

Hallo miteinander, wir , das sind : Aaron, Bashar und ich, Immanuel, haben uns mit einer Suchmaschine für politische Reden beschäftigt.

Wir wollten dabei interessierten Nutzern, also beispielsweise Journalisten, generell politisch Interessierten, Schülern und Studenten oder auch Lobbyisten :wink: ein Werkzeug an die Hand zu geben, um politisch-motivierte Informationen, also entscheidungsrelevante Prämissen, Thesen und deren Argumentation , sowie der Stellungnahme bestimmter Politiker aus deren Primärquelle zur Verfügung zu stellen, Aussagen im Kontext überprüfen zu können, um so mit das eigene politische Meinungsbild zu schärfen, Recherchen für eben jene Journalisten oder Lobbyisten zu erleichtern und insgesamt damit auch Desinformationskampagnen und Framing entgegenzuwirken.

Dabei gab es im wesentlichen zwei Herausforderungen:

Zum einen die inhaltliche Analyse der Reden: Also der Frage „Wer hat was wann und wo zu welchem Thema gesagt?“ Welche relevante Themen tauchen auf und welche werden angesprochen, auf wen nehmen die Redner*innen Bezug und welche Ausdrücke verwenden sie dazu

Und zum anderen die rhetorische Analyse, um herauszufinden, in welchem Verhältnis das gesprochene Wort zu der vermittelten Aussage der Redner*innen steht. Also werde etwa bewusst eindeutig konnotierte Wörter benutzt, um bei dem Zuhörer eine bestimmte emotionale Regung auszulösen. Welche Metaphern oder populäre Redewendungen werden benutzt, um „die Sprache des Volkes zu bedienen“ und werden etwa logische Fehlschlüsse oder Lügen verbreitet, um den Zuhörer zu einer bestimmten Meinung zu überreden.

Architektur

Unsere Suchmaschine besteht aus drei Komponenten, die alle über REST-APIs untereinander kommunizieren.

Da wäre zum einen die Lucene-Implementierung Solr als Backend zu nennen, welche unser Dataset gespeichert, geparsed und indexiert hat.

Dann natürlich das User-Interface auf Basis von AngularJS, welches auf einem NGINX-Webserver läuft und für das Query-Processing mit Solr kommuniziert.

Als dritte Komponente haben wir noch einen Evaluationsserver, der die Queries und deren Results loggt und mit dem Frontend im Evaluationsmodus das Relevanzfeedback speichert.