Semantic Search (Part III)

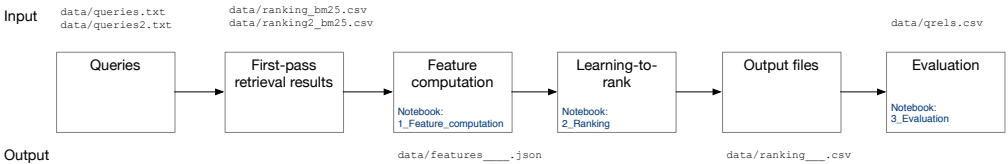
[DAT640] Information Retrieval and Text Mining

Krisztian Balog
University of Stavanger

October 21, 2019

Assignment 2B

Overview



- **Scenario 1**: The model is trained using cross-validation, that is on 4/5 of queries, then applied on the remaining 1/5 of queries (repeated 5 times)
- **Scenario 2**: The model is trained on all available training data

Search API


You
Search API
(running on the UiS
Unix network)
Elasticsearch
(hosted on the cloud)

Discussion

Question

Why should we consider the first-pass retrieval results when computing features and learning the model?

Entity retrieval

Recap

- Ad hoc entity retrieval
  - Given a keyword query, return a ranked list of entities from an entity catalog (knowledge base)
  - Idea: Construct term-based representations of entities (entity description documents), which can then be ranked the same way as documents
  - Specific techniques: catch-all field, predicate folding, URI resolution

Example

Name	Audi A4
Name variants	Audi A4 ... Audi A4 Allroad
Attributes	The Audi A4 is a compact executive car produced since late 1994 by the German car manufacturer Audi, a subsidiary of the Volkswagen Group [...] ... 1996 ... 2002 ... 2005 ... 2007
Types	Product ... Front wheel drive vehicles ... Compact executive cars ... All wheel drive vehicles
Outgoing relations	Volkswagen Passat (B5) ... Audi 80
Incoming relations	Audi A5
`<foaf:name>`	Audi A4
`<dbo:abstract>`	The Audi A4 is a compact executive car produced since late 1994 by the German car manufacturer Audi, a subsidiary of the Volkswagen Group [...]
Catch-all	Audi A4 ... Audi A4 ... Audi A4 Allroad ... The Audi A4 is a compact executive car produced since late 1994 by the German car manufacturer Audi, a subsidiary of the Volkswagen Group [...] ... 1996 ... 2002 ... 2005 ... 2007 ... Product ... Front wheel drive vehicles ... Compact executive cars ... All wheel drive vehicles ... Volkswagen Passat (B5) ... Audi 80 ... Audi A5

Ranking term-based entity representations

Overview

- Unstructured retrieval models
  - LM, BM25, **SDM**
- Fielded retrieval models
  - MLM, BM25F, **PRMS**, **FSDM**

Mixture of Language Models (MLM)

- Idea: Build a separate language model for each field, then take a linear combination of them

$$P(t|\theta_d) = \sum_i w_i P(t|\theta_{d_i})$$

- where
  - i corresponds to the field index
  - w_i is the field weight (such that $\sum_i w_i = 1$)
  - $P(t|\theta_{d_i})$ is the field language model

Probabilistic Retrieval Model for Semistructured data (PRMS)

- Extension to MLM for dynamic field weighting
- To key ideas
  - Instead of using a fixed (static) field weight for all terms, field weights are determined dynamically on a term-by-term basis
  - Field weights can be established based on the term distributions of the respective fields
- Replace the static weight w_i with a *mapping probability* $P(f|t)$

$$P(t|\theta_d) = \sum_f P(f|t)P(t|\theta_{d_f})$$

  - Note: we now use field f instead of index i when referring to fields

Estimating the mapping probability

- By applying Bayes' theorem and using the law of total probability:

$$P(f|t) = \frac{P(t|f)P(f)}{P(t)} = \frac{P(t|f)P(f)}{\sum_{f' \in \mathcal{F}} P(t|f')P(f')}$$

- where
  - $P(f)$ is a prior that can be used to incorporate, for example, domain-specific background knowledge, or left to be uniform
  - $P(t|f)$ is conveniently estimated using the background language model of that field $P(t|C_f)$

Example

$t =$ ``Meg"		$t =$ ``Ryan"		$t =$ ``war"		$t =$ ``redemption"	
f	$P(f\|t)$	f	$P(f\|t)$	f	$P(f\|t)$	f	$P(f\|t)$
cast	0.407	cast	0.601	genre	0.927	title	0.983
team	0.381	team	0.381	title	0.070	location	0.017
title	0.187	title	0.017	location	0.002	year	0.000

Table: Example mapping probabilities computed on the IMDB collection, taken from Kim et al., 2009.

Exercise #0

- Getting term probabilities from Elasticsearch
- Code skeleton on GitHub: `exercises/lecture_15/exercise_0.ipynb` (make a local copy)

Exercise #1

- Implementing PRMS
- Code skeleton on GitHub: `exercises/lecture_15/exercise_1.ipynb` (make a local copy)

Reading

- Entity-Oriented Search (Balog)[1]
  - Chapter 3

[1] PDF: https://rd.springer.com/content/pdf/10.1007%2F978-3-319-93935-3.pdf