Knowledge Graphs and Natural Language Processing

Lecture at ESSLLI 2022

DOI: 10.5281/zenodo.6667766

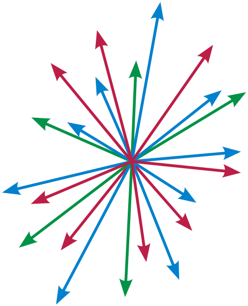
Outline

Section 1

Introduction

Introduction

- So far we have mostly considered *undirected* graphs, but many important linguistic relations are *directed*
- How can we extract, reason, enrich, and express the knowledge about the real world and use it in Natural Language Processing (NLP) applications?
- We will discuss **Knowledge Graphs** *aka* **Semantic Networks** in various forms and applications



Source: Alexandrov (2007)

Core Idea: Feel the Direction

The world is complex and we have to select the subset of this complexity in our applications.

Relations

Previously, we worked mainly with some of the *symmetric* relations:

- **synonymy**: big $=$ large
- **relatedness**: water $\approx$ liquid
- **antonymy:** big $\leftrightarrow$ small

In the real world, there are many *asymmetric* relations:

- **hypernym–hyponym** *aka* `is-a`: cat $\prec$ mammal
- **holonym–meronym** *aka* `has-a`: cat $\ni$ tail
- **class–instance** *aka* `instance-of`: ESSLLI 2022 $\prec$ summer school

The list is non-exhaustive.

Ontologies

Lassila et al. (2001) showed an *ontology spectrum*:

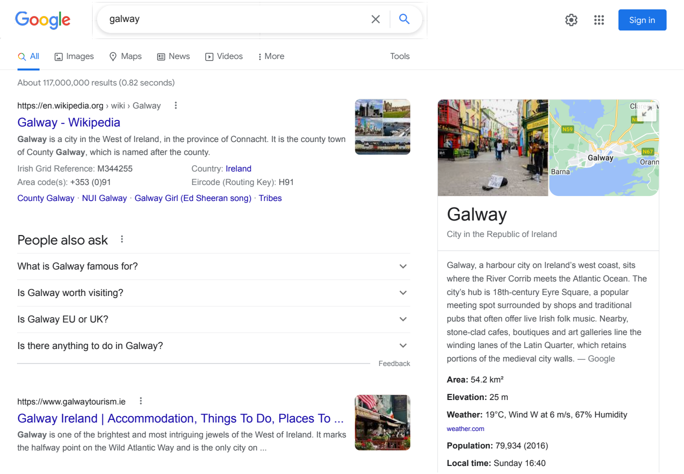
- **vocabulary**, a finite list of terms
- **glossary**, a list of terms and their textual meanings
- **thesaurus**, a glossary with informal semantic relations
- **taxonomy**, a thesaurus with formal relations and `is-a` transitivity
- ...
- **formal ontology** with finite controlled vocabulary, unambiguous interpretation of classes and term relationships, and strict hierarchical subclass relationships (and more)

Definition by Hogan et al. (2021)

Knowledge Graph is a graph of data intended to accumulate and convey knowledge of the real world, whose nodes represent entities of interest and whose edges represent potentially different relations between these entities.

Motivation I

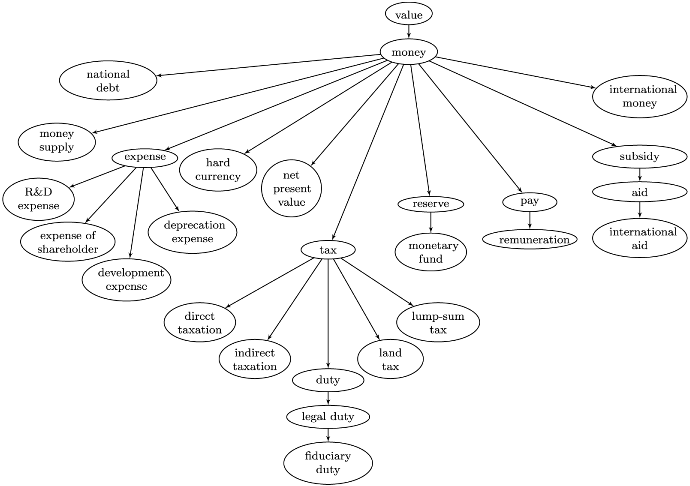
Knowledge Graphs power some of our everyday digital tools.



Source: https://www.google.com/search?q=galway

Motivation II

We can teach machines to accumulate and employ knowledge.



Source: Velardi et al. (2013)

Section 2

Taxonomies

Taxonomies

- Such taxonomies and thesauri as WordNet (Fellbaum, 1998), BabelNet (Navigli et al., 2012), and ConceptNet (Speer et al., 2017) are popular kinds of NLP resources
- These resources reflect real-world *things* and relations between them and are useful both in benchmarks and downstream applications
- We will see a case of a classical taxonomy induction method and then a few modern machine learning approaches useful for hypernym discovery

Source: Buissinne (2016)

WordNet: Example

Source: http://wordnetweb.princeton.edu/perl/webwn

How Taxonomies Are Built

- Most popular taxonomies are **manually built by groups of experts**, e.g., WordNet (Fellbaum, 1998), Cyc (Lenat, 1995)
- This laborious process can be automated by **building a taxonomy from scratch** using raw `is-a` pairs, e.g., Knowledge Harvesting (Kozareva et al., 2010) and OntoLearn (Velardi et al., 2013)
- Possible alternatives include **matching and translating** the available datasets (Navigli et al., 2012), designing a **crowdsourcing** pipeline (Biemann, 2013), etc.

Source: Free-Photos (2016)

Hearst Patterns

Hearst (1992) proposed extracting hypernyms from large text corpora with *lexico-syntactic patterns*:

- `NP`$_0$ `such as {NP`$_1$`, NP`$_2$`, ..., (and|or)} NP`$_n$
- `such NP as {NP ,}* {(or|and)} NP`
- `NP {, NP}*{,} or other NP`

Alternatives: logistic regression on dependency trees (Snow et al., 2004), word-concept lattices (Velardi et al., 2013, Section 3.2), etc.

- Despite the simplicity, it shows good results in practice after filtering
- The main challenge is in achieving *recall*
- The extracted pairs do not necessarily form a taxonomy!

Hearst Patterns: Revisited

Roller et al. (2018) used 4.5M `is-a` pairs and found that the truncated SVD (Hansen, 1987) embeddings outperform all other methods in three tasks:

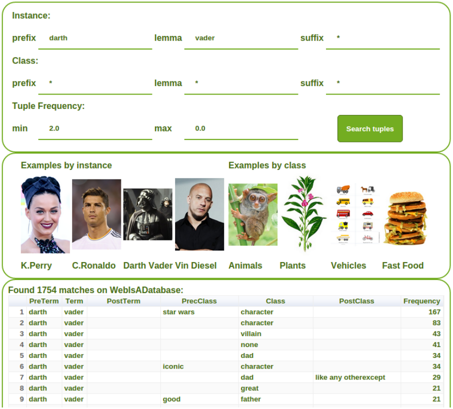
- **Detection:** whether the words are in an `is-a` relation?
- **Direction:** which term is broader in a given pair of words?
- **Graded Entailment:** to what degree the relation holds?

	Detection (AP)					**Direction** (Ac)			**Graded** (ρ_s)
	BLESS	EVAL	LEDS	Shwartz	WBLESS	BLESS	WBLESS	BiBless	Hyperlex
Cosine	.12	.29	.71	.31	.53	.00	.54	.52	.14
WeedsPrec	.19	.39	.87	.43	.68	.63	.59	.45	.43
invCL	.18	.37	**.89**	.38	.66	.64	.60	.47	.43
SLQS	.15	.35	.60	.38	.69	.75	.67	.51	.16
$\mathrm{p}(x, y)$	.49	.38	.71	.29	.74	.46	.69	.62	**.62**
$\mathrm{ppmi}(x, y)$	.45	.36	.70	.28	.72	.46	.68	.61	.60
$\mathrm{sp}(x, y)$	.66	.45	.81	.41	.91	**.96**	.84	.80	.51
$\mathrm{spmi}(x, y)$	**.76**	**.48**	.84	**.44**	**.96**	**.96**	**.87**	**.85**	.53

Source: Roller et al. (2018)

WebIsA Database

- Seitner et al. (2016) created a database of 400M hypernymy pairs called **WebIsADb**
- The input Common Crawl corpus contained 2.1B Web pages, totaling in 168TB of compressed data; https://commoncrawl.org/
- 58 manually-defined patterns and pre-processing: part-of-speech tagging, duplicate removal, text normalization



	PreTerm	Term	PostTerm	PrecClass	Class	PostClass	Frequency
1	darth	vader		star wars	character		167
2	darth	vader			character		83
3	darth	vader			villain		43
4	darth	vader			none		41
5	darth	vader			dad		34
6	darth	vader		iconic	character		34
7	darth	vader			dad	like any otherexcept	29
8	darth	vader			great		21
9	darth	vader		good	father		21

Source: Seitner et al. (2016) & http://webdatacommons.org/isadb/

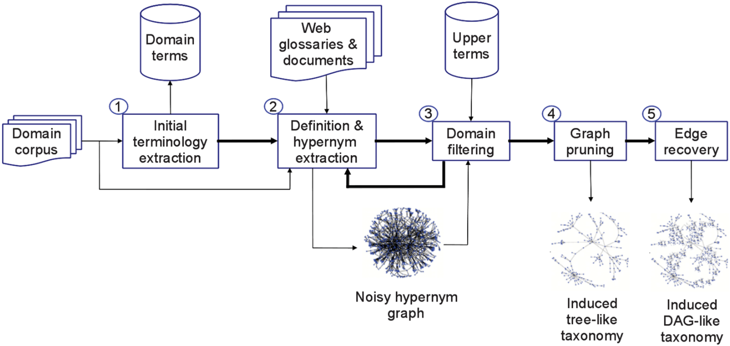
Taxonomy Induction

- Popular ontologies and taxonomies, such as Cyc (Lenat, 1995) and WordNet (Fellbaum, 1998), are built by teams of experts
- It is difficult to keep them up-to-date and to expand them to the new domains and languages
- This urges the task of taxonomy induction, in which given the set of `is-a` pairs, organize the terms in a directed acyclic graph (Bordea et al., 2016)

Source: Kittner (2015)

OntoLearn

OntoLearn is a graph-based method for building taxonomies from scratch (Velardi et al., 2013), comprised of multiple processing steps.



Source: Velardi et al. (2013)

OntoLearn is a good example of a sophisticated workflow for a specific NLP task.

OntoLearn: Extraction and Filtering

Terminology Extraction.

- Harvest single- and multi-word terms for the domain from the corpus

Hypernym Extraction.

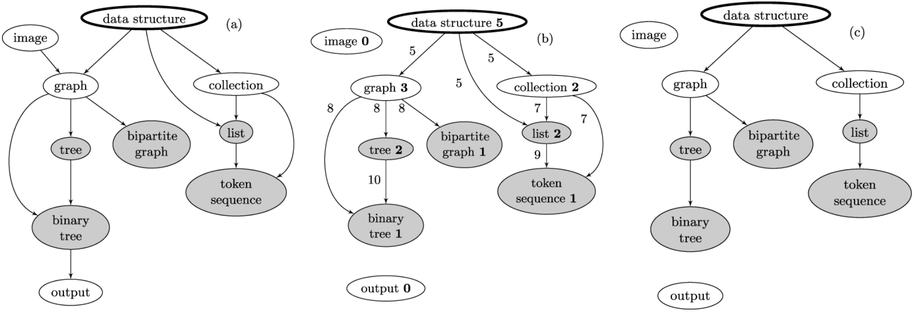
- Manually choose a set of *upper terms* from, e.g., WordNet
- Extract definition candidates from the domain corpus and the Web, perform part-of-speech tagging
- Apply word-concept lattices to extract `is-a` pairs

Domain Filtering.

- Exclude from the retrieved sentences non-domain definitions in multiple iterations
- Populate the noisy hypernym graph with the kept pairs
- Connect all the upper terms to a virtual top node to obtain the *backbone component*

OntoLearn: Graph Pruning and Edge Recovery

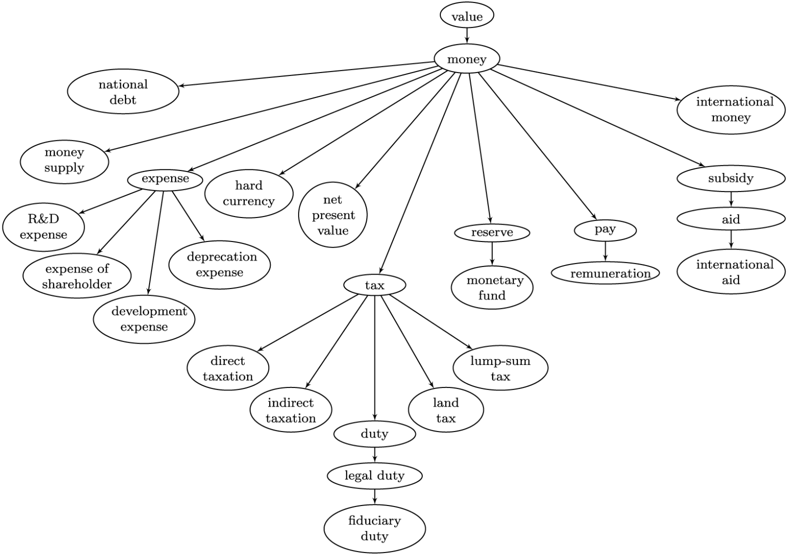
Graph Pruning. Produce a complete taxonomy by graph trimming, edge weighting, optimal branching, and pruning recovery.



Source: Velardi et al. (2013)

Edge Recovery. Transform the built tree into a directed acyclic graph by evaluating the lengths of shortest paths from the nodes in removed paths to the taxonomy root.

OntoLearn: Example



Source: Velardi et al. (2013)

OntoLearn: Discussion

Pros:

- Can be applied to building domain taxonomies from scratch
- The approach is language-independent
  (as soon as all the steps can be executed)

Cons:

- Assumes that terms and hypernyms co-occur in the same sentence
- Does not discriminate in-domain ambiguity and systematic polysemy
- Taxonomies are difficult to evaluate

Resources:

- http://lcl.uniroma1.it/ontolearn_reloaded/
- http://lcl.uniroma1.it/wcl/

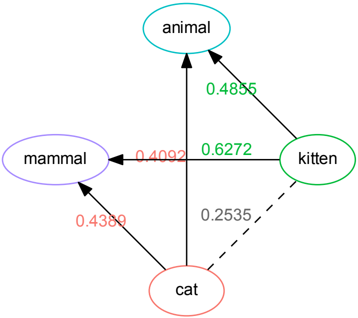
Learning Hypernym Representations

An earlier idea was to split the word embedding space into $K \in \mathbb{N}$ clusters and then approximate a matrix Φ_k that transforms the hyponym vector $\vec{x}$ into the hypernym vector $\vec{y}$ (Fu et al., 2014):

$$\Phi_k^* = \arg\min_{\Phi_k} \frac{1}{N_k} \sum_{(x,y)\in C_k} \|\Phi_k \vec{x} - \vec{y}\|^2 + \lambda R$$

Quality can be increased with regularizations (Ustalov et al., 2017), $\lambda \in \mathbb{R}^+$:

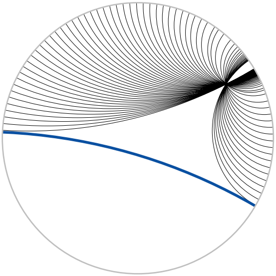
- $R_{\text{asym}} = \frac{1}{N_k} \sum_{(x,\_)\in C_k} (\Phi_k \Phi_k \vec{x} \cdot \vec{x})^2$
- $R_{\text{neg}} = \frac{1}{N_k} \sum_{(x,\_)\in C_k, z\in N(x)} (\Phi_k \Phi_k \vec{x} \cdot \vec{z})^2$



Source: Ustalov et al. (2017)

Poincaré Embeddings

- Asymmetric relations form a hierarchy, but linear embeddings of graphs require too many dimensions to model them properly
- We can preserve the relationship by embedding taxonomies and other trees in a *hyperbolic space* instead of the Euclidean space
- Nickel et al. (2017) proposed embedding hierarchies in a Poincaré ball, measuring the distance between two points as



Source: Trevorgoodchild (2008)

$$d(\vec{u}, \vec{v}) = \operatorname{arcosh}\left(1 + 2\frac{\|\vec{u} - \vec{v}\|}{(1 - \|\vec{u}\|^2)(1 - \|\vec{v}\|^2)}\right)$$

This allows using gradient-based optimization methods on Riemannian manifolds, such as Riemannian stochastic gradient descent *aka* RSGD (Bonnabel, 2013).

Poincaré Embeddings: Estimation

Nickel et al. (2017) minimized the cross-entropy loss function using the RSGD to estimate the point coordinates, $\Theta \in \mathbb{R}^{|V| \times d}$:

$$\sum_{(u,v) \in E} \log \frac{\exp\left(-d(\vec{u}, \vec{v})\right)}{\sum_{v' \in \mathcal{N}(u)} \exp\left(-d(\vec{u}, \vec{v}')\right)}$$

We randomly sample for $u \in V$ a set of negative examples:

$$\mathcal{N}(u) = \{v' \in V : (u, v') \notin E\} \cup \{v\}$$

- During the *burn-in* period in the first $e \in \mathbb{N}$ epochs, we use a lower learning rate $\eta > 0$ and sample the more popular objects $u \in V$ more often using the alias method (Walker, 1977)
- For the rest of the training, we sample uniformly and use a higher learning rate

Poincaré Embeddings: Algorithm

Input: graph $G = (V, E)$, dimensions $d \ll |V|$, burn-in epochs $e \in \mathbb{N}$, learning rate $\eta > 0$, burn-in rate coefficient $c > 0$, batch size $b \in \mathbb{N}$

Output: embedding $\vec{u} \in \mathbb{R}^d, \forall u \in V$

$\Theta \leftarrow \texttt{random}([-0.001, 0.001]^{|V| \times d})$

while not converged **do**

  if epoch $\leq e$ **then** $\triangleright$ Burn-in; use alias sampling for $u \in V$

    $T_{\text{batch}} \leftarrow \{(u, v, \mathcal{N}(u)) : (u, v) \in E\}$ s.t. $|T_{\text{batch}}| = b$

    $\Theta \leftarrow \text{RSGD}(\Theta, \frac{\eta}{c}, T_{\text{batch}})$

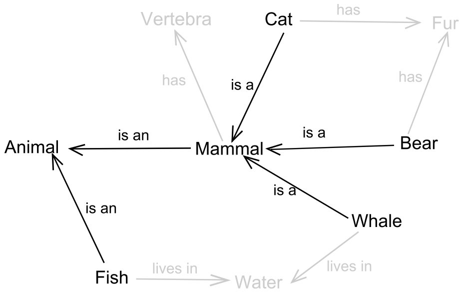
  else $\triangleright$ Use uniform sampling for $u \in V$

    $T_{\text{batch}} \leftarrow \{(u, v, \mathcal{N}(u)) : (u, v) \in E\}$ s.t. $|T_{\text{batch}}| = b$

    $\Theta \leftarrow \text{RSGD}(\Theta, \eta, T_{\text{batch}})$

return $\vec{u}_i \rightarrow \Theta_i$ **for all** $1 \leq i \leq |V|$

Poincaré Embeddings: Example



Source: Konstable (2006)

$$\Theta = \begin{pmatrix} -0.06 & -0.12 \\ 0.17 & 0.33 \\ -0.36 & -0.15 \\ -0.06 & -0.10 \\ 0.08 & -0.39 \\ 0.21 & 0.41 \end{pmatrix}$$

- $d($Cat, Bear$) = 0.603$
- $d($Cat, Fish$) = 0.367$
- $d($Cat, Animal$) = 0.398$
- $d($Cat, Mammal$) = 0.607$
- $d($Fish, Mammal$) = 0.443$
- $d($Animal, Mammal$) = 0.490$

Poincaré Embeddings: Discussion

Pros:

+ Handles hierarchical datasets natively
+ Outperforms Hearst patterns in the setup of Roller et al. (2018)

Cons:

- Polysemy, cycles, and noisy hypernyms affect the result
- Can be stabilized with the Lorentz model (Le et al., 2019)
- If the data are not a tree, just use Word2Vec

Implementation:

- `https://github.com/facebookresearch/poincare-embeddings` (Nickel et al., 2017)
- `https://radimrehurek.com/gensim/` (Řehůřek et al., 2010)

Taxonomies: Wrap-Up

- Classical methods rely on graph pruning (Velardi et al., 2013), while the modern methods learn hierarchical representations (Nickel et al., 2017; Le et al., 2019)
- Polysemeous words are an obstacle; they can be handled with crowdsourcing (Biemann, 2013), clustering (Ustalov et al., 2019), embeddings (Bartunov et al., 2016)
- Building a full-fledged taxonomy is a good benchmark, but the research community now focusing on a simpler task of *hypernymy discovery* (Camacho-Collados et al., 2018)

Source: rawpixel (2017)

Section 3

Knowledge Representation

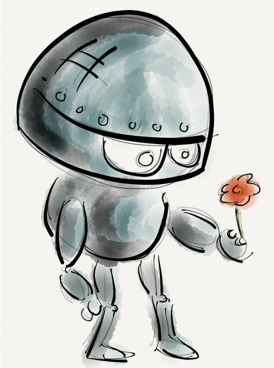
Knowledge Representation

We often need to express *metadata* on the Internet to obtain better ranking and representation in search engines and social media.

- Open Graph Protocol, `https://ogp.me/`
- Schema.org, `https://schema.org/`
- Dublin Core, `https://www.dublincore.org/`

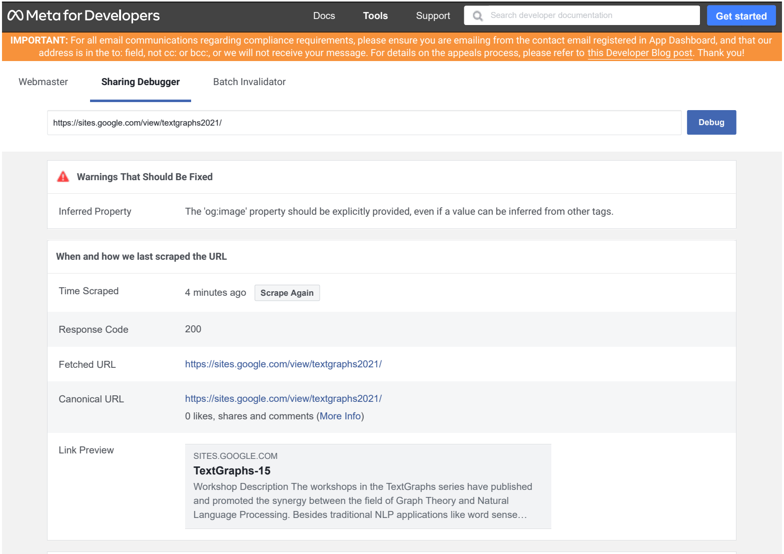
These standards allow the representation of machine-readable information about Web documents with clearly-defined types and meanings.

But we can do it for any identifiable object.



Source: bamenny (2016)

Open: Example Graph Protocol



Source: `https://developers.facebook.com/tools/debug/`

Resource Description Framework

Resource Description Framework (RDF) is a framework for representing information on the Web.

- Everything is represented as triples of **subject**, **predicate**, and **object**; e.g., (expert, study, problem)
- Each part of the triple is identified via a uniform resource identifier (URI), e.g., https://2022.esslli.eu/
- A set of triples forms a labeled, directed multigraph, enabling access to machine-readable information through the Web (*aka* **the Semantic Web**)
- RDF is just a data model; we need the serialization syntax and a language to express the knowledge

https://www.w3.org/TR/rdf11-concepts/

RDF Serialization Syntaxes

- **RDF in Attributes (RDFa):** https://www.w3.org/TR/rdfa-primer/
- **Terse RDF Triple Language (Turtle):** https://www.w3.org/TR/turtle/
- **JSON for Linked Data (JSON-LD):** https://www.w3.org/TR/json-ld/
- **RDF/XML:** https://www.w3.org/TR/rdf-syntax-grammar/

Turtle: Example

```
@prefix cc: <https://creativecommons.org/ns#> .
@prefix dc: <https://purl.org/dc/elements/1.1/> .

<https://zenodo.org/record/6667766>
    dc:title "Graphs, Computation, and Language" ;
    dc:creator "Dmitry Ustalov" ;
    cc:license <http://creativecommons.org/licenses/
                  by-nc-sa/4.0/> .
```

RDF Models and Languages

- **RDF Schema** (RDFS) is an RDF vocabulary allowing modeling data; https://www.w3.org/TR/rdf-schema/
- **Web Ontology Language** (OWL) is a formal way to express things and relations between them; https://www.w3.org/TR/owl2-overview/
- **Simple Knowledge Organization System** (SKOS) is a model for thesauri, taxonomies and similar semantic resources; https://www.w3.org/TR/skos-reference/
- **SPARQL Protocol and RDF Query Language** is a query language for RDF, similar to SQL for relational data; https://www.w3.org/TR/sparql11-query/
- **Shapes Constraint Language** is a graph validation language; https://www.w3.org/TR/shacl/
- ... and much, much more!

RDF: Not a Silver Bullet

- n-ary relations like *temporarily true facts* require workarounds, such as compound value types (CVTs) or similar techniques (Pellissier Tanon et al., 2016)
- RDF works well with factual assertions, but it is less appropriate for other kinds of knowledge, e.g., representing the difference between jazz music and blues (Dong et al., 2014)

Source: Rahman Rony (2016)

Software

The most popular open-source software is listed below.

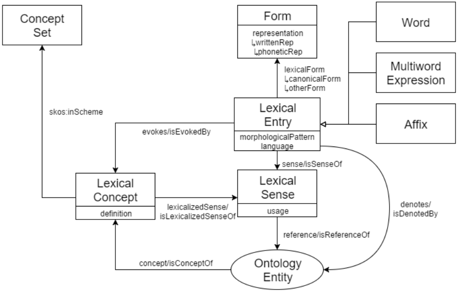
- **Protégé**, https://protege.stanford.edu/
- **Apache Jena**, https://jena.apache.org/
- **RDFLib** for Python, https://github.com/RDFLib
- **Eclipse RDF4J**, https://rdf4j.org/
- **OpenLink Virtuoso**, https://github.com/openlink/virtuoso-opensource

Many commercial vendors offer implementations of these standards.

Lemon

Lemon is a model for machine-readable dictionaries; http://lemon-model.net/.

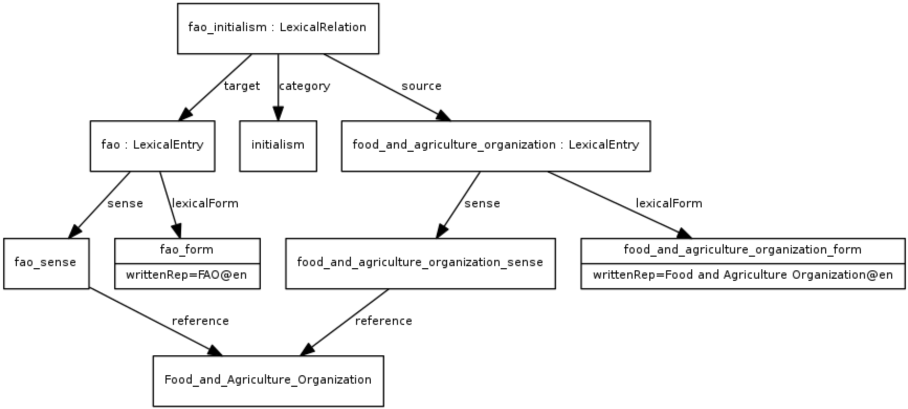
- Leverages the existing Semantic Web technologies
- Enables expressing WordNet-like databases as Linked Data
- Uses the LexInfo model to describe properties of linguistic objects; http://lexinfo.net/



Source: Cimiano et al. (2016)

https://www.w3.org/2016/05/ontolex/

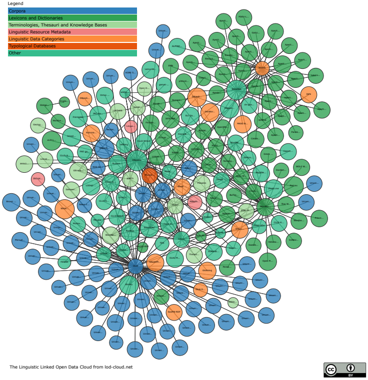
Lemon: Example



Source: Cimiano et al. (2016)

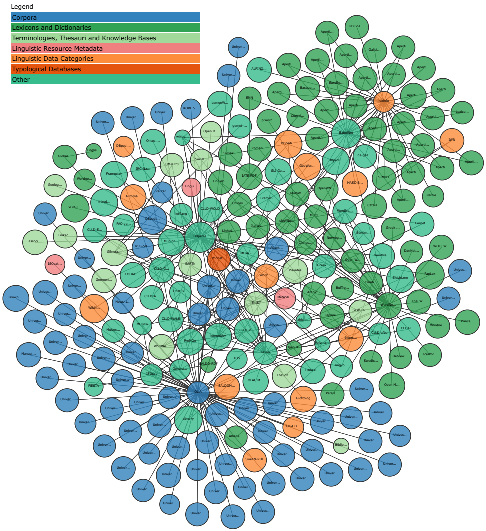
Linguistic Linked Open Data Cloud

- Thanks to the defined schemata and semantics, we can link different linguistic databases to each other using OWL, Lemon, and other vocabularies
- Example of such an initiative is **Linguistic Linked Open Data Cloud** *aka* LLOD (Cimiano et al., 2020)
- Cross-reference is possible, e.g., BabelNet can link to DBpedia, Wiktionary, WordNet, etc.



Source: `https://linguistic-lod.org/llod-cloud`

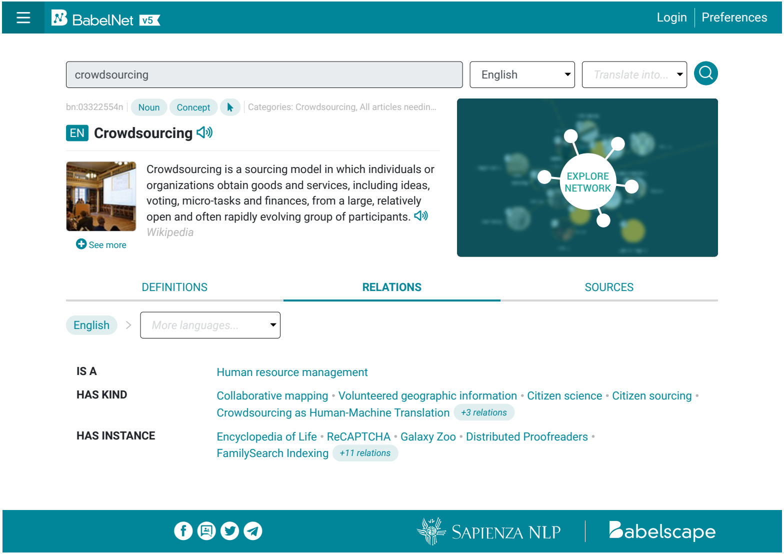
Linguistic Linked Open Data Cloud: Example



The Linguistic Linked Open Data Cloud from lod-cloud.net

Source: `https://linguistic-lod.org/llod-cloud`

BabelNet: Example



Source: https://babelnet.org/synset?id=bn:03322554n&lang=EN

DBpedia: Example

DBpedia | Browse using | Formats | Faceted Browser | Sparql Endpoint

About: Saint Petersburg

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

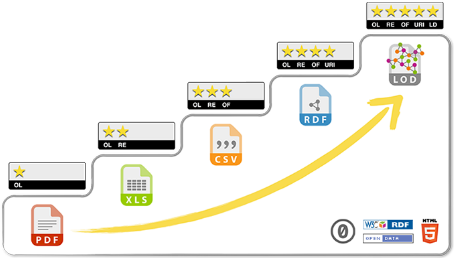
Saint Petersburg (Russian: Санкт-Петербург, tr. Sankt-Peterburg, IPA: [ˈsankt pʲɪtʲɪrˈburk] ()), formerly known as Petrograd (Петроград) (1914–1924), then Leningrad (Ленинград) (1924–1991), is a city situated on the Neva River, at the head of the Gulf of Finland on the Baltic Sea. It is Russia's second-largest city after Moscow. With over 5.3 million inhabitants as of 2018, it is the fourth-most populous city in Europe, as well as being the northernmost megalopolis. As an important Russian port on the Baltic Sea, it is governed as a federal city.

Property	Value
dbo:PopulatedPlace/areaTotal	▪ 1439.0
dbo:PopulatedPlace/populationDensity	▪ 3699.31
dbo:abstract	▪ Saint Petersburg (Russian: Санкт-Петербург, tr. Sankt-Peterburg, IPA: [ˈsankt pʲɪtʲɪrˈburk] ()), formerly known as Petrograd (Петроград) (1914–1924), then Leningrad (Ленинград) (1924–1991), is a city situated on the Neva River, at the head of the Gulf of Finland on the Baltic Sea. It is Russia's second-largest city after Moscow. With over 5.3 million inhabitants as of 2018, it is the fourth-most populous city in Europe, as well as being the northernmost megalopolis. As an important Russian port on the Baltic Sea, it is governed as a federal city. The city was founded by Tsar Peter the Great on 27 May [O.S. 16 May] 1703, on the site of a captured Swedish fortress. It served as a capital of the Russian Tsardom and the subsequent Russian Empire from 1713 to 1918 (being replaced by Moscow for a short period of time between 1728 and 1730). After the October Revolution, the Bolsheviks moved their government to Moscow. In modern times, Saint Petersburg is considered the Northern Capital and serves as a home to some federal government bodies such as the Constitutional Court of Russia and the Heraldic Council of the President of the Russian Federation. It is also a seat for the National Library of Russia and a planned location for the Supreme Court of the Russian Federation. The Historic Centre of Saint Petersburg and Related Groups of Monuments constitute a UNESCO World Heritage Site, so it's also referred to as Russia's cultural capital. Saint Petersburg is home to the Hermitage, one of the largest art museums in the world, and the Lakhta Center, the tallest skyscraper in Europe. Many foreign consulates, international corporations, banks and businesses have offices in Saint Petersburg. (en)
dbo:areaTotal	▪ 1439000000.000000 (xsd:double)
dbo:country	▪ dbr:Russia

Source: https://dbpedia.org/page/Saint_Petersburg

Five-Star Open Data

- When publishing data on the Internet, think of the usability and accessibility of the information
- It is useful in the practice to consult the available RDF vocabularies to represent the data in a well-thought way
- Most of these standards have been developing for years and contain non-trivial intricacies of the subject domains



Source: https://5stardata.info/en/

https://www.w3.org/DesignIssues/LinkedData.html

Knowledge Representation: Wrap-Up

- Knowledge Engineering is tangentially related to our course, but *people invested a lot in these standards and resources*, so it is wise to re-use them
- These standards enable interoperability between different applications and domains, but the adoption is not clearly incentivized
- Linked Data as a means for explainable artificial intelligence? (Hitzler, 2021)

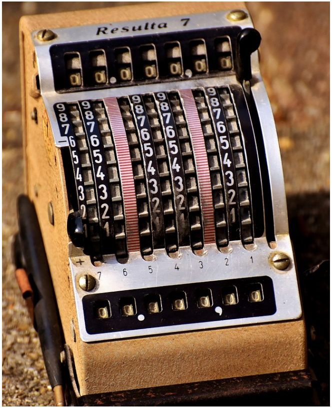
Source: Finnsson (2017)

Section 4

Knowledge Graph Embeddings

Knowledge Graph Embeddings

- Can we embed a knowledge graph using the methods we already know?
- Sure, StarSpace can learn a mapping between entities and relations (Wu et al., 2018) and DeepWalk can be extended for multi-relational data (Ristoski et al., 2018)
- However, many other methods take into account these graphs natively



Source: Alexas_Fotos (2017)

Multi-Relational Data and Graphs

Since we have multiple types of relations, we need to adjust our graph definitions accordingly.

Definition

For multi-relational data, we will consider as a multigraph a triple $G = (V, E, R)$, where V is a set of nodes, $E \subseteq V \times R \times V$ are labeled edges, and R is a set of relations.

- Following the RDF terminology, subjects and objects are nodes, predicates are relations, and triples are the directed labels
- The nodes are often called *entities* and the relations are called *links*
- Since the links are directed, the subject is referred to as the *head* and the object is related to as the *tail* of the relation

TransE

Translating Embeddings (TransE *aka* $\mathrm{DistAdd}$) is a method that models relationships by interpreting them as translations operating on the low-dimensional embeddings of the entities (Bordes et al., 2013).

- **Assumption:** in a triplet $(h, l, t) \in E, l \in R$ translates embeddings of $h \in V$ to $t \in V$, i.e., $\vec{h} + \vec{l} \approx \vec{t}$
- **Dissimilarity** $d : \mathbb{R}^k \times \mathbb{R}^k \to \mathbb{R}$ can be Manhattan (L_1), Euclidean (L_2) or any other distance in a k-dimensional space
- Samples **corrupted triplets** from the training set $S \subseteq E$ by replacing either head or tail by a random entity: $S'_{(h,l,t)} = \{(h', l, t | h' \in V)\} \cup \{(h, l, t' | t' \in V)\}$

TransE uses a *margin ranking loss*, where $[\dots]_+$ denotes the ReLU function:

$$\sum_{(h,l,t) \in S} \sum_{((h,l,t),(h',l,t')) \in S'_{(h,l,t)}} [\gamma + \underbrace{d(\vec{h} + \vec{l}, \vec{t})}_{\substack{\text{distance}\\\text{to correct}}} - \underbrace{d(\vec{h'} + \vec{l}, \vec{t'})}_{\substack{\text{distance}\\\text{to corrupted}}}]_+$$

TransE: Algorithm

Input: multigraph $G = (V, E, R)$, dissimilarity d, margin $\gamma > 0$,
dimensions $k \in \mathbb{N}$, batch size $b \in \mathbb{N}$

Output: embeddings $\vec{u} \in \mathbb{R}^k, \forall u \in V \cup R$

$\vec{l} \leftarrow \texttt{random}([-\frac{6}{\sqrt{k}}, \frac{6}{\sqrt{k}}]^k)$ **for all** $l \in R$

$\vec{l} \leftarrow \frac{\vec{l}}{\|\vec{l}\|}$ **for all** $l \in R$

$\vec{u} \leftarrow \texttt{random}([-\frac{6}{\sqrt{k}}, \frac{6}{\sqrt{k}}]^k)$ **for all** $u \in V$

while not converged **do**

  $\vec{u} \leftarrow \frac{\vec{u}}{\|\vec{u}\|}$ **for all** $u \in V$

  $T_{\text{batch}} \leftarrow \emptyset$

  for all $(h, l, t) \in \texttt{sample}(S, b)$ **do** ▷ Sample a minibatch

    $(h', l, t') \leftarrow \texttt{random}(S'_{(h,l,t)})$ ▷ Sample a corrupted triplet

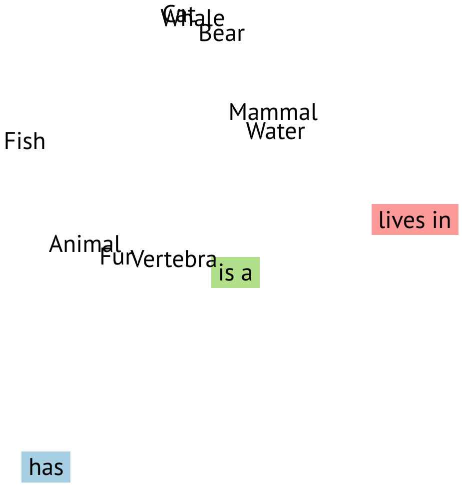
    $T_{\text{batch}} \leftarrow T_{\text{batch}} \cup \{((h, l, t), (h', l, t'))\}$

  Update w.r.t. $\sum_{((h,l,t),(h',l,t')) \in T_{\text{batch}}} \nabla[\gamma + d(\vec{h} + \vec{l}, \vec{t}) - d(\vec{h'} + \vec{l}, \vec{t'})]_+$

return $\vec{u} \in \mathbb{R}^k$ **for all** $u \in V \cup R$

TransE: Example

- $d(\text{Cat}, \text{Bear}) = 0.493$
- $d(\text{Cat}, \text{Fish}) = 2.245$
- $d(\text{Cat}, \text{Animal}) = 2.578$
- $d(\text{Cat}, \text{Mammal}) = 1.536$
- $d(\text{Fish}, \text{Mammal}) = 2.209$
- $d(\text{Animal}, \text{Mammal}) = 2.541$
- $\overrightarrow{\text{has}} = (-0.83, -2.64)^\top$
- $\overrightarrow{\text{is a}} = (0.68, -1.09)^\top$
- $\overrightarrow{\text{lives in}} = (2.11, -0.67)^\top$



TransE: Discussion

Pros:

- Simple and useful, handles most of the relations in practice
- Has a nice geometric interpretation: $\vec{h} + \vec{l} \approx \vec{t}$

Cons:

- Fails to handle symmetric and 1-to-N relations, see ComplEx (Trouillon et al., 2016)
- Assumes that all relations are in a single semantic vector space (not always true)

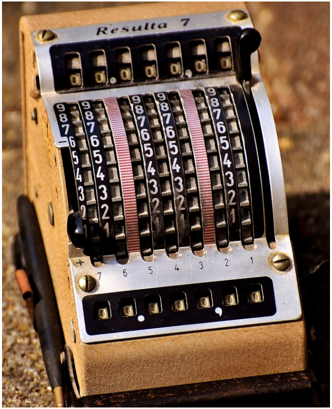
Implementations:

- https://github.com/glorotxa/SME

Instead of element-wise subtraction with a bias one may apply $\mathrm{DistMult}$ that *uses weighted element-wise dot product* (Yang et al., 2015): $\sum_{i=1}^{k} \vec{h}_i \cdot \vec{l}_i \cdot \vec{t}_i$.

Knowledge Graph Embeddings: Wrap-Up

- Typical applications for knowledge graph embeddings are link prediction, entity prediction, and relation prediction (called *knowledge graph completion*)
- TransE is not the only method; there are many approaches, including RESCAL (Nickel et al., 2011), ComplEx (Trouillon et al., 2016), R-GCN (Schlichtkrull et al., 2018), and many others (Ji et al., 2022)
- The choice of hyper-parameters **greatly** affects the model performance (Ruffinelli et al., 2020)



Source: Alexas_Fotos (2017)

Section 5

Case Studies

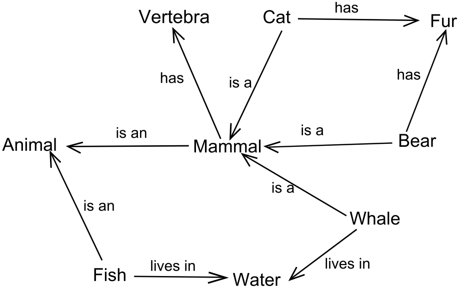
Case Studies

- Word Sense Disambiguation (Moro et al., 2014)
- Retrofitting Word Embeddings (Faruqui et al., 2015)
- Question Answering (Heo et al., 2022)

Source: Merrill (2014)

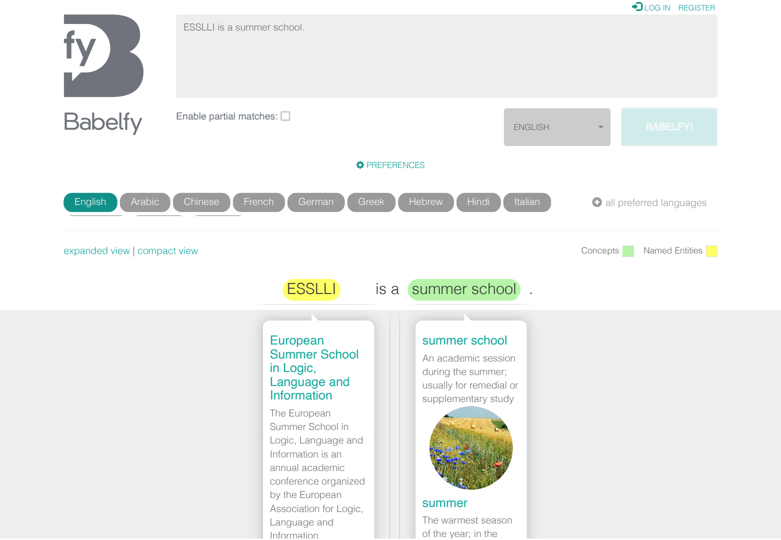
Word Sense Disambiguation

- **Babelfy** is an unsupervised graph-based approach for disambiguating the word senses and linking them to a knowledge graph (Moro et al., 2014), such as WordNet or BabelNet; http://babelfy.org/
- It performs part-of-speech tagging of each input sentence, extracts the candidates from the knowledge graph, and maps the best-matching concepts to the text fragments



Source: Konstable (2006)

Babelfy: Example



Source: http://babelfy.org/

Babelfy: Semantic Signatures

For preprocessing, Babelfy computes semantic signatures for all the concepts in the knowledge graph.

In a graph $G = (V, E)$ each edge $(v, v') \in E$ is weighted by the number of directed triangles it appears in:

$$w(v, v') = 1 + \left|(v, v', v'') \in V^3 : (v, v') \in E, (v', v'') \in E, (v'', v) \in E\right|$$

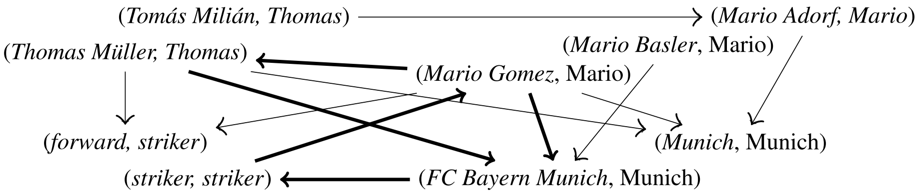
Then, a random walk with restart *aka* RWR (Tong et al., 2008) runs with transition probability:

$$P(v'|v) = \frac{w(v, v')}{\sum_{v'' \in V} w(v, v'')}$$

Finally, the signature $\mathrm{SemSign}(v) \subseteq V$ is composed of a set of nodes visited at least $\mu \in \mathbb{N}$ times by the random walk.

Babelfy: Candidate Disambiguation

1. Build a semantic interpretation graph for the sentence
2. Prune the graph with the *densest subgraph heuristic*
3. Link the text fragment to the candidate with the highest score



Source: Moro et al. (2014)

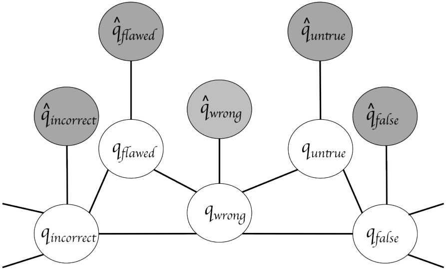
Babelfy showed state-of-the-art performance on many benchmarks; more modern methods apply pre-trained language models (Orlando et al., 2021).

Retrofitting

Faruqui et al. (2015) invented a method for incorporating knowledge from taxonomies into word embeddings.

- Given the taxonomy $G = (V, E)$ and a k-dimensional vector $\hat{q}_i \in \mathbb{R}^k$ for word $i \in V$, estimate the inferred vector $q_i \in \mathbb{R}^k$:

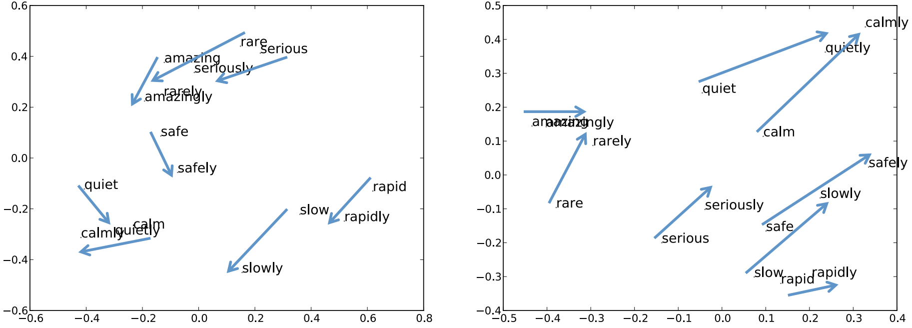
$$q_i = \frac{\sum_{j \in V:(i,j) \in E} \beta_{ij} q_j + \alpha_i \hat{q}_i}{\sum_{j \in V:(i,j) \in E} \beta_{ij} + \alpha_i}$$



Source: Faruqui et al. (2015)

Values of α and β control the strengths of the association; in practice they are set to $\alpha_i = 1, \forall i \in V$ and $\beta_{ij} = \deg(i)^{-1}, \forall i \in V, j \in V$.

Retrofitting: Example



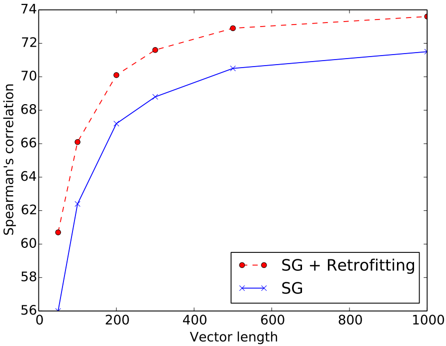
Source: Faruqui et al. (2015)

- Retrofitting allows aligning the analogy vectors, such as “adjective to verb,” in the same direction
- Euclidean distance for adjacent nodes after ten iterations are changed by less than 10^{-2}

Retrofitting: Results

Retrofitting adjusts the word vectors to be closer if the external semantic lexicon says so.

- Experiments on word similarity and syntactic relation tasks on different datasets show substantial improvement of vector quality, especially on smaller dimensions
- The effect is consistent across the different types of word embedding models, Word2Vec, GloVe, etc.
- Implementation is available at https://github.com/mfaruqui/retrofitting



Source: Faruqui et al. (2015)

Question Answering

The *knowledge-aware visual question answering* (KVQA) task requires world knowledge about named entities in images (Shah et al., 2019).

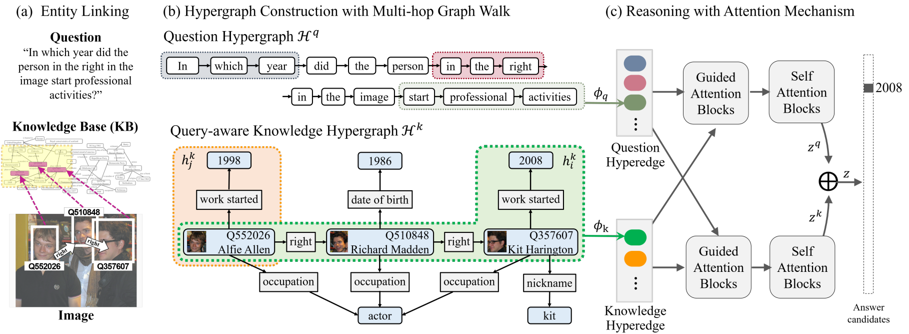
- **?** Which country produces the most of the fruit in the hand of the person you can see in the picture?
- **!** https://www.wikidata.org/wiki/Q668 (India)

Heo et al. (2022) proposed a method called **Hypergraph Transformer** that:

- constructs question and knowledge hypergraphs
- encodes their intra- and inter-associations

Source: McGuire (2015)

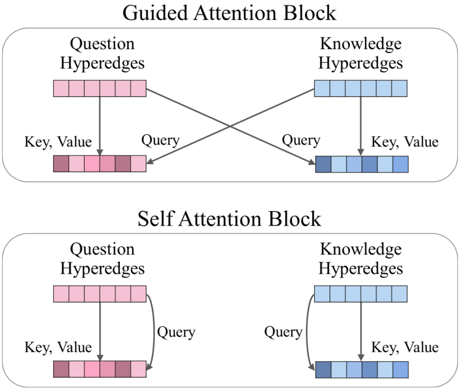
A **directed hypergraph** is a graph $\mathcal{H} = (V, E)$ defined by a set of nodes V and a set of hyperedges $E \subseteq 2^V \setminus \{\emptyset\}$; each hyperedge $e \in E$ has partial order.



Source: Heo et al. (2022)

Hypergraph Transformer: Prediction

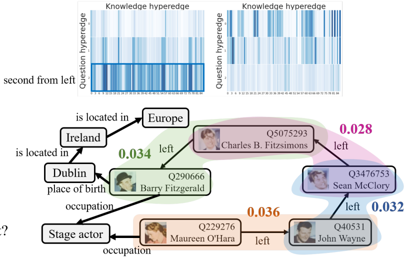
- Reasoning is performed using guided attention (Tsai et al., 2019) and self-attention blocks (Vaswani et al., 2017)
- The joint representation of question and knowledge is used to predict the answer
- Cross-entropy between prediction and ground truth is the loss function
- Hypergraph Transformer can mitigate the over-smoothing problem (Chen et al., 2020) by using hyperedge matching instead of message passing



Source: Heo et al. (2022)

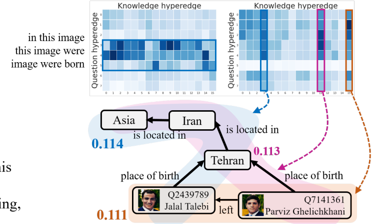
Hypergraph Transformer: Results

Question: Who is second from left?
Question type: Spatial, 1-hop



Prediction : Q5075293 (Charles B. Fitzsimons) ✔ (correct)

Question: How many people in this image were born in Asia?
Question type: Multi-hop, Counting, Multi-Entity, Multi-Relation



Prediction : 2 ✔ (correct)

Source: Heo et al. (2022)

Section 6

Conclusion

Conclusion

- Knowledge Graphs surround us every day in applications like search (Singhal, 2012) and help address difficult NLP tasks
- We discussed methods for building and embedding taxonomies, using multigraphs for representing multi-relational data, and embedding them
- Not covered here: entailment, rules, veracity, querying, and much, much more (Hogan et al., 2021)

Source: Adamovich (2015)

Resources I

Journals:

- Semantic Web Journal
- Journal of Web Semantics

Books:

- Knowledge Graphs (Kejriwal et al., 2021)
- Linguistic Linked Data (Cimiano et al., 2020)

Courses:

- Stanford CS224W: Machine Learning with Graphs

Resources II

Conferences:

- **ISWC**, International Conference on Semantic Web, http://iswc2022.semanticweb.org/
- **ESWC**, European Semantic Web Conference, https://www.eswc-conferences.org/
- **CIKM**, Conference on Information and Knowledge Management, http://www.cikmconference.org/
- **TheWebConf**, International World Wide Web Conference, https://thewebconf.org/
- **TextGraphs**, the Workshop on Graph-Based Algorithms for NLP, http://www.textgraphs.org/

Resources III

Implementations:

- PyKEEN (Ali et al., 2021)
- LibKGE (Ruffinelli et al., 2020)
- OpenKE (Han et al., 2018)

The list is non-exhaustive.

Questions?

Contacts

Dr. **Dmitry Ustalov**

https://github.com/dustalov

mailto:dmitry.ustalov@gmail.com

0000-0002-9979-2188

Revision: 5d35748

References I

Ali M. et al. (2021). PyKEEN 1.0: A Python Library for Training and Evaluating Knowledge Graph Embeddings. *Journal of Machine Learning Research*, vol. 22, no. 82, pp. 1–6. URL: `https://jmlr.org/papers/v22/20-825.html`.

Bartunov S. et al. (2016). Breaking Sticks and Ambiguities with Adaptive Skip-gram. *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*. AISTATS 2016. Cadiz, Spain: PMLR, pp. 130–138. URL: `https://proceedings.mlr.press/v51/bartunov16.html`.

Biemann C. (2013). Creating a system for lexical substitutions from scratch using crowdsourcing. *Language Resources and Evaluation*, vol. 47, no. 1, pp. 97–122. DOI: `10.1007/s10579-012-9180-5`.

Bonnabel S. (2013). Stochastic Gradient Descent on Riemannian Manifolds. *IEEE Transactions on Automatic Control*, vol. 58, no. 9, pp. 2217–2229. DOI: `10.1109/TAC.2013.2254619`.

Bordea G., Lefever E., and Buitelaar P. (2016). SemEval-2016 Task 13: Taxonomy Extraction Evaluation (TExEval-2). *Proceedings of the 10th International Workshop on Semantic Evaluation*. SemEval-2016. San Diego, CA, USA: Association for Computational Linguistics, pp. 1081–1091. DOI: `10.18653/v1/S16-1168`.

Bordes A. et al. (2013). Translating Embeddings for Modeling Multi-relational Data. *Advances in Neural Information Processing Systems 26*. NIPS 2013. Lake Tahoe, NV, USA: Curran Associates, Inc., pp. 2787–2795. URL: `https://papers.nips.cc/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Paper.pdf`.

Camacho-Collados J. et al. (2018). SemEval-2018 Task 9: Hypernym Discovery. *Proceedings of The 12th International Workshop on Semantic Evaluation*. SemEval-2018. New Orleans, LA, USA: Association for Computational Linguistics, pp. 712–724. DOI: `10.18653/v1/S18-1115`.

Chen D. et al. (2020). Measuring and Relieving the Over-Smoothing Problem for Graph Neural Networks from the Topological View. *Proceedings of the AAAI Conference on Artificial Intelligence*, AAAI-20 vol. 34, no. 5, pp. 3438–3445. DOI: `10.1609/aaai.v34i04.5747`.

Cimiano P. et al. (2020). Linguistic Linked Data: Representation, Generation and Applications. Springer International Publishing. ISBN: 978-3-030-30224-5. DOI: `10.1007/978-3-030-30225-2`.

Dong X. et al. (2014). Knowledge Vault: A Web-Scale Approach to Probabilistic Knowledge Fusion. *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '14. New York, NY, USA: Association for Computing Machinery, pp. 601–610. DOI: `10.1145/2623330.2623623`.

Faruqui M. et al. (2015). Retrofitting Word Vectors to Semantic Lexicons. *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. NAACL-HLT 2015. Denver, CO, USA: Association for Computational Linguistics, pp. 1606–1615. DOI: `10.3115/v1/N15-1184`.

Fellbaum C. (1998). WordNet: An Electronic Database. Massachusetts, MA, USA: MIT Press. ISBN: 978-0-262-06197-1. DOI: `10.7551/mitpress/7287.001.0001`.

References II

Fu R. et al. (2014). Learning Semantic Hierarchies via Word Embeddings. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics Volume 1: Long Papers*. ACL 2014. Baltimore, MD, USA: Association for Computational Linguistics, pp. 1199–1209. DOI: 10.3115/v1/P14-1113.

Han X. et al. (2018). OpenKE: An Open Toolkit for Knowledge Embedding. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. EMNLP 2018. Brussels, Belgium: Association for Computational Linguistics, pp. 139–144. DOI: 10.18653/v1/D18-2024.

Hansen P. C. (1987). The truncated*SVD* as a method for regularization. *BIT Numerical Mathematics*, vol. 27, no. 4, pp. 534–553. DOI: 10.1007/BF01937276.

Hearst M. A. (1992). Automatic Acquisition of Hyponyms from Large Text Corpora. *Proceedings of the 14th Conference on Computational Linguistics - Volume 2*. COLING 1992. Nantes, France: Association for Computational Linguistics, pp. 539–545. DOI: 10.3115/992133.992154.

Heo Y.-J. et al. (2022). Hypergraph Transformer: Weakly-Supervised Multi-hop Reasoning for Knowledge-based Visual Question Answering. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. ACL 2022. Dublin, Ireland: Association for Computational Linguistics, pp. 373–390. DOI: 10.18653/v1/2022.acl-long.29.

Hitzler P. (2021). A Review of the Semantic Web Field. *Communications of the ACM*, vol. 64, no. 2, pp. 76–83. DOI: 10.1145/3397512.

Hogan A. et al. (2021). Knowledge Graphs. *ACM Computing Surveys*, vol. 54, no. 4, pp. 1–37. DOI: 10.1145/3447772.

Ji S. et al. (2022). A Survey on Knowledge Graphs: Representation, Acquisition, and Applications. *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 2, pp. 494–514. DOI: 10.1109/TNNLS.2021.3070843.

Kejriwal M., Knoblock C. A., and Szekely P. (2021). Knowledge Graphs: Fundamentals, Techniques, and Applications. MIT Press. ISBN: 9780262045094. URL: https://mitpress.mit.edu/books/knowledge-graphs.

Kozareva Z. and Hovy E. (2010). A Semi-Supervised Method to Learn and Construct Taxonomies Using the Web. *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. EMNLP 2010. Cambridge, MA, USA: Association for Computational Linguistics, pp. 1110–1118. URL: https://aclanthology.org/D10-1108.

Lassila O. and McGuinness D. (2001). The Role of Frame-Based Representation on the Semantic Web. *Linköping Electronic Articles in Computer and Information Science*, vol. 6, no. 5. URL: https://ep.liu.se/ea/cis/2001/005/index.asp.

Le M. et al. (2019). Inferring Concept Hierarchies from Text Corpora via Hyperbolic Embeddings. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. ACL 2019. Florence, Italy: Association for Computational Linguistics, pp. 3231–3241. DOI: 10.18653/v1/P19-1313.

Lenat D. B. (1995). CYC: A Large-Scale Investment in Knowledge Infrastructure. *Communications of the ACM*, vol. 38, no. 11, pp. 33–38. DOI: 10.1145/219717.219745.

References III

Moro A., Raganato A., and Navigli R. (2014). Entity Linking meets Word Sense Disambiguation: a Unified Approach. *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 231–244. DOI: 10.1162/tacl_a_00179. the content is released under a CC BY-NC-ND license, but used with the permission of The MIT Press.

Navigli R. and Ponzetto S. P. (2012). BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, vol. 193, pp. 217–250. DOI: 10.1016/j.artint.2012.07.001.

Nickel M., Tresp V., and Kriegel H.-P. (2011). A Three-Way Model for Collective Learning on Multi-Relational Data. *Proceedings of the 28th International Conference on International Conference on Machine Learning*. ICML'11. Bellevue, WA, USA: Omnipress, pp. 809–816. URL: http://www.icml-2011.org/papers/438_icmlpaper.pdf.

Nickel M. and Kiela D. (2017). Poincaré Embeddings for Learning Hierarchical Representations. *Advances in Neural Information Processing Systems 30*. NIPS 2017. Vancouver, BC, Canada: Curran Associates, Inc., pp. 6341–6350. URL: https://proceedings.nips.cc/paper/2017/file/59dfa2df42d9e3d41f5b02bfc32229dd-Paper.pdf.

Orlando R. et al. (2021). AMuSE-WSD: An All-in-one Multilingual System for Easy Word Sense Disambiguation. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. EMNLP 2021. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 298–307. DOI: 10.18653/v1/2021.emnlp-demo.34.

Pellissier Tanon T. et al. (2016). From Freebase to Wikidata: The Great Migration. *Proceedings of the 25th International Conference on World Wide Web*. WWW '16. Montréal, Québec, Canada: International World Wide Web Conferences Steering Committee, pp. 1419–1428. DOI: 10.1145/2872427.2874809.

Řehůřek R. and Sojka P. (2010). Software Framework for Topic Modelling with Large Corpora. *Proceedings of the Workshop New Challenges for NLP Frameworks (NLPFrameworks 2010)*. Valletta, Malta: European Language Resources Association (ELRA), pp. 46–50. URL: http://lrec-conf.org/proceedings/lrec2010/workshops/W10.pdf.

Ristoski P. et al. (2018). RDF2Vec: RDF graph embeddings and their applications. *Semantic Web*, pp. 1–32. DOI: 10.3233/SW-180317.

Roller S., Kiela D., and Nickel M. (2018). Hearst Patterns Revisited: Automatic Hypernym Detection from Large Text Corpora. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. ACL 2018. Melbourne, VIC, Australia: Association for Computational Linguistics, pp. 358–363. DOI: 10.18653/v1/P18-2057.

Ruffinelli D., Broscheit S., and Gemulla R. (2020). You CAN Teach an Old Dog New Tricks! On Training Knowledge Graph Embeddings. *8th International Conference on Learning Representations*. ICLR 2020. Virtual: OpenReview.net. URL: https://openreview.net/forum?id=BkxSmlBFvr.

Schlichtkrull M. et al. (2018). Modeling Relational Data with Graph Convolutional Networks. *The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings*. Vol. 10843. Lecture Notes in Computer Science. Cham, Switzerland: Springer International Publishing, pp. 593–607. DOI: 10.1007/978-3-319-93417-4_38.

References IV

Seitner J. et al. (2016). A Large DataBase of Hypernymy Relations Extracted from the Web. *Proceedings of the Tenth International Conference on Language Resources and Evaluation*. LREC 2016. Portorož, Slovenia: European Language Resources Association (ELRA), pp. 360–367. URL: https://aclanthology.org/L16-1056.

Shah S. et al. (2019). KVQA: Knowledge-Aware Visual Question Answering. *Proceedings of the AAAI Conference on Artificial Intelligence*, AAAI-19 vol. 33, no. 1, pp. 8876–8884. DOI: 10.1609/aaai.v33i01.33018876.

Singhal A. (2012). Introducing the Knowledge Graph: things, not strings. Google. URL: https://blog.google/products/search/introducing-knowledge-graph-things-not/.

Snow R., Jurafsky D., and Ng A. (2004). Learning Syntactic Patterns for Automatic Hypernym Discovery. *Advances in Neural Information Processing Systems 17*. NIPS 2004. Vancouver, BC, Canada: A Bradford Book, pp. 1297–1304. URL: https://proceedings.nips.cc/paper/2004/file/358aee4cc897452c00244351e4d91f69-Paper.pdf.

Speer R., Chin J., and Havasi C. (2017). ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1. DOI: 10.1609/aaai.v31i1.11164.

Tong H., Faloutsos C., and Pan J.-Y. (2008). Random walk with restart: fast solutions and applications. *Knowledge and Information Systems*, vol. 14, no. 3, pp. 327–346. DOI: 10.1007/s10115-007-0094-2.

Trouillon T. et al. (2016). Complex Embeddings for Simple Link Prediction. *Proceedings of The 33rd International Conference on Machine Learning*. ICML 2016. New York, NY, USA: PMLR, pp. 2071–2080. URL: https://proceedings.mlr.press/v48/trouillon16.html.

Tsai Y.-H. H. et al. (2019). Multimodal Transformer for Unaligned Multimodal Language Sequences. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. ACL 2019. Florence, Italy: Association for Computational Linguistics, pp. 6558–6569. DOI: 10.18653/v1/P19-1656.

Ustalov D. et al. (2017). Negative Sampling Improves Hypernymy Extraction Based on Projection Learning. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. EACL 2017. Valencia, Spain: Association for Computational Linguistics, pp. 543–550. DOI: 10.18653/v1/E17-2087.

Ustalov D. et al. (2019). Watset: Local-Global Graph Clustering with Applications in Sense and Frame Induction. *Computational Linguistics*, vol. 45, no. 3, pp. 423–479. DOI: 10.1162/COLI_a_00354.

Vaswani A. et al. (2017). Attention Is All You Need. *Advances in Neural Information Processing Systems 30*. NIPS 2017. Vancouver, BC, Canada: Curran Associates, Inc., pp. 6000–6010. URL: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

References V

Velardi P., Faralli S., and Navigli R. (2013). OntoLearn Reloaded: A Graph-Based Algorithm for Taxonomy Induction. *Computational Linguistics*, vol. 39, no. 3, pp. 665–707. DOI: 10.1162/COLI_a_00146. the content is released under a CC BY-NC-ND license, but used with the permission of The MIT Press.

Walker A. J. (1977). An Efficient Method for Generating Discrete Random Variables with General Distributions. *ACM Transactions on Mathematical Software*, vol. 3, no. 3, pp. 253–256. DOI: 10.1145/355744.355749.

Wu L. et al. (2018). StarSpace: Embed All The Things! *The Thirty-Second AAAI Conference on Artificial Intelligence*. AAAI-18. Association for the Advancement of Artificial Intelligence, pp. 5569–5577. URL: https://ojs.aaai.org/index.php/AAAI/article/view/11996.

Yang B. et al. (2015). Embedding Entities and Relations for Learning and Inference in Knowledge Bases. *3rd International Conference on Learning Representations, Conference Track Proceedings*. ICLR 2015. San Diego, CA, USA. arXiv: 1412.6575 [cs.CL].

Supplementary Media I

Adamovich O. (September 3, 2015). Girls Whispering Best Friends. Pixabay. URL: `https://pixabay.com/images/id-914823/`. Licensed under Pixabay License.

Alexandrov O. (July 14, 2007). Illustration of a Vector space. Wikimedia Commons. URL: `https://commons.wikimedia.org/wiki/File:Vector_space_illust.svg`. Licensed under Public Domain.

Alexas_Fotos (October 7, 2017). Calculating Machine Resulta Old. Pixabay. URL: `https://pixabay.com/images/id-2825179/`. Licensed under Pixabay License.

bamenny (February 24, 2016). Robot Flower Technology. Pixabay. URL: `https://pixabay.com/images/id-1214536/`. Licensed under Pixabay License.

Buissinne S. (August 25, 2016). Dictionary Reference Book Learning. Pixabay. URL: `https://pixabay.com/images/id-1619740/`. Licensed under Pixabay License.

Cimiano P., McCrae J. P., and Buitelaar P. (May 10, 2016). Lexicon Model for Ontologies. Community Report. Ontology-Lexicon Community Group. URL: `https://www.w3.org/2016/05/ontolex/`. Used with author's permission.

Finnsson I. (May 19, 2017). Books Covers Book Case. Pixabay. URL: `https://pixabay.com/images/id-2321934/`. Licensed under Pixabay License.

Free-Photos (August 9, 2016). Person Mountain Top Achieve. Pixabay. URL: `https://pixabay.com/images/id-1245959/`. Licensed under Pixabay License.

Kittner L. (October 26, 2015). Cook Cooking School Pan. Pixabay. URL: `https://pixabay.com/images/id-1002505/`. Licensed under Pixabay License.

Konstable (November 9, 2006). An example of a Semantic Network. Wikimedia Commons. URL: `https://commons.wikimedia.org/wiki/File:Semantic_Net.svg`. Licensed under Public Domain.

McGuire R. (March 24, 2015). Suit Business Man. Pixabay. URL: `https://pixabay.com/images/id-673697/`. Licensed under Pixabay License.

Merrill B. (July 24, 2014). Pedestrians People Busy. Pixabay. URL: `https://pixabay.com/images/id-400811/`. Licensed under Pixabay License.

Rahman Rony M. (May 31, 2016). Mad Max Fury Car Monster. Pixabay. URL: `https://pixabay.com/images/id-1426796/`. Licensed under Pixabay License.

rawpixel (April 18, 2017). Calm Freedom Location. Pixabay. URL: `https://pixabay.com/images/id-2218409/`. Licensed under Pixabay License.

Trevorgoodchild (April 26, 2008). Poincare disc hyperbolic parallel lines. Wikimedia Commons. URL: `https://commons.wikimedia.org/wiki/File:Poincare_disc_hyperbolic_parallel_lines.svg`. Licensed under Public Domain.