Clustering und Summarization Services für DIPAS

Daniel Bremer, Daniel Fischer, Simon Jordan, Fin Töter

19. Februar 2021

1 Summarization Service

Der Summarization Service dient dazu, relevante Schlüsselworte aus den einzelnen Beiträgen zu erfassen und auszugeben. Ziel ist es, einen Überblick über Themen eines Beitrages zu gewinnen, indem nur eine kurze Zusammenfassung von n Worten betrachtet wird.

1.1 Konzept

Zur Evaluierung eines Wortes bezüglich seiner Wichtigkeit wird zuerst eine Eingabe benötigt, anhand derer die Frequenz eines Wortes bewertet werden kann und somit die relative Wichtigkeit ermittelt wird. Anschließend gilt es lediglich die Worte eines Beitrages gegen seine Frequenz zu mappen und zu entscheiden, ob ein Wort „wichtig genug“ ist, um in der Ausgabe zu landen.
Hier gibt es die Ansätze Term Frequency (TF) und Term Frequency - Inverse Document Frequency (TF-IDF). Term Frequency stellt den einfachen Ansatz des „Zähle, wie oft ein Wort in einem Text vorkommt“ dar. Zusätzlich kann hier eine Normalisierung erfolgen, um die Vergleichbarkeit von Worten zu erhöhen. TF-IDF auf der anderen Hand betrachtet die Spezifität eines Wortes über alle verfügbaren Dokumente, indem es den Wert des TF Ansatzes mit dem Wert „In wie vielen Dokumenten taucht dieses Wort auf“ skaliert.

Unser Ansatz implementiert eine Mischung aus beiden Ansätzen. Basieren tut der genutzte Algorithmus auf TF, jedoch wird die Frequenztabelle nicht über ein einzelnes Dokument gebildet, sondern über die Summe aller Dokumente. Ein Dokument ist in unserem Fall ein einzelner Beitrag.
Dieser Schritt wurde gewählt, da die einzelnen Beiträge in der Regel recht kurz sind, die Häufigkeitstabelle somit verzerrt wird. Durch erweitern dieses „Trainingsdokumentes“ wird die Präzision stark erhöht und bessere Ergebnisse werden erreicht. Beim späteren Extrahieren der wichtigen Worte wird jedoch nur der Beitrag als Eingabe eingegeben, da nur dieser für die Zusammenfassung relevant ist.
Weiterhin wird für jedes Verfahren eine eigene Frequenztabelle gebaut. Dies hat zum Vorteil, dass Verfahrensspezifische Schwerpunkte besser erfasst werden. So werden „Fahrrad“ und „Verkehr“ bezogene Worte besser im Verfahren „Radverkehr Eimsbüttel“, während der Fokus beim Verfahren „Spielplatz Op'n Hainholt“ auf anderen Themenbereichen liegen kann.

In diesem Bereich gibt es noch ein offenes ToDo: Die API, mit der Beiträge aus dem System extrahiert werden können, wird aktuell umgebaut. In der neuen Version wird es nicht nur möglich sein, alle Beiträge einzeln abzurufen, sondern auch nur einzelne Beiträge abzufragen. Weiterhin gibt es zusätzlich die Möglichkeit Kommentare auf Beiträge zu laden. Dies kann genutzt werden, um einen Beitrag und seine Kommentare zu einem Dokument zu konkatenieren und somit mehr Worte zur Analyse zu haben. Diskutierte Themenbereiche werden somit noch leichter erkannt und die Zusammenfassung wird präziser.

1.2 Schnittstelle zu DIPAS

Zur einfacheren Abfrage der Daten wurde eine Schnittstelle zur DIPAS API gebaut, welche durch Angabe des Verfahrensnamens entweder alle Beiträge oder einen einzelnen Beitrag (durch Download aller Beiträge und anschließender Filterung) zurückliefert. Diese Schnittstelle wird zur neuen Version des DIPAS Systems angepasst, um weiterhin kompatibel zu sein.

1.2.1 Caching

Im Cache landen Daten, für deren Download oder Berechnung viel Zeit benötigt wird. Somit wird die Menge aller Beiträge eines Verfahrens gecached und zusätzlich die Summarizer-Objekte, die auch die Frequenz-Tabellen beinhalten.12 Der Download aller Beiträge kann recht langsam sein, da es sich je nach Verfahren um eine größere Anzahl handeln kann. Um dies zu verbessern wurde Caching implementiert. Durch die neue API wird dieser Cache eventuell obsolet.
Weiterhin wird der Summarizer für jedes Projekt gecached, sodass die Frequenztabelle nicht für jede Anfrage neu berechnet wird.
Jeder Cache gilt für eine bestimmte Dauer, die in der Konfigurationsdatei gesetzt ist, standardmäßig für 24 Stunden. Der erste Aufruf einer Zusammenfassung nach Ablauf der Cache-Frist führt zu einer Neuberechnung der betroffenen Caches, jedoch nicht aller Caches.

2 Summarizer

Wie schon in 1.1 beschrieben, verwendet der Summarizer TF mit der Abwandlung, dass die Bildung der Frequenztabelle über die gesamten Beiträge passiert. Mit Hilfe des `nltk` Paketes werden Stopwords gefunden. Weiterhin wird der Textkorpus von allen nicht-Text-Zeichen bereinigt und zu Kleinbuchstaben konvertiert, um die Streuung der Worte zu verringern. Ebenso wird versucht ähnliche Worte zu finden, indem versucht wird ähnliche Worte wie „finde“ und „finden“ auf ein Wort zu mappen.
Deutsche Sprache benutzt einige (unwichtige) Füllworte und Hilfsworte. Für diese gibt es Listen, welche nach Bedarf erweitert werden können und Modalworte (vielleicht, eigentlich, wie, ...) oder Hilfverben (werden, haben, ...) herausfiltern, da diese für die spätere Zusammenfassung keine Informationen liefern.

3 Nutzung

Das Programm ist als Flask App implementiert, liefert somit eine REST Schnittstelle. Erreichbar ist der Dienst unter „`http://ip:5000/summarize`“. Durch Angabe der Parameter `nid`, `proj` und `n` können die ID des Beitrages, der Bezeichner des Projektes und die Länge der Zusammenfassung eingestellt werden. Ein beispielhafter Aufruf für das Projekt „Radverkehr Eimsbüttel“ und den Beitrag mit ID 52, der auf 10 Worte zusammengefasst wird, lautet somit „`http://ip:5000/summarize?nid=52&proj=radverkehr-eimsbuettel&n=10`“

Laufen lassen kann man den Kram auch mit Docker, ein entsprechendes Dockerfile ist im Projekt beigelegt.