IE1 - SKB Bericht - monteric, kabasbet

# Motivation

Da wir zusammen KI1 & KI2 besucht haben, war es für uns naheliegend für diesen Auftrag gelernte Algorithmen aus dem Machine Learning-Bereich zu verwenden. Mittels Unsupervised Learning Verfahren konnten wir die Dokumente und die Queries automatisch nach Themenbereichen (Topics) kategorisieren. Durch das automatische Kategorisieren der vorhandenen Daten können so Suchbegriffe mittels Kategorisierung irrelevante Kategorien "prunnen" -> weniger Suchaufwand

# Verwendete Systeme

## Python mit Jupyter Notebook

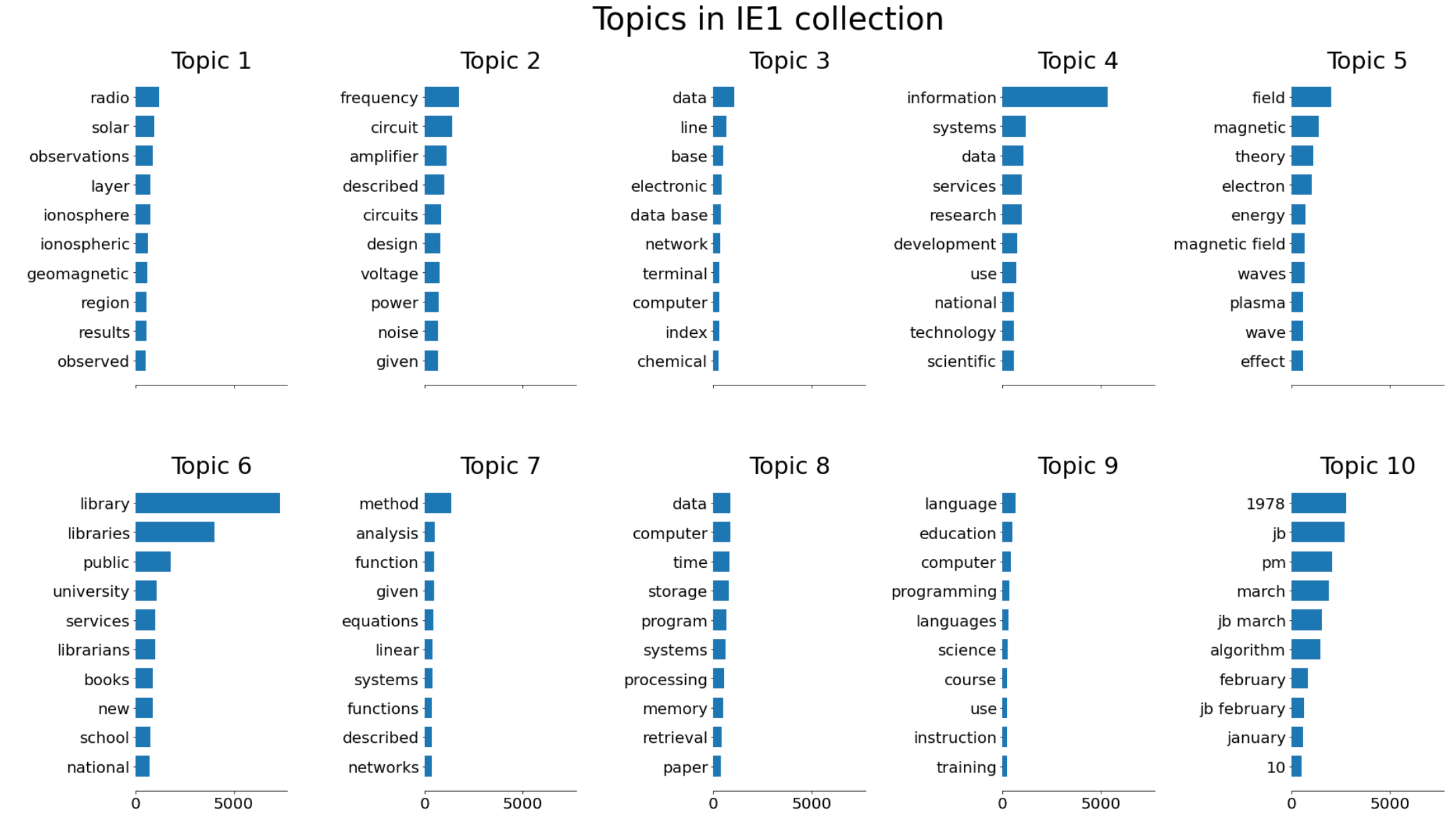
Ermöglicht uns kleinere Codeteile/Cells zu schreiben, zu dokumentieren und direkt zu testen, ohne das ganze Programm und die Datenverarbeitung neuzustarten.

## Sklearn

Eine ML-Bibliothek, die verschiedene Klassifikations-/Regressions- und Clusteringalgorithmen beinhaltet. Wir haben folgende verwendet:

* CountVectorizer: <add short desc. what it is used for>
* TfIdfVectorizer: <add short desc. what it is used for>
* LatentDirichletAllocation: <add short desc. what is is used for>
* pairwise\_distance: <add short desc. what is is used for>

# Verfahren

1. Lesen und Parsen der Inputfiles in ein Dictionary
2. Das System trainieren: Die Dokumente werden mit Hilfe eines Vectorizers (Count-oder TfIdf-Vecorizer) in eine Matrix transformiert. Dabei werden zuerst Stoppwörter eliminiert und die restlichen Wörter, je nach Verfahren, anders gewichtet bzw. ein Score gegeben. Ausserdem können wir steuern, wie viele Wörter/Feautures wir gewichten wollen.
3. Aus diesen Scores kann das System mittels LatentDirichletAllocation (LDA) die Dokumente in n-Topics gruppieren. Das sieht dann so aus:  
     
   Die Wörter auf der Y-Achse sind die Features, sortiert nach dem Score, der im Schritt 2 vergeben wurde.
4. Die Dokumente werden dann den Topics zugewiesen und erhalten einen Score. Je höher der Score, desto eher passt das Dokument zum Topic
5. Das gleiche Verfahren in Schritt 2 & 3 wird dann auch für die Queries verwendet. Das heisst jede Query wird auch einem Topic zugewiesen.
6. Im letzten Schritt sucht das System für jede Query, mit Hilfe des Topics, nach möglichen dazugehörigen Dokumenten und berechnet die Distanz zwischen Query und Dokument. Je kleiner die Distanz desto eher passt die Query zum Dokument.

# Verwendeter Index

Kei ahnig was er da wot? Eventuell eifach sverfahre erkläre?

# Resultate

Chasch da eifach churz ineschribe was du vo de Resultat haltisch und obs grossi unterschied geh het mit Count oder Tf Vecorizer?

## Mit Count Vecorizer

## Mit Tf-Idf Vecorizer