LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Eagle Eye

Sommersemester 2017
Big Data Science Praktikum

©Zhenyu Chen - Wentao Hua - Guoliang Xue - Bernhard Fabry - Daly

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Agenda

- Brief Introduction
- Pre-processiong of dataset
- Front-end Design
- Back-end Design (BM25)
- Synonym implementation
- Naive Bayes Implementation
- Improvement

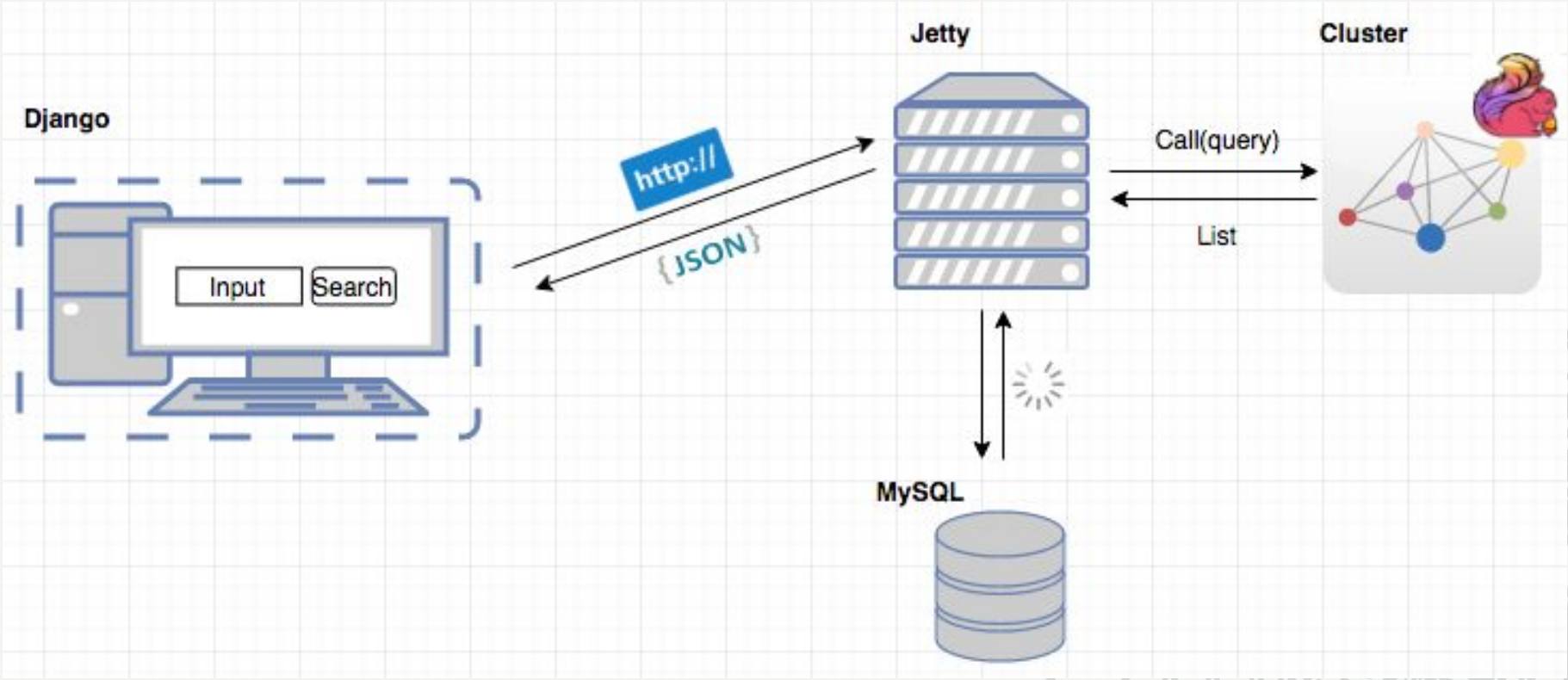
Techniques

Jersey

RESTful Web Services in Java.

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Architecture



LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

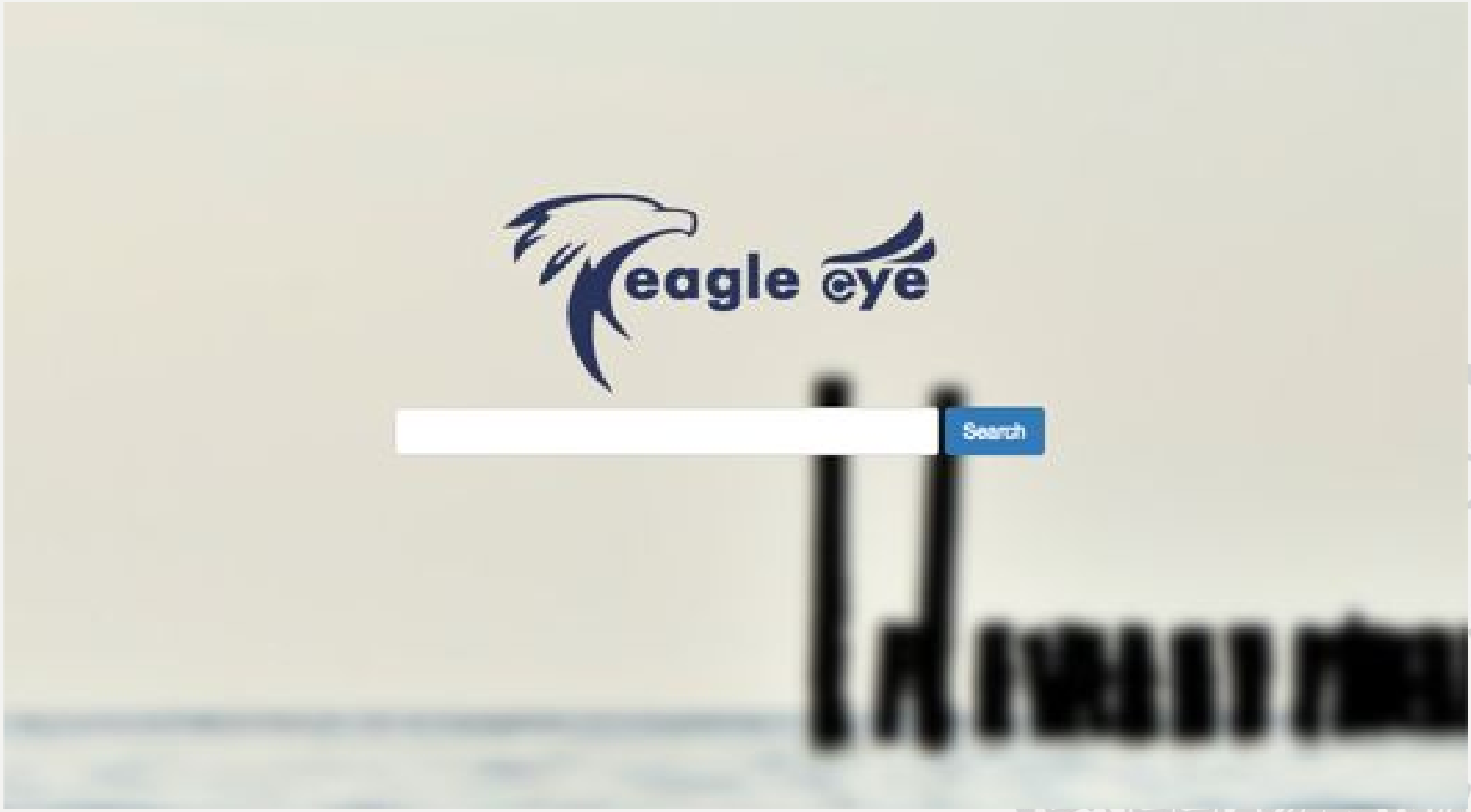
Data Preprocessing

- "xml.etree.ElementTree" module
- divide and save them into separated CSV-extension file respectively by "PostTypeId"(1 means question, 2 means answer)
- Regular Expression filter
- CSV extension files(question, answer)

	Id	Title	Body	CreationDate
0	1	Comments are a code smell	A coworker of mine believes that any use of...	2010-09-01T19:34:48.000
1	4	Getting non-programmers to understand the deve...	When starting a project for a company that's ...	2010-09-01T19:37:39.957
2	9	Hor		
3	16	Do		
4	18	Wh		

	Id	ParentId	CreationDate	Body	OwnerUserId
0	3	1	2010-09-01T19:36:50.053	Ideally code should be so well coded that it...	11
1	7	1	2010-09-01T19:42:16.797	I think the answer is the usual "It depends" ...	21
2	12	4	2010-09-01T19:44:47.413	IMO I've found that the transparency offered...	21
3	13	1	2010-09-01T19:45:33.183	Only if the comment describes what the code i...	4
4	20	9	2010-09-01T19:48:20.170	Stress. I give in to temptation and surf Stac...	6

Frontend

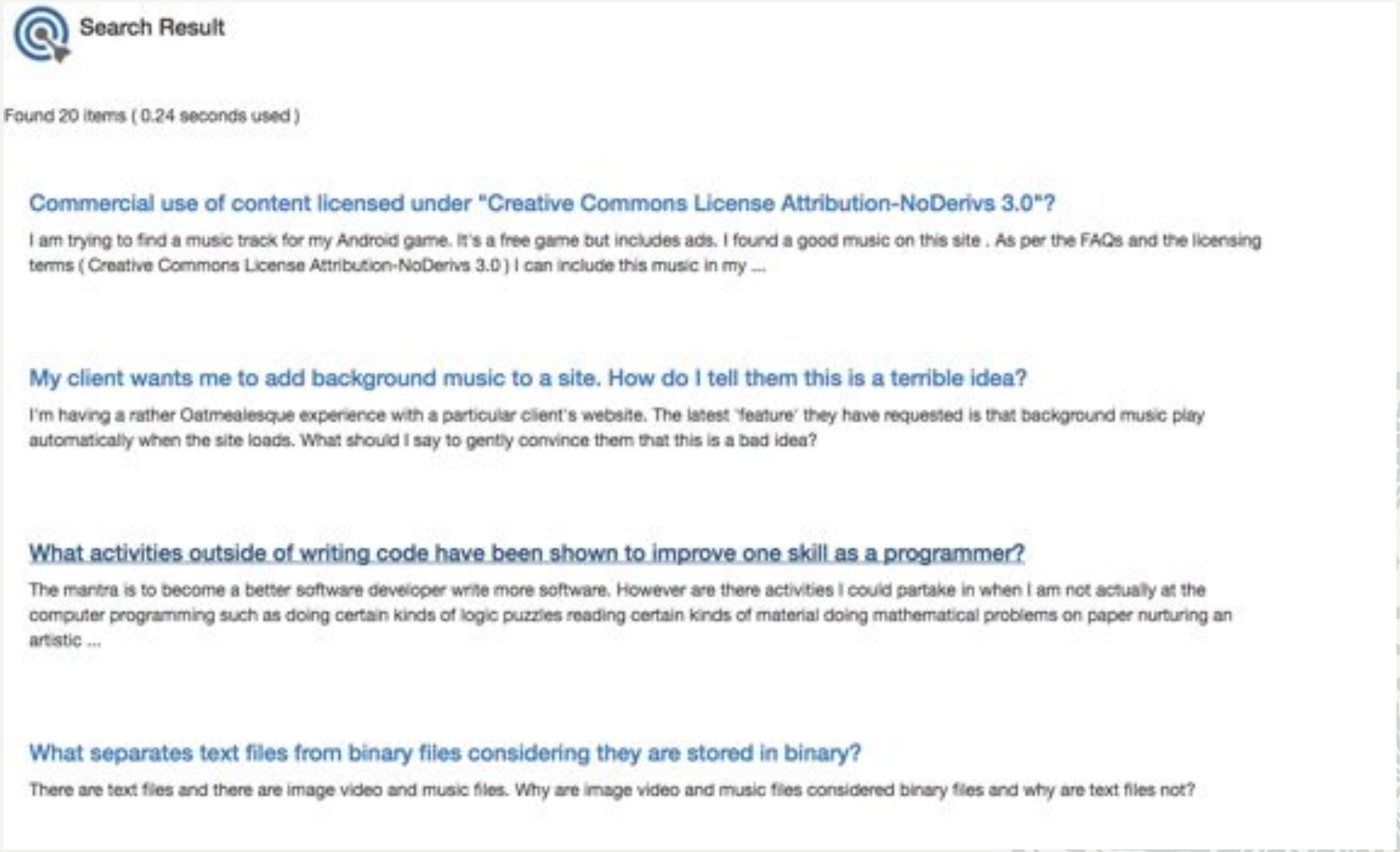


Features

- real-time suggestion
- spelling correction

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

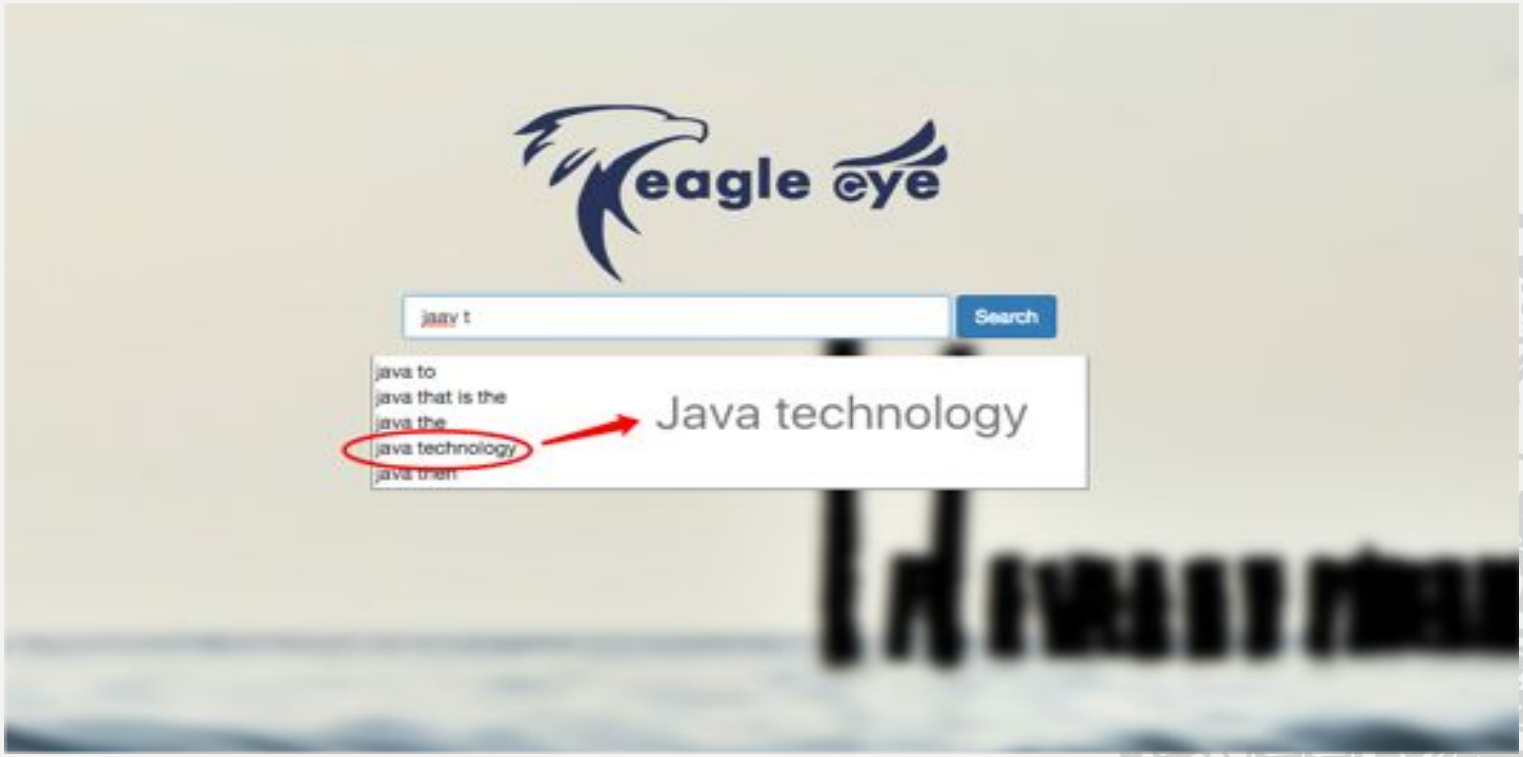
Search Result



LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Real-time Suggestion

When user input search query in the webpage, system will give the suggestion below the input box. This is a kind of prediction, System will predict the whole query according to the prefix.



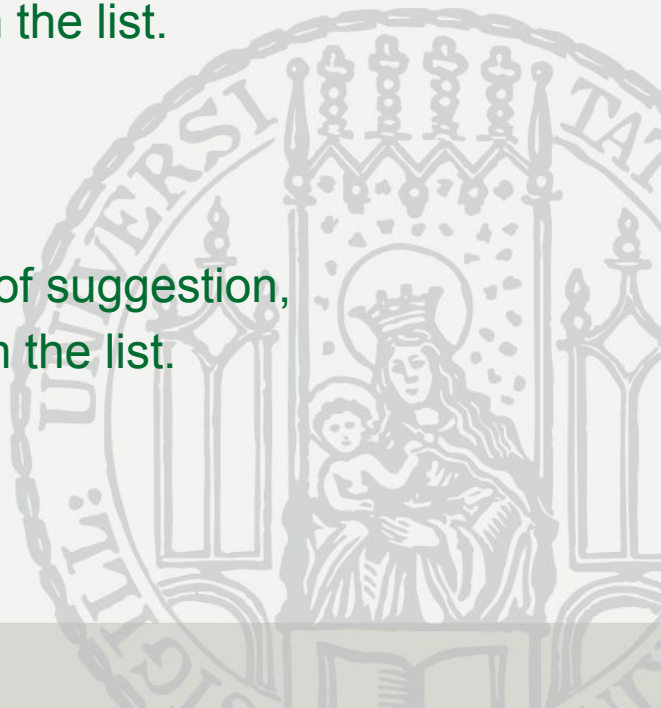
LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Realtime Suggestion

- **Unigram (N-gram n=1)**
  input: ja, use prefix to find suggestion

- **Bigram (N-gram n=2)**
  input: jaav to
  first spelling correct, use **java** to find a list of suggestion,
  then use **to** as prefix to find suggestion from the list.

- **Trigram (N-gram n=3)**
  input: java and sc
  first spelling correct, use **java and** find a list of suggestion,
  then use **sc** as prefix to find suggestion from the list.



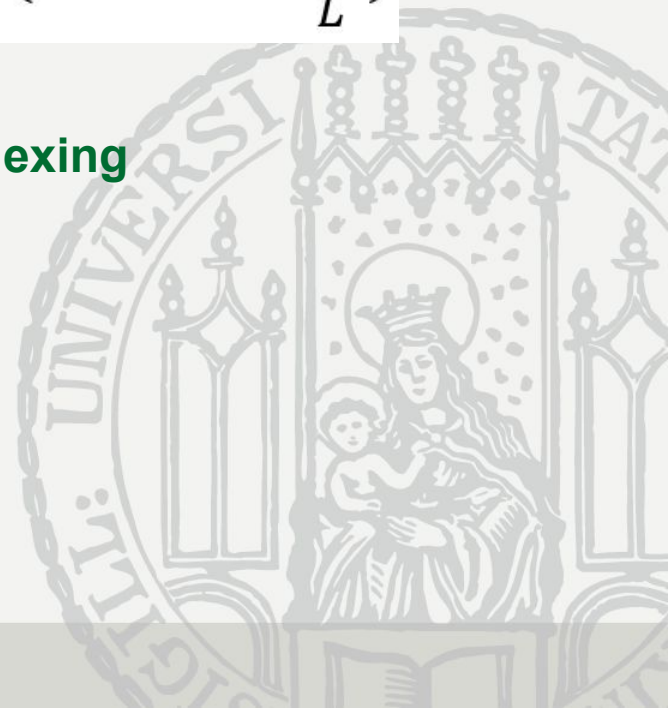
LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

BM25 Scoring Model

- **Implementation of BM25 with Java Flink**

$$\text{score}(\text{D}, \text{Q}) = \sum_{i=1}^{n} IDF(q_i) \frac{TF(q_i, D) \cdot (k_1 + 1)}{TF(q_i, D) + k_1 \cdot (1 - b + b \cdot \frac{|D|}{\bar{L}})}$$

- **IDF and TF are available with the help of Indexing**
- **Free parameters remain fixed**
- **Document length added to the index**



LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Eagle Eye vs. Google&Bing

Google

Document ID	Google rank	Eagle Eye rank
304445	1	3302
90203	2	3842
254279	3	82
267846	4	-
287800	5	181
301114	6	1097
212593	7	527
305956	8	2190
131995	9	3064
254984	10	3701

Bing

Document ID	Bing rank	Eagle Eye rank
304445	1	3302
90203	2	3842
137172	3	1233
304579	4	-
267846	5	-
279216	6	1445
125275	7	1171
58998	8	377
348984	9	-
206860	10	2425

Problems

Bottleneck

- The performance of BM25 scoring model is unsatisfactory
- The time to perform a query is too long
- Unexpected I/O exceptions

Solution

- Use a new model to calculate the score of ranking
- Distribution using Apache Flink on server

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

The New Scoring Model

- Add new features to the previous model

$$score = \sum_{i=1}^{N} q_i f_i$$

- Each value of features is multiplied by a weight
- Features are:
  - BM25 score
  - Quality of questions
  - Quality of answers
  - Other factors
- Use machine learning method to train the model

Training The New Model

Construction of training data

- Query 45 questions in Google and get ID of top 10 results
- Identify noises and delete them manually
- Get the training matrix of 342*10

Train the model

- A task of solving linear regression

$$\vec{y} = X\vec{\beta}$$

- Use Ordinary Least Square to solve the equation

$$\hat{\beta} = (X^T X)^{-1} X^T \vec{y}$$

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Performance Of The New Model

Document ID	Google rank	Eagle Eye rank (before opt.)	Eagle Eye rank (after opt.)	Comparison
307805	0	7	3	4↑
219320	1	1465	1448	17↑
332017	2	4098	4093	5↑
260207	3	2393	2387	6↑
270891	4	2669	2666	3↑
230060	5	24	22	2↑
238896	6	-1	-1	-
131397	7	51	51	-
275631	8	841	833	8↑
264598	9	338	339	1↓

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Ontology extraction

Idea:
The query the user generated might not deliver exactly what he wanted

Goal:
Finding exactly what the user is looking for

Achieved by:
Extraction of synonyms out of an automatically constructed ontology

“Cut my text into pieces”

Check for synonyms in ontology

“Split my string into tokens”

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Definition

What is an ontology:

An ontology is a set of concepts and categories *in a subject area or domain* that shows their properties and the relations between them

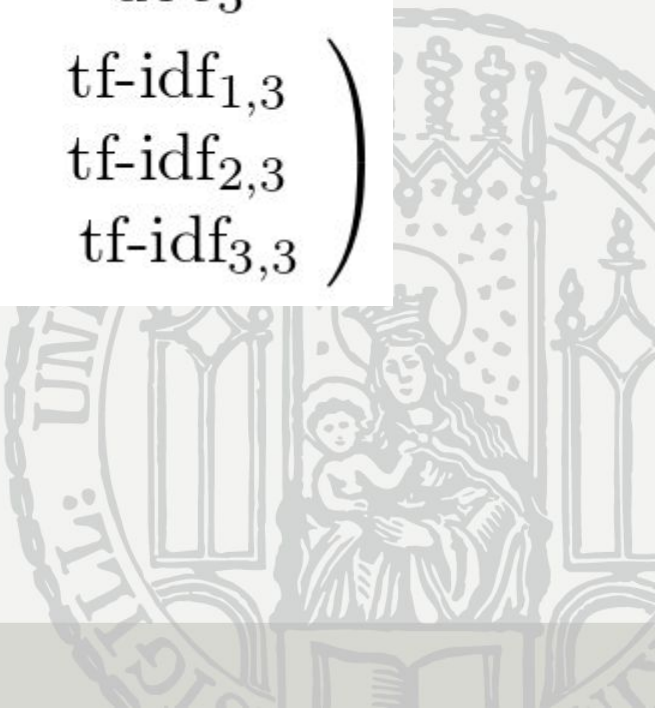
What we achieved:

Automatically constructed set of concepts in a domain containing terms and their individual degree of affection to this concept

Preparation Of Data

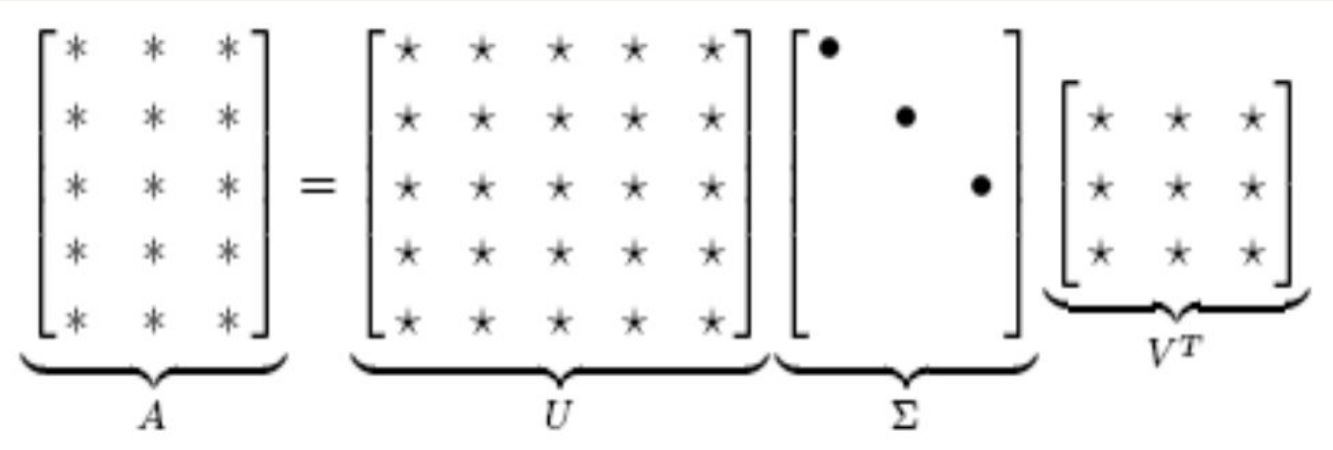
Create document-term-matrix with corresponding tf-idf as entries

$$\mathbf{M} = \begin{array}{c} \\ \text{term}_1 \\ \text{term}_2 \\ \text{term}_3 \end{array} \begin{array}{c} \begin{array}{ccc} \text{doc}_1 & \text{doc}_2 & \text{doc}_3 \end{array} \\ \begin{pmatrix} \text{tf-idf}_{1,1} & \text{tf-idf}_{1,2} & \text{tf-idf}_{1,3} \\ \text{tf-idf}_{2,1} & \text{tf-idf}_{2,2} & \text{tf-idf}_{2,3} \\ \text{tf-idf}_{3,1} & \text{tf-idf}_{3,2} & \text{tf-idf}_{3,3} \end{pmatrix} \end{array}$$



Singular Value Decomposition

Create the Singular Value Decomposition of this matrix



Further Processing

For further computation we only need the matrix *U*

$$\mathbf{U} = \begin{array}{c} \\ \text{term}_1 \\ \text{term}_2 \\ \text{term}_3 \end{array} \begin{array}{c} \begin{array}{ccc} \text{concept}_1 & \text{concept}_2 & \text{concept}_3 \end{array} \\ \begin{pmatrix} \text{affection}_{1,1} & \text{affection}_{1,2} & \text{affection}_{1,3} \\ \text{affection}_{2,1} & \text{affection}_{2,2} & \text{affection}_{2,3} \\ \text{affection}_{3,1} & \text{affection}_{3,2} & \text{affection}_{3,3} \end{pmatrix} \end{array}$$

Result

Name the concepts by concatenating the three most affected terms and write them to a .csv file

server-data-web	2	view	439	0.065098348890849 53
server-data-web	2	client	65	0.091679783213342 01
server-data-web	2	api	16	0.078033518556892 35
server-data-web	2	web	442	0.104308347392825 64
server-data-web	2	server	350	0.116936975743107 03
server-data-web	2	database	97	0.081733407487163 6
server-data-web	2	application	18	0.060723476772032 9
server-data-web	2	table	393	0.067348646949379 36
server-data-web	2	model	258	0.095219656646997 37
server-data-web	2	data	96	0.109238335949646 71
server-data-web	2	service	351	0.087962006245238 03
server-data-web	2	user	431	0.085894680419986 67

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Problems

- **Implementation of SVD in flink:**
  - Proper indexing of terms and documents
  - Representing/creating “0” entries in the matrix
  - Understanding the algorithms and concepts used
  - Results were crooked matrices and strange values
- **Switching to external library**
  - Uses double [ ][ ] to represent a matrix
  - Very high memory consumption
  - Very time consuming
- **No real synonyms were found**
  - Writing style in forums differs from scientific papers
  - Papers have more textbody

Naïve Bayes For Stack Overflow Questions

- Classes: android, ios, java, jquery, javascript, html, c#, php, python, c++, c

- Observations: Stack Overflow programming language questions

Naïve Bayes For Stack Overflow Questions

Goals:

- Enlarge the search engine by predicting programming language tags of the new Dataset
- Reduce time response

Main Steps

- **Dataset Preprocessing**
- **Training of the program :**

$$\Pr oglang^{*} = \arg\max_{\Pr oglang} [\sum_{i} (\log P(W_i \mid \Pr oglang)) + \log P(\Pr oglang)]$$

- **Test on another dataset and calculate the accuracy**

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Results

	No Merge	Merge in Training Phase	Merge in Test Phase	Merge in both Phases
Use 1st Part of the Dataset for training and 3rd part for Testing	37,71%	37,07%	41,44%	40,73%
Use 3rd part of the Dataset for training and 1stpart for Testing	42,53%	42,07%	46,61%	46,40%
Use 10 first parts for training and 11th and 13th for testing	34,76%	33,51%	38.90%	37,73%
Use 11th and 13th parts for training and 10 first parts for testing	47,46%	47,2%	55,65%	55,59%

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Improvement

- Improve user experience, response time should be less
- Work on the whole data dump of StackExchange
- Integrate the Naive, Synonyms algorithm into Eagle Eye project

DEMO

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Thank you for your attention