Fielded Sequential Dependence Model for Ad-Hoc Entity Retrieval in the Web of Data

Nikita Zhiltsov [1,2] Alexander Kotov [3] Fedor Nikolaev [3]

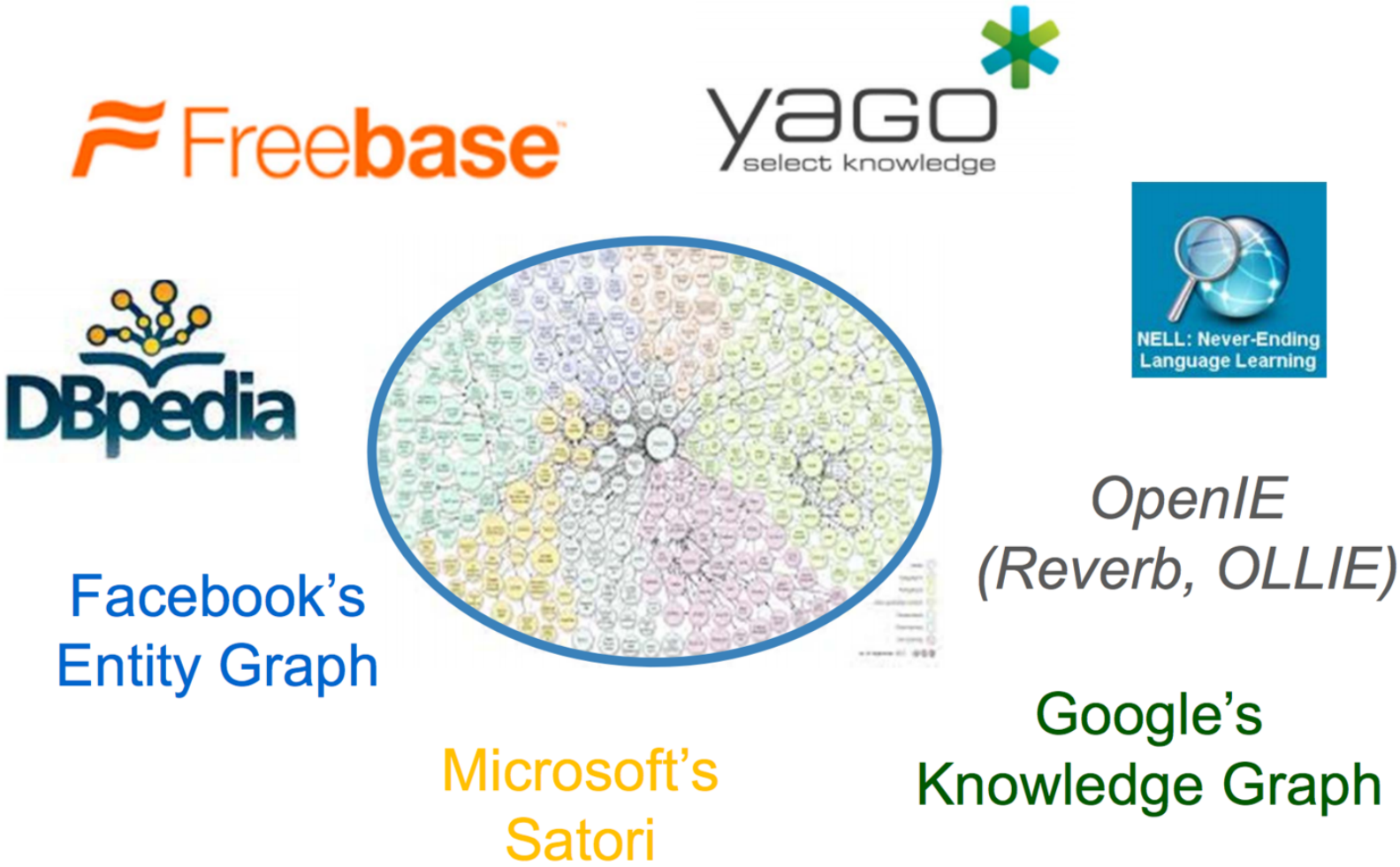
[1]Kazan Federal University

[2]Textocat

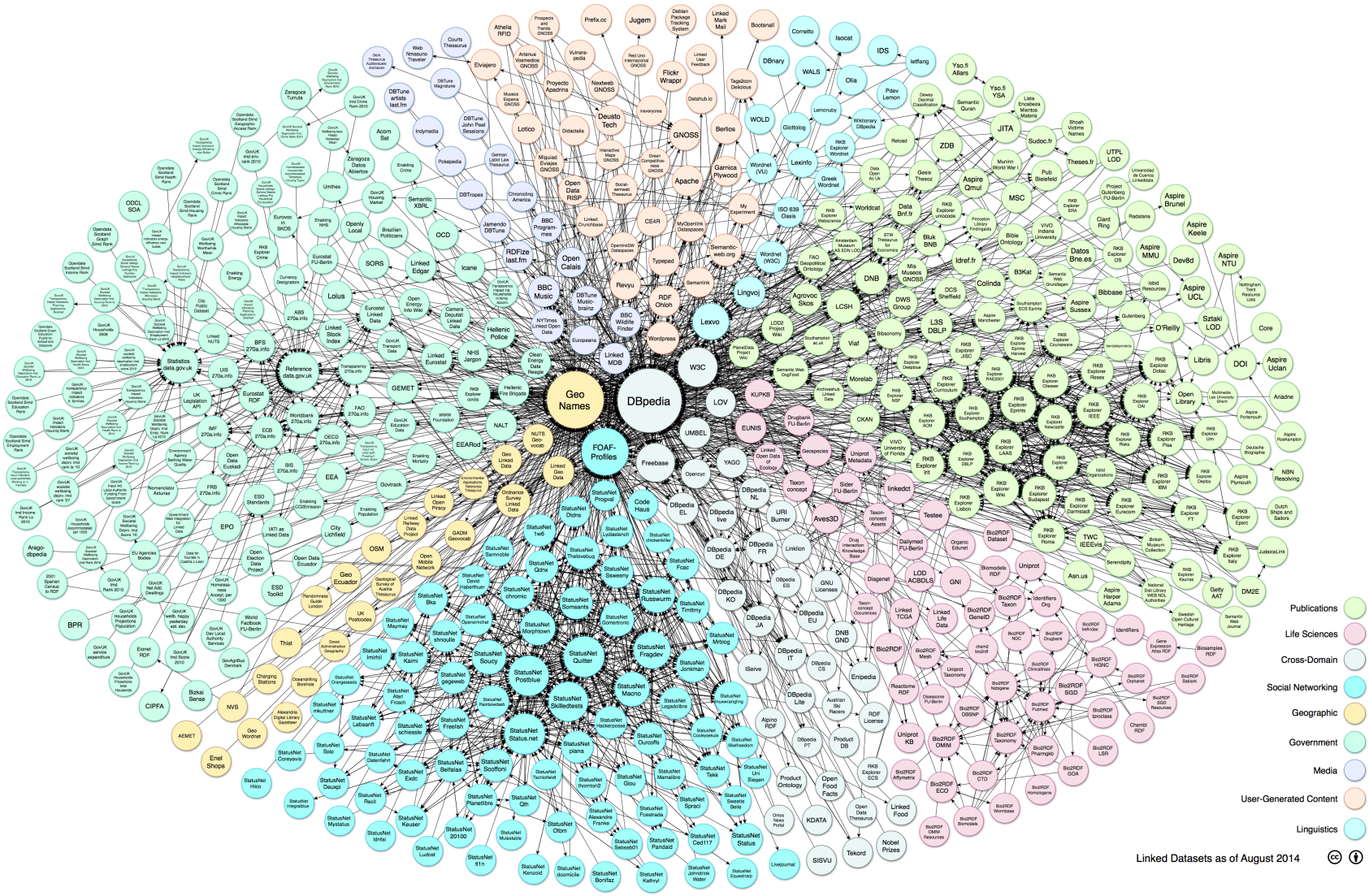
[3]Textual Data Analytics Lab, Department of Computer Science, Wayne State University

Overview

Knowledge Graphs



Linked Open Data (LOD) Cloud



Entities

- Material objects or concepts in the real world or fiction (e.g. people, movies, conferences etc.)
- Are connected with other entities by *relations* (e.g. hasGenre, actedIn, isPCmemberOf etc.)
- Subject-Predicate-Object (SPO) triple: subject=entity; object=entity (or primitive data value); predicate=relationship between subject and object
- Many SPO triples $\rightarrow$ *knowledge graph*

DBPedia entity page example

About: Barack Obama

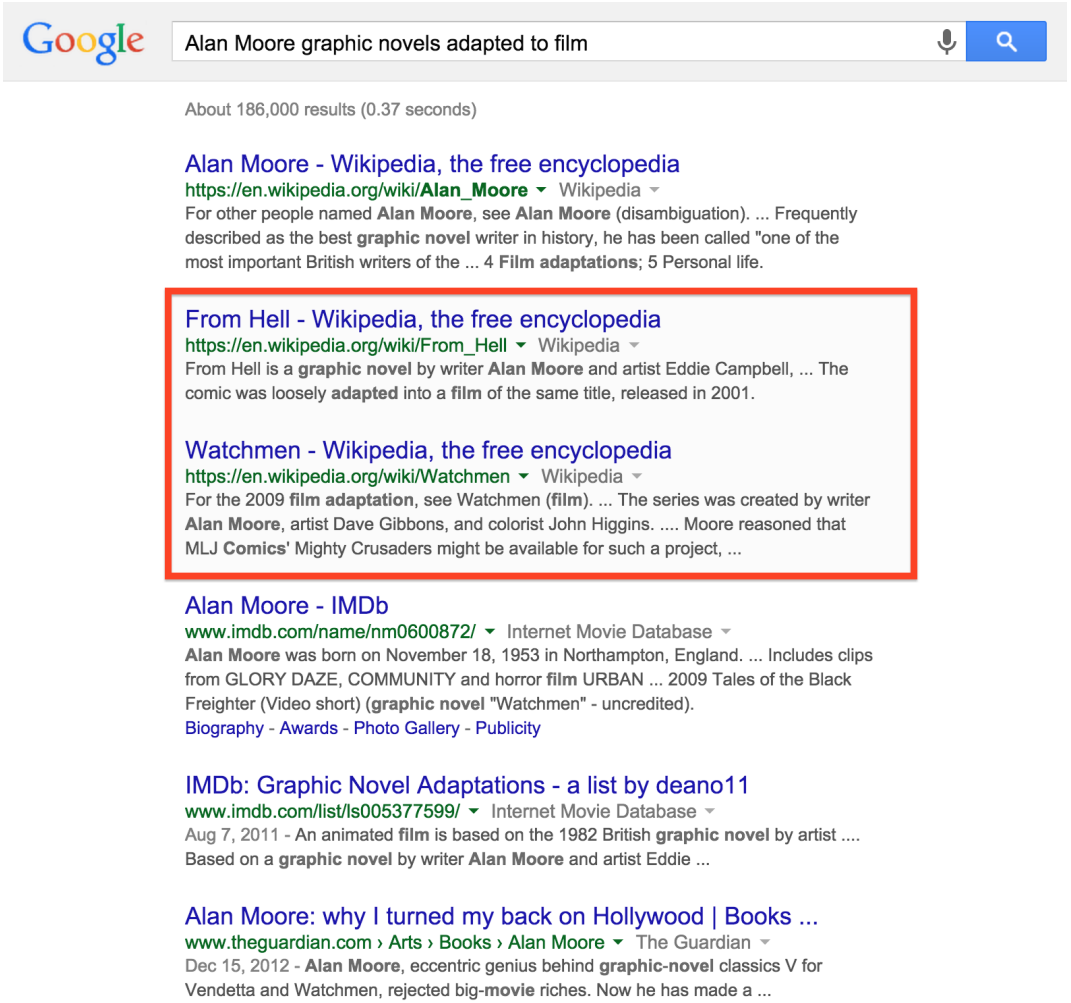
An Entity of Type : office holder, from Named Graph : http://dbpedia.org, within Data Space : dbpedia.org

Barack Hussein Obama II (/bəˈrɑːk huːˈseɪn oʊˈbɑːmə/; born August 4, 1961) is the 44th and current President of the United States, and the first African American to hold the office. Born in Honolulu, Hawaii, Obama is a graduate of Columbia University and Harvard Law School, where he served as president of the Harvard Law Review. He was a community organizer in Chicago before earning his law degree.

rdfs:comment	▪ Barack Hussein Obama II (/bəˈrɑːk huːˈseɪn oʊˈbɑːmə/; born August 4, 1961) is the 44th and current President of the United States, and the first African American to hold the office. Born in Honolulu, Hawaii, Obama is a graduate of Columbia University and Harvard Law School, where he served as president of the Harvard Law Review. He was a community organizer in Chicago before earning his law degree. ▪ Barack Hussein Obama II [bəˈɹɑːk hʊˈseɪn oʊˈbɑːmə] (* 4. August 1961 in Honolulu, Hawaii) ist seit dem 20. Januar 2009 der 44. Präsident der Vereinigten Staaten. Er ist der erste Afroamerikaner in diesem Amt und wurde 2012 für eine zweite Amtsperiode wiedergewählt. Am 10.
rdfs:label	▪ Barack Obama ▪ Barack Obama
is dbpedia-owl:author of	▪ dbpedia:Dreams_from_My_Father ▪ dbpedia:The_Audacity_of_Hope ▪ dbpedia:Of_Thee_I_Sing_(book)
is dbpedia-owl:child of	▪ dbpedia:Ann_Dunham ▪ dbpedia:Lolo_Soetoro ▪ dbpedia:Barack_Obama,_Sr.
dc:description	▪ American politician, 44th President of the United States ▪ American politician, 44th President of the United States
dcterms:subject	▪ category:African-American_academics ▪ category:American_civil_rights_lawyers ▪ category:Illinois_State_Senators ▪ category:United_Church_of_Christ_members ▪ category:1961_births ▪ category:20th-century_American_writers ▪ category:21st-century_American_writers ▪ category:African-American_Christians

ENTITY RETRIEVAL FROM KNOWLEDGE GRAPH(S)

- Graph KBs are perfectly suited for addressing the information needs that aim at finding specific objects (entities) rather than documents
- Given the user's information need expressed as a keyword query, retrieve *a relevant set of objects from the knowledge graph(s)*



Typical ERWD tasks

- **Entity Search**
  Queries refer to a particular entity.
  - *"Ben Franklin"*
  - *"England football player highest paid"*
  - *"Einstein Relativity theory"*
- **List Search**
  Complex queries with several relevant entities.
  - *"US presidents since 1960"*
  - *"animals lay eggs mammals"*
- **Question Answering**
  Queries are questions in natural language.
  - *"Who is the mayor of Santiago?"*
  - *"For which label did Elvis record his first album?"*

Fundamental problems in ERWD

- Designing effective and concise entity representations
  - Pound, Mika et al. Ad-hoc Object Retrieval in the Web of Data, WWW'10
  - Blanco, Mika et al. Effective and Efficient Entity Search in RDF Data, ISWC'11
  - Neumayer, Balog et al. On the Modeling of Entities for Ad-hoc Entity Search in the Web of Data, ECIR'12
- Developing accurate retrieval models
  - Mostly adaptations of standard unigram bag-of-words retrieval models, such as BM25F, MLM

Overview

Entity document

An entity is represented as a structured (multi-fielded) document:

names
: Conventional names of the entities, such as the name of a person or the name of an organization

attributes
: All entity properties, other than names

categories
: Classes or groups, to which the entity has been assigned

similar entity names
: Names of the entities that are very similar or identical to a given entity

related entity names
: Names of the entities that are part of the same RDF triple

Entity document example

Multi-fielded entity document for the entity *Barack Obama*.

Field	**Content**
names	barack obama barack hussein obama ii
attributes	44th current president united states birth place honolulu hawaii
categories	democratic party united states senator nobel peace prize laureate christian
similar entity names	barack obama jr barak hussein obama barack h obama ii
related entity names	spouse michelle obama illinois state predecessor george walker bush

Overview

Motivation

Previous research in ad-hoc IR has focused on two major directions:

- unigram bag-of-words retrieval models for multi-fielded documents
  - Ogilvie and Callan. Combining Document Representations for Known-item Search, SIGIR'03
  - Robertson et al. Simple BM25 Extension to Multiple Weighted Fields, CIKM'04
- retrieval models incorporating term dependencies
  - Metzler and Croft. A Markov Random Field Model for Term Dependencies, SIGIR'05
  - Huston and Croft. A Comparison of Retrieval Models using Term Dependencies, CIKM'14

Goal: to develop a retrieval model that captures both document structure and term dependencies

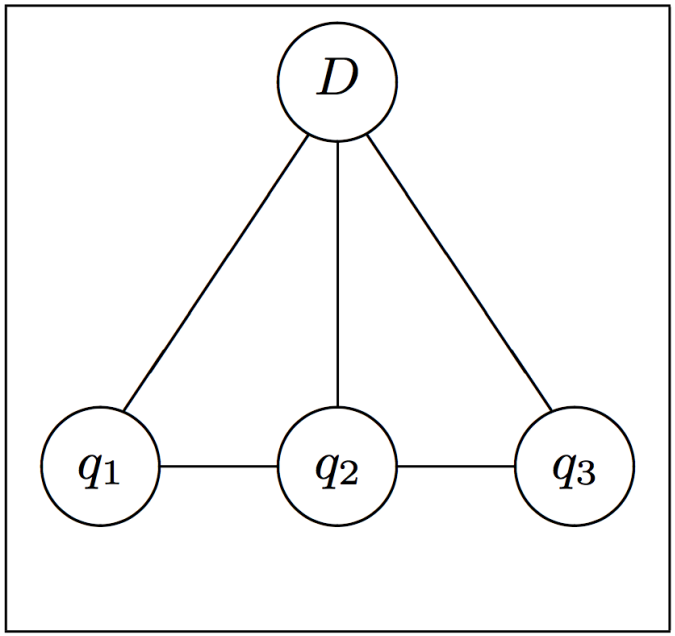
MLM

$$P(Q|D) \overset{rank}{=} \prod_{q_i \in Q} P(q_i|\theta_D)^{tf(q_i)},$$

where

$$P(q_i|\theta_D) = \sum_j w_j P(q_i|\theta_D^j)$$

SDM



Ranks w.r.t. $P_\Lambda(D|Q) = \sum_{i \in \{T,U,O\}} \lambda_i f_i(Q,D)$
Potential function for unigrams is QL:

$$f_T(q_i, D) = \log P(q_i|\theta_D) = \log \frac{tf_{q_i,D} + \mu \frac{cf_{q_i}}{|C|}}{|D| + \mu}$$

FSDM ranking function

FSDM incorporates document structure and term dependencies with the following ranking function:

$$P_\Lambda(D|Q) \overset{rank}{=} \lambda_T \sum_{q \in Q} \tilde{f}_T(q_i, D) + \lambda_O \sum_{q \in Q} \tilde{f}_O(q_i, q_{i+1}, D) + \lambda_U \sum_{q \in Q} \tilde{f}_U(q_i, q_{i+1}, D)$$

Separate MLMs for bigrams and unigrams give FSDM the flexibility to adjust the document scoring depending on the query type

MLM is a special case of FSDM, when $\lambda_T = 1, \lambda_O = 0, \lambda_U = 0$

FSDM ranking function

FSDM incorporates document structure and term dependencies with the following ranking function:

$$P_\Lambda(D|Q) \overset{rank}{=} \lambda_T \sum_{q \in Q} \tilde{f}_T(q_i, D) + \lambda_O \sum_{q \in Q} \tilde{f}_O(q_i, q_{i+1}, D) + \lambda_U \sum_{q \in Q} \tilde{f}_U(q_i, q_{i+1}, D)$$

Separate MLMs for bigrams and unigrams give FSDM the flexibility to adjust the document scoring depending on the query type

MLM is a special case of FSDM, when $\lambda_T = 1, \lambda_O = 0, \lambda_U = 0$

FSDM ranking function

FSDM incorporates document structure and term dependencies with the following ranking function:

$$P_\Lambda(D|Q) \overset{rank}{=} \lambda_T \sum_{q \in Q} \tilde{f}_T(q_i, D) + \lambda_O \sum_{q \in Q} \tilde{f}_O(q_i, q_{i+1}, D) + \lambda_U \sum_{q \in Q} \tilde{f}_U(q_i, q_{i+1}, D)$$

Separate MLMs for bigrams and unigrams give FSDM the flexibility to adjust the document scoring depending on the query type

MLM is a special case of FSDM, when $\lambda_T = 1, \lambda_O = 0, \lambda_U = 0$

FSDM ranking function

FSDM incorporates document structure and term dependencies with the following ranking function:

$$P_\Lambda(D|Q) \overset{rank}{=} \lambda_T \sum_{q \in Q} \tilde{f}_T(q_i, D) + \lambda_O \sum_{q \in Q} \tilde{f}_O(q_i, q_{i+1}, D) + \lambda_U \sum_{q \in Q} \tilde{f}_U(q_i, q_{i+1}, D)$$

Separate MLMs for bigrams and unigrams give FSDM the flexibility to adjust the document scoring depending on the query type

MLM is a special case of FSDM, when $\lambda_T = 1, \lambda_O = 0, \lambda_U = 0$

FSDM ranking function

Potential function for unigrams in case of FSDM:

$$\tilde{f}_T(q_i, D) = \log \sum_j w_j^T P(q_i|\theta_D^j) = \log \sum_j w_j^T \frac{tf_{q_i,D^j} + \mu_j \frac{cf_{q_i}^j}{|C_j|}}{|D^j| + \mu_j}$$

Example

apollo astronauts who walked on the moon

FSDM ranking function

Potential function for unigrams in case of FSDM:

$$\tilde{f}_T(q_i, D) = \log \sum_j w_j^T P(q_i|\theta_D^j) = \log \sum_j w_j^T \frac{tf_{q_i,D^j} + \mu_j \frac{cf_{q_i}^j}{|C_j|}}{|D^j| + \mu_j}$$

Example

apollo astronauts who walked on the moon

category

FSDM ranking function

Potential function for unigrams in case of FSDM:

$$\tilde{f}_T(q_i, D) = \log \sum_j w_j^T P(q_i|\theta_D^j) = \log \sum_j w_j^T \frac{tf_{q_i,D^j} + \mu_j \frac{cf_{q_i}^j}{|C_j|}}{|D^j| + \mu_j}$$

Example

apollo astronauts who walked on the moon
category attribute

Parameters of FSDM

Overall, FSDM has $3 * F + 3$ free parameters: $\langle \boldsymbol{w}^T, \boldsymbol{w}^O, \boldsymbol{w}^U, \boldsymbol{\lambda} \rangle$.

Properties of ranking function

1. Linearity with respect to $\boldsymbol{\lambda}$.

   We can apply any linear learning-to-rank algorithm to optimize the ranking function with respect to $\boldsymbol{\lambda}$.

2. Linearity with respect to $\boldsymbol{w}$ of the arguments of monotonic $\tilde{f}(\cdot)$ functions.

   Optimization of the arguments as linear functions with respect to $\boldsymbol{w}$, leads to optimization of each function $\tilde{f}(\cdot)$.

Overview

Optimization algorithm

1: $Q \leftarrow$ Training queries
2: **for** $s \in \{T, O, U\}$ **do** // *Optimize field weights of LMs independently*
3: $\quad \boldsymbol{\lambda} = \boldsymbol{e}_s$
4: $\quad \hat{\boldsymbol{w}}^s \leftarrow CoordAsc(Q, \boldsymbol{\lambda})$
5: **end for**
6: $\hat{\boldsymbol{\lambda}} \leftarrow CoordAsc(Q, \hat{\boldsymbol{w}}_T, \hat{\boldsymbol{w}}_O, \hat{\boldsymbol{w}}_U)$ // *Optimize* $\boldsymbol{\lambda}$

The unit vectors $\boldsymbol{e}_T = (1, 0, 0)$, $\boldsymbol{e}_O = (0, 1, 0)$, $\boldsymbol{e}_U = (0, 0, 1)$ are the corresponding settings of the parameters $\boldsymbol{\lambda}$ in the formula of FSDM ranking function.

$\Rightarrow$ direct optimization w.r.t. target metric, e.g. MAP

Overview

Collection

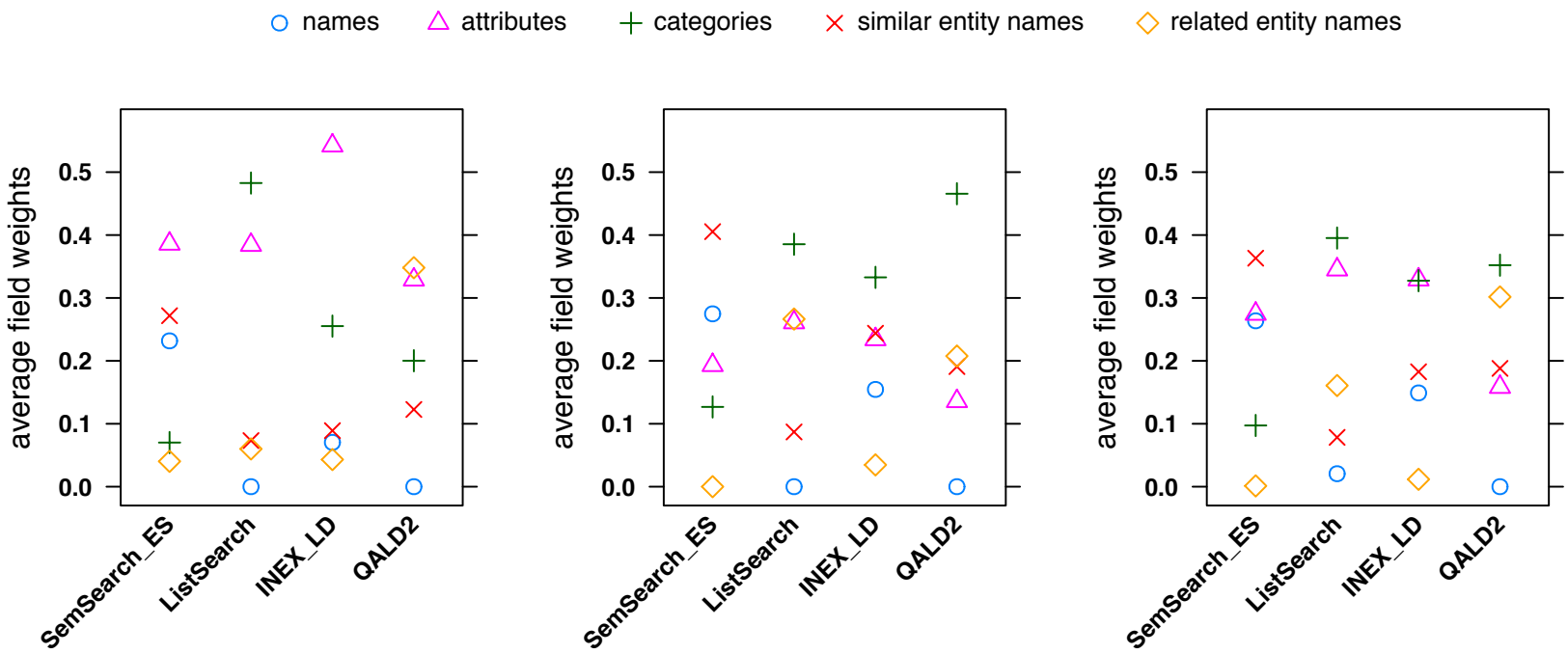
- DBPedia 3.7 was used as a collection in all experiments
- Structured version of on-line encyclopedia Wikipedia
- Provides the descriptions of over 3.5 million entities belonging to 320 classes

Query Sets

Balog and Neumayer. A Test Collection for Entity Search in DBpedia, SIGIR'13.

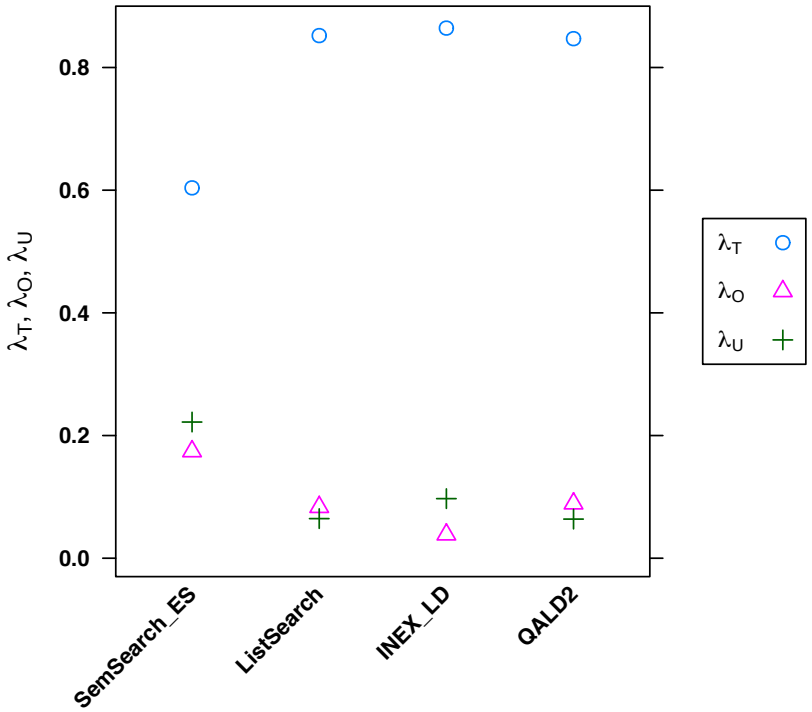
Query set	**Amount**	**Query types [Pound et al., 2010]**
SemSearch ES	130	Entity
ListSearch	115	Type
INEX-LD	100	Entity, Type, Attribute, Relation
QALD-2	140	Entity, Type, Attribute, Relation

Tuning field weights

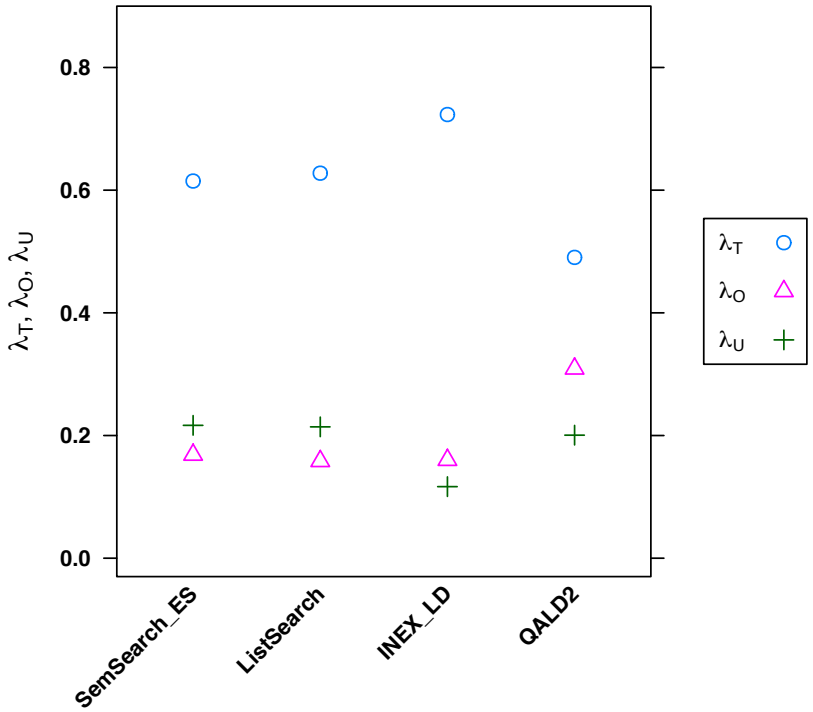


- *Attributes* field is consistently considered to be a very valuable for both unigrams and bigrams.
- The *names* field as well as the *similar entity names* field are highly important for queries aiming at finding named entities.
- Distinguishing *categories* from *related entity names* is particularly important for type queries.

Tuning λ



(a) SDM



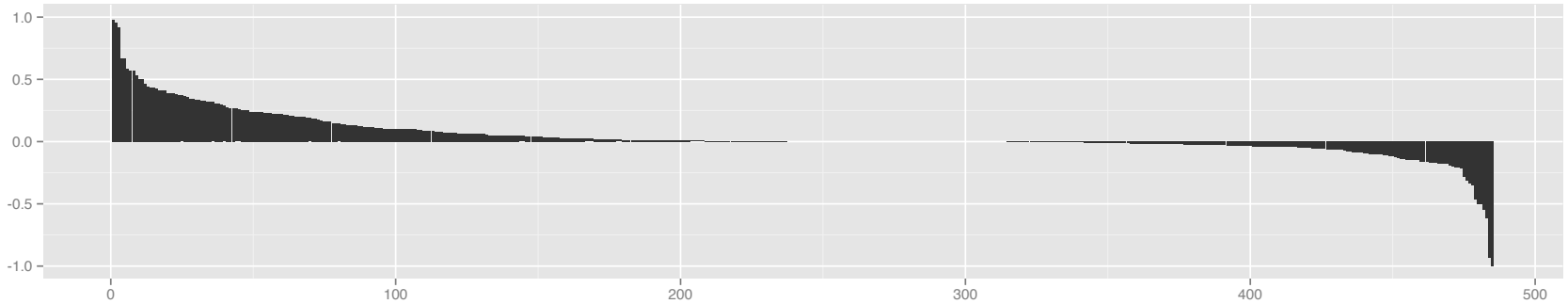
(b) FSDM

- Bigram matches are important for named entity queries.
- Transformation of SDM into FSDM increases the importance of bigram matches, which ultimately improves the retrieval performance, as we will demonstrate next.

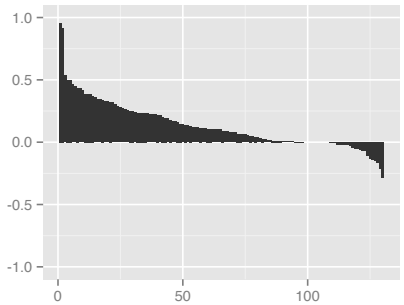
Experimental results

Query set	Method	MAP	P@10	P@20	b-pref
SemSearch ES	MLM-CA	0.320	0.250	0.179	0.674
	SDM-CA	0.254*	0.202*	0.149*	0.671
	FSDM	**0.386**$^{*}_{\dagger}$	**0.286**$^{*}_{\dagger}$	**0.204**$^{*}_{\dagger}$	**0.750**$^{*}_{\dagger}$
ListSearch	MLM-CA	0.190	0.252	0.192	0.428
	SDM-CA	0.197	0.252	0.202	**0.471***
	FSDM	**0.203**	**0.256**	**0.203**	0.466*
INEX-LD	MLM-CA	0.102	0.238	0.190	0.318
	SDM-CA	**0.117***	0.258	0.199	0.335
	FSDM	0.111*	**0.263***	**0.215**$^{*}_{\dagger}$	**0.341***
QALD-2	MLM-CA	0.152	0.103	0.084	0.373
	SDM-CA	0.184	0.106	0.090	0.465*
	FSDM	**0.195***	**0.136**$^{*}_{\dagger}$	**0.111***	**0.466***
All queries	MLM-CA	0.196	0.206	0.157	0.455
	SDM-CA	0.192	0.198	0.155	0.495*
	FSDM	**0.231**$^{*}_{\dagger}$	**0.231**$^{*}_{\dagger}$	**0.179**$^{*}_{\dagger}$	**0.517**$^{*}_{\dagger}$

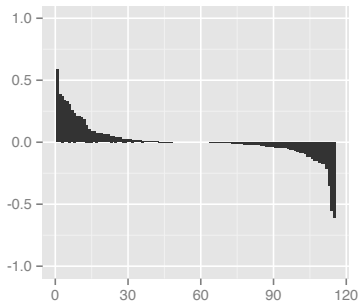
Topic-Level differences between SDM and FSDM



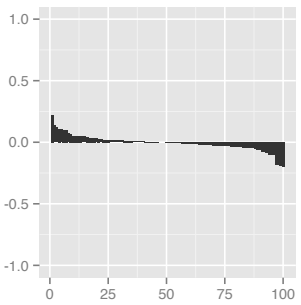
(a) All queries



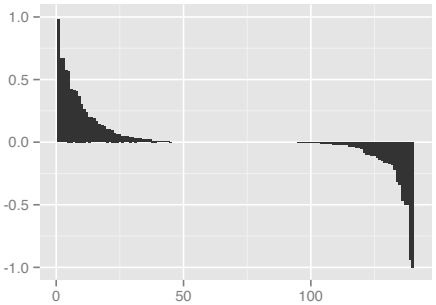
(b) SemSearch ES



(c) ListSearch



(d) INEX-LD



(e) QALD-2

Topic-level differences in average precision between FSDM and SDM. Positive values indicate FSDM is better.

Overview

Conclusion

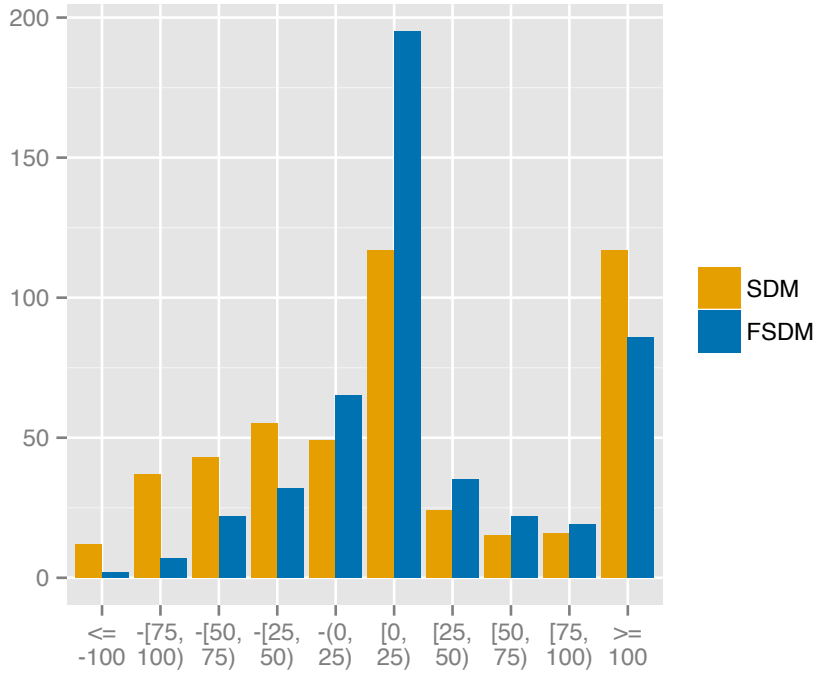
- We proposed Fielded Sequential Dependence Model, a novel retrieval model, which incorporates term dependencies into structured document retrieval
- We proposed a two-stage algorithm to directly optimize the parameters of FSDM with respect to the target retrieval metric
- We experimentally demonstrated that having different field weighting schemes for unigrams and bigrams is effective for different types of ERWD queries
- Experimental evaluation of FSDM on a standard publicly available benchmark showed that it consistently and, in most cases, statistically significantly outperforms MLM and SDM for the task of ERWD

Code and runs are available at

`github.com/teanalab/FieldedSDM`

Questions?

Robustness



- FSDM is more robust compared to SDM
- FSDM improves the performance of 50% of the queries with respect to MLM-CA, compared to 45% of the queries improved by SDM
- FSDM decreases the performance of only 26% of the queries, while SDM degrades the performance of 40% of the queries

Various Levels of Difficulty

Level	Model	MAP	P@10	P@20	b-pref
Difficult queries	SDM	0.213	**0.067**	0.042	0.599
	FSDM	**0.239**	0.065	**0.043**	**0.621**
Medium queries	SDM	0.209	0.224	0.165	0.532
	FSDM	**0.264**$_\dagger$	**0.272**$_\dagger$	**0.191**$_\dagger$	**0.559**$_\dagger$
Easy queries	SDM	0.139	0.298	0.262	0.316
	FSDM	**0.166**$_\dagger$	**0.345**$_\dagger$	**0.309**$_\dagger$	**0.330**

Creating sophisticated entity descriptions is not sufficient for answering *difficult queries* in entity retrieval scenario and better capturing the semantics of query terms is required to further improve the precision of FSDM for difficult queries.

Failure Analysis

- SDM errors
  - Overestimation of importance of matches in the fields other than *names*
    - *"city of charlotte"*
    - *"give me all soccer clubs in the premier league"*
    - *"us presidents since 1960"*
- FSDM errors
  - Neglecting the important query terms
    - *"members of the beaux arts trio"*
    - *"who created goofy"*
    - *"where is the residence of the prime minister of spain?"*
  - Lack of semantic knowledge.
    - *"did nicole kidman have any siblings"*