten. Positiv, Neutral, Negativ.



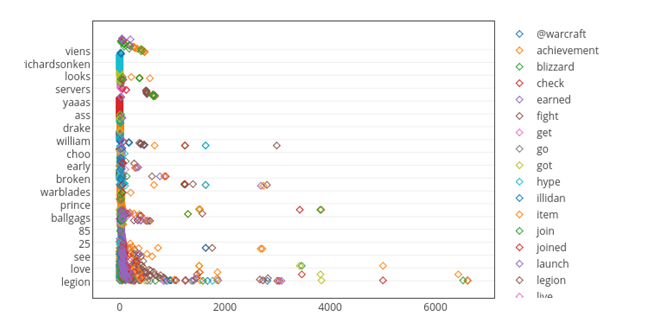
Die Sentimentanalyse lässt sich ebenfalls On-Top für 3 Sprachen sowie alle zusammen darstellen.

Die Wortzusammenhänge und -häufigkeiten werden in Form mehrerer Wordclouds dargestellt, hierzu werden die Dictionaries in eine Unterfunktion der Wordcloud Lib geschoben und zusätzlich das Twitterlogo als Mask sowie eine passende Font geladen.

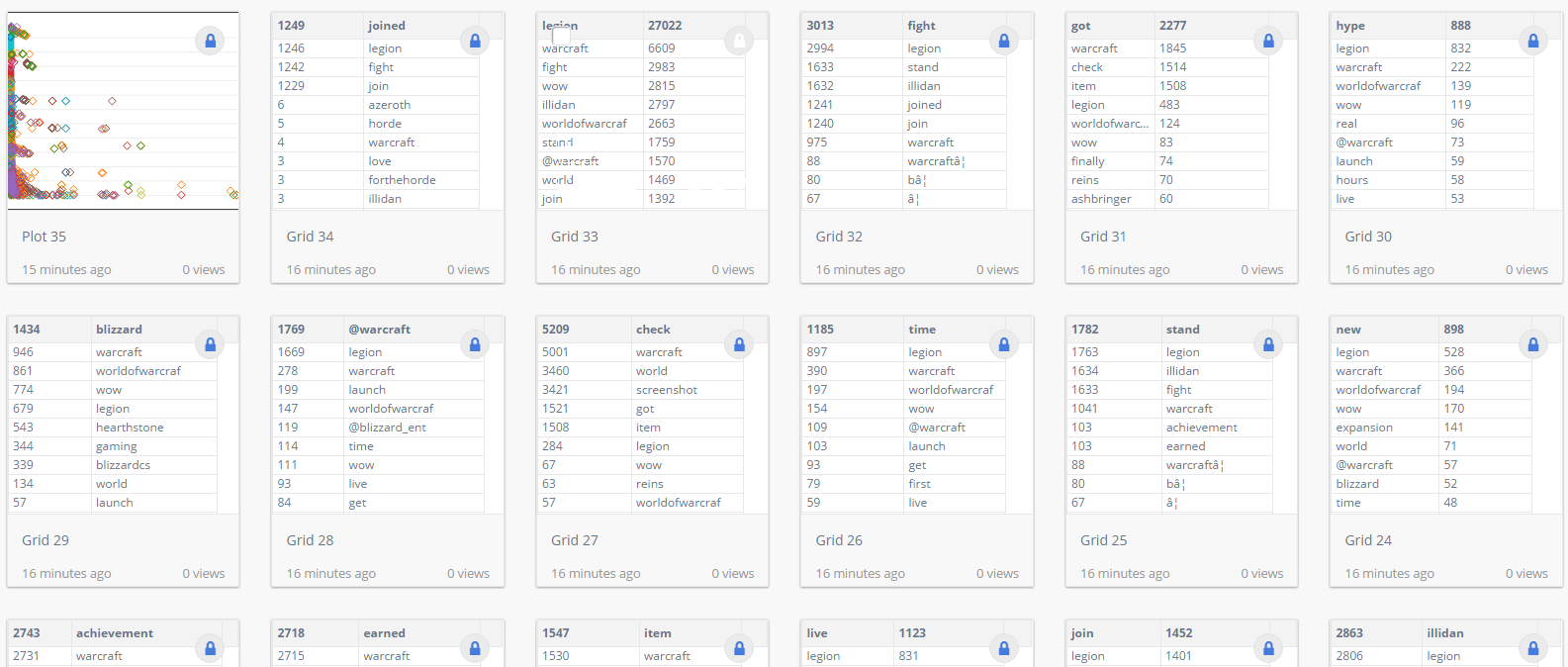


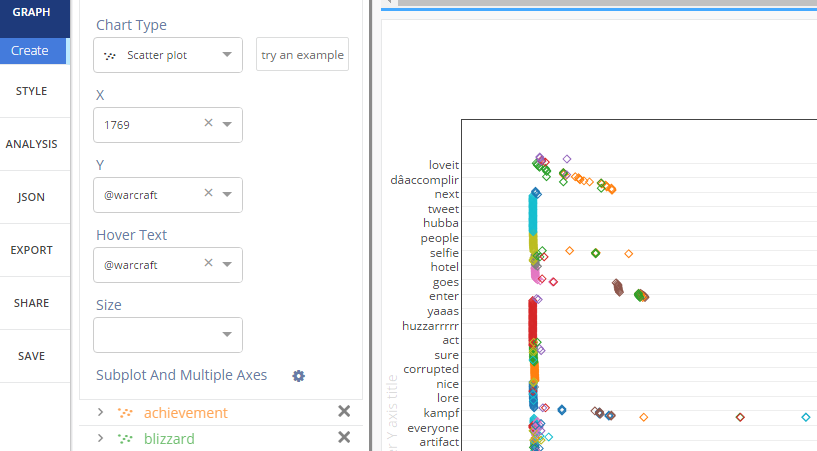
Das oben gezeigte Bild zeigt die am häufigsten mit dem Begriff „warcraft“ zusammen in einem Tweet vorkommenden Wörter an. Je größer das Wort, desto häufiger kam es mit „warcraft“ zusammen vor.

Zusätzlich zu den Wordclouds wurde ein interaktiver Scatterplot angelegt, welcher alle Worte den Toptermen zugeordnet in einer Grafik anzeigt.



Hierzu wurden die Daten der Wordclouds in eine .csv Datei gespeichert und manuell auf einzelne Dateien aufgeteilt um diese dann in die Seite plot.ly einzuspeisen und zu konfigurieren. Das Ergebnis wird als html Datei gespeichert und ist somit interaktiv in Form von zoomen, verschieben und skalieren verfügbar.





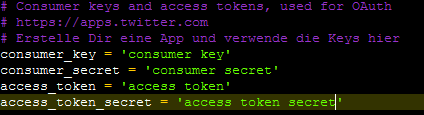
Für den PHP-Teil…

**Bedienungsanleitung**

Für die erfolgreiche Ausführung des Programmes werden die folgenden Libs benötigt:

* Codecs
* Collections
* Matplotlib
* NLTK
* Numpy
* OS
* PIL
* RE
* Tweepy
* Unicodecsv
* Wordcloud

Zum Starten des Streams müssen die Twitter-API Daten in die twitter\_keys.py eingetragen werden.



Um mit dem Tweetseinsammeln zu beginnen, muss das Programm save\_Tweets\_Client.py oder save\_Tweets\_Stream.py ausgeführt werden.

Für alle anderen Funktionen des Projektes wird die main.py ausgeführt. Nicht erwünschte Aktionen können im Quellcode einfach mit einem Hashtag auskommentiert werden.