Learning

Friday, July 26, 2019 8:07 PM

Lesson 1.1: Natural Language Content Analysis21m

Lesson 1.2: Text Access9m

Lesson 1.3: Text Retrieval Problem26m

Lesson 1.4: Overview of Text Retrieval Methods10m

Lesson 1.5: Vector Space Model - Basic Idea9m

Lesson 1.6: Vector Space Retrieval Model - Simplest Instantiation17m

From <https://www.coursera.org/learn/text-retrieval>

Lesson 2.1: Vector Space Model - Improved Instantiation16m

Lesson 2.2: TF Transformation9m

Lesson 2.3: Doc Length Normalization18m

Lesson 2.4: Implementation of TR Systems21m

Lesson 2.5: System Implementation - Inverted Index Construction18m

Lesson 2.6: System Implementation - Fast Search17m

From <https://www.coursera.org/learn/text-retrieval>

Lesson 3.1: Evaluation of TR Systems10m

Lesson 3.2: Evaluation of TR Systems - Basic Measures12m

Lesson 3.3: Evaluation of TR Systems - Evaluating Ranked Lists - Part 115m

Lesson 3.4: Evaluation of TR Systems - Evaluating Ranked Lists - Part 210m

Lesson 3.5: Evaluation of TR Systems - Multi-Level Judgements10m

Lesson 3.6: Evaluation of TR Systems - Practical Issues15m

From <https://www.coursera.org/learn/text-retrieval>

Lesson 4.1: Probabilistic Retrieval Model - Basic Idea12m

Lesson 4.2: Statistical Language Model17m

Lesson 4.3: Query Likelihood Retrieval Function12m

Lesson 4.4: Statistical Language Model - Part 112m

Lesson 4.5: Statistical Language Model - Part 29m

Lesson 4.6: Smoothing Methods - Part 19m

Lesson 4.7: Smoothing Methods - Part 213m

From <https://www.coursera.org/learn/text-retrieval>

Lesson 5.1: Feedback in Text Retrieval6m

Lesson 5.2: Feedback in Vector Space Model - Rocchio12m

Lesson 5.3: Feedback in Text Retrieval - Feedback in LM19m

Lesson 5.4: Web Search: Introduction & Web Crawler11m

Lesson 5.5: Web Indexing17m

Lesson 5.6: Link Analysis - Part 19m

Lesson 5.7: Link Analysis - Part 217m

Lesson 5.8: Link Analysis - Part 35m

From <https://www.coursera.org/learn/text-retrieval>

Lesson 6.1: Learning to Rank - Part 15m

Lesson 6.2: Learning to Rank - Part 210m

Lesson 6.3: Learning to Rank - Part 34m

Lesson 6.4: Future of Web Search13m

Lesson 6.5: Recommender Systems: Content-Based Filtering - Part 112m
Lesson 6.6: Recommender Systems: Content-Based Filtering - Part 210m
Lesson 6.7: Recommender Systems: Collaborative Filtering - Part 16m
Lesson 6.8: Recommender Systems: Collaborative Filtering - Part 212m
Lesson 6.9: Recommender Systems: Collaborative Filtering - Part 34m
Lesson 6.10: Course Summary9m

From <https://www.coursera.org/learn/text-retrieval>

Monday, July 29, 2019 5:23 PM

Create sentence dictionary from the Tax chapters
Create sentence vectors for train dataset and train model based on this
Create sentence vectors for test dataset and find the output from there

Code review

Sunday, August 11, 2019 12:37 AM

Comments :

1. Test dataset to drive the entire exercise?
2. In the below block, code and comments aren't in synch:
   ```
   #Drop the newly added column
   stack = (test["split"].apply(pd.Series).stack())
   #Converting the ID's to Numeric Data
   test["qid"] = pd.to_numeric(stack[:,0])
   test["ParaId"] = pd.to_numeric(stack[:,1])
   test = test.drop(['split'], axis =1)
   ```
3. Just taking the test dataset may skew the results
4. Why the ngrams are limited to 2 only
5. Comments don't tell any functional purpose of the code block which makes it difficult to understand
6. Didn't use any pre-trained models to embed words or sentences, does she have idea around the same?
7. How did she arrive at the priority in sorting?
8. Why didn't use the train dataset?