The Semantic Knowledge Graph:

A compact, auto-generated model for real-time traversal and ranking of any relationship within a domain

Trey Grainger
SVP of Engineering
Lucidworks

Khalifeh AlJadda
Lead Data Scientist
CareerBuilder

Mohammed Korayem
Data Scientist
CareerBuilder

Andries Smith
Software Engineer
CareerBuilder

About Me

Trey Grainger

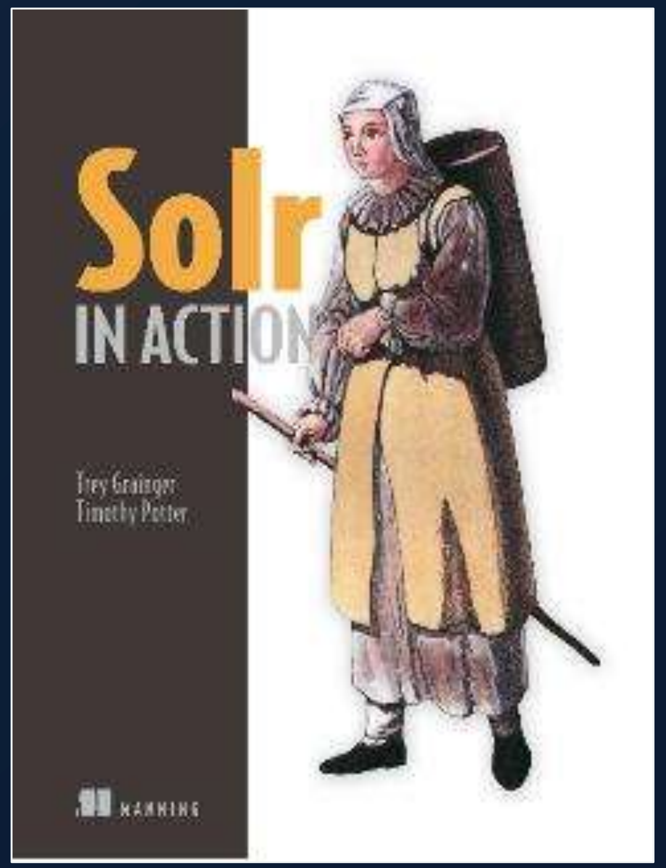
SVP of Engineering

Lucidworks

- Previously Director of Engineering @ **CareerBuilder**
- MBA, **Management of Technology – Georgia Tech**
- BA, **Computer Science**, **Business**, & **Philosophy – Furman University**
- **Information Retrieval & Web Search - Stanford University**

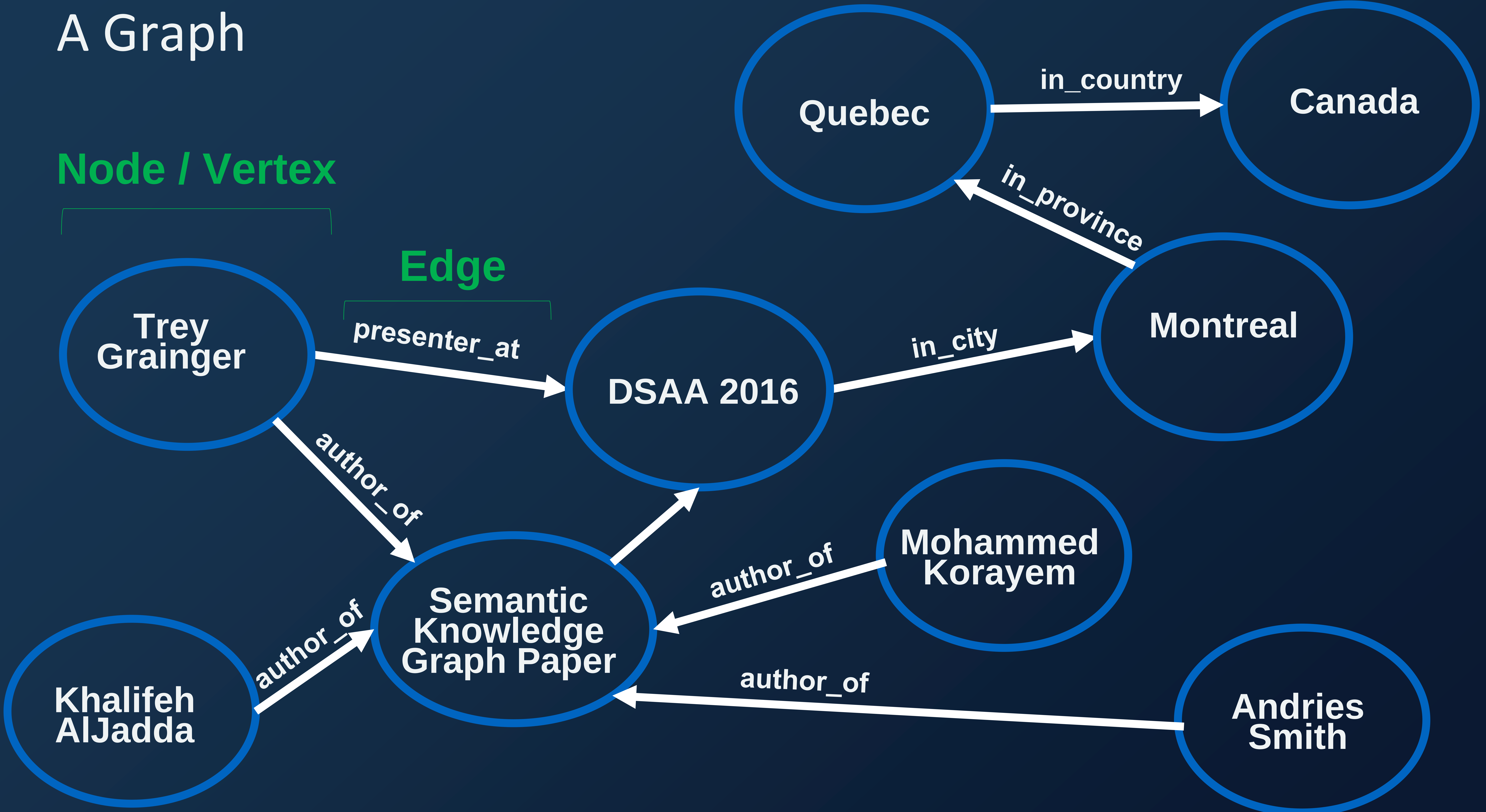
Fun outside of CB:

- Co-author of *Solr in Action*, plus a handful of research papers
- Frequent conference speaker
- Founder of Celiaccess.com, the gluten-free search engine
- Lucene/Solr contributor



Terminology / Background

A Graph

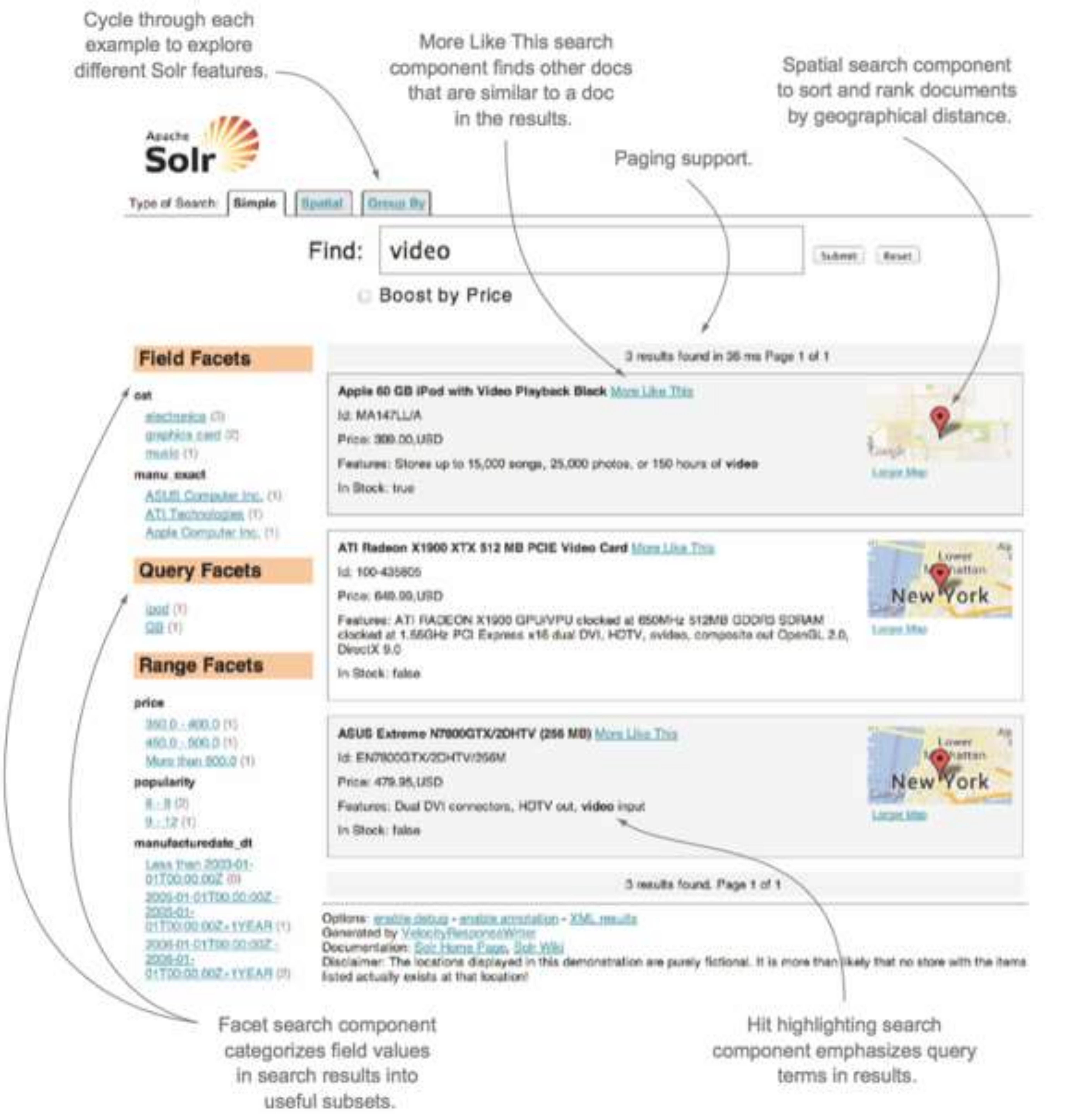


Node / Vertex
Edge
Trey Grainger
presenter_at
DSAA 2016
in_city
Montreal
in_province
Quebec
in_country
Canada
author_of
Semantic Knowledge Graph Paper
author_of
Mohammed Korayem
author_of
Khalifeh AlJadda
author_of
Andries Smith

“Solr is the popular, blazing-fast, open source enterprise search platform built on Apache Lucene™.”

Key Solr Features:

- Multilingual Keyword search
- Relevancy Ranking of results
- Faceting & Analytics
- Highlighting
- Spelling Correction
- Autocomplete/Type-ahead Prediction
- Sorting, Grouping, Deduplication
- Distributed, Fault-tolerant, Scalable
- Geospatial search
- Complex Function queries
- Recommendations (More Like This)
- … many more



**source: Solr in Action, chapter 2*

The inverted index

What you SEND to Lucene/Solr:

Document	Content Field
doc1	once upon a time, in a land far, far away
doc2	the cow jumped over the moon.
doc3	the quick brown fox jumped over the lazy dog.
doc4	the cat in the hat
doc5	The brown cow said “moo” once.
…	…

How the content is INDEXED into Lucene/Solr (conceptually):

Term	Documents
a	doc1 [2x]
brown	doc3 [1x] , doc5 [1x]
cat	doc4 [1x]
cow	doc2 [1x] , doc5 [1x]
…	…
once	doc1 [1x], doc5 [1x]
over	doc2 [1x], doc3 [1x]
the	doc2 [2x], doc3 [2x], doc4[2x], doc5 [1x]
…	…

Matching queries to documents

IEEE/ACM DSAA'2016
17 - 19 Oct 2016, Montréal, Canada

*/solr/select/?q=*apache solr

Term	Documents
…	…
apache	doc1, doc3, doc4, doc5
…	
hadoop	doc2, doc4, doc6
…	…
solr	doc1, doc3, doc4, doc7, doc8
…	…

Related Work

Related Work

- Primarily related to ontology Learning.

- Recently, large-scale knowledge bases that utilize ontologies (FreeBase [4], DBpedia [5], and YAGO [6, 7]) have been constructed using structured sources such as Wikipedia infoboxes.

- Other approaches (DeepDive [8], Nell2RDF [9], and PROSPERA [10]) crawl the web and use machine learning and natural language processing to build web-scale knowledge graphs.

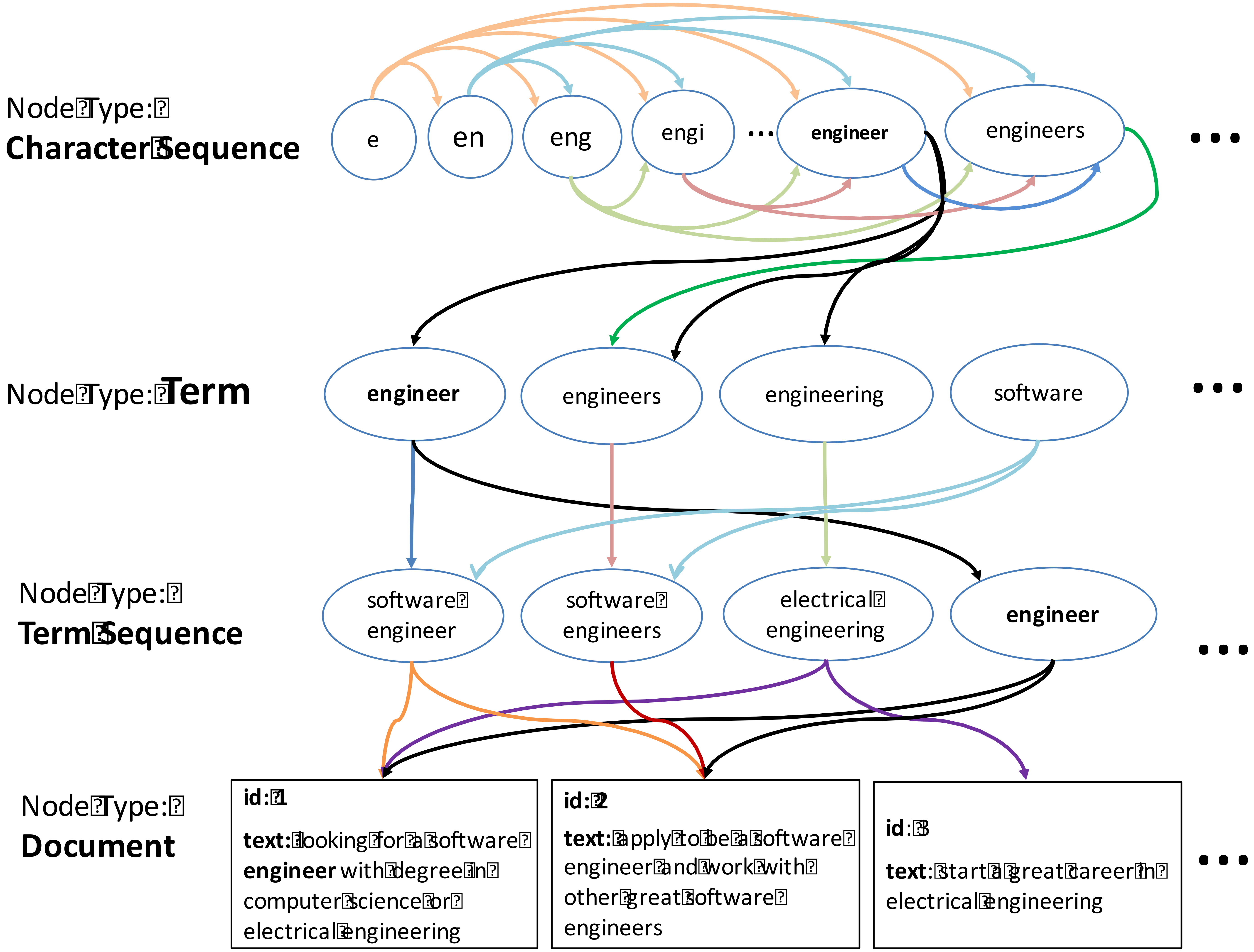
Problem Description

Challenges we are solving

Because current knowledge bases / ontology learning systems typically requires **explicitly modeling nodes and edges into a graph** ahead of time, this unfortunately **presents several limitations** to the use of such a knowledge graph:

- **Entities not modeled explicitly** as nodes **have no known relationships** to any other entities.
- Edges exist between nodes, but not between arbitrary combinations of nodes, and therefore such a graph is **not ideal for representing nuanced meanings of an entity** when appearing within different contexts, as is common within natural language.
- Substantial **meaning is encoded in the linguistic representation of the domain that is lost** when the underlying textual representation is not preserved: phrases, interaction of concepts through actions (i.e. verbs), positional ordering of entities and the phrases containing those entities, variations in spelling and other representations of entities, the use of adjectives to modify entities to represent more complex concepts, and aggregate frequencies of occurrence for different representations of entities relative to other representations.
- It can be **an arduous process to create robust ontologies**, map a domain into a graph representing those ontologies, **and ensure the generated graph is compact**, **accurate**, **comprehensive**, and **kept up to date**.

Semantic Data Encoded into Free Text Content



Model

Knowledge Graph

Documents

id: 1
job_title: Software Engineer
desc: software engineer at a great company
skills: .Net, C#, java

id: 2
job_title: Registered Nurse
desc: a registered nurse at hospital doing hard work
skills: oncology, phlebotemy

id: 3
job_title: Java Developer
desc: a software engineer or a java engineer doing work
skills: java, scala, hibernate

Docs-Terms Uninverted Index

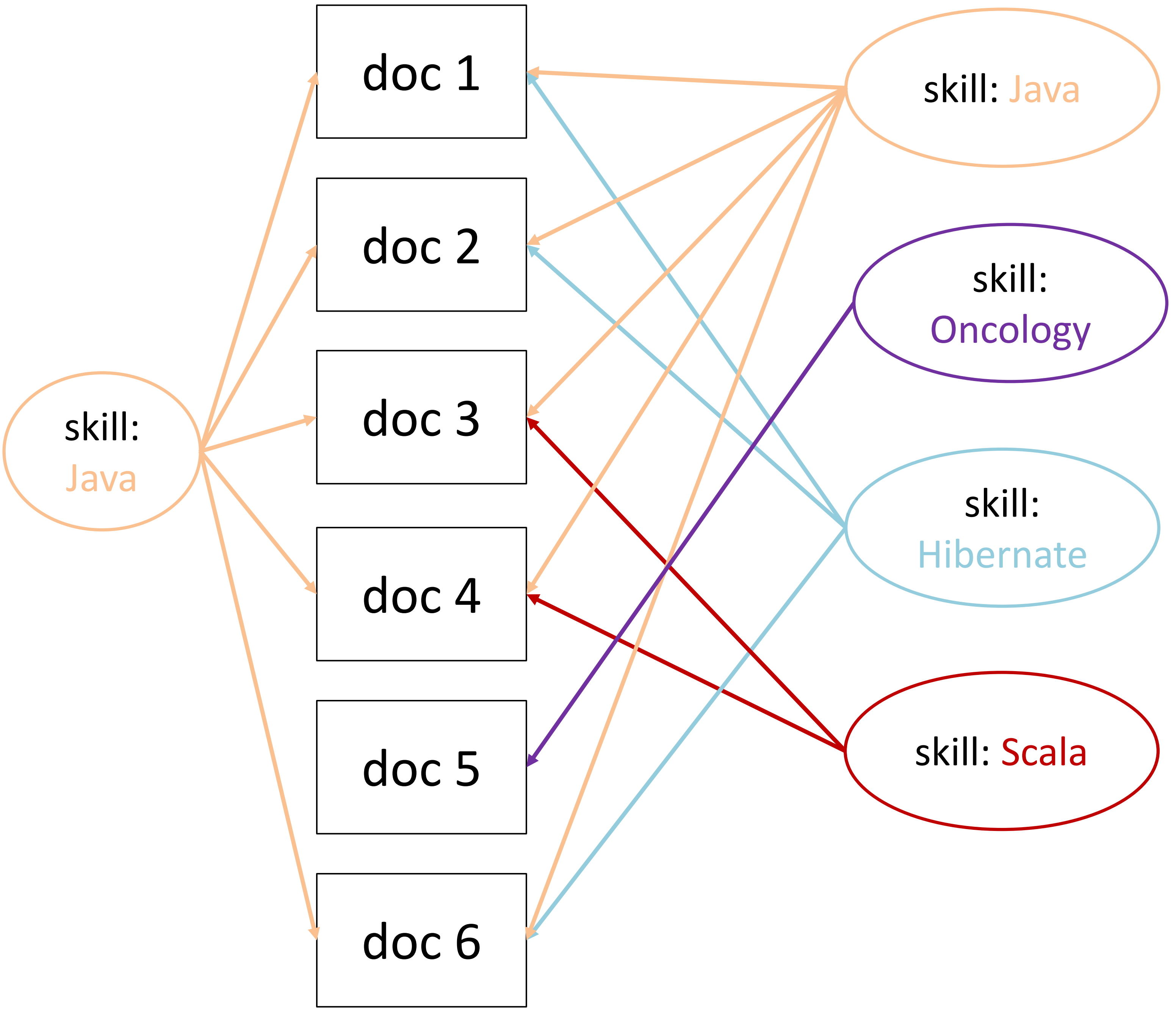
field	doc	term
desc	1	a
		at
		company
		engineer
		great
		software
	2	a
		at
		doing
		hard
		hospital
		nurse
		registered
		work
	3	a
		doing
		engineer
		java
		or
		software
		work
job_title	1	Software Engineer
…	…	…

Terms-Docs Inverted Index

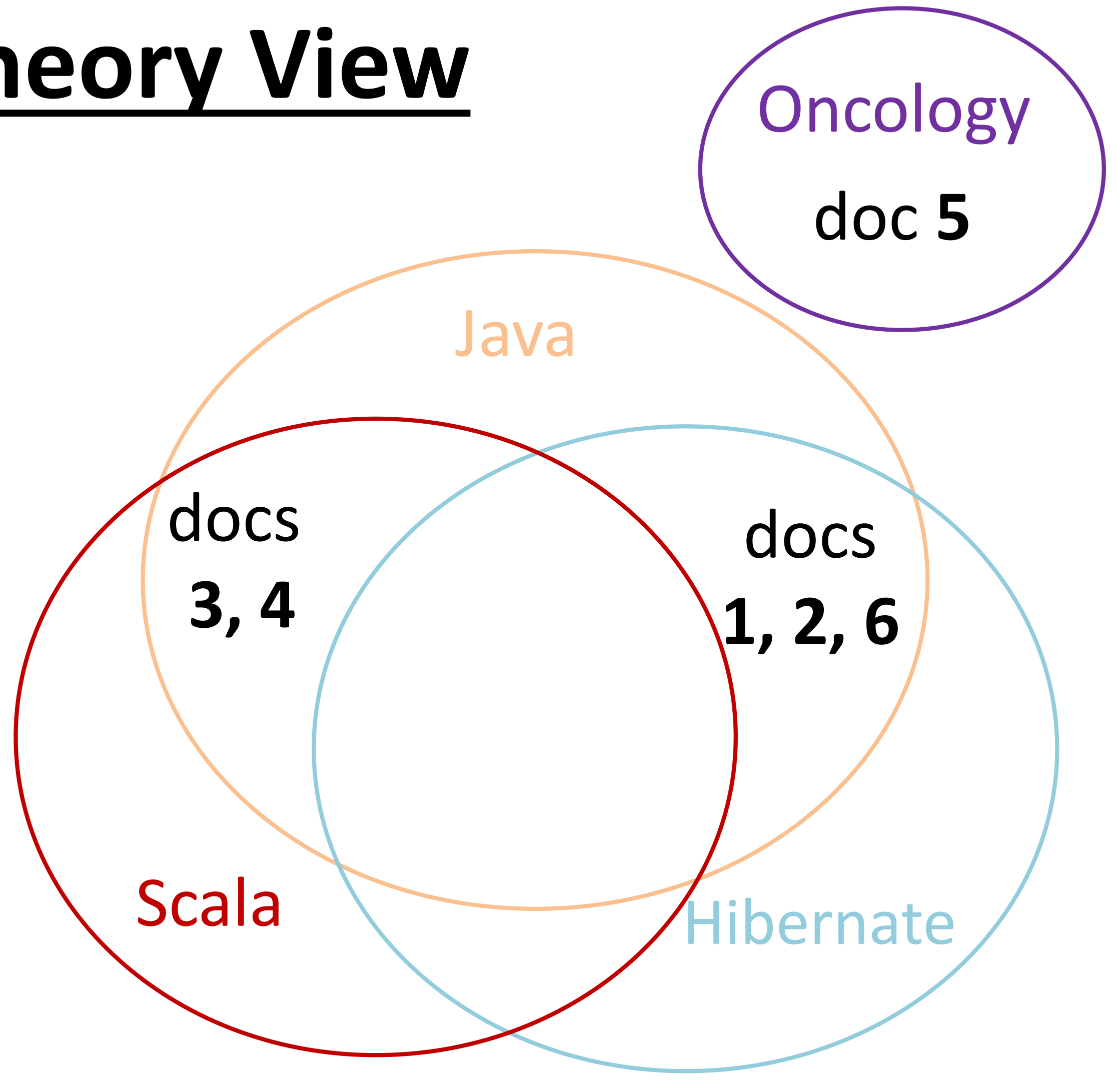
field	term	postings list	
		doc	pos
desc	a	1	4
		2	1
		3	1, 5
	at	1	3
		2	4
	company	1	6
	doing	2	6
		3	8
	engineer	1	2
		3	3, 7
	great	1	5
	hard	2	7
	hospital	2	5
	java	3	6
	nurse	2	3
	or	3	4
	registered	2	2
	software	1	1
		3	2
	work	2	10
		3	9
job_title	java developer	3	1
…	…	…	…

How the Graph Traversal Works

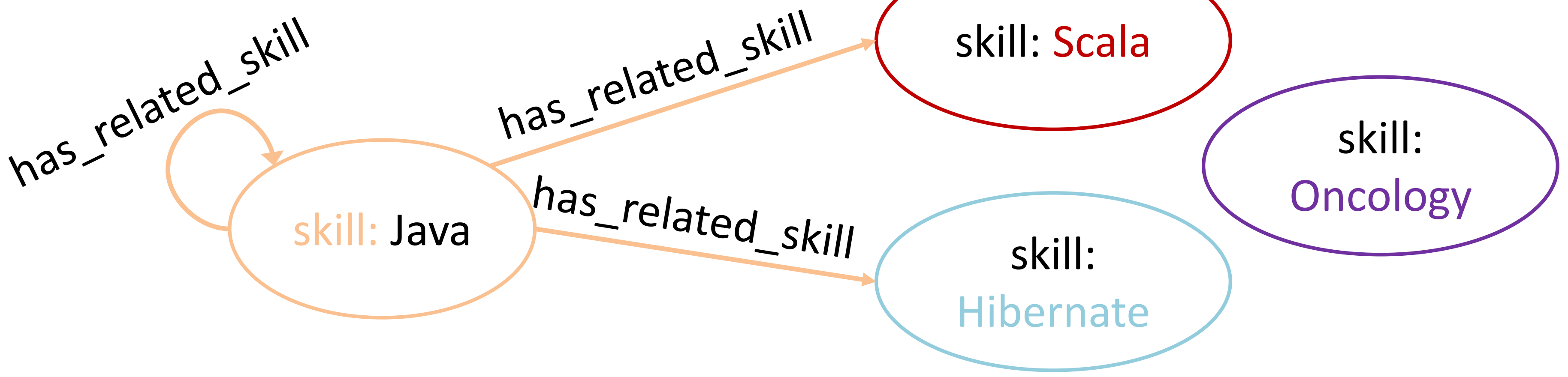
Data Structure View



Set-theory View



Graph View



Graph Model

Structure:

Consider an undirected graph $G = (V, E)$ where V and $E \subset V \times V$ denote the sets of nodes and edges, respectively. We define the following:

- $D = \{d_1, d_2, ..., d_m\}$ is a set of documents that represent a corpus that the Semantic Knowledge Graph will utilize to extract and score semantic relationships.
- $X = \{x_1, x_2, ..., x_k\}$ is a set of all items stored in D. These items could be keywords, phrases, or any arbitrary linguistic representation found within D.
- $d_i = \{x | x \in X\}$ where we can think of each document $d \in D$ as a set of items.
- $T = \{t_1, t_2, ...t_n\}$ where t_i is a tag which assigns an entity type to an item such as keyword, title, location, company, school, person, etc.

Given the previous notations, the set of nodes V in our graph can be defined as $V = \{v_1, v_2, .., v_n\}$ where v_i stores an item $x_i \in X$ tagged with tag $t_j \in T$. While $D_{v_i} = \{d | x_i \in d, d \in D\}$ is a set of documents that contains item x_i with its appropriate tag t_j. Finally, we define e_{ij} as an edge between (v_i, v_j) with a function $f(e_{ij}) = \{d \in D_{v_i} \cap D_{v_j}\}$ that stores on each edge the set of documents that contain both items x_i and x_j with their tags. On the other hand, we define $g(e_{ij}, v_k) = \{d : d \in f(e_{ij}) \cap D_{v_k}\}$ that stores on the edge e_{jk} the common set of documents between $f(e_{ij})$ and D_k.

Single-level Traversal / Scoring:

The simple use case for scoring semantic relationships is to score directly connected nodes v_i and v_j. In this case we query the terms-docs inverted index for item x_i tagged with t_j, and as a result we get back D_{vi}. Then we query the terms-docs inverted index again for x_j tagged with t_k to get D_{vj}. An edge e_{ij} will be created between v_i and v_j if $f(e_{ij}) \neq \phi$. We call the D_{vi} our $foreground$ document set D_{FG}, while $D_{BG} \subseteq D$ is our $background$ document set. The hypothesis behind our scoring technique is that if x_i tends to be semantically related to x_j, then the presence of x_j in the $foreground$ document set D_{FG} should be above the average presence of x_j in D_{BG}. We utilize The z score to evaluate this hypothesis:

$$z(v_i, v_j) = \frac{y - n * p}{\sqrt{n * p(1-p)}}$$

Where $n = |D_{FG}|$ is the number of documents in our $foreground$ document set, $p = \frac{|D_{v_j}|}{|D_{BG}|}$ is the probability of finding the term x_j with tag t_k in the $background$ document set, and $y = |f(e_{ij})|$ is the number of documents containing both x_i and x_j.

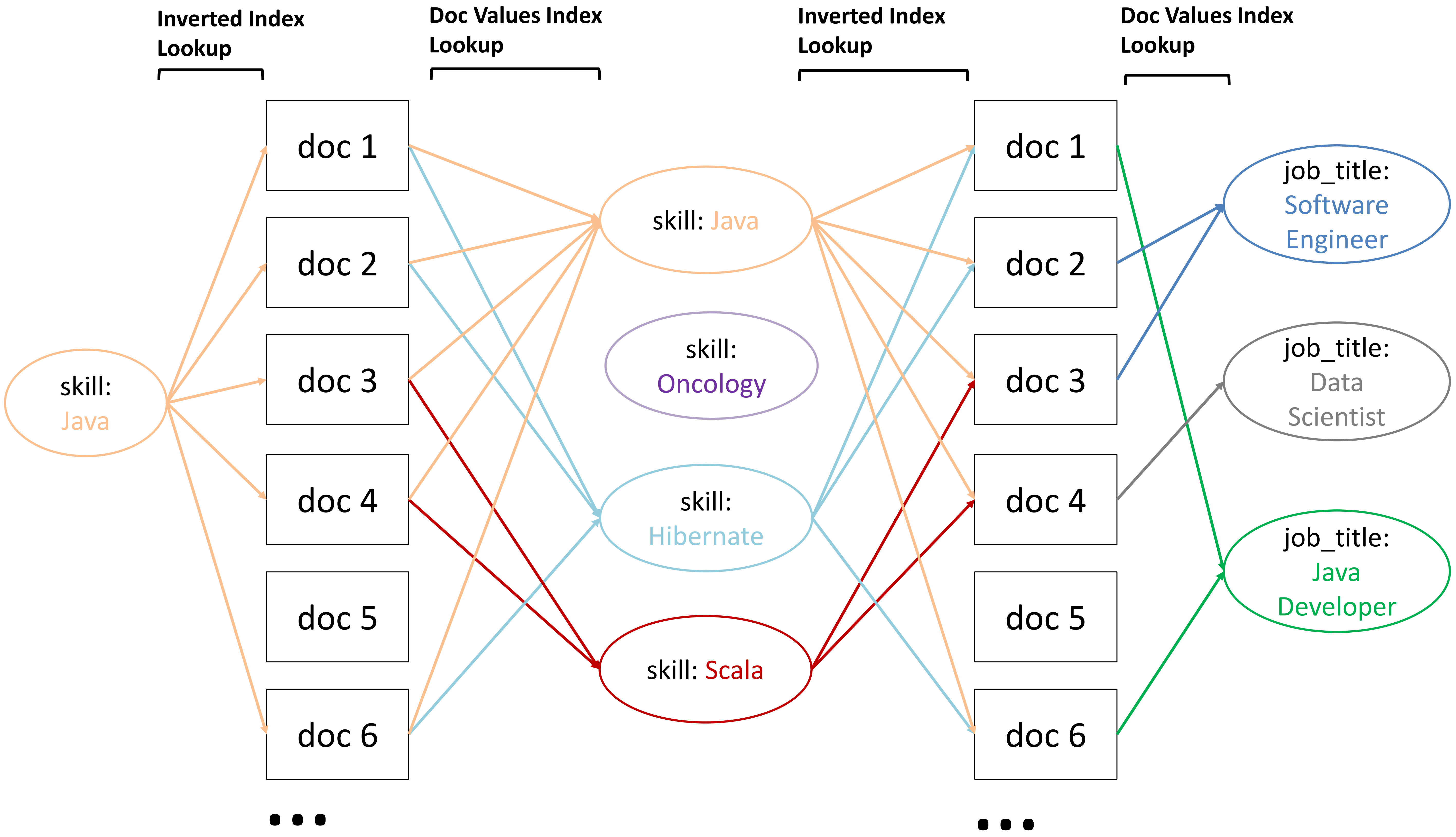
Multi-level Traversal / Scoring:

$$D_{FG} = \begin{cases} f(e_{ij}) & \text{if } n = 3 \\ \{\bigcap_{i=1, j=i+1, k=j+1}^{n-3} g(e_{ij}, D_{v_k})\} & \text{if } n > 3 \end{cases}$$

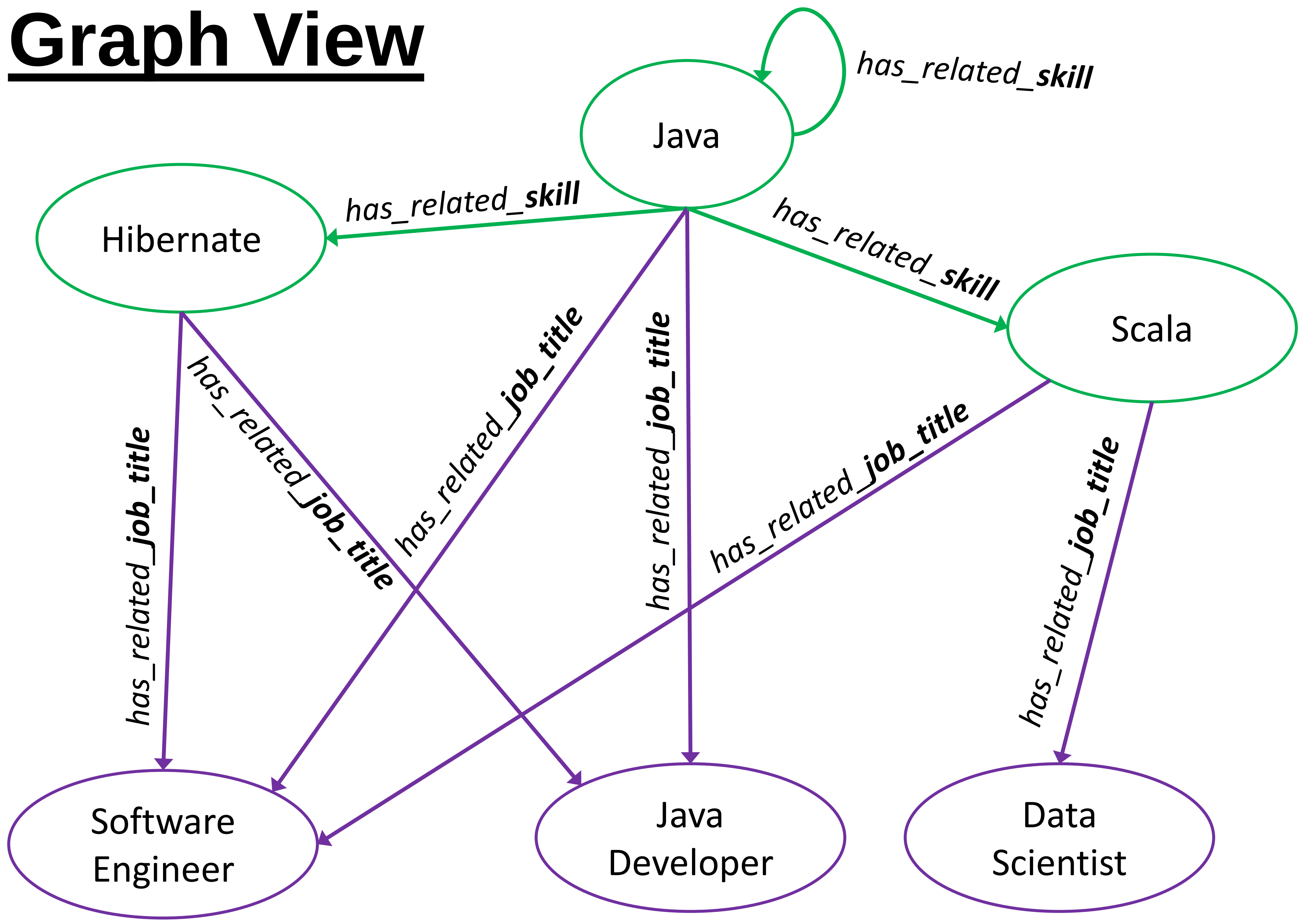
while $y = |D_{FG} \cap D_{v_n}|$. We normalize the z score using a sigmoid function to bring the scores in the range $[-1, 1]$.

Multi-level Traversal

Data Structure View



Graph View



Scoring nodes in the Graph

Knowledge Graph

Foreground vs. Background Analysis

Every term scored against it’s context. The more commonly the term appears within it’s foreground context versus its background context, the more relevant it is to the specified foreground context.

$$z = \frac{countFG(x) - totalDocsFG * probBG(x)}{sqrt(totalDocsFG * probBG(x) * (1 - probBG(x)))}$$

Foreground Query:

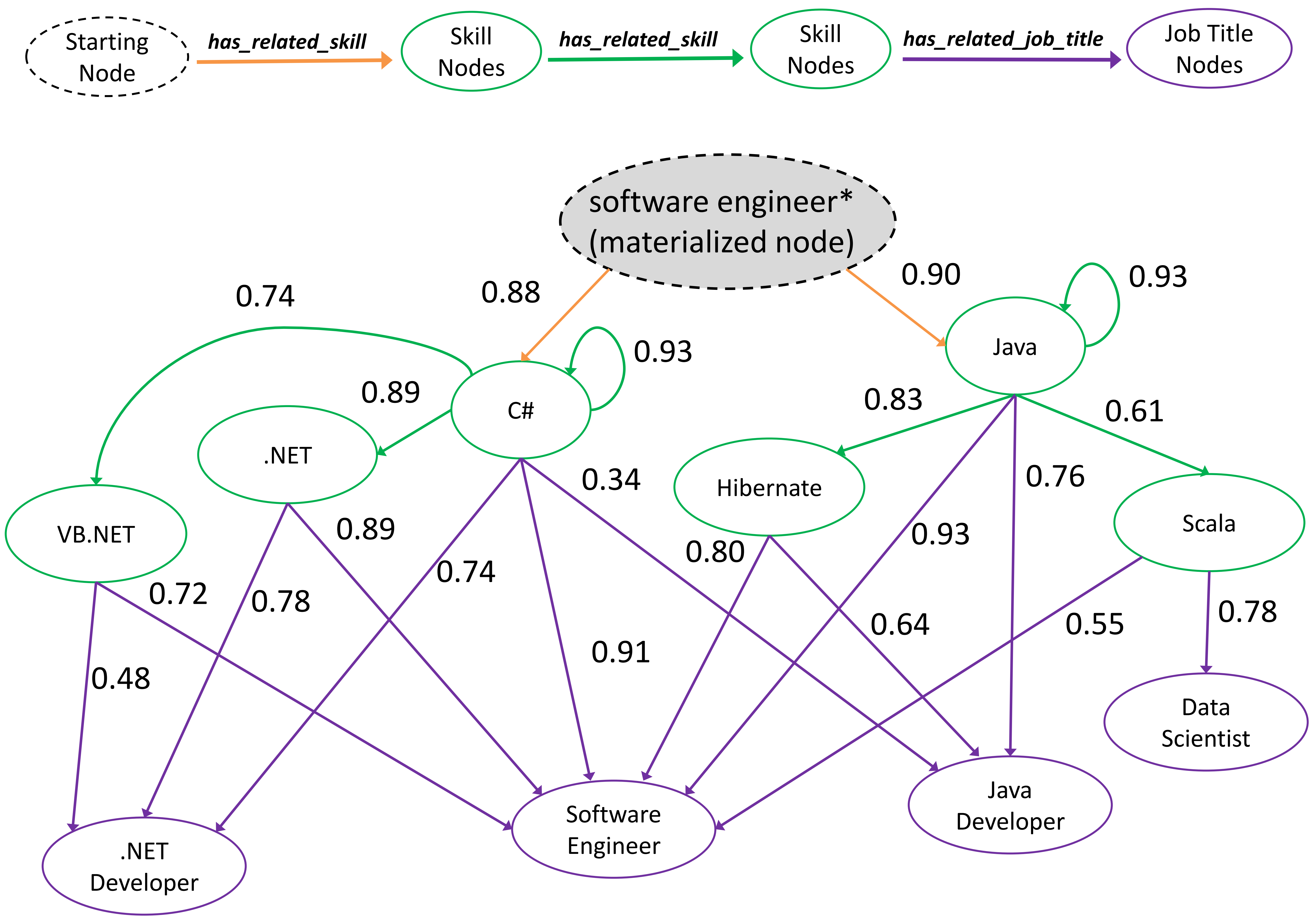
"Hadoop"

+

-

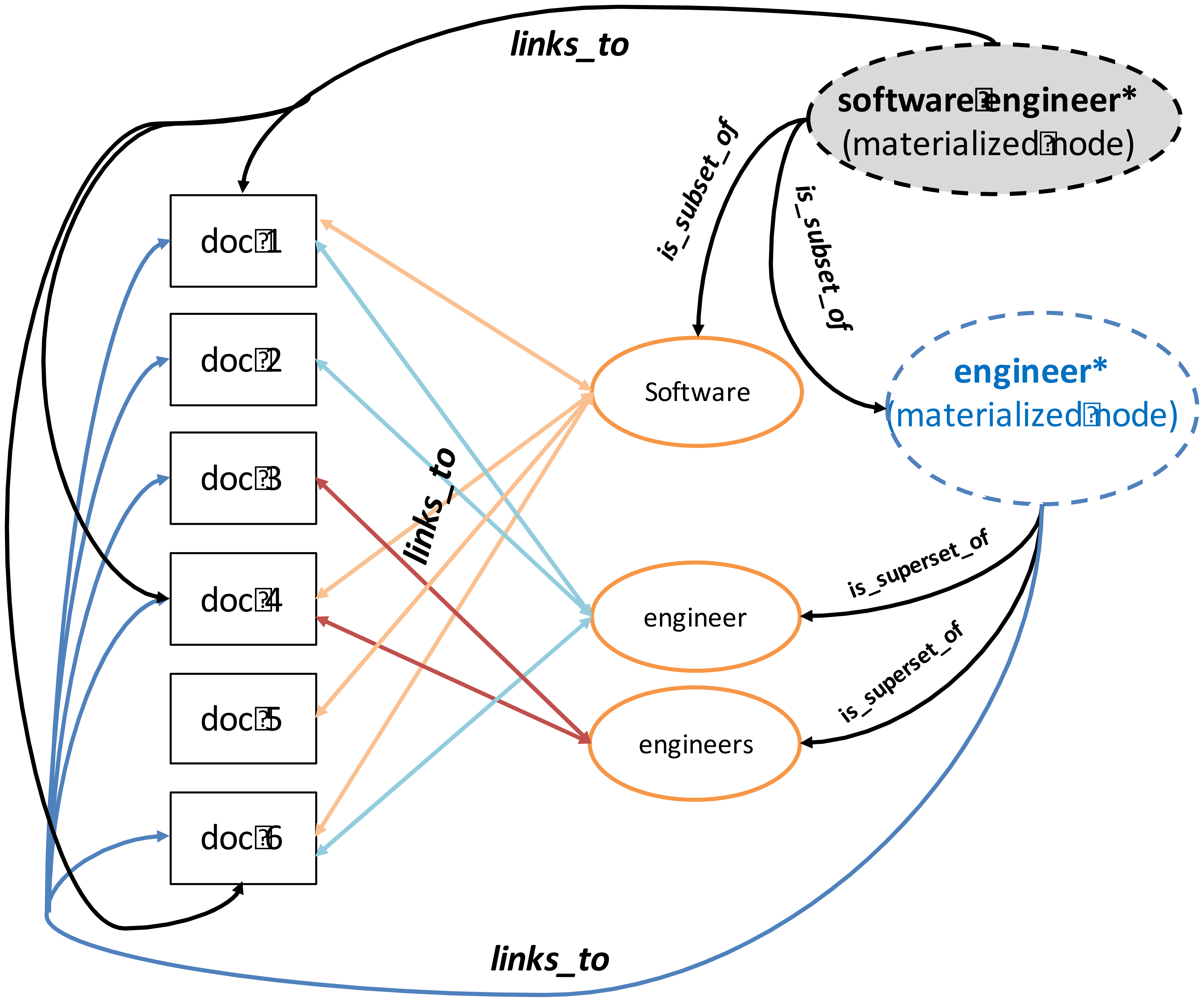
{ "type":"keywords”, "values":[
{ "value":"hive", "relatedness": 0.9765, "popularity":369 },
{ "value":”spark", "relatedness": 0.9634, "popularity":15653 },
{ "value":".net", "relatedness": 0.5417, "popularity":17683 },
{ "value":"bogus_word", "relatedness": 0.0, "popularity":0 },
{ "value":"teaching", "relatedness": -0.1510, "popularity":9923 },
{ "value":"CPR", "relatedness": -0.4012, "popularity":27089 }] }

Multi-level Graph Traversal with Scores



Source: *Trey Grainger, Khalifeh AlJadda, Mohammed Korayem, Andries Smith.“The Semantic Knowledge Graph: A compact, auto-generated model for real-time traversal and ranking of any relationship within a domain”. DSAA 2016.*

Materialization of new nodes through shared documents



Implementation

Open Sourced!

Populating the Graph

```
curl —H 'Content—type:application/json'
http://localhost:8983/solr/semantic—knowledge—graph/update —d
'[{ "id" : "job1",
        "title" : "Data Scientist",
        "skills": ["machine learning","spark"],
        "keywords": "Seeking a senior—level data scientist with experience with
                  spark and machine learning..."},
   { "id" : "job2",
        "title" : "Registered Nurse",
        "skills": ["er","trauma", "phlebotomy"],
        "keywords": "Come join the top—rated hospital in the region..."}
]'
```

GitHub, Inc. [US] https://github.com/careerbuilder/semantic-knowledge-graph

This repository Search Pull requests Issues Gist

careerbuilder / semantic-knowledge-graph

Unwatch 14 Star 15 Fork 7

Code Issues 0 Pull requests 0 Projects 0 Wiki Pulse Graphs Settings

Knowledge Graph

Usage (examples from the job search domain):

Request:

```
curl -X POST http://localhost:8983/solr/skg/rel \
-H "Content-Type: application/json" \
-d \
'{
  "queries": [
    "keywords:\"data scientist\""
  ],
  "compare": [
    {
      "type": "jobtitle",
      "limit": 1,
      "compare": [
        {
          "type": "skills",
          "limit": 5,
          "discover_values": true,
          "values": [
            "java (programming language)"
          ]
        }
      ]
    }
  ]
}'
```

Knowledge Graph

Usage (examples from the job search domain):

Response:


```
{ "data": [
  {
    "type": "jobtitle",
    "values": [
      {
        "id": "",
        "name": "Data Scientist",
        "relatedness": 0.989,
        "popularity": 86.0,
        "foreground_popularity": 86.0,
        "background_popularity": 142.0,
        "compare": [
          {
            "type": "skills.v3",
            "values": [
              {
                "id": "",
                "name": "Machine Learning",
                "relatedness": 0.97286,
                "popularity": 54.0,
                "foreground_popularity": 54.0,
                "background_popularity": 356.0
              },
              {
                "id": "",
                "name": "Predictive Modelling",
                "relatedness": 0.94565,
                "popularity": 27.0,
                "foreground_popularity": 27.0,
                "background_popularity": 384.0
              },
              {
                "id": "",
                "name": "Artificial Neural Networks",
                "relatedness": 0.94416,
                "popularity": 10.0,
                "foreground_popularity": 10.0,
                "background_popularity": 57.0
              },
              {
                "id": "",
                "name": "Apache Hadoop",
                "relatedness": 0.94274,
                "popularity": 50.0,
                "foreground_popularity": 50.0,
                "background_popularity": 1418.0
              },
              {
                "id": "",
                "name": "Java (Programming Language)",
                "relatedness": 0.76606,
                "popularity": 37.0,
                "foreground_popularity": 37.0,
                "background_popularity": 17442.0
              }
            ]
          }
        ]
      }
    ]
  }
]
}
```

Experiments

Data Cleansing

Knowledge Graph

Foreground Query: "Hadoop"

{ "type":"keywords", "values":[
{ "value":"hive", "relatedness": 0.9765, "popularity":369 },
{ "value":"spark", "relatedness": 0.9634, "popularity":15653 },
{ "value":".net", "relatedness": 0.5417, "popularity":17683 },
{ "value":"bogus_word", "relatedness": 0.0, "popularity":0 },
{ "value":"teaching", "relatedness": -0.1510, "popularity":9923 },
{ "value":"CPR", "relatedness": -0.4012, "popularity":27089 }] }

Experiment: Data analyst manually annotated 500 pairs of terms found together in real query logs as "relevant" or "not relevant"

Results: SKG removed 78% of the terms while maintaining a 95% accuracy at removing the correct noisy pairs from the input data.

TABLE III. SAMPLES FOR THE CO-TERMS CLEANED BY SKG

Term	Co-term	Blacklisted?
system support	it manager	Yes
senior buyer	customer service manager	Yes
leasing consultant	manufacturing manager	Yes
programmer	engineering manager	Yes
product requirement documents	sows	No
events	wedding coordinator	No
electrical engineering	cad designer	No

Predictive Analytics

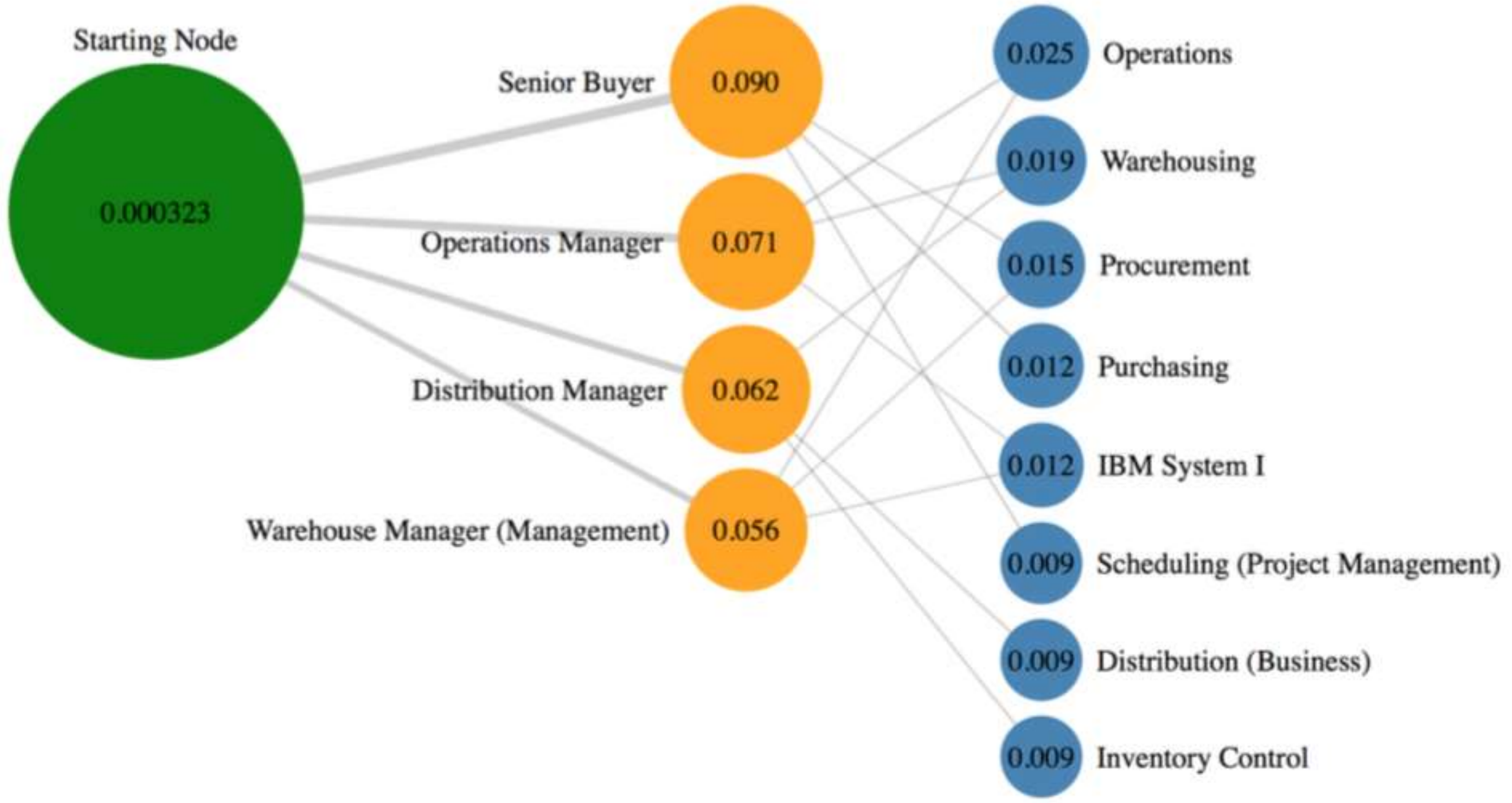


Fig. 4. **Predictive analytics (consequent scoring)**. Assume a jobseeker has a job title of *Logistics Manager*, the skill of *Distribution (Business)*, and additionally some experience with the keyword *purchasing*. The figure shows this starting materialized node with its support on the left. The figure highlights the results for the top five predicted job titles with the middle circles, with the highest confidence job title being *Senior Buyer* with a confidence of 0.09. The top skills are predicted jointly with the job title in the circles on the right, with *Operations* as the highest confidence skill, with confidence of 0.025.

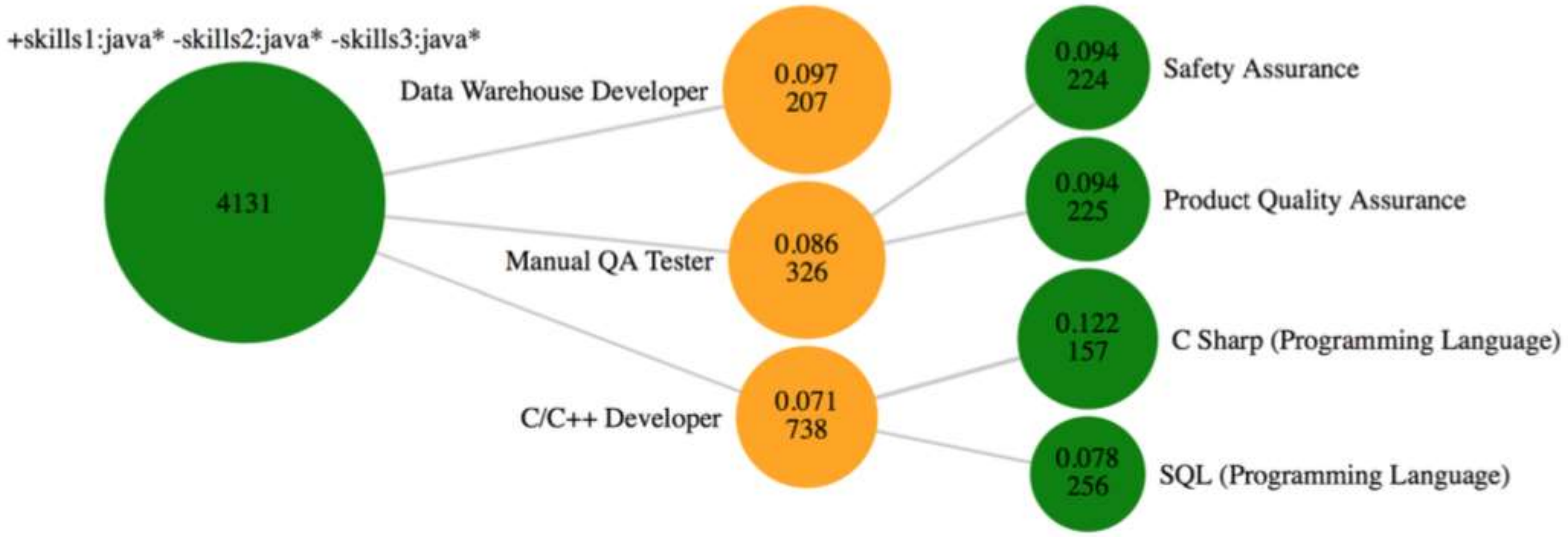


Fig. 5. **Predictive analytics (antecedent scoring)**. An example of query expansion for a q_0 of *skills:Java**. The nodes joined by edges on the middle and right form the combined antecedent, with the query result set forming the consequent. The top number on the rightmost nodes equals the confidence of the combined antecedent $\rightarrow$ starting node rule, while the top number for the middle column represents the confidence of the single-item antecedent $\rightarrow$ starting node rule. Correspondingly, the bottom number indicates the support (times one million) for each rule.

Search Expansion

Experiment: Take an initial query, and expand keyword phrases to include the most related entities to that query

Example:

Query:

hadoop

☑ **Dynamically identify unknown keywords**

OR

Document:

Job title here ...

add some text here to extract keywords ....

Version:

next

Submit

Raw Response | Table View

⊞ parsed_input
⊞ extracted_keywords
⊟ job_titles

name	id	weight
Software Engineer	15.0	1
Hadoop DevOps Engineer	15.192	0.91
Java Developer	15.2	0.45
ETL Developer	15.44	0.18
Data Consultant	15.218	0.17
Data Architect	15.34	0.12

⊞ occupations
⊟ related_keywords

name	weight
hadoop developer	1
map/reduce	0.9
hive	0.8
hbase	0.78
pig	0.75
big data	0.7
obiee	0.45

The Semantic Search Problem

User's Query:

machine learning research and development Portland, OR software engineer AND hadoop, java

---

Traditional Query Parsing:

(machine AND learning AND research AND development AND portland)
OR (software AND engineer AND hadoop AND java)

Semantic Query Parsing:

"machine learning" AND "research and development" AND "Portland, OR"
AND "software engineer" AND hadoop AND java

Semantically Expanded Query:

("machine learning"^10 OR "data scientist" OR "data mining" OR "artificial intelligence")
AND ("research and development"^10 OR "r&d") AND
AND ("Portland, OR"^10 OR "Portland, Oregon" OR {!geofilt pt=45.512,-122.676 d=50 sfield=geo})
AND ("software engineer"^10 OR "software developer")
AND (hadoop^10 OR "big data" OR hbase OR hive) AND (java^10 OR j2ee)

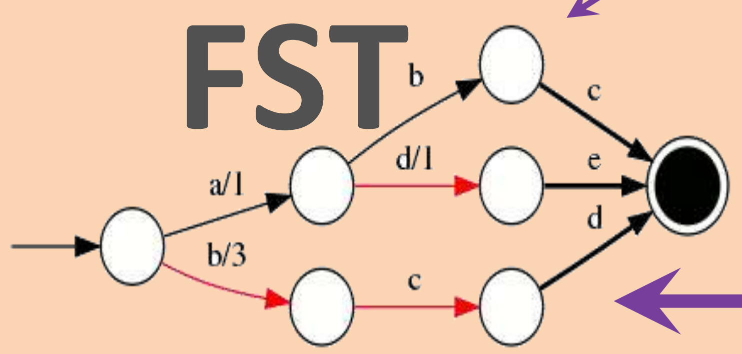
Query Expansion

Keywords:

machine learning

Semantic Interpretation

FST



Known keyword phrases

java developer
machine learning
registered nurse

Related Phrases

machine learning:
{ data mining .9,
matlab .8,
data scientist .75,
artificial intelligence .7,
neural networks .55 }

Related Occupations

machine learning:
{15-1031.00 .58
Computer Software Engineers, Applications
15-1011.00 .55
Computer and Information Scientists, Research
15-1032.00 .52
Computer Software Engineers, Systems Software }

Common Job Titles

machine learning:
{ software engineer .65,
data manager .3,
data scientist .25,
hadoop engineer .2, }

hadoop + Spark

Search Behavior, Application Behavior, etc.

Knowledge Graph in Solr

Job Title Classifier, Skills Extractor, Job Level Classifier, etc.

Modified Query:

keywords:((machine learning)^10 OR
{ AT_LEAST_2: ("data mining"^0.9, matlab^0.8,
"data scientist"^0.75, "artificial intelligence"^0.7,
"neural networks"^0.55)) }
{ BOOST_TO_TOP: (job_title:(
"software engineer" OR "data manager" OR
"data scientist" OR "hadoop engineer")) }

Solr


Find Candidates
Keywords
senior software engineer perl hadoop big data
senior software engineer +10 more
AND
perl +10 more
AND
hadoop +10 more
AND
big data +10 more
1
1 - 20 out of 34
Actions
View
related terms
map/reduce
hive
hadoop developer
hbase
pig
big data
obiee
lucene
solr
nutch
Demo Resume - Name Omitted
Hadoop Developer - 11.5 years of in-depth IT Experience
Experience: 11
Degree Level: Master's Degree
Recent Position: Hadoop Developer
US-GA
and Technology

Document Summarization

Knowledge Graph

Experiment: Pass in raw text (extracting phrases as needed), and rank their similarity to the documents using the SKG.

Additionally, can traverse the graph to “related” entities/keyword phrases NOT found in the original document

Applications: Content-based and multi-modal recommendations (no cold-start problem), data cleansing prior to clustering or other ML methods, semantic search / similarity scoring

Request	Response	
Job Title: Big Data Engineer REQUIREMENTS: Bachelor’s degree in Computer Science or related discipline... 2+ years of hands—on implementation experience(preferably lead engineer) working with a combination of the following technologies: Hadoop, Map Reduce, Pig, Hive, Impala, ... IDEAL ADDITIONAL EXPERIENCE: Strong knowledge of noSQL of at—least one noSQL database like HBase and Cassandra. 3+ years’ programming/scripting languages Java and Scala, python, R, Pig 2+ years’ experience with spring framework Experience in developing the full life—cycle of a Hadoop solution. This includes creating the requirements analysis, design of the technical architecture, design of the application design and development, testing, and deployment of the proposed solution... Understanding of Machine Learning skills (like Mahout) Experience with Visualization Tools such as Tableau ...	data engineer hive pig hadoop mapreduce nosql hbase impala python cassandra scala machine learning tableau mahout analytics java	0.96 0.82 0.82 0.8 0.79 0.71 0.66 0.6 0.56 0.56 0.56 0.49 0.39 0.37 0.36 0.36

TABLE IV. DOCUMENT SUMMARIZATION

Document Enrichment – Find / Score Relationships

Submit a query OR a document

Query:

Keywords here ...

☑ **Dynamically identify unknown keywords**

OR

Document:

Accounts Payable Clerk

Company in the far western suburbs is looking for a Accounts Payable Clerk. This Accounts Payable Clerk will be in charge of entering purchase orders accurately, 3-way match and resolve invoice discrepancies. This Accounts Payable Clerk must have strong written and verbal communication skills because there will be a lot of interaction with vendors. Company is currently going through a system conversion so experience with Epicor is a plus. Strong Excel is

Version:

next

Submit

Raw Response | Table View

⊞ related_keywords

⊟ job_titles

name	id	weight
Accounts Payable Clerk	43.9	0.5

⊟ occupations

name	id	weight
Bookkeeping, Accounting, and Auditing Clerks	43-3031.00	0.97
Billing, Cost, and Rate Clerks	43-3021.02	0.51

⊟ skills

name	weight
Bookkeeping	0.98
Accounts Payable	0.88
Finance	0.88
Accounting	0.88

⊞ versions

Document Summarization – Rank / Clean Keywords

Query:

Keywords here ...

☑ **Dynamically identify unknown keywords**

OR

Document:

Big Data Engineer

Required Skills:
Must have demonstrable, programming proficiency in one or more of the following: Java, C/C++, or Python.
Deep understanding of Map Reduce framework & Hadoop.
Fluent in Pig and/or Hive with experience in building UDFs, strong scripting ability.
Proven expertise and understanding of ETL techniques.
Knowledge of Azkaban, Oozie or Hamake for

Version:

next

Submit

Raw Response | Table View


```
{
  "extracted_keywords": [
    {
      "name": "data engineer",
      "weight": 0.95,
      "type": "job_title",
      "relationships": {}
    },
    {
      "name": "big data",
      "weight": 0.92,
      "type": "skill",
      "relationships": {}
    },
    {
      "name": "hadoop",
      "weight": 0.92,
      "type": "skill",
      "relationships": {}
    },
    {
      "name": "hive",
      "weight": 0.92,
      "type": "skill",
      "relationships": {}
    },
    {
      "name": "mapreduce",
      "weight": 0.91,
      "type": "skill",
      "relationships": {}
    },
    {
      "name": "pig",
```

Future Work

- Semantic Search (more experiments)
- Search Engine Relevancy Algorithms
- Trending Topics
- Recommendation Systems
- Root Cause Analysis
- Abuse Detection

Conclusion

Applications:

The Semantic Knowledge Graph has numerous applications, including automatically building ontologies, identification of trending topics over time, predictive analytics on timeseries data, root-cause analysis surfacing concepts related to failure scenarios from free text, data cleansing, document summarization, semantic search interpretation and expansion of queries, recommendation systems, and numerous other forms of anomaly detection.

Main contribution of this paper:

The introduction (and open sourcing) of the the Semantic Knowledge Graph, a novel and compact new graph model
that can dynamically materialize and score the relationships between any arbitrary combination of entities represented within a corpus of documents.

References

[1] Tom Gruber. Ontology. In Ling Liu and M. Tamer Özsu, editors, *Encyclopedia of Database Systems*. 2009.

[2] Chris Biemann. Ontology learning from text: A survey of methods. In *LDV forum*, volume 20.

[3] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11).

[4] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM GMOD/PODS*.

[5] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, et al. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2).

[6] Johannes Hoffart, Fabian M Suchanek, Klaus Berberich, and Gerhard Weikum. Yago2: A spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence*, 194.

[7] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of WWW*.

[8] Feng Niu, Ce Zhang, Christopher Ré, and Jude W Shavlik. Deepdive: Web-scale knowledge-base construction using statistical learning and inference. *VLDS*, 12.

[9] Antoine Zimmermann, Christophe Gravier, Julien Subercaze, and Quentin Cruzille. Nell2rdf: Read the web, and turn it into rdf. In *KNOW@ LOD*.

[10] Ndapandula Nakashole, Martin Theobald, and Gerhard Weikum. Scalable knowledge harvesting with high precision and high recall. In *Proceedings of WSDM 2011*.

[11] Natalya Fridman Noy, Mark A Musen, et al. Algorithm and tool for automated ontology merging and alignment. In *Proceedings of the AAAI 2000.*, 2000.

[11] Natalya Fridman Noy, Mark A Musen, et al. Algorithm and tool for automated ontology merging and alignment. In *Proceedings of the AAAI 2000.*, 2000.

[12] Meghan Daly, Fletcher Grow, Mackenzie Peterson, Jeremy Rhodes, and Robert L Nagel. Development of an automated ontology generator for analyzing customer concerns. In *Systems and Information Engineering Design Symposium (SIEDS), 2015*.

[13] AA Krizhanovsky and AV Smirnov. An approach to automated construction of a general-purpose lexical ontology based on wiktionary. *Journal of Computer and Systems Sciences International*, 52(2).

[14] Chang-Shing Lee, Yuan-Fang Kao, Yau-Hwang Kuo, and Mei-Hui Wang. Automated ontology construction for unstructured text documents. *Data & Knowledge Engineering*, 60(3).

[15] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of KDD 2014*.

[16] Roberto Navigli and Paola Velardi. Learning domain ontologies from document warehouses and dedicated web sites. *Computational Linguistics*, 30(2).

[17] Jan Daciuk and Dawid Weiss. Smaller representation of finite state automata. In *Implementation and Application of Automata*.

[18] Crowdflower data science report 2016. http://visit.crowdflower.com/rs/416-ZBE-142/images/CrowdFlower_DataScienceReport_2016.pdf, 2016.

[19] Khalifeh AlJadda, Mohammed Korayem, Trey Grainger, and Chris Russell. Crowdsourced query augmentation through semantic discovery of domain-specific jargon. In *IEEE Big Data 2014*.

[20] K. AlJadda, M. Korayem, C. Ortiz, T. Grainger, J. A. Miller, and W. S. York. Pgmhd: A scalable probabilistic graphical model for massive hierarchical data problems. In *Big Data (Big Data), 2014 IEEE International Conference on*, pages 55–60, Oct 2014.

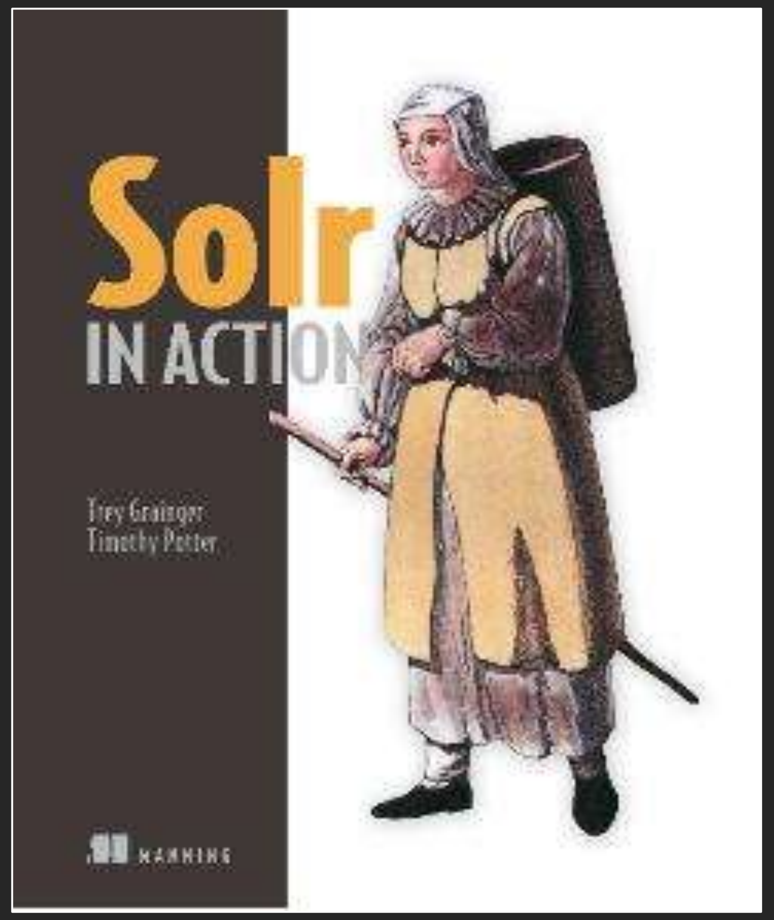
[21] Jochen Hipp, Ulrich Güntzer, and Gholamreza Nakhaeizadeh. Algorithms for association rule mining-a general survey and comparison. *ACM sigkdd explorations newsletter*, 2(1).

[22] Juan Ramos. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, 2003.

Contact Info

Trey Grainger

trey.grainger@lucidworks.com
@treygrainger



Other presentations:

http://www.treygrainger.com

http://solrinaction.com