ANT – R&D Platform for Entity-Oriented Search

IEEE SYP 2018 Porto

Last update: Jul 26, 2018

José Devezas MAP-i 2016/2017 INESC TEC & FEUP InfoLab

Supervisor:	Sérgio Nunes	U.Porto
External advisor:	Bruno Martins	U.Lisboa

Universidade do Minho

universidade de aveiro

U.PORTO

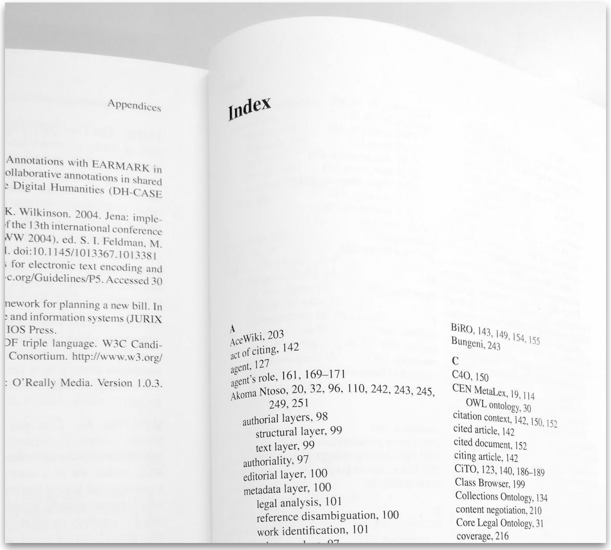
Contents

Introduction

What is entity-oriented search and why does it matter?

Keyword-based vs entity-oriented search

- Keyword-based search was modeled after the back-of-the-book index.
- Finding relevant content involved:
  1. Selecting one or several keywords;
  2. Jumping to the indicated pages;
  3. Reading passages and using knowledge, either internal or external to the book, to assess the relevance.



Appendices

Annotations with EARMARK in
ollaborative annotations in shared
e Digital Humanities (DH-CASE

K. Wilkinson. 2004. Jena: imple-
f the 13th international conference
WW 2004), ed. S. I. Feldman, M.
I. doi:10.1145/1013367.1013381
s for electronic text encoding and
-c.org/Guidelines/P5. Accessed 30

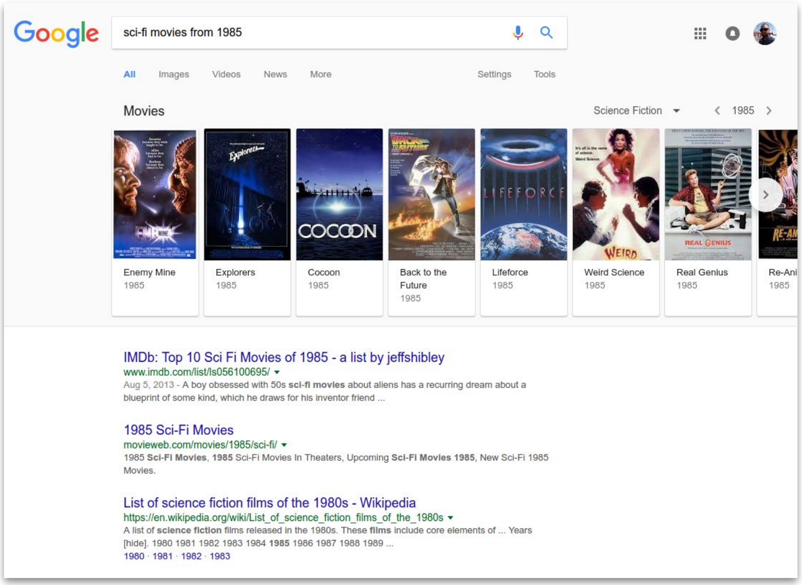
nework for planning a new bill. In
e and information systems (JURIX
IOS Press.
DF triple language. W3C Candi-
Consortium. http://www.w3.org/

: O'Really Media. Version 1.0.3.

Index

Keyword-based vs entity-oriented search

- Entity-oriented search makes use of:
  - Natural language understanding:
    - For queries;
    - And documents.
  - Structured data from knowledge bases.
- Making it possible to answer queries like:
  - `[ sci-fi movies from 1985 ]`
- By returning a combination of:
  - Text documents;
  - And entities (e.g., movies).



IMDb: Top 10 Sci Fi Movies of 1985 - a list by jeffshibley
www.imdb.com/list/ls056100695/
Aug 5, 2013 - A boy obsessed with 50s **sci-fi movies** about aliens has a recurring dream about a blueprint of some kind, which he draws for his inventor friend ...

1985 Sci-Fi Movies
movieweb.com/movies/1985/sci-fi/
1985 **Sci-Fi Movies**, **1985** Sci-Fi Movies In Theaters, Upcoming **Sci-Fi Movies 1985**, New Sci-Fi 1985 Movies.

List of science fiction films of the 1980s - Wikipedia
https://en.wikipedia.org/wiki/List_of_science_fiction_films_of_the_1980s
A list of **science fiction** films released in the 1980s. These **films** include core elements of ... Years [hide]. 1980 1981 1982 1983 1984 **1985** 1986 1987 1988 1989 ...
1980 · 1981 · 1982 · 1983

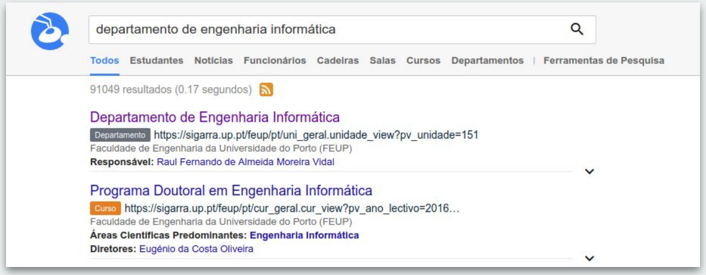
The relevance of entities in search

- In queries:.
  - A study of the AOL Query Log showed that:
    - 18-39% queries directly reference entities;
    - 73-87% queries contain at least one entity.
- In documents:
  - The annotated CoNLL 2003 English training set contained:
    - 14,987 sentences;
    - 23,499 entities;
    - Resulting in 1.6 entities per sentence.

ANT

Searching for information at the University of Porto.

ANT



departamento de engenharia informática

Todos | Estudantes | Notícias | Funcionários | Cadeiras | Salas | Cursos | Departamentos | Ferramentas de Pesquisa

91049 resultados (0.17 segundos)

Departamento de Engenharia Informática
Departamento https://sigarra.up.pt/feup/pt/uni_geral.unidade_view?pv_unidade=151
Faculdade de Engenharia da Universidade do Porto (FEUP)
Responsável: Raul Fernando de Almeida Moreira Vidal

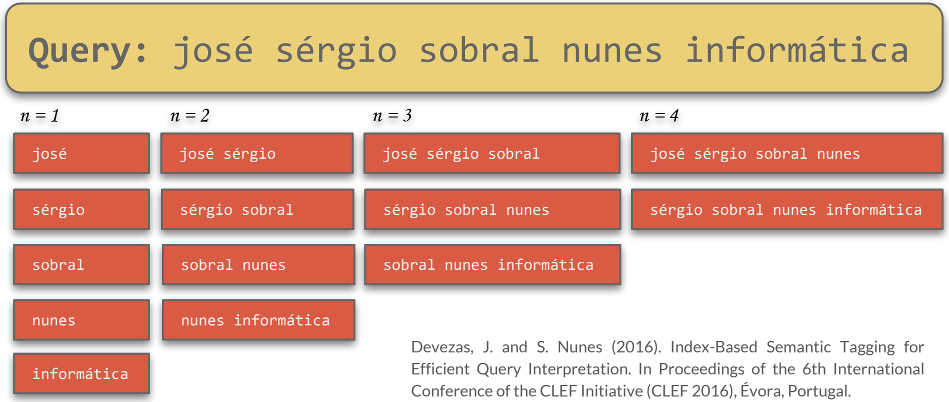
Programa Doutoral em Engenharia Informática
Curso https://sigarra.up.pt/feup/pt/cur_geral.cur_view?pv_ano_lectivo=2016...
Faculdade de Engenharia da Universidade do Porto (FEUP)
Áreas Científicas Predominantes: **Engenharia Informática**
Diretores: Eugénio da Costa Oliveira

Ad hoc search of eNtities and Text.

- ANT is an entity-oriented search engine, built to support the five query categories defined by Pound et al. (2010):
  - Entity query;
  - Type query;
  - Attribute query;
  - Relation query;
  - Keyword query.
- It is supported by two Lucene indexes:
  - Query analysis index;
  - Entity index.
- And a Virtuoso RDF triplestore:
  - Useful for relation queries.

How does ANT understand queries?

- Query segmentation based on the retrieval of matching entities for all query n-grams up to a maximum value of n.
- Semantic tagging of query segments based on the probability of associating a given type of entity to an n-gram.



Devezas, J. and S. Nunes (2016). Index-Based Semantic Tagging for Efficient Query Interpretation. In Proceedings of the 6th International Conference of the CLEF Initiative (CLEF 2016), Évora, Portugal.

How does ANT understand queries?

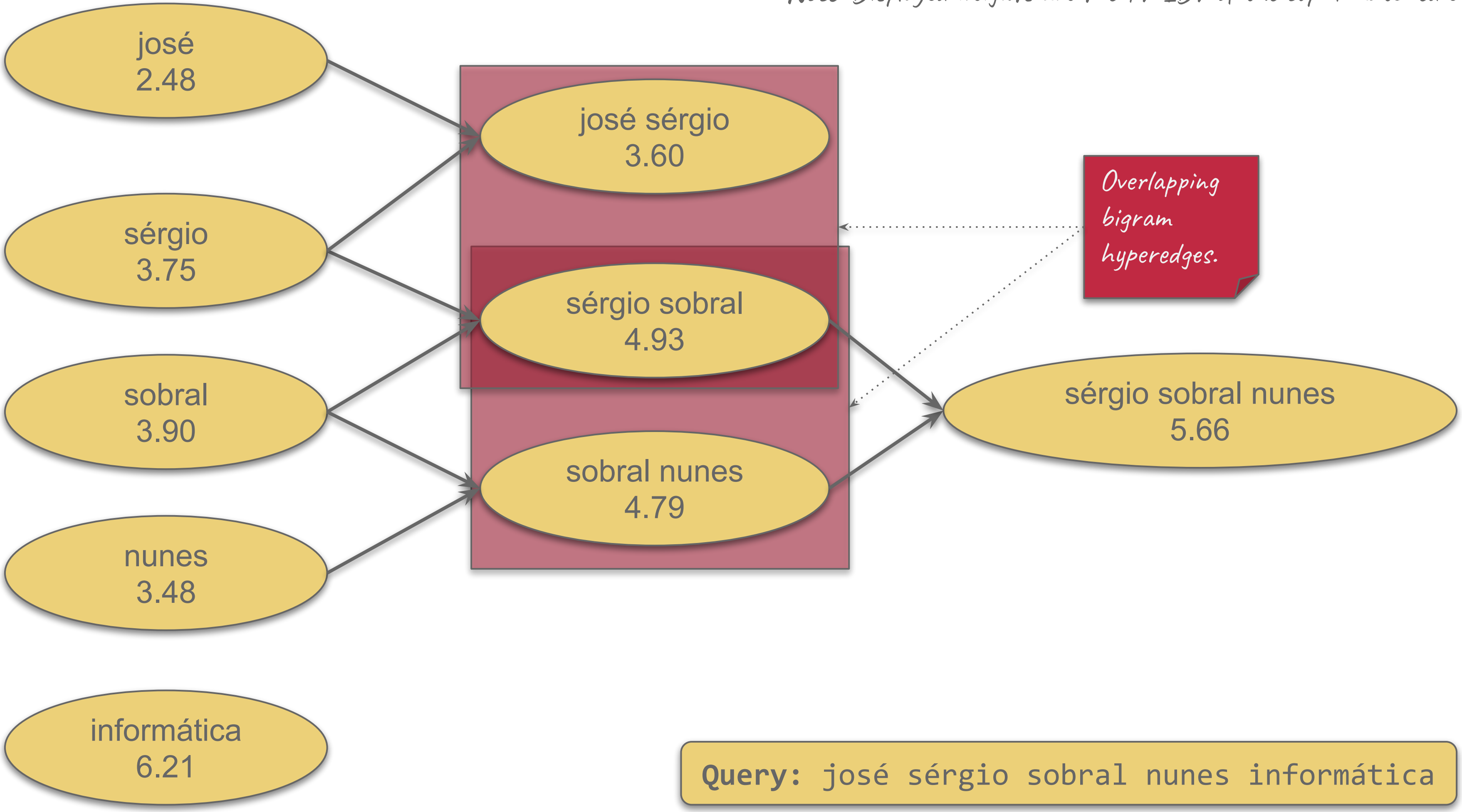
- The **actual method** we ended up using is a variation of this that we called "**Score Hypergraph**".
  - TF-IDF scores instead of probabilities.
  - Dedicated query analysis index to search for entities matching n-grams.
  - Hypergraph* of n-grams to resolve query segment overlaps and to fix bugs with the previous approach.

* A hypergraph is a generalization of a graph, where edges can have an arbitrary number of nodes.
Think of a social network modeling binary friendship relations (a social graph).
And then think of a social network that also models groups of multiple friends (a social hypergraph).

Score Hypergraph

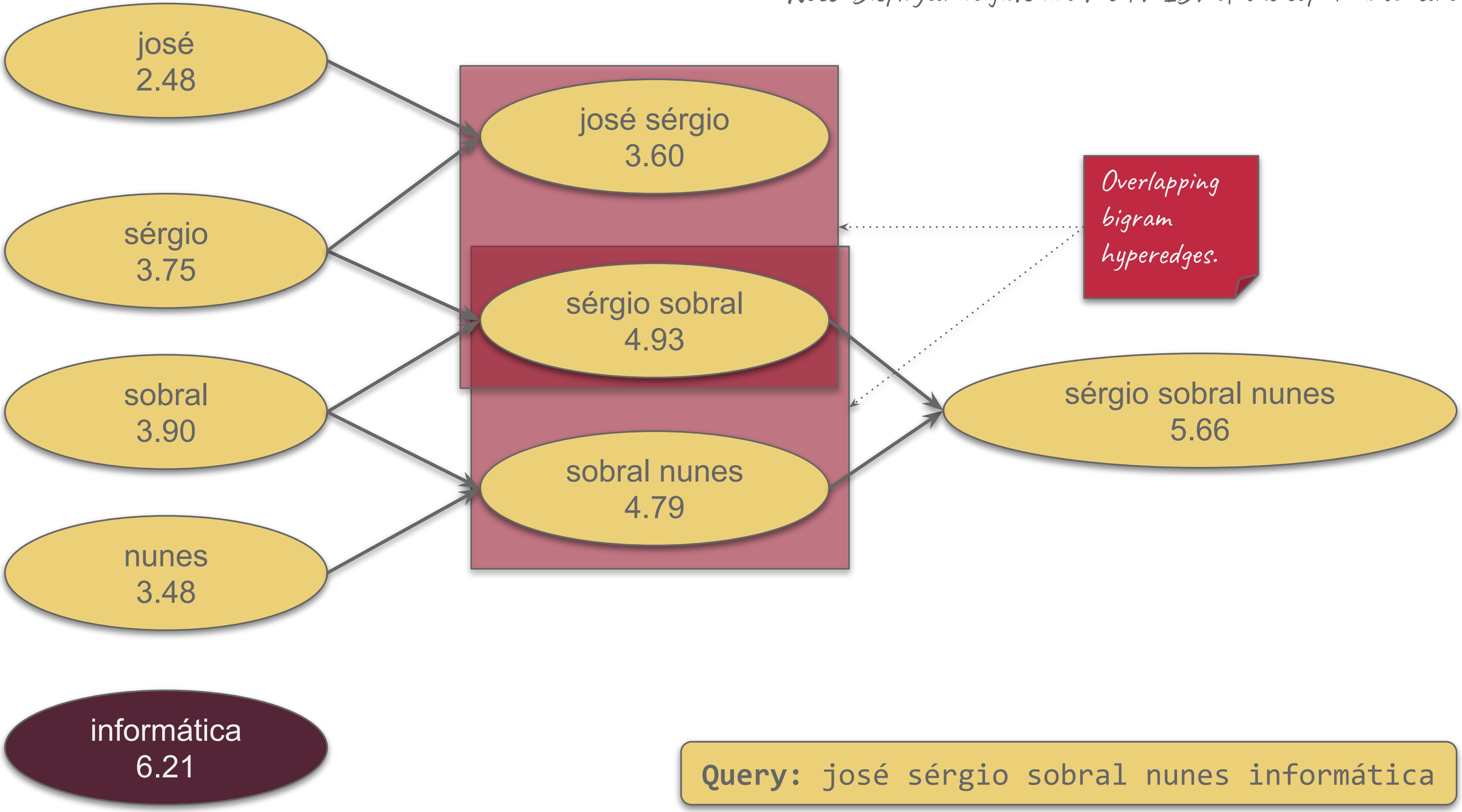
Query segmentation and semantic tagging in ANT.

Note: *Displayed weights are the TF-IDF of the top-1 matched entity.*



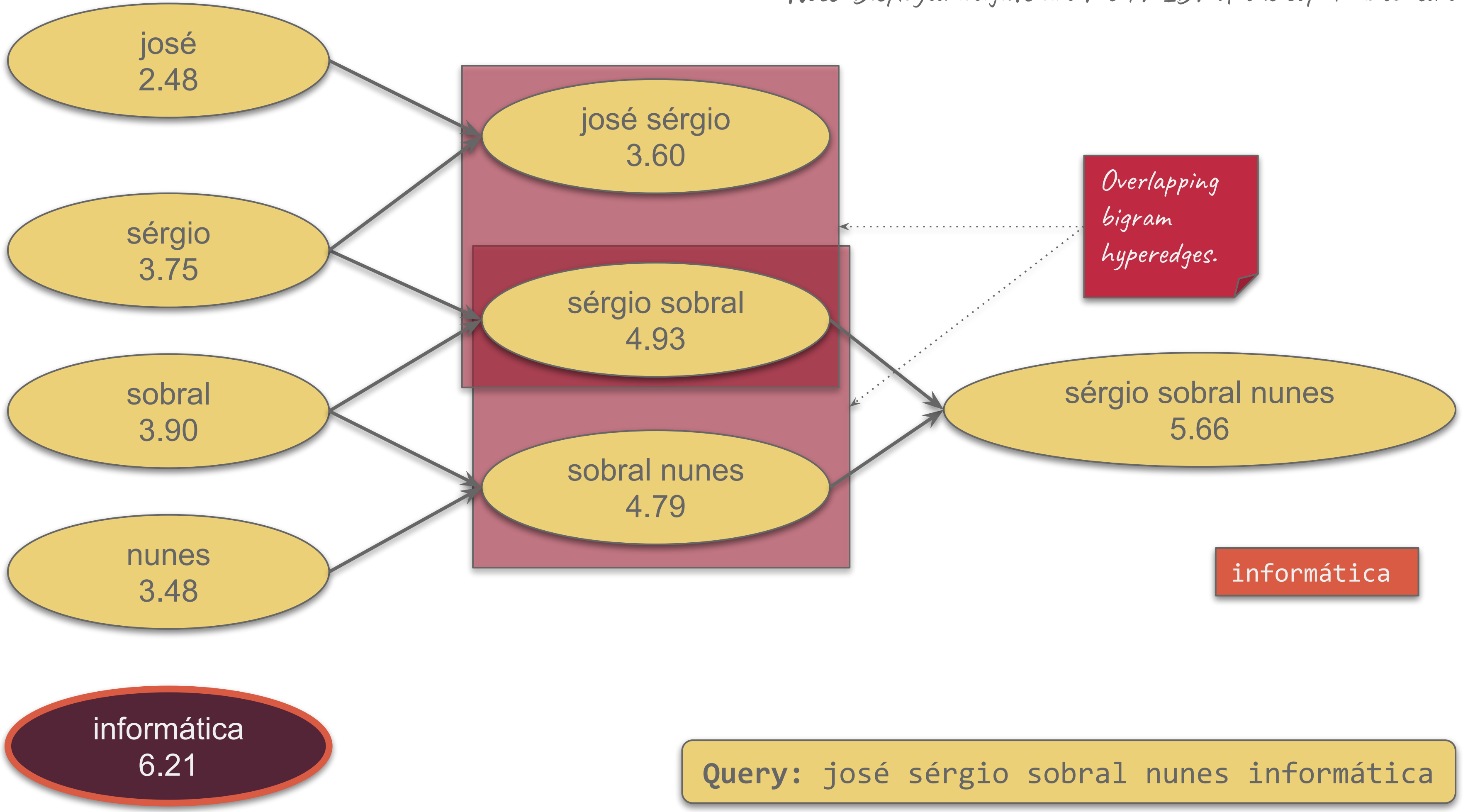
Query: `josé sérgio sobral nunes informática`

Note: Displayed weights are the TF-IDF of the top-1 matched entity.



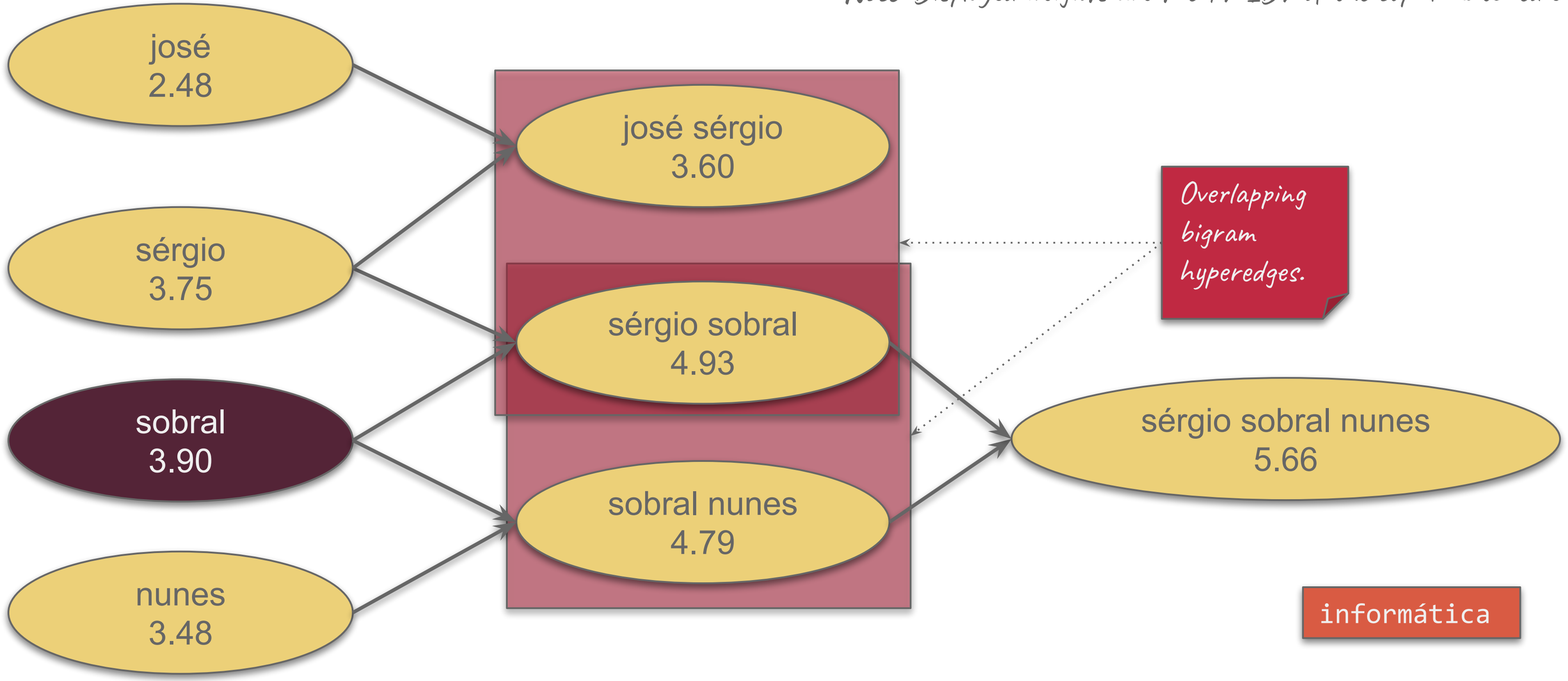
Query: `josé sérgio sobral nunes informática`

Note: Displayed weights are the TF-IDF of the top-1 matched entity.



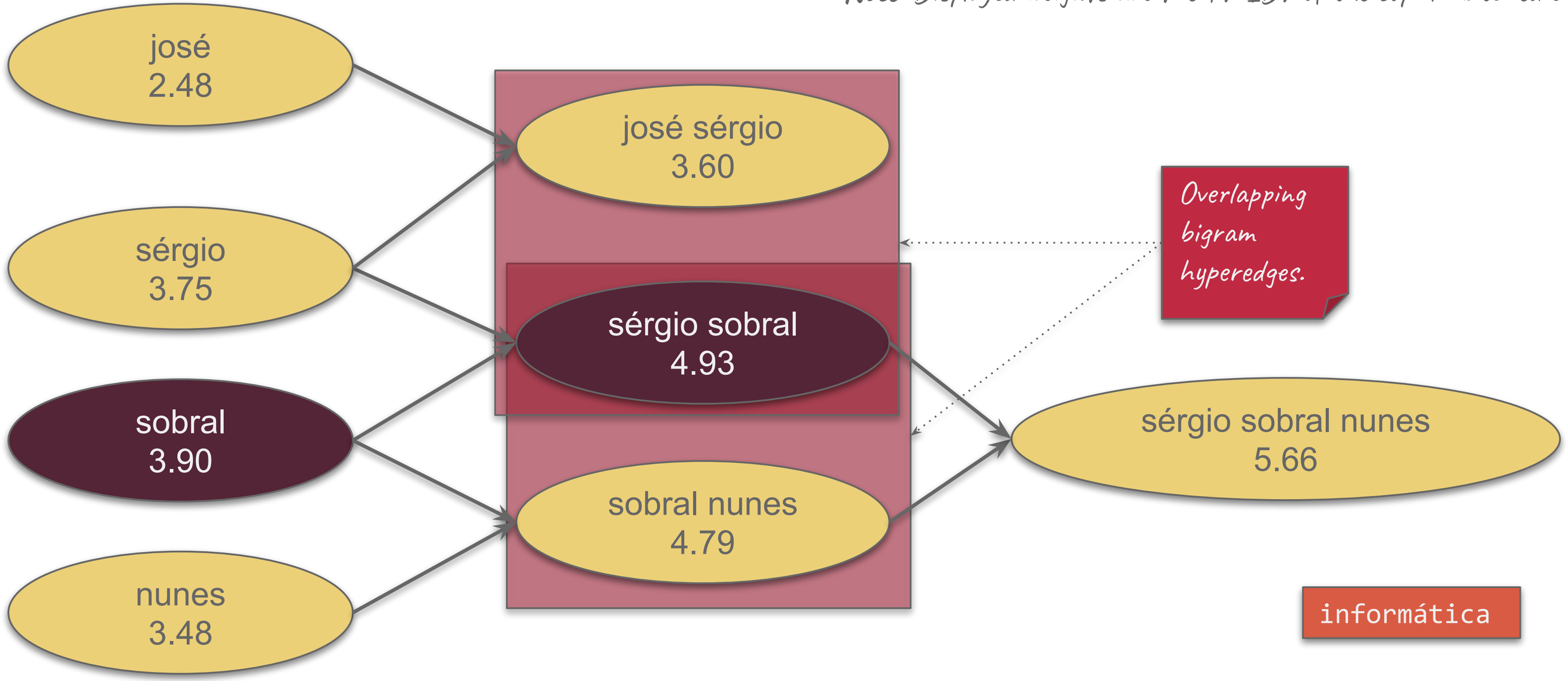
Query: josé sérgio sobral nunes informática

Note: Displayed weights are the TF-IDF of the top-1 matched entity.



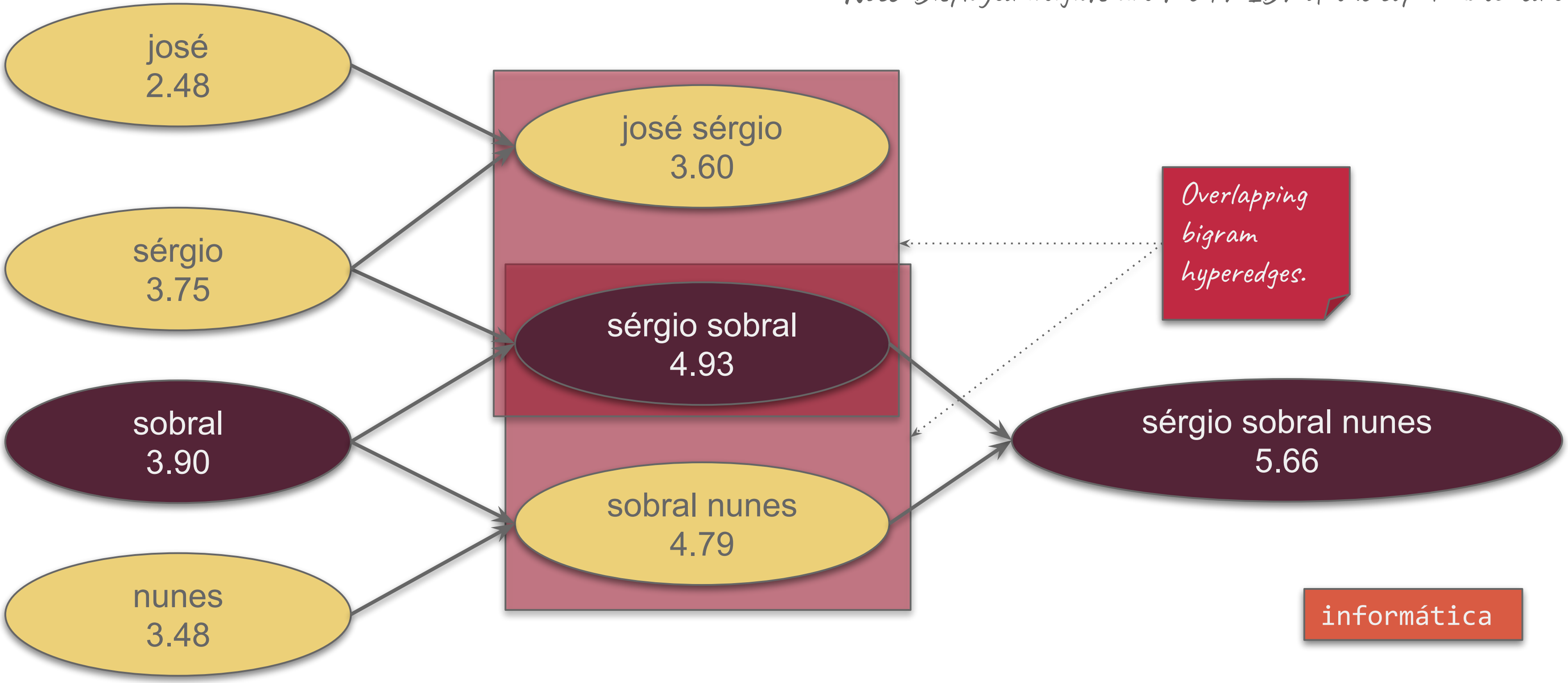
Query: `josé sérgio sobral nunes informática`

Note: Displayed weights are the TF-IDF of the top-1 matched entity.



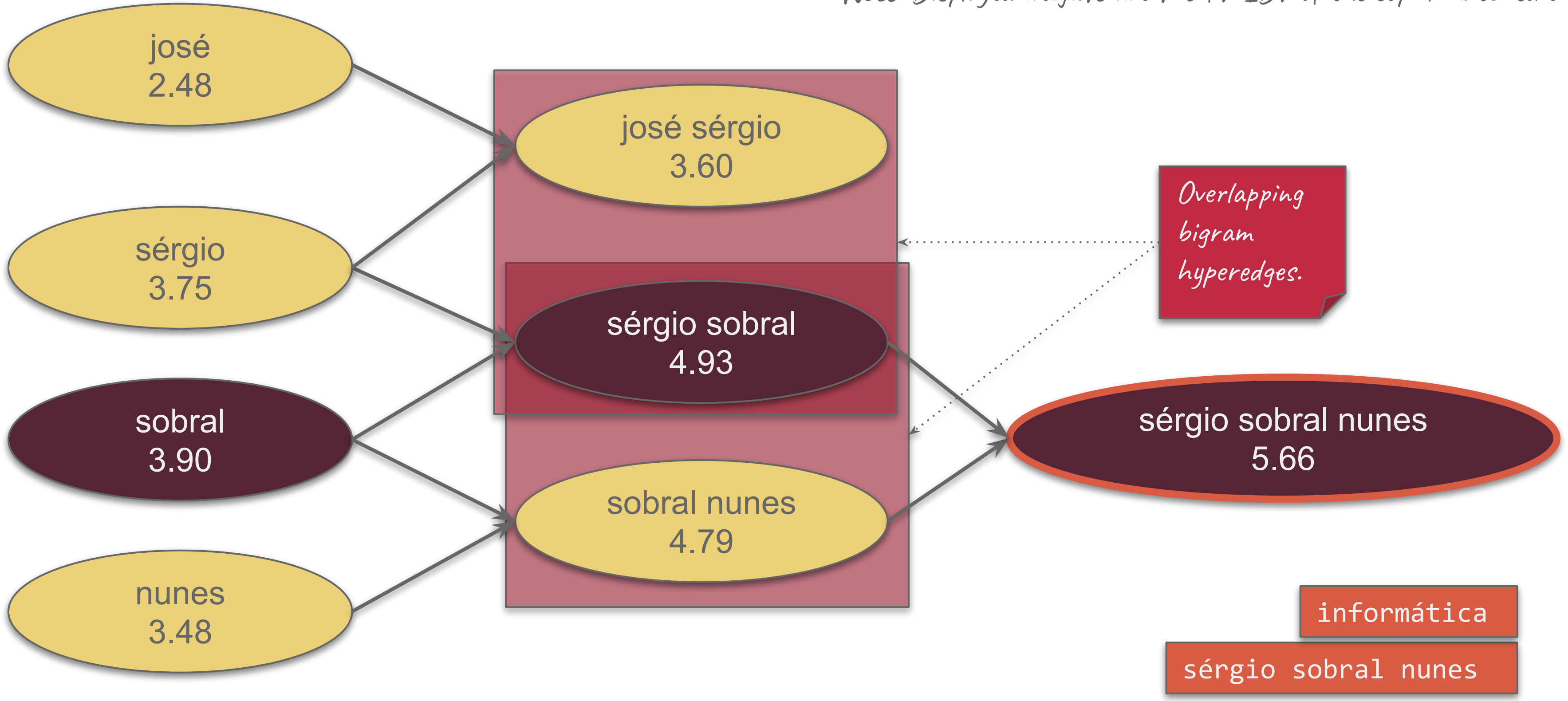
Query: josé sérgio sobral nunes informática

Note: Displayed weights are the TF-IDF of the top-1 matched entity.



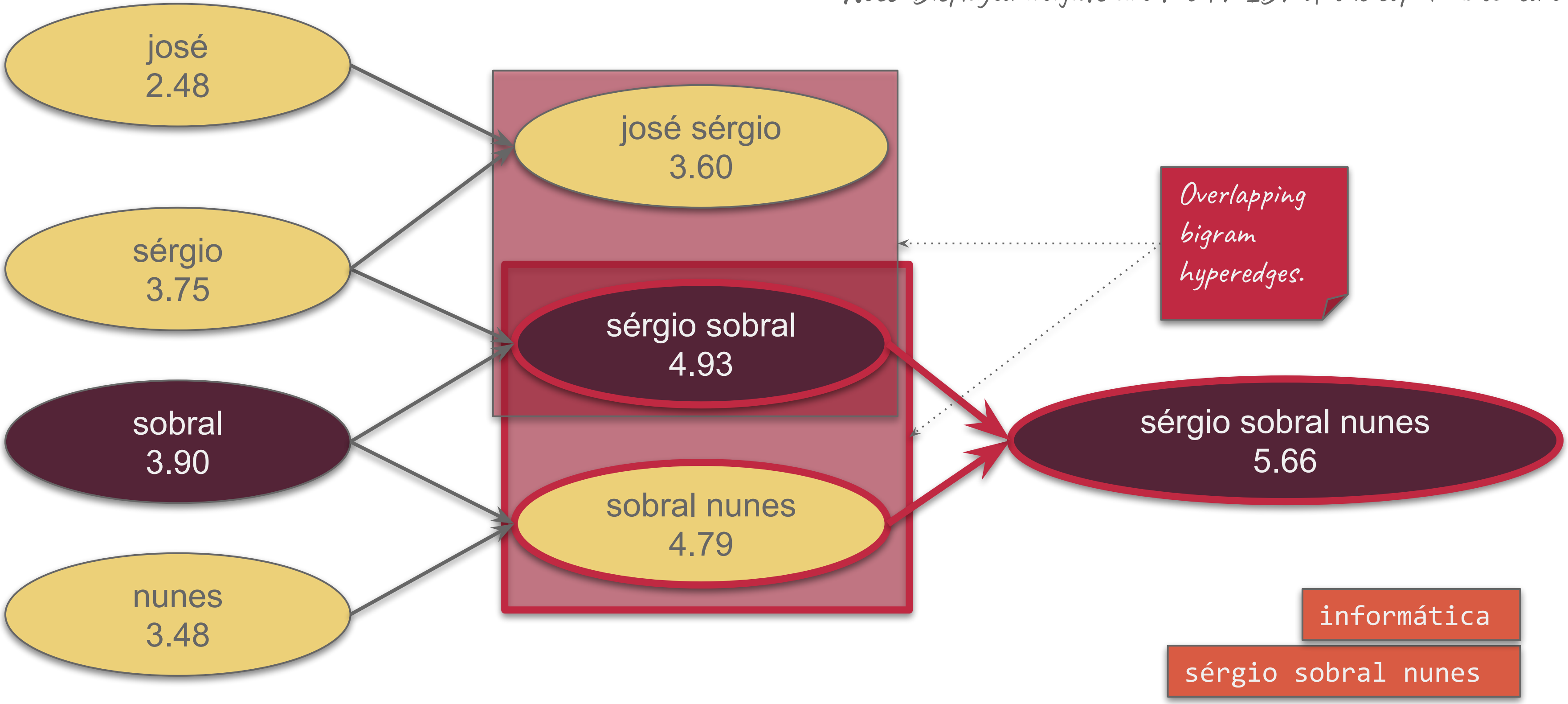
Query: `josé sérgio sobral nunes informática`

Note: *Displayed weights are the TF-IDF of the top-1 matched entity.*



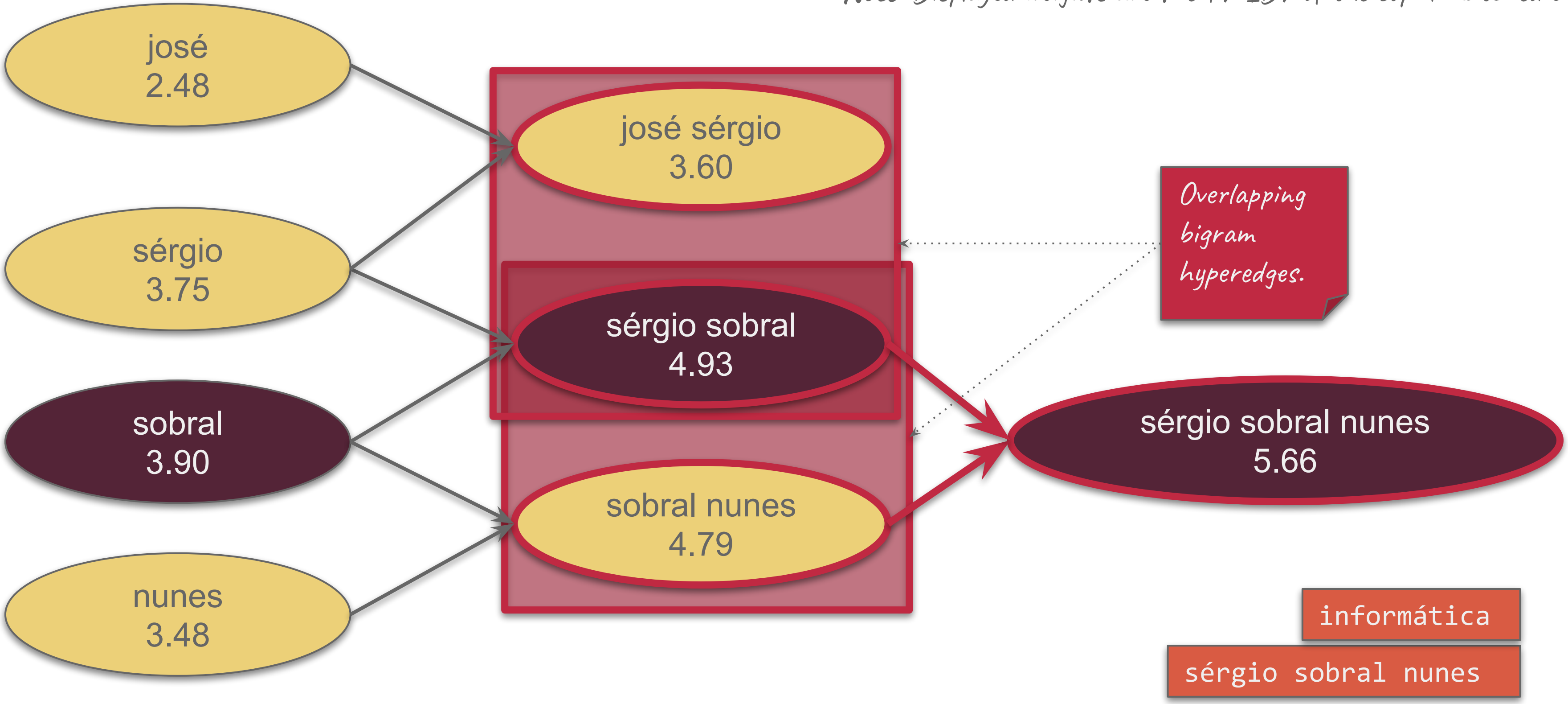
Query: `josé sérgio sobral nunes informática`

***Note:** Displayed weights are the TF-IDF of the top-1 matched entity.*



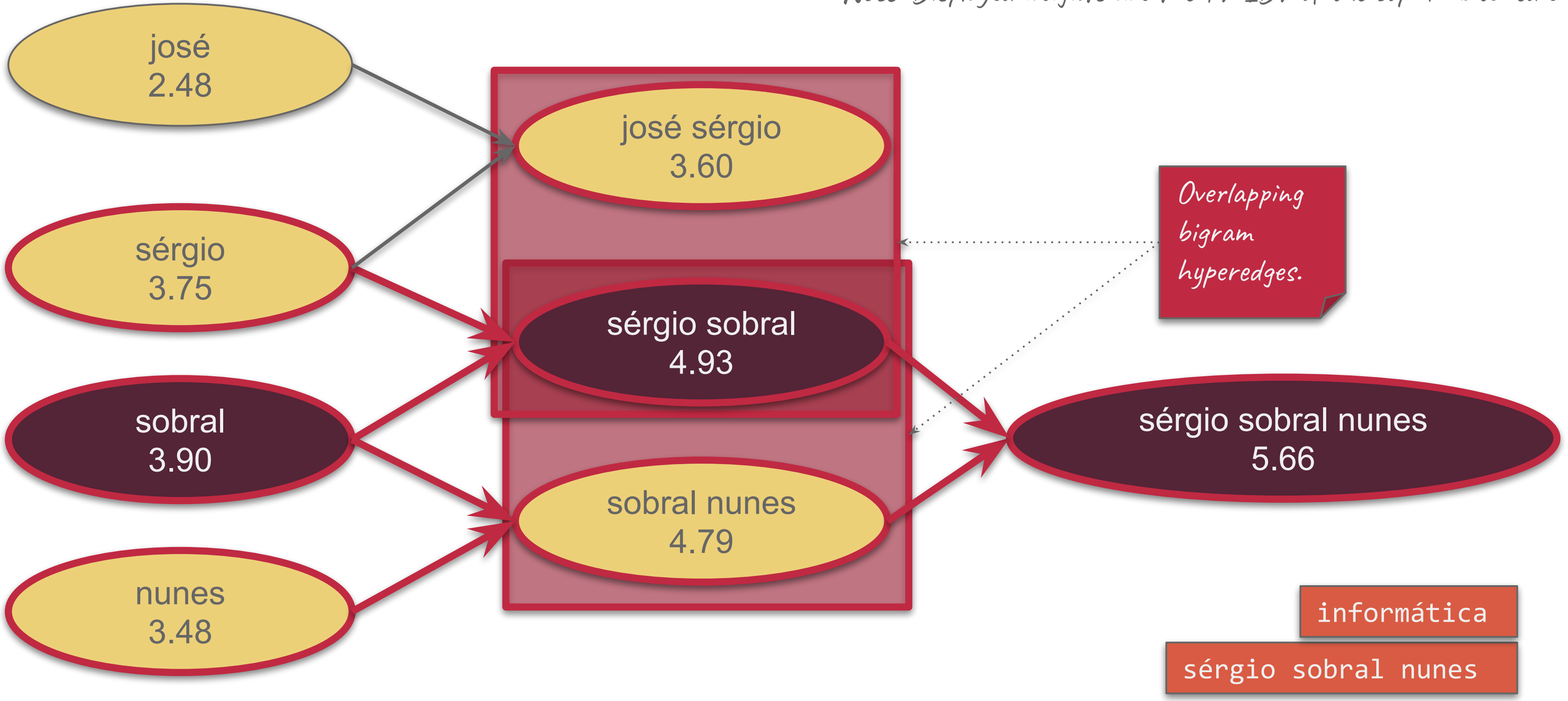
Query: `josé sérgio sobral nunes informática`

Note: Displayed weights are the TF-IDF of the top-1 matched entity.



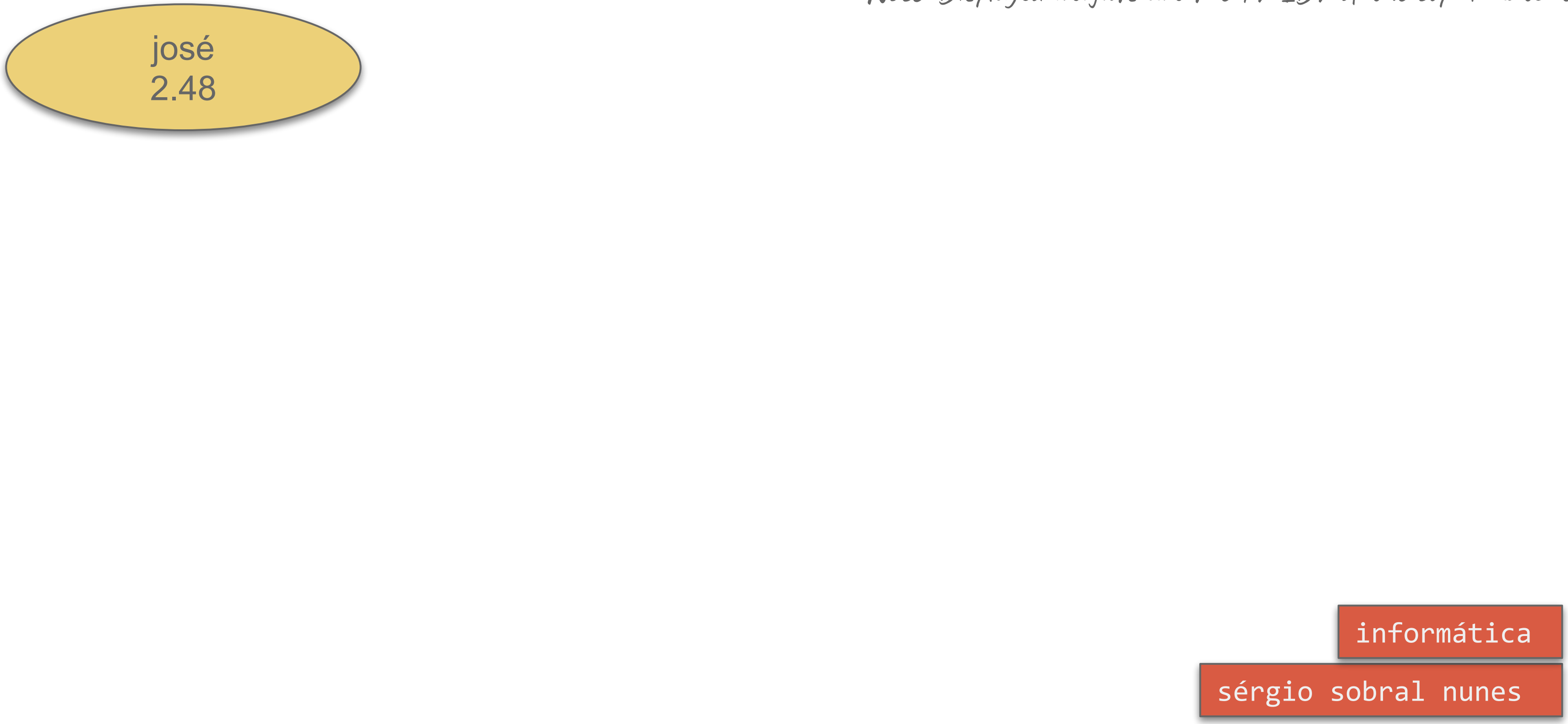
Query: josé sérgio sobral nunes informática

Note: Displayed weights are the TF-IDF of the top-1 matched entity.



Query: `josé sérgio sobral nunes informática`

Note: *Displayed weights are the TF-IDF of the top-1 matched entity.*



Query: josé sérgio sobral nunes informática

Note: *Displayed weights are the TF-IDF of the top-1 matched entity.*

josé
2.48

informática

sérgio sobral nunes

Query: josé sérgio sobral nunes informática

Note: *Displayed weights are the TF-IDF of the top-1 matched entity.*

josé
2.48
informática
sérgio sobral nunes
josé

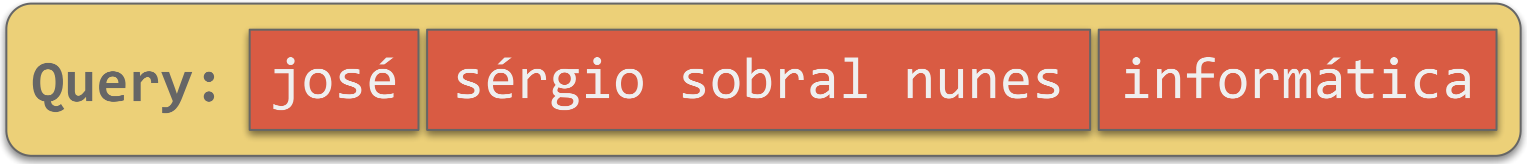
Query: josé sérgio sobral nunes informática

informática

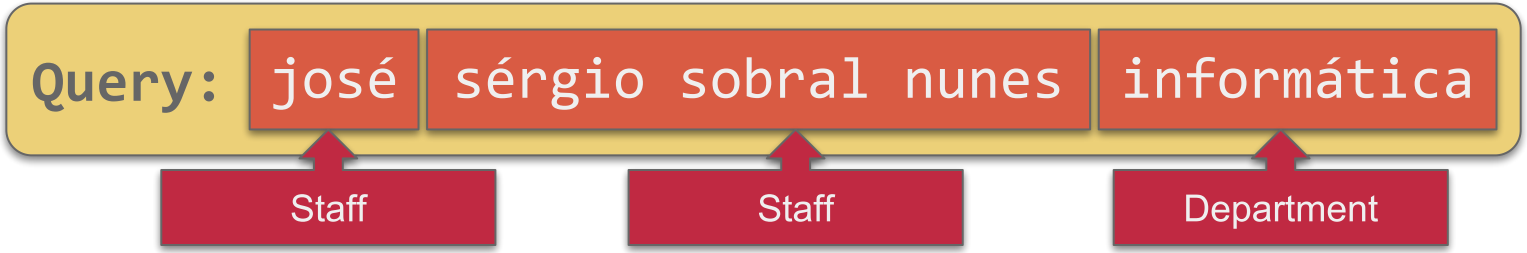
sérgio sobral nunes

josé

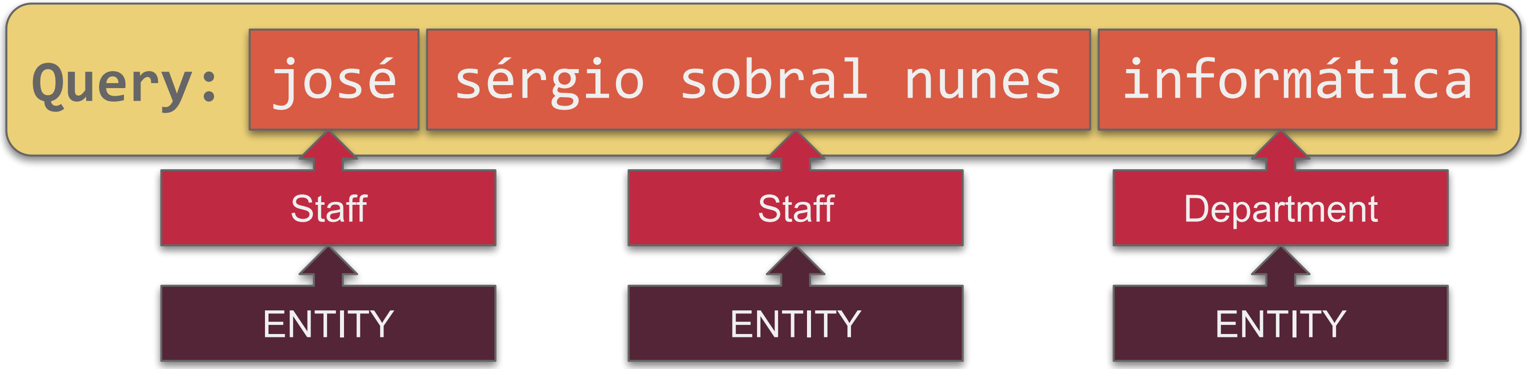
Query: josé sérgio sobral nunes informática



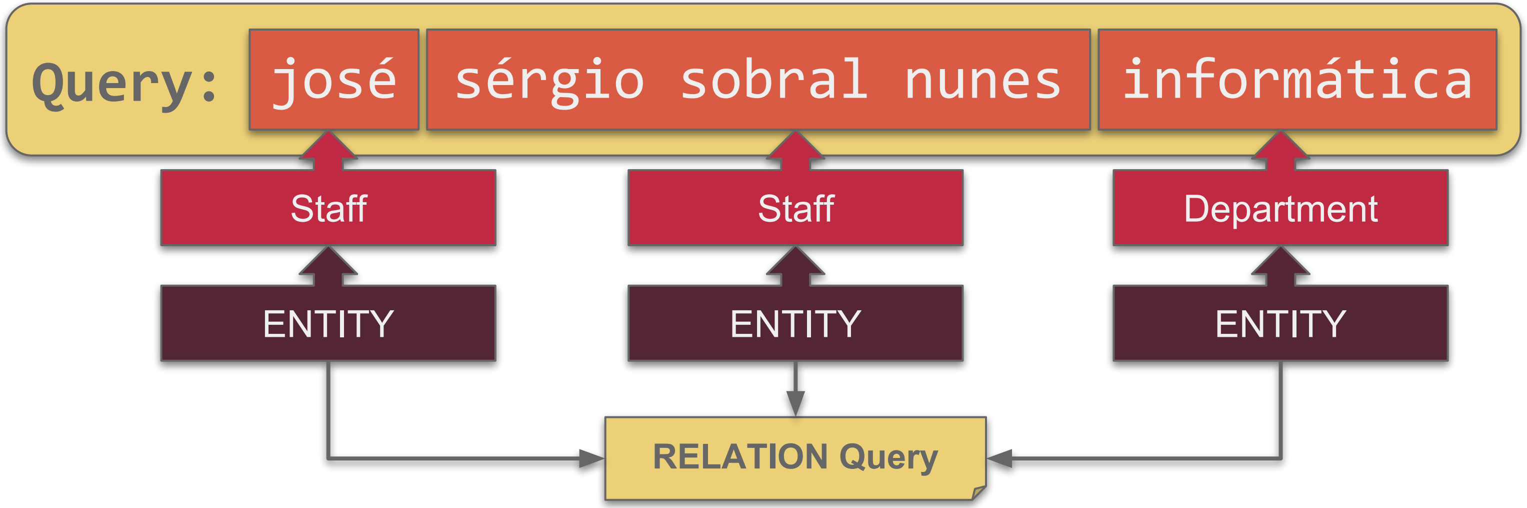
1. The query was segmented based on the *n*-grams with the highest-scoring entities.



2. The query was assigned semantic tags based on the type of the highest-scoring entity.



3. From the semantic tag, we directly derived a higher level tag that could either be ENTITY (e.g., instance of Staff class), ATTRIBUTE (e.g., property) or TYPE (e.g., Staff class).

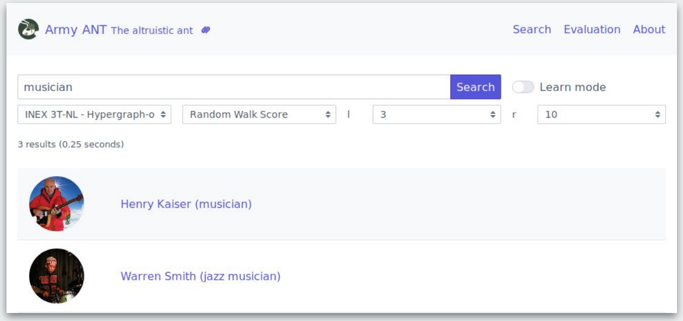


4. Based on the combination of higher level tags, we conditionally obtained the query category.

Army ANT

Researching entity-oriented search.

Army ANT


Army ANT The altruistic ant

Search Evaluation About

musician

Search

Learn mode

INEX 3T-NL - Hypergraph-o | Random Walk Score | l 3 | r 10

3 results (0.25 seconds)

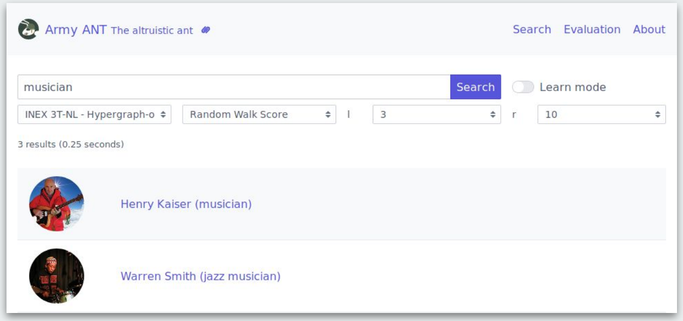
Henry Kaiser (musician)

Warren Smith (jazz musician)

A workbench for innovation in entity-oriented search.

- Indexing unit: documents with *doc_id*, *text* and *triples.*
- Able to define **readers** that work as iterators of documents.
- Able to implement retrieval models **(engines)** by implementing *index()* and *search()* methods.

Army ANT



musician

INEX 3T-NL - Hypergraph-o | Random Walk Score | l 3 | r 10

3 results (0.25 seconds)

Henry Kaiser (musician)

Warren Smith (jazz musician)

A workbench for innovation in entity-oriented search.

Front-end provides:

- Standard **search interface**, where you can select an index and a ranking function.
- **Learn mode** interface with:
  - Results without metadata;
  - Score component visualization;
  - Trace for the active query;
  - Ranking function details;
  - Collection description.
- **Evaluation interface** supporting:
  - Topics+Assessments (INEX Ad Hoc and INEX XML Entity Ranking);
  - Topics (TREC Common Core);
  - Living Labs API (TREC OpenSearch).

Command Line Interface

`index | search | inspect | analysis | sampling | features | extras | evaluation | server`

./army-ant.py

- `index`
  - Index a supported collection (i.e., based on an implemented reader), using one of the available engines.
- `search`
  - Search one of the supported indexes (has an interactive mode to avoid preload latency).
- `inspect`
  - Extract several features from a particular index (supported features depend on the engine).
- `analysis`
  - `rws-rank-concordance`
    - Analyze rank concordance for Random Walk Score (Hypergraph-of-Entity).
- `sampling`
  - Create a subset of one of the supported collections.
- `features`
  - Extract features (usually from a collection), such as word embeddings and similarities.

./army-ant.py

- `extras`
  - `fetch-wikipedia-images`
    - Obtain the Wikipedia image URL for documents stored in the database.
  - `word2vec-knn`
    - Return a ranked list of the k-nearest neighbors for a given word.
  - `word2vec-sim`
    - Measure the similarity between the embeddings for two words.
- `evaluation`
  - Queue and run an evaluation task using a supported evaluator (for now, it supports INEX and Living Labs API; soon it will support TREC qrels).
- `server`
  - Launch the web server with a search interface, a learn mode and an evaluation panel.

Configuration

Based on YAML and mostly used by the web interface.

Example file

Global settings for metadata storage, evaluation metrics and location, and reserved heap space for Java-based engines.

```
defaults:
  db:
    location: mongo
    name: army_ant
    type: mongo
  eval:
    metrics:
      favorite:
        - GMAP
        - MAP
        - NDCG@10
        - P@10
    location: /home/army-ant/data/eval
  service:
    ner:
      entity_list: /home/army-ant/data/people.txt
  depend:
    stanford-ner: /opt/stanford-ner-2015-12-09
  jvm:
    memory: 5120
    other_args: -XX:+UseConcMarkSweepGC
```

Example file

Lucene index and ranking functions configuration.

```
engines:
  lucene-inex-3t-nl:
    name: INEX 3T-NL - Lucene
    db:
      name: inex
    index:
      type: lucene
      location: /home/army-ant/data/indexes/lucene
    ranking:
      default:
        id: tf_idf
     functions:
        tf_idf:
          name: TF-IDF
        bm25:
          name: BM25
          params:
            k1: [1.2, 1, 1.8]
            b: [0.75, 0.5, 1]
        dfr:
          name: DFR
          params:
            BM: [BE, G, P, D, In, Ine, IF]
            AE: [L, B, Disabled]
            N: [H1, H2, H3, Z, Disabled]
```

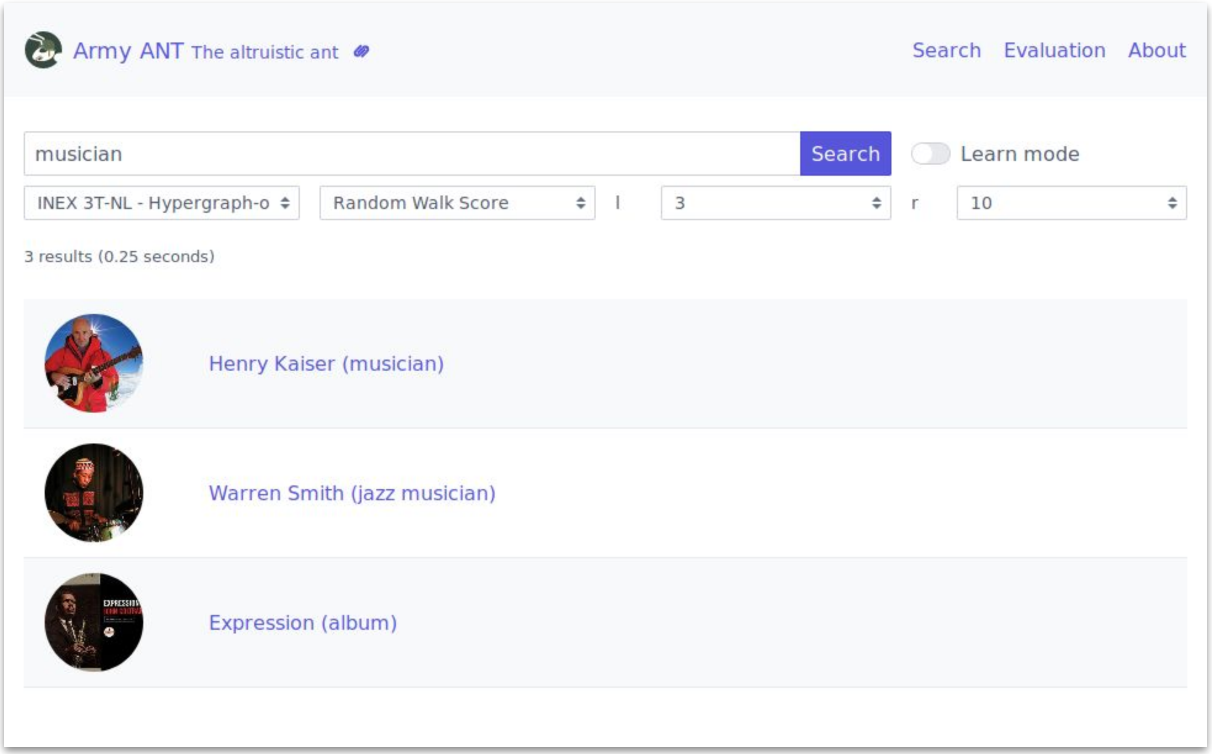
Example file

Hypergraph-of-entity index and ranking functions configuration.

```
engines:
  hgoe-inex-3t-nl:
    name: INEX 3T-NL - Hypergraph-of-Entity
    db:
      name: inex
    index:
      type: hgoe
      location: /home/army-ant/data/indexes/hgoe
      preload: true
    ranking:
      default:
        id: random_walk
        params:
          l: 2
          r: 10
      functions:
        jaccard:
          name: Jaccard Score
        random_walk:
          name: Random Walk Score
          params:
            l: [1, 2, 3, 4, 5, 6]
            r: [10, 25, 50, 100, 1000]
```

Web Interface

Searching, learning and evaluating.



musician

INEX 3T-NL - Hypergraph-o

Random Walk Score

l 3

r 10

3 results (0.25 seconds)

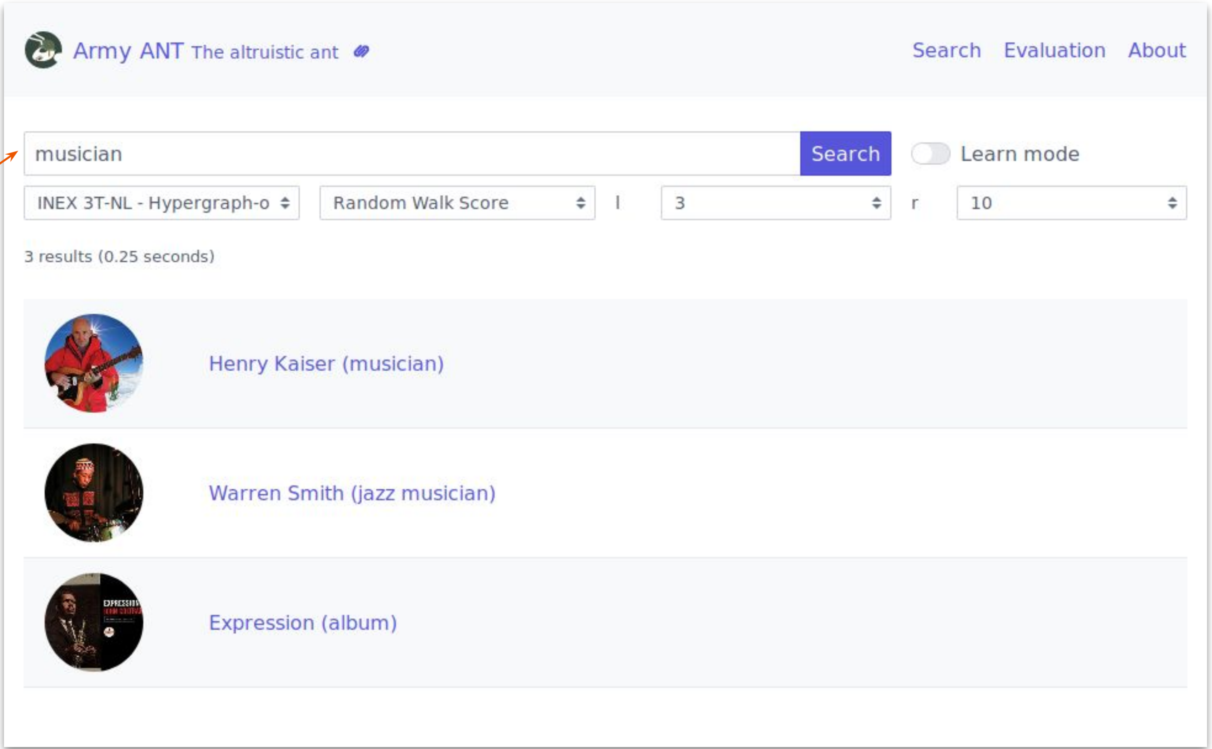
Henry Kaiser (musician)

Warren Smith (jazz musician)

Expression (album)

1. **Search interface**, showing a query over an hypergraph-of-entity index, using the random walk score as the ranking function, with $\ell = 3$ and $r = 10$.

There's something new over here, that we will show in the demo.


Army ANT The altruistic ant

Search Evaluation About

musician

Search

Learn mode

INEX 3T-NL - Hypergraph-o

Random Walk Score

l 3

r 10

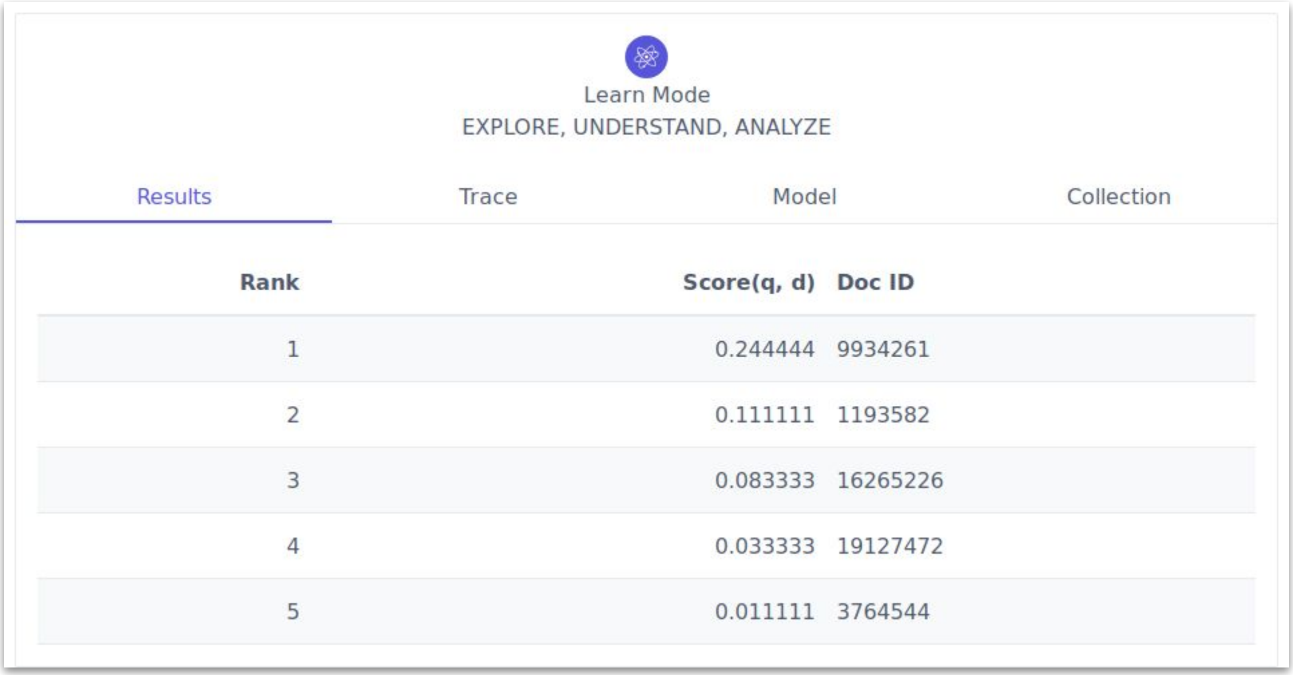
3 results (0.25 seconds)

Henry Kaiser (musician)

Warren Smith (jazz musician)

Expression (album)

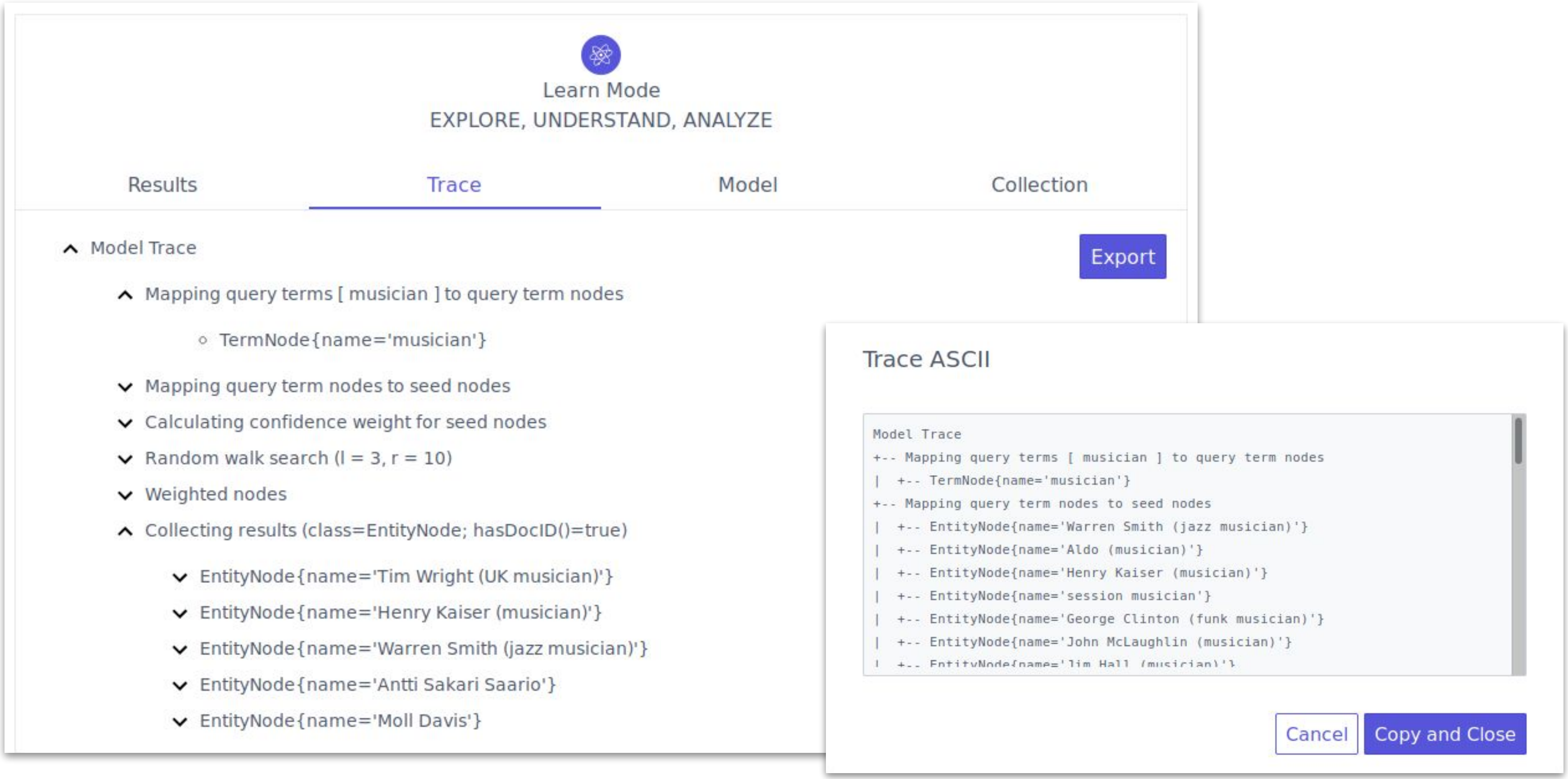
1. **Search interface**, showing a query over an hypergraph-of-entity index, using the random walk score as the ranking function, with $\ell = 3$ and $r = 10$.



Learn Mode
EXPLORE, UNDERSTAND, ANALYZE

Rank	Score(q, d)	Doc ID
1	0.244444	9934261
2	0.111111	1193582
3	0.083333	16265226
4	0.033333	19127472
5	0.011111	3764544

2. **Learn mode**, showing the results with the ranking, score and document ID.



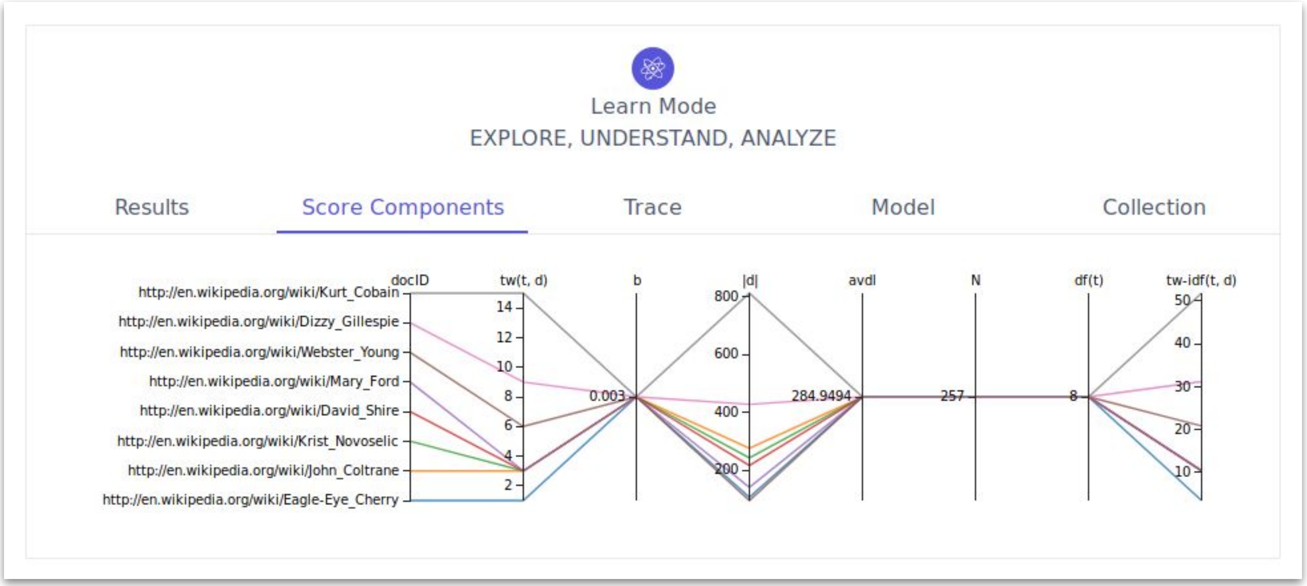
Model Trace

- Mapping query terms [musician] to query term nodes
  - TermNode{name='musician'}
- Mapping query term nodes to seed nodes
- Calculating confidence weight for seed nodes
- Random walk search (l = 3, r = 10)
- Weighted nodes
- Collecting results (class=EntityNode; hasDocID()=true)
  - EntityNode{name='Tim Wright (UK musician)'}
  - EntityNode{name='Henry Kaiser (musician)'}
  - EntityNode{name='Warren Smith (jazz musician)'}
  - EntityNode{name='Antti Sakari Saario'}
  - EntityNode{name='Moll Davis'}

Trace ASCII

```
Model Trace
+-- Mapping query terms [ musician ] to query term nodes
|   +-- TermNode{name='musician'}
+-- Mapping query term nodes to seed nodes
|   +-- EntityNode{name='Warren Smith (jazz musician)'}
|   +-- EntityNode{name='Aldo (musician)'}
|   +-- EntityNode{name='Henry Kaiser (musician)'}
|   +-- EntityNode{name='session musician'}
|   +-- EntityNode{name='George Clinton (funk musician)'}
|   +-- EntityNode{name='John McLaughlin (musician)'}
|   +-- EntityNode{name='Jim Hall (musician)'}
```

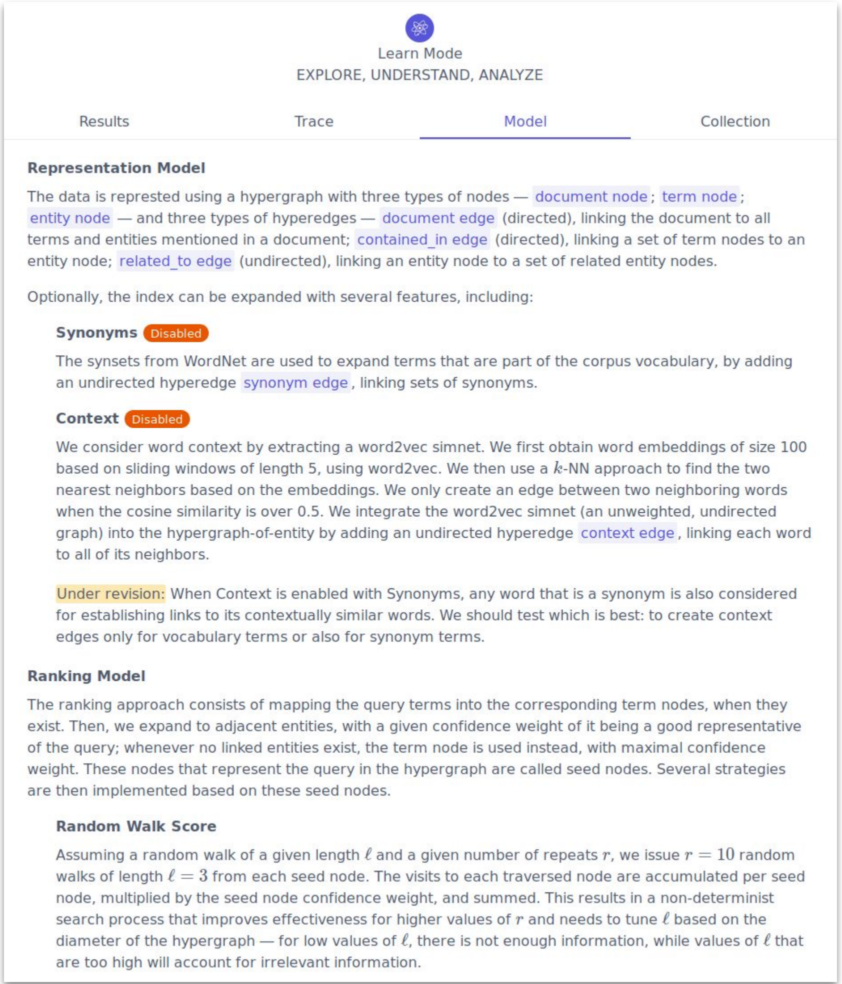
3. **Learn mode**, showing a trace (“instanced explain”) and respective ASCII export for the hypergraph-of-entity engine.


Learn Mode
EXPLORE, UNDERSTAND, ANALYZE
Results
Score Components
Trace
Model
Collection
docID
tw(t, d)
b
|d|
avdl
N
df(t)
tw-idf(t, d)
http://en.wikipedia.org/wiki/Kurt_Cobain
http://en.wikipedia.org/wiki/Dizzy_Gillespie
http://en.wikipedia.org/wiki/Webster_Young
http://en.wikipedia.org/wiki/Mary_Ford
http://en.wikipedia.org/wiki/David_Shire
http://en.wikipedia.org/wiki/Krist_Novoselic
http://en.wikipedia.org/wiki/John_Coltrane
http://en.wikipedia.org/wiki/Eagle-Eye_Cherry
0.003
284.9494
257
8

4. **Learn mode**, showing the score components visualization based on the parallel coordinates system. Displayed score components are based on the graph-of-entity and the entity weight ranking function.

5. **Learn mode:** model

Illustrated with the description of the hypergraph-of-entity, dynamically showing which index extensions are enabled and which parameter values were used in the active query.


Learn Mode
EXPLORE, UNDERSTAND, ANALYZE

Representation Model

The data is represted using a hypergraph with three types of nodes — document node; term node; entity node — and three types of hyperedges — document edge (directed), linking the document to all terms and entities mentioned in a document; contained_in edge (directed), linking a set of term nodes to an entity node; related_to edge (undirected), linking an entity node to a set of related entity nodes.

Optionally, the index can be expanded with several features, including:

Synonyms Disabled

The synsets from WordNet are used to expand terms that are part of the corpus vocabulary, by adding an undirected hyperedge synonym edge, linking sets of synonyms.

Context Disabled

We consider word context by extracting a word2vec simnet. We first obtain word embeddings of size 100 based on sliding windows of length 5, using word2vec. We then use a k-NN approach to find the two nearest neighbors based on the embeddings. We only create an edge between two neighboring words when the cosine similarity is over 0.5. We integrate the word2vec simnet (an unweighted, undirected graph) into the hypergraph-of-entity by adding an undirected hyperedge context edge, linking each word to all of its neighbors.

Under revision: When Context is enabled with Synonyms, any word that is a synonym is also considered for establishing links to its contextually similar words. We should test which is best: to create context edges only for vocabulary terms or also for synonym terms.

Ranking Model

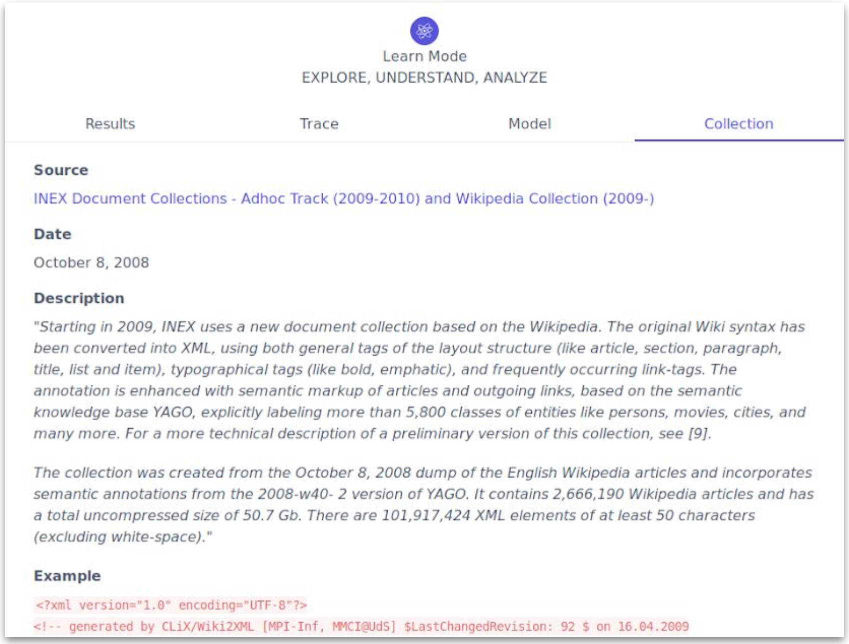
The ranking approach consists of mapping the query terms into the corresponding term nodes, when they exist. Then, we expand to adjacent entities, with a given confidence weight of it being a good representative of the query; whenever no linked entities exist, the term node is used instead, with maximal confidence weight. These nodes that represent the query in the hypergraph are called seed nodes. Several strategies are then implemented based on these seed nodes.

Random Walk Score

Assuming a random walk of a given length ℓ and a given number of repeats r, we issue $r = 10$ random walks of length $\ell = 3$ from each seed node. The visits to each traversed node are accumulated per seed node, multiplied by the seed node confidence weight, and summed. This results in a non-determinist search process that improves effectiveness for higher values of r and needs to tune ℓ based on the diameter of the hypergraph — for low values of ℓ, there is not enough information, while values of ℓ that are too high will account for irrelevant information.

6. **Learn mode:** collection

Shows the description of the indexed collection, including the source of the data, the temporal coverage, a free text description, an example of the data and a reference paper.



Learn Mode
EXPLORE, UNDERSTAND, ANALYZE

Source

INEX Document Collections - Adhoc Track (2009-2010) and Wikipedia Collection (2009-)

Date

October 8, 2008

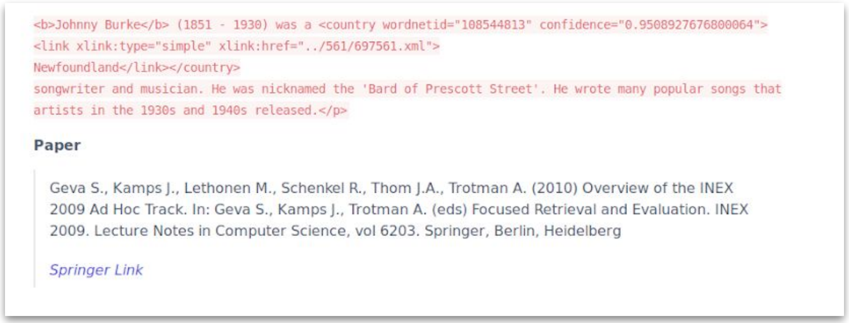
Description

"Starting in 2009, INEX uses a new document collection based on the Wikipedia. The original Wiki syntax has been converted into XML, using both general tags of the layout structure (like article, section, paragraph, title, list and item), typographical tags (like bold, emphatic), and frequently occurring link-tags. The annotation is enhanced with semantic markup of articles and outgoing links, based on the semantic knowledge base YAGO, explicitly labeling more than 5,800 classes of entities like persons, movies, cities, and many more. For a more technical description of a preliminary version of this collection, see [9].

The collection was created from the October 8, 2008 dump of the English Wikipedia articles and incorporates semantic annotations from the 2008-w40- 2 version of YAGO. It contains 2,666,190 Wikipedia articles and has a total uncompressed size of 50.7 Gb. There are 101,917,424 XML elements of at least 50 characters (excluding white-space)."

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- generated by CLiX/Wiki2XML [MPI-Inf, MMCI@UdS] $LastChangedRevision: 92 $ on 16.04.2009
```

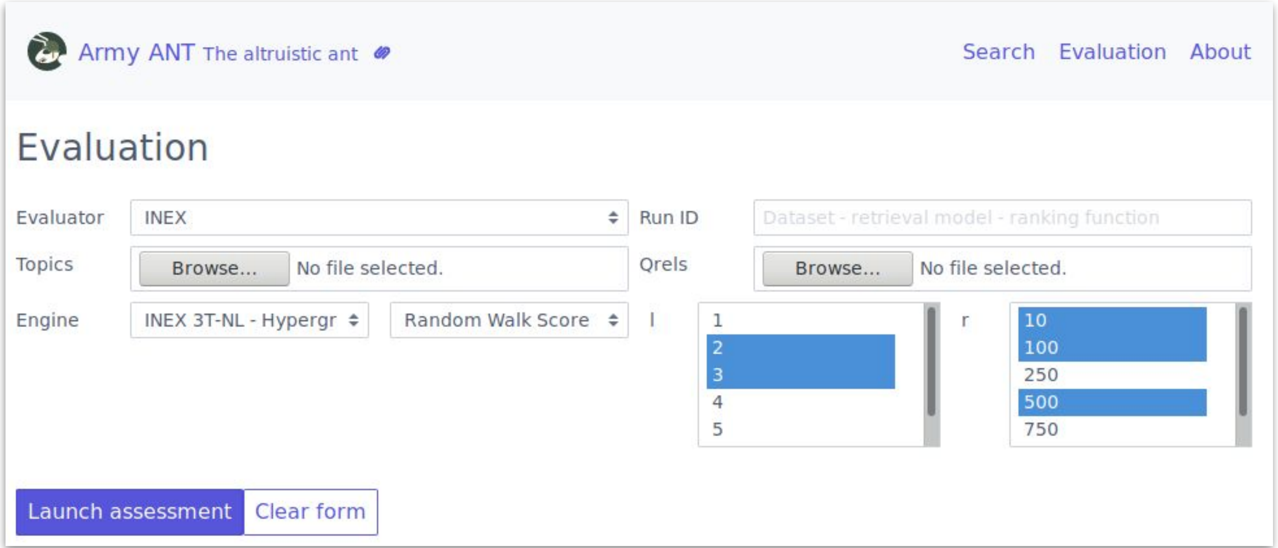


```
<b>Johnny Burke</b> (1851 - 1930) was a <country wordnetid="108544813" confidence="0.9508927676800064">
<link xlink:type="simple" xlink:href="../561/697561.xml">
Newfoundland</link></country>
songwriter and musician. He was nicknamed the 'Bard of Prescott Street'. He wrote many popular songs that
artists in the 1930s and 1940s released.</p>
```

Paper

Geva S., Kamps J., Lethonen M., Schenkel R., Thom J.A., Trotman A. (2010) Overview of the INEX 2009 Ad Hoc Track. In: Geva S., Kamps J., Trotman A. (eds) Focused Retrieval and Evaluation. INEX 2009. Lecture Notes in Computer Science, vol 6203. Springer, Berlin, Heidelberg

Springer Link



Army ANT The altruistic ant

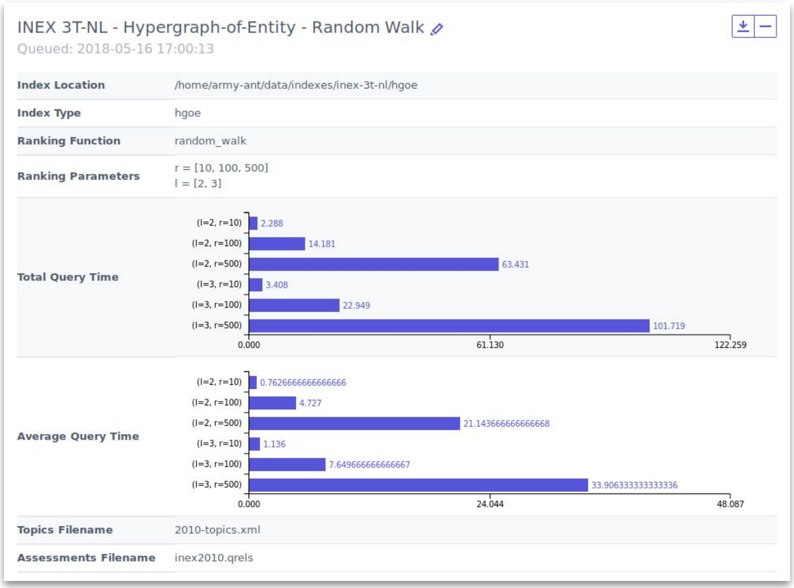
Search Evaluation About

Evaluation

Evaluator	INEX	Run ID	Dataset - retrieval model - ranking function
Topics	Browse... No file selected.	Qrels	Browse... No file selected.
Engine	INEX 3T-NL - Hypergr / Random Walk Score	l	1, [✓ 2], [✓ 3], 4, 5
		r	[✓ 10], [✓ 100], 250, [✓ 500], 750

Launch assessment Clear form

7. **Evaluation module**, showing the task launching form.



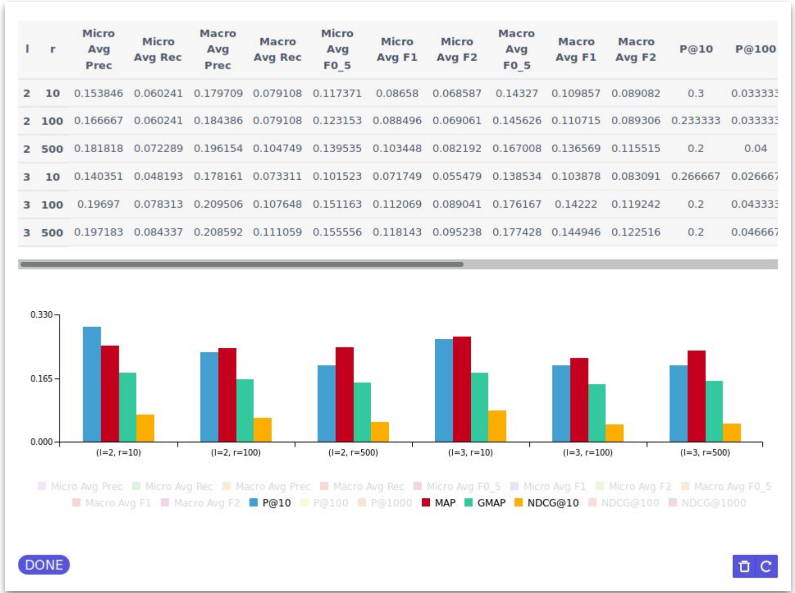
INEX 3T-NL - Hypergraph-of-Entity - Random Walk

Queued: 2018-05-16 17:00:13

Index Location	/home/army-ant/data/indexes/inex-3t-nl/hgoe
Index Type	hgoe
Ranking Function	random_walk

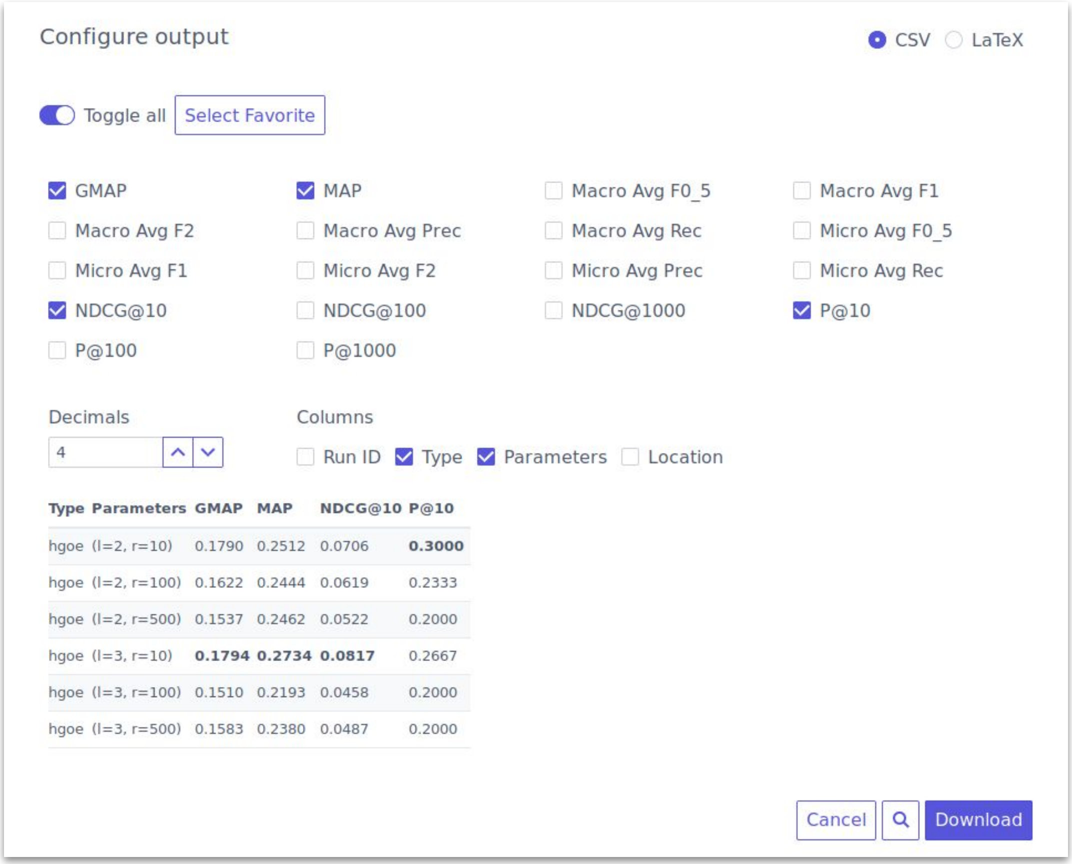
Ranking Parameters	r = [10, 100, 500]
	l = [2, 3]

Total Query Time	
Average Query Time	
Topics Filename	2010-topics.xml
Assessments Filename	inex2010.qrels



l	r	Micro Avg Prec	Micro Avg Rec	Macro Avg Prec	Macro Avg Rec	Micro Avg F0_5	Micro Avg F1	Micro Avg F2	Macro Avg F0_5	Macro Avg F1	Macro Avg F2	P@10	P@100
2	10	0.153846	0.060241	0.179709	0.079108	0.117371	0.08658	0.068587	0.14327	0.109857	0.089082	0.3	0.03333
2	100	0.166667	0.060241	0.184386	0.079108	0.123153	0.088496	0.069061	0.145626	0.110715	0.089306	0.233333	0.03333
2	500	0.181818	0.072289	0.196154	0.104749	0.139535	0.103448	0.082192	0.167008	0.136569	0.115515	0.2	0.04
3	10	0.140351	0.048193	0.178161	0.073311	0.101523	0.071749	0.055479	0.138534	0.103878	0.083091	0.266667	0.02666
3	100	0.19697	0.078313	0.209506	0.107648	0.151163	0.112069	0.089041	0.176167	0.14222	0.119242	0.2	0.04333
3	500	0.197183	0.084337	0.208592	0.111059	0.155556	0.118143	0.095238	0.177428	0.144946	0.122516	0.2	0.04666

8. **Evaluation module**, showing a finished task and its expanded results.



Configure output

[✓ CSV] LaTeX

[✓ Toggle all]

[✓ GMAP]	[✓ MAP]	Macro Avg F0_5	Macro Avg F1
Macro Avg F2	Macro Avg Prec	Macro Avg Rec	Micro Avg F0_5
Micro Avg F1	Micro Avg F2	Micro Avg Prec	Micro Avg Rec
[✓ NDCG@10]	NDCG@100	NDCG@1000	[✓ P@10]
P@100	P@1000		

Decimals: 4

Columns: Run ID [✓ Type] [✓ Parameters] Location

Type	Parameters	GMAP	MAP	NDCG@10	P@10
hgoe	(l=2, r=10)	0.1790	0.2512	0.0706	**0.3000**
hgoe	(l=2, r=100)	0.1622	0.2444	0.0619	0.2333
hgoe	(l=2, r=500)	0.1537	0.2462	0.0522	0.2000
hgoe	(l=3, r=10)	**0.1794**	**0.2734**	**0.0817**	0.2667
hgoe	(l=3, r=100)	0.1510	0.2193	0.0458	0.2000
hgoe	(l=3, r=500)	0.1583	0.2380	0.0487	0.2000

9. **Evaluation module**, showing the global evaluation export, for comparing the results of multiple parameter configurations among different tasks.

Conclusions

Final remarks and interactive demo.

Final remarks

- The ANT search engine is serving the local academic community and giving us a test platform.
- Army ANT is serving the research needs in the area and supporting my PhD.
- By the way, my thesis topic is “Graph-Based Entity-Oriented Search”:
    - If you’re interested on the topic, feel free to look me up on ANT and contact me.
    - I’m also exploring hypergraphs as an alternative, higher-level, representation model.
    - The goal is to integrate text and knowledge in a joint model.
    - And to provide a generalized model to support entity-oriented search tasks.

This work has been developed at FEUP InfoLab and INESC TEC and I have been supported by research grant PD/BD/128160/2016, provided by the Portuguese funding agency, Fundação para a Ciência e a Tecnologia (FCT).

Installing Army ANT demo using Docker

- First install Docker Compose:
  - https://docs.docker.com/compose/install/
- And then get Army ANT install repository:
  - https://github.com/feup-infolab/army-ant-install
- Follow the instructions to launch Army ANT and explore the included Lucene and hypergraph-of-entity indexes:
  - git clone git@github.com:feup-infolab/army-ant-install.git
  - cd army-ant-install
  - git checkout ieee-syp-2018
  - docker-compose up

Thank you!

https://ant.fe.up.pt
https://github.com/feup-infolab/army-ant