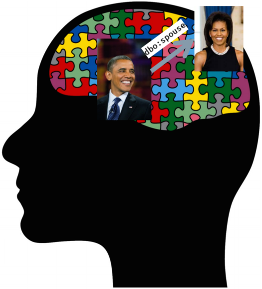
Joint Word and Entity Embeddings for Entity Retrieval from a Knowledge Graph

Fedor Nikolaev[1,2] and Alexander Kotov[1]

[1] Textual Data Analytics (TEANA) Lab, Wayne State University, USA

[2] Kazan Federal University, Russia

Knowledge graphs



- *Knowledge graphs* are a way to represent knowledge as a set of subject-predicate-object (SPO) triples
- An *entity* is an abstract or material object designated by an identifier (e.g. URI `http://dbpedia.org/resource/Barack_Obama`, in the case of DBpedia)
- *Entities* are always *subjects* in SPO triples
- Entities are connected with other entities, literals or scalars by relations or *predicates* (e.g. *dbo:genre*, *dbo:knownFor*, *dbo:spouse*, *dbp:memberOf*, etc.)
- Each SPO triple represents a simple fact (e.g. $\textit{dbr:Barack\_Obama} \xrightarrow{\textit{dbo:spouse}} \textit{dbr:Michelle\_Obama}$)

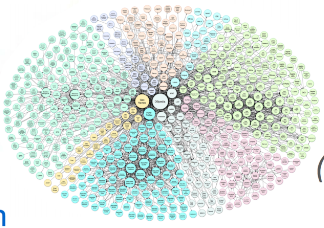
Existing knowledge graphs

WIKIDATA

yago
select knowledge

NELL: Never-Ending Language Learning

DBpedia



Facebook's Entity Graph

OpenIE (Reverb, OLLIE)

Microsoft's Satori

Google's Knowledge Graph

DBpedia entity page (rendered)

dbpedia.org/page/Barack_Obama

DBpedia   Browse using   Formats   Faceted Browser   Sparql Endpoint

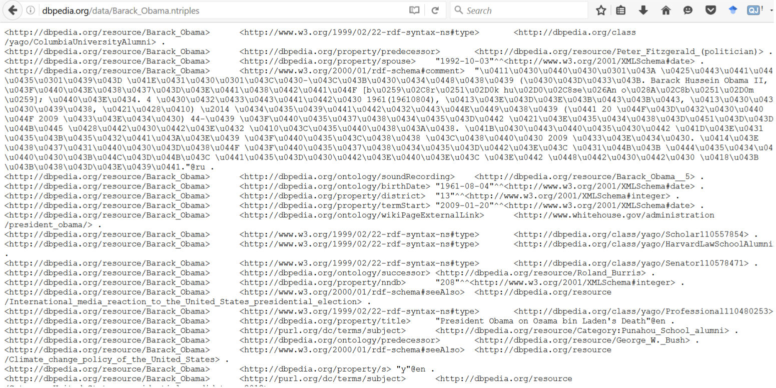
About: Barack Obama

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

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

Property	Value
dbo:abstract	▪ Barack Hussein Obama II (US /bəˈrɑːk huːˈseɪn oʊˈbɑːmə/; born August 4, 1961) is the 44th and current President of the United States, and the first African American to hold the office. Born in Honolulu, Hawaii, Obama is a graduate of Columbia University and Harvard Law School, where he served as president of the Harvard Law Review. He was a community organizer in Chicago before earning his law degree. He worked as a civil rights attorney and taught constitutional law at University of Chicago Law School between 1992 and 2004. He served three terms representing the 13th District in the Illinois Senate from 1997 to 2004, running unsuccessfully for the United States House of Representatives in 2000.In 2004, Obama received national attention during his campaign to represent Illinois in the United States Senate with his victory in the March Democratic Party primary, his keynote address at the Democratic National Convention in July, and his election to the Senate in November. He began his presidential campaign in 2007 and, after a close primary campaign against Hillary Rodham Clinton in 2008, he won sufficient delegates in the Democratic Party primaries to receive the presidential nomination. He then defeated Republican nominee John McCain in the general election, and was inaugurated as president on January 20, 2009. Nine months after his inauguration, Obama was named the 2009 Nobel Peace Prize laureate.During his first two

DBpedia entity page (RDF triples)




```
<http://dbpedia.org/resource/Barack_Obama>	<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>	<http://dbpedia.org/class
/yago/ColumbiaUniversityAlumni> .
<http://dbpedia.org/resource/Barack_Obama>	<http://dbpedia.org/property/predecessor>	<http://dbpedia.org/resource/Peter_Fitzgerald_(politician)> .
<http://dbpedia.org/resource/Barack_Obama>	<http://dbpedia.org/property/spouse>	"1992-10-03"^^<http://www.w3.org/2001/XMLSchema#date> .
<http://dbpedia.org/resource/Barack_Obama>	<http://www.w3.org/2000/01/rdf-schema#comment>	"\u0411\u0430\u0440\u0430\u0301\u043A \u0425\u0443\u0441\u044
\u0435\u0301\u0439\u043D \u041E\u0431\u0430\u0301\u043C\u0430-\u043C\u043B\u0430\u0434\u0448\u0438\u0439 (\u0430\u043D\u0433\u043B. Barack Hussein Obama II,
\u043F\u0440\u043E\u0438\u0437\u043D\u043E\u0441\u0438\u0442\u0441\u044F [b\u0259\u02C8r\u0251\u02D0k hu\u02D0\u02C8se\u026An o\u028A\u02C8b\u0251\u02D0m
\u0259]; \u0440\u043E\u0434. 4 \u0430\u0432\u0433\u0443\u0441\u0442\u0430 1961(19610804), \u0413\u043E\u043D\u043E\u043B\u0443\u043B\u0443, \u0413\u0430\u043
\u0430\u0439\u0438, \u0421\u0428\u0410) \u2014 \u0434\u0435\u0439\u0441\u0442\u0432\u0443\u044E\u0449\u0438\u0439 (\u0441 20 \u044F\u043D\u0432\u0430\u0440
\u044F 2009 \u0433\u043E\u0434\u0430) 44-\u0439 \u043F\u0440\u0435\u0437\u0438\u0434\u0435\u043D\u0442 \u0421\u043E\u0435\u0434\u0438\u043D\u0451\u043D\u043D
\u044B\u0445 \u0428\u0442\u0430\u0442\u043E\u0432 \u0410\u043C\u0435\u0440\u0438\u043A\u0438. \u041B\u0430\u0443\u0440\u0435\u0430\u0442 \u041D\u043E\u0431
\u0435\u043B\u0435\u0432\u0441\u043A\u043E\u0439 \u043F\u0440\u0435\u043C\u0438\u0438 \u043C\u0438\u0440\u0430 2009 \u0433\u043E\u0434\u0430. \u0414\u043E
\u0438\u0437\u0431\u0440\u0430\u043D\u0438\u044F \u043F\u0440\u0435\u0437\u0438\u0434\u0435\u043D\u0442\u043E\u043C \u0431\u044B\u043B \u0444\u0435\u0434\u04
\u0440\u0430\u043B\u044C\u043D\u044B\u043C \u0441\u0435\u043D\u0430\u0442\u043E\u0440\u043E\u043C \u043E\u0442 \u0448\u0442\u0430\u0442\u0430 \u0418\u043B
\u043B\u0438\u043D\u043E\u0439\u0441."@ru .
<http://dbpedia.org/resource/Barack_Obama>	<http://dbpedia.org/ontology/soundRecording>	<http://dbpedia.org/resource/Barack_Obama__5> .
<http://dbpedia.org/resource/Barack_Obama>	<http://dbpedia.org/ontology/birthDate> "1961-08-04"^^<http://www.w3.org/2001/XMLSchema#date> .
<http://dbpedia.org/resource/Barack_Obama>	<http://dbpedia.org/property/district>	"13"^^<http://www.w3.org/2001/XMLSchema#integer> .
<http://dbpedia.org/resource/Barack_Obama>	<http://dbpedia.org/property/termStart> "2009-01-20"^^<http://www.w3.org/2001/XMLSchema#date> .
<http://dbpedia.org/resource/Barack_Obama>	<http://dbpedia.org/ontology/wikiPageExternalLink>	<http://www.whitehouse.gov/administration
/president_obama/> .
<http://dbpedia.org/resource/Barack_Obama>	<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>	<http://dbpedia.org/class/yago/Scholar110557854> .
<http://dbpedia.org/resource/Barack_Obama>	<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>	<http://dbpedia.org/class/yago/HarvardLawSchoolAlumni
.
<http://dbpedia.org/resource/Barack_Obama>	<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>	<http://dbpedia.org/class/yago/Senator110578471> .
<http://dbpedia.org/resource/Barack_Obama>	<http://dbpedia.org/ontology/successor> <http://dbpedia.org/resource/Roland_Burris> .
<http://dbpedia.org/resource/Barack_Obama>	<http://dbpedia.org/property/nndb>	"208"^^<http://www.w3.org/2001/XMLSchema#integer> .
<http://dbpedia.org/resource/Barack_Obama>	<http://www.w3.org/2000/01/rdf-schema#seeAlso>	<http://dbpedia.org/resource
/International_media_reaction_to_the_United_States_presidential_election> .
<http://dbpedia.org/resource/Barack_Obama>	<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>	<http://dbpedia.org/class/yago/Professional110480253>
<http://dbpedia.org/resource/Barack_Obama>	<http://dbpedia.org/property/title>	"President Obama on Osama bin Laden's Death"@en .
<http://dbpedia.org/resource/Barack_Obama>	<http://purl.org/dc/terms/subject>	<http://dbpedia.org/resource/Category:Punahou_School_alumni> .
<http://dbpedia.org/resource/Barack_Obama>	<http://dbpedia.org/ontology/predecessor>	<http://dbpedia.org/resource/George_W._Bush> .
<http://dbpedia.org/resource/Barack_Obama>	<http://www.w3.org/2000/01/rdf-schema#seeAlso>	<http://dbpedia.org/resource
/Climate_change_policy_of_the_United_States> .
<http://dbpedia.org/resource/Barack_Obama>	<http://dbpedia.org/property/s> "y"@en .
<http://dbpedia.org/resource/Barack_Obama>	<http://purl.org/dc/terms/subject>	<http://dbpedia.org/resource
```

DBpedia structural components

- **Entities**
  - *dbr:Barack_Obama*
  - *dbr:Michelle_Obama*
- **Categories**
  - *dbc:Presidents_of_the_United_States*
  - *dbc:Critics_of_Islamophobia*
- **Literals**
  - *dbr:Barack_Obama dbo:birthDate* **"1961-08-04"**
  - *dbr:Barack_Obama foaf:gender* **"male"**
- **Predicates**
  - *dbo:birthDate*
  - *dbo:spouse*

Entity retrieval from a knowledge graph

- **Entity Search**: finding an entity based on its description
  - *"Ben Franklin"*
  - *"Einstein Relativity theory"*
- **List Search**: finding a set of entities based on their description
  - *"Formula 1 drivers who won the Monaco Grand Prix"*
  - *"animals lay eggs mammals"*
- **Attribute Search**: find a property of an entity
  - *"When was Intel founded?"*
  - *"What is the elevation of Karakoram?"*

Term-based KG entity retrieval

Traditionally, entities are represented as multi-field documents and retrieved using structured document retrieval models:

- Fielded Sequential Dependence Model (FSDM) [Zhiltsov et al., SIGIR 2015]
- Parametrized Fielded Sequential Dependence Model (PFSDM) [Nikolaev et al., SIGIR 2016]
- BM25F [Robertson and Zaragoza, Foundations and Trends in IR, 2009]

Key limitation: matching of queries to entities is performed at the word level

Network embedding methods

- Aim to embed network nodes into a low-dimensional vector space
- **Main idea:** apply of word embedding methods to sequences obtained using *random walks* on a given network
- Popular methods:
  - DeepWalk [Perozzi et al., KDD 2014]
  - LINE [Tang et al., WWW 2015]
  - node2vec [Grover and Leskovec, KDD 2016]
  - struc2vec [Ribeiro et al., KDD 2017]

Problems with network embeddings

1. We can apply network embeddings to knowledge graphs, but can't utilize entity embedding obtained this way directly in word-based retrieval models
2. We can use only word embeddings, but they utilize no information from a given knowledge graph

Proposed method

We propose **K**nowledge graph **E**ntity and **W**ord **E**mbeddings for **R**etrieval (KEWER), a method that given a KG G:

- learns distributed representations of words (in predicates, literals, entity and category names) as well as entities and categories in G in the same embedding space
- utilizes the local structure of G when learning these embeddings

KEWER steps

KEWER consists of three steps:

1. **Random Walks from Knowledge Graph Entities**
   Starting from each KG entity, generate γ random walks of length $\leq t$.
   Example:
   $$\textit{dbr:Pierre\_Curie} \xrightarrow{\textit{dbp:spouse}} \textit{dbr:Marie\_Curie} \xrightarrow{\textit{dbp:knownFor}} \textit{dbr:Radioactivity}$$

KEWER steps

KEWER consists of three steps:

1. **Random Walks from Knowledge Graph Entities**
   Starting from each KG entity, generate γ random walks of length $\leq t$.
   Example:
   $dbr{:}Pierre\_Curie \xrightarrow{dbp{:}spouse} dbr{:}Marie\_Curie \xrightarrow{dbp{:}knownFor} dbr{:}Radioactivity$
2. **Replacement with Surface Forms**
   Randomly replace entity and category URIs with their surface forms (i.e. word tokens) in sequences of entity and category URIs, predicates and literals generated by random walks on G. The surface form of an entity or category for URI replacement is chosen uniformly at random from a set of available surface forms.

KEWER objective

3. **Learn Embeddings**
   Learn embeddings of words, entities and categories by maximizing the log-likelihood of observing other KG elements (word, entity or category) ξ_{i+j} in the context of each KG element ξ_i:

$$\frac{1}{T}\sum_{i=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(\xi_{i+j}|\xi_i),\ \xi_{1\ldots T}\in\Xi,$$

$$\Xi = E\cup N\begin{cases}\cup\, K, \text{ if } \textbf{categories} \text{ are used}\\ \cup\, V, \text{ if } \textbf{literals} \text{ are used}\\ \cup\, P, \text{ if } \textbf{predicates} \text{ are used.}\end{cases}$$

where $p(\xi_O|\xi_I)$ is defined using softmax:

$$p(\xi_O|\xi_I) = \frac{\exp(\mathbf{v}'^{\top}_{\xi_O}\mathbf{v}_{\xi_I})}{\sum_{k=1}^{|\Xi|}\exp(\mathbf{v}'^{\top}_{\xi_k}\mathbf{v}_{\xi_I})}.$$

Entity retrieval using KEWER embeddings

Embedding of a query q is a weighted sum of the embeddings of individual query words $\mathbf{v}_{q_i}$ [Arora et al., ICLR 2017]:

$$\mathbf{q} = \sum_{i=1}^{k} \frac{a}{p(q_i) + a} \mathbf{v}_{q_i}$$

Entity retrieval using KEWER embeddings

Embedding of a query q is a weighted sum of the embeddings of individual query words $\mathbf{v}_{q_i}$ [Arora et al., ICLR 2017]:

$$\mathbf{q} = \sum_{i=1}^{k} \frac{a}{p(q_i) + a} \mathbf{v}_{q_i}$$

Entities are scored according to the cosine similarity between entity embedding and query embedding:

$$KEWER(q, e) = \cos(\mathbf{q}, \mathbf{v}_e)$$

Entity retrieval using KEWER embeddings

Embedding of a query q is a weighted sum of the embeddings of individual query words $\mathbf{v}_{q_i}$ [Arora et al., ICLR 2017]:

$$\mathbf{q} = \sum_{i=1}^{k} \frac{a}{p(q_i) + a} \mathbf{v}_{q_i}$$

Entities are scored according to the cosine similarity between entity embedding and query embedding:

$$KEWER(q, e) = \cos(\mathbf{q}, \mathbf{v}_e)$$

These scores can be interpolated with BM25F scores:

$$MM(q, e) = \beta KEWER(q, e) + (1 - \beta) BM25F(q, e), \;\; 0 \leq \beta \leq 1$$

Utilizing entity linking

To fine-tune query's vector representation, we can perform entity linking on a query and add embeddings of the linked entities to the query embedding:

$$\mathbf{q}_{el} = \sum_{i=1}^{k} \frac{a}{p(q_i) + a} \mathbf{v}_{q_i} + \sum_{i=1}^{m} s(e_i) \mathbf{v}_{e_i},$$

where $s(e_i)$ is the entity linker's annotation score for the entity e_i.

Jointly

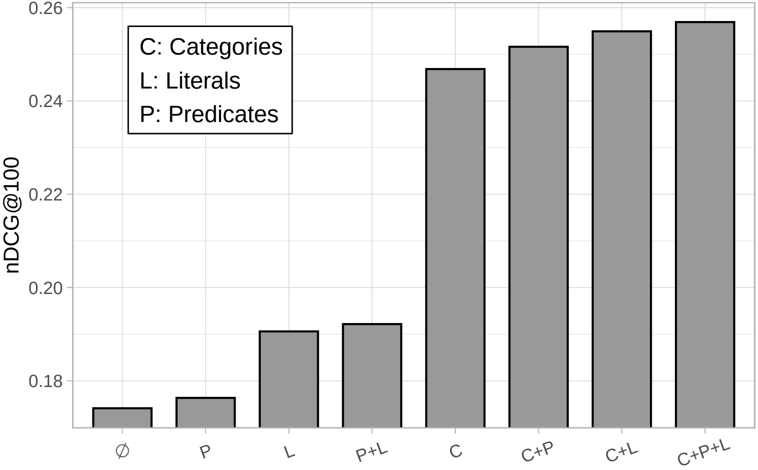
As a baseline, we used our implementation of the Jointly word and entity embedding method [Wang et al., EMNLP 2014]:

$$\mathcal{L}_J = \mathcal{L}_K + \mathcal{L}_T + \mathcal{L}_A$$

- Knowledge component loss $\mathcal{L}_K$ is a translation-based loss for triples (similar to TransE [Bordes et al., NIPS 2013]).
- Text component loss $\mathcal{L}_T$ corresponds to CBOW word embeddings trained on entity abstracts.
- Alignment loss $\mathcal{L}_A$ aligns embeddings for words and entities based on entity abstracts.

Several similar models [Xie et al., AAAI 2016; Zhong et al., EMNLP 2015] were proposed for KG link prediction and triplet classification tasks.

Usefulness of KG structural components



$nDCG_{100}$ when using different combinations of categories, literals and predicates to train KEWER embeddings

Retrieval performance with different entity linkers

Sp stands for DBpedia Spotlight [Daiber et al., I-SEMANTICS 2013], *SM* for SMAPH [Cornolti et al., WWW 2016], *N* for Nordlys [Hasibi et al., SIGIR 2017].

Model	nDCG_{10}	nDCG_{100}	MAP
KEWER	0.2102	0.2569	0.1449
$\mathrm{KEWER}_{el\text{-}Sp}$	0.2417	0.2803	0.1579
$\mathrm{KEWER}_{el\text{-}SM}$	**0.2704**	**0.3098**	**0.1780**
$\mathrm{KEWER}_{el\text{-}N}$	0.2660	0.3083	0.1775
Jointly (desp)	0.0486	0.0547	0.0211
$\mathrm{Jointly}_{el\text{-}Sp}$ (desp)	0.1603	0.1587	0.0838
$\mathrm{Jointly}_{el\text{-}SM}$ (desp)	**0.1981**	**0.1924**	**0.1014**
$\mathrm{Jointly}_{el\text{-}N}$ (desp)	0.1870	0.1814	0.0981
Jointly (sf)	0.0291	0.0393	0.0137
$\mathrm{Jointly}_{el\text{-}Sp}$ (sf)	0.1365	0.1357	0.0684
$\mathrm{Jointly}_{el\text{-}SM}$ (sf)	0.1685	0.1627	0.0795
$\mathrm{Jointly}_{el\text{-}N}$ (sf)	0.1624	0.1598	0.0836

Re-ranking performance

Statistically significant improvements (determined by a randomized test with $\alpha = 0.05$) over BM25F and BM25F+word2vec are indicated by "$\star$" and "$\dagger$", respectively.

SemSearch ES

Model	$nDCG_{10}$	$nDCG_{100}$	MAP
BM25F	0.6606	0.7391	0.5693
BM25F+word2vec	**0.6798***	**0.7445**	**0.5712**
BM25F+KEWER	0.6606	0.7333	0.5627
BM25F+$KEWER_{el\text{-}SM}$	0.6619	0.7409	0.5690

INEX-LD

Model	$nDCG_{10}$	$nDCG_{100}$	MAP
BM25F	0.4456	0.5127	0.3271
BM25F+word2vec	0.4591	0.5227	0.3406*
BM25F+KEWER	**0.4676***	**0.5298***	**0.3417***
BM25F+$KEWER_{el\text{-}SM}$	0.4577*	0.5215*	0.3363*

ListSearch

Model	$nDCG_{10}$	$nDCG_{100}$	MAP
BM25F	0.4287	0.4989	0.3506
BM25F+word2vec	0.4235	0.5055*	0.3551
BM25F+KEWER	0.4402†	0.5210*†	0.3752*†
BM25F+$KEWER_{el\text{-}SM}$	**0.4451***†	**0.5251***†	**0.3777***†

QALD-2

Model	$nDCG_{10}$	$nDCG_{100}$	MAP
BM25F	0.3442	0.4375	0.2861
BM25F+word2vec	0.3567*	0.4504*	0.2986*
BM25F+KEWER	**0.3859***†	**0.4743***†	**0.3154***†
BM25F+$KEWER_{el\text{-}SM}$	0.3800*†	0.4700*†	0.3081*†

All queries

Model	$nDCG_{10}$	$nDCG_{100}$	MAP
BM25F	0.4631	0.5416	0.3792
BM25F+word2vec	0.4730*	0.5504*	0.3874*
BM25F+KEWER	**0.4831***†	**0.5602***†	**0.3955***†
BM25F+$KEWER_{el\text{-}SM}$	0.4807*†	0.5601*†	0.3944*†

Example query

Top 10 entities for the query *"wonders of the ancient world"* when using term-based retrieval with BM25F and cosine similarity based on query and entity embeddings. Relevant results are *italicized* and highly relevant results are **boldfaced**.

BM25F	KEWER
Seven Wonders of the Ancient World	**Colossus of Rhodes**
7 Wonders of the Ancient World (video game)	**Statue of Zeus at Olympia**
Wonders of the World	**Temple of Artemis**
Seven Ancient Wonders	List of archaeoastronomical sites by country
The Seven Fabulous Wonders	**Hanging Gardens of Babylon**
The Seven Wonders of the World (album)	Antikythera mechanism
Times of India's list of seven wonders of India	Timeline of ancient history
Lighthouse of Alexandria	*Wonders of the World*
7 Wonders (board game)	*Lighthouse of Alexandria*
Colossus of Rhodes	**Great Pyramid of Giza**

Conclusions

1. Using all KG structural components (entities, categories, literals, and predicates) to learn KEWER embeddings results in the highest retrieval accuracy on DBpedia-Entity v2.
2. KEWER is particularly suitable for improving the ranking of results of complex entity search queries, such as question answering, list search, and keyword queries, where it can provide semantic relevance signal not captured by the retrieval models based on term matching.

Code, runs, and embeddings are available at
`https://github.com/teanalab/kewer`

Thank you! Questions?