Learning Embeddings to lexicalise RDF Properties

Laura Perez-Beltrachini   Claire Gardent

CNRS/LORIA
Nancy, France

15 November 2016
ILCC, School of Informatics, University of Edinburgh

Lexicalisation of RDF properties

- Generating text from RDF data involves a serie of subtasks
  - Property lexicalisation subtask

$$\textit{RDF property} \xrightarrow{lex} \textit{Natural Language Phrase(s)}$$

$$\textsc{hasWonPrize} \xrightarrow{lex} \{\ \textit{“was honoured with”}\ ,\ \textit{“received”}\ \}$$

- Challenges

indirect	$\textsc{routeEnd} \xrightarrow{lex} \{\ \textit{“finishes at”}\ \}$
opaque	$\textsc{crew1up} \xrightarrow{lex} \{\ \textit{“is the commander of”}\ \}$
variety	find alternative lexicalisations

Lexicalisation of RDF properties

- Generating text from RDF data involves a serie of subtasks
  - Property lexicalisation subtask

$$\textit{RDF property} \xrightarrow{\textit{lex}} \textit{Natural Language Phrase(s)}$$

$$\textsc{hasWonPrize} \xrightarrow{\textit{lex}} \{\ \textit{“was honoured with”}\ ,\ \textit{“received”}\ \}$$

- Challenges

indirect $\textsc{routeEnd} \xrightarrow{\textit{lex}} \{\ \textit{“finishes at”}\ \}$

opaque $\textsc{crew1up} \xrightarrow{\textit{lex}} \{\ \textit{“is the commander of”}\ \}$

variety find alternative lexicalisations

Lexicalisation of RDF properties

- Generating text from RDF data involves a serie of subtasks
  - Property lexicalisation subtask

$$\textit{RDF property} \xrightarrow{lex} \textit{Natural Language Phrase(s)}$$

$$\textsc{hasWonPrize} \xrightarrow{lex} \{ \text{“\textit{was honoured with}”} , \text{“\textit{received}”} \}$$

- Challenges

indirect	$\textsc{routeEnd} \xrightarrow{lex} \{ \text{“\textit{finishes at}”} \}$
opaque	$\textsc{crew1up} \xrightarrow{lex} \{ \text{“\textit{is the commander of}”} \}$
variety	find alternative lexicalisations

Existing approaches

- words appearing in relation names or labels

  Quelo [Trevisan, 2010]

  CREW1UP $\xrightarrow{lex}$ { *"is the crew 1 up of"*}

- distant supervision ideas – linking named entities

  DBlexipedia$_e$ [Walter et al., 2014a, Walter et al., 2014b]

  SPOUSE $\xrightarrow{lex}$ { *"divorced from"*}

- open information (relation) extraction
  - search for relation mentions in text / unrestricted
  - exception: clustering phase + link to DBPedia properties

    Patty [Nakashole et al., 2012]

Our approach is inspired by the work of [Bordes et al., 2014]

- Question Answering over general purpose Knowledge Bases (KB)
- distributed word representations, synthetic data, multi-task training with paraphrases

Existing approaches

- words appearing in relation names or labels
  Quelo [Trevisan, 2010]
  CREW1UP $\xrightarrow{lex}$ { *"is the crew 1 up of"*}
- distant supervision ideas – linking named entities
  DBlexipedia$_e$ [Walter et al., 2014a, Walter et al., 2014b]
  SPOUSE $\xrightarrow{lex}$ { *"divorced from"*}
- open information (relation) extraction
  - search for relation mentions in text / unrestricted
  - exception: clustering phase + link to DBPedia properties
    Patty [Nakashole et al., 2012]

Our approach is inspired by the work of [Bordes et al., 2014]

- Question Answering over general purpose Knowledge Bases (KB)
- distributed word representations, synthetic data, multi-task training with paraphrases

Existing approaches

- words appearing in relation names or labels
  Quelo [Trevisan, 2010]
  CREW1UP $\xrightarrow{lex}$ { *"is the crew 1 up of"*}
- distant supervision ideas – linking named entities
  DBlexipedia$_e$ [Walter et al., 2014a, Walter et al., 2014b]
  SPOUSE $\xrightarrow{lex}$ { *"divorced from"*}
- open information (relation) extraction
  - search for relation mentions in text / unrestricted
  - exception: clustering phase + link to DBPedia properties
    Patty [Nakashole et al., 2012]

Our approach is inspired by the work of [Bordes et al., 2014]

- Question Answering over general purpose Knowledge Bases (KB)
- distributed word representations, synthetic data, multi-task training with paraphrases

Lexicalisation with embeddings: Intuition

Embedding RDF triples and NL phrases in the same continuous space

- $\vec{t}$ vector representation for triple $t = (s, p, o)$
- $\vec{v}$ vector representation for NL phrase $v =$ *"S relation mention O"*
- similarity scoring function $S_{t/v}$ over $\vec{t}$ and $\vec{v}$

✓ $\xrightarrow{lex}$ (S, HASWONPRIZE, O) *"S was honoured with O"* (high $S_{t/v}$)

* $\xrightarrow{lex}$ (S, HASWONPRIZE, O) *"S broke O"* (low $S_{t/v}$)

Rest of the talk

Introduction

Lexicalisation approach

Evaluation and Results

Conclusion

Approach overview

RDF property $\xrightarrow{lex}$ { ??? }

1. Learn embeddings of RDF triples and NL phrases

   Similarity function $S_{t/v}(t, v)$

2. Build sets of candidate NL phrases (Lex_p)

3. Rank candidate phrases using the scoring similarity function

$$\hat{v}(t) = \arg\max_{v\prime \in Lex_p} \quad S_{t/v}(t, v\prime)$$

4. Extract lexicalisations from top ranked candidates

Embeddings model

$$\boxed{S_{t/v}(t, v) = f(t)^\top . g(v)}$$

$$f(t) = K^\top . \phi(t)$$

$$g(v) = W^\top . \psi(v)$$

$K \in \mathbb{R}^{n_k \times d}$ embedding matrix for KB symbols
$W \in \mathbb{R}^{n_w \times d}$ embedding matrix for words

Training

- $\mathcal{T} = \{(t_i, v_i); i = 1, \cdots, |\mathcal{T}|\}$
  - automatic generation of NL phrases ($\approx$ 5 per triple)

t_i (ARISTOTLE, INFLUENCED, CHRISTIAN_PHILOSOPHY)
v_i *"Christian philosophy is influenced by Aristotle."*

- data corruption

t' (ARISTOTLE, COMPUTINGMEDIA, CHRISTIAN_PHILOSOPHY)
v_i *"Christian philosophy is influenced by Aristotle."*

- Ranking loss function

$$\forall i, \forall t' \neq t_i,\ [1 - S_{s/v}(t_i, v_i) + S_{s/t}(t', v_i)]$$

Multitask training of word embeddings on paraphrases

- extend vocabulary coverage
- cover alternative lexicalisations
- auxiliary task: paraphrases should have similar embeddings

$$\boxed{S_p(p_i, p_j) = g(p_i)^\top . g(p_j)}$$

$$g(p) = W^\top . \psi(p)$$

➡ word embedding matrix W is shared by $S_{t/v}$ and S_p

Multitask training of word embeddings on paraphrases

- extend vocabulary coverage
- cover alternative lexicalisations
- auxiliary task: paraphrases should have similar embeddings

$$\boxed{S_p(p_i, p_j) = g(p_i)^\top . g(p_j)}$$

$$g(p) = W^\top . \psi(p)$$

➡ word embedding matrix W is shared by $S_{t/v}$ and S_p

Multitask training

- $\mathcal{P} = \{(p_i, p_j), i, j = 1; \cdots, |\mathcal{P}|\}$
  - PPDB dataset [Bannard and Callison-Burch, 2005]
  - WikiAnswers [Fader et al., 2013]

    (transformed question paraphrases)

    p_i *"much coca cola be buy per year"*

    p_j *"much do a consumer pay for coca cola"*
  - DBPP a custom dataset

    (bridge between entity names and common nouns)

    p_i *"Amsterdam"*

    p_j *"Place"*
- data corruption

  p_l *"information on neem plant"*
- Ranking loss function

$$\forall i, j, l, \forall [1 - S_p(p_i, p_j) + S_p(p_i, p_l)]$$

Approach overview

RDF property $\xrightarrow{lex}$ { ??? }

1. Learn embeddings of RDF triples and NL phrases
   Similarity function $S_{t/v}(t, v)$
2. Build sets of candidate NL phrases (Lex_p)
3. Rank candidate phrases using the scoring similarity function

$$\hat{v}(t) = \underset{v\prime \in Lex_p}{\arg\max} \quad S_{t/v}(t, v\prime)$$

4. Extract lexicalisations from top ranked candidates

Candidate lexicalisation sets

- L-LEX$_p$ *lexically-related* candidates
  Wikipedia sentences $\cap$
  WordNet (related synsets and derivationally related words)

p = CROSSES

WordNet Synset	(v) cross, traverse, span, sweep
L-Candidate	*"Old Blenheim Bridge spans Schoharie Creek"*

- E-LEX$_p$ *extensionally-related* candidates
  Wikipedia sentences $\cap$
  Semantic annotation of text (entity linking) [Walter et al., 2014a]

p = CREW1UP

RDF Triple	⟨ STS-130, CREW1UP, GEORGE_D._ZAMKA ⟩
E-Candidate	*Zamka served as the commander of mission STS-130*

Candidate lexicalisation sets

- L-LEX$_p$ *lexically-related* candidates
  Wikipedia sentences ∩
  WordNet (related synsets and derivationally related words)

p = CROSSES

WordNet Synset (v) crossbreed, cross, hybridize, hybridise, interbreed
*L-Candidate *"Shellbark hickory hybridizes with pecan"*

- E-LEX$_p$ *extensionally-related* candidates
  Wikipedia sentences ∩
  Semantic annotation of text (entity linking) [Walter et al., 2014a]

p = SPOUSE

RDF Triple ⟨ CHUCK_TRAYNOR, SPOUSE, LINDA_LOVELACE ⟩
*E-Candidate *Chuck Traynor was recently divorced from Linda Lovelace*

Approach overview

RDF property $\xrightarrow{lex}$ { ??? }

1. Learn embeddings of RDF triples and NL phrases
   Similarity function $S_{t/v}(t, v)$
2. Build sets of candidate NL phrases (Lex_p)
3. Rank candidate phrases using the scoring similarity function

$$\hat{v}(t) = \arg\max_{v\prime \in Lex_p} \quad S_{t/v}(t, v\prime)$$

4. Extract lexicalisations from top ranked candidates

Approach overview

RDF property $\xrightarrow{lex}$ { ??? }

1. Learn embeddings of RDF triples and NL phrases
   Similarity function $S_{t/v}(t, v)$
2. Build sets of candidate NL phrases (Lex_p)
3. Rank candidate phrases using the scoring similarity function
   $$\hat{v}(t) = \arg\max_{v\prime \in Lex_p} S_{t/v}(t, v\prime)$$
4. Extract lexicalisations from top ranked candidates

Experimental setup

Data:

- *Triples and Sentences* ($\mathcal{T}$) dataset ~300k pairs from DBPedia
  from 53384 DBPedia triples from 149 relations
- Paraphrases ($\mathcal{P}$ dataset ~3.5M pairs)
  PPDB M size lexical and phrasal sets + trans. WikiAnswers + custom DBPP

Implementation:

- emb. dimension 100
- KB embedding randomly initialised
- word embeddings initialised with pre-trained GloVe vectors
- training with SGD

Comparison

- 30 DBPedia properties
- gold lexicon developed manually for DBPedia properties [McCrae et al., 2011]

  `https://github.com/ag-sc/lemon.dbpedia`

- 3 automatic lexicons: Quelo, DBlexipedia_e, Patty
- various model variations:
  - (L/E)-LEX_p candidate sets: single, union and intersection
  - thresholds: top 10, third quartile, frequency re-ranked, and combinations thereof

Results

System/goldLemonDBPPatterns	Avg.NB	Recall	Precision	F1
L-LEX(k=10)	9.9	0.3611	**0.0875**	**0.1409**
L-LEX(FreqQ3Limit(7,25))	21.8	0.4583	0.0505	0.0909
L-LEX(All)	687.4	**0.8194**	0.0029	0.0057
E-LEX(k=10)	10	0.3333	0.0800	**0.1290**
E-LEX(FreqQ3Limit(7,25))	23.3	0.5000	0.0514	0.0933
E-LEX(All)	1557	**0.8056**	0.0012	0.0025
union(k=10)	10	0.3889	0.0933	0.1505
union(FreqQ3Limit(7,25))	10.8	0.4861	**0.1080**	**0.1768**
union(All)	2162.5	0.9444	0.0010	0.0021
L-LEXRandom(k=10)	9.9	0.2083	0.0505	0.0813
E-LEXRandom(k=10)	10	0.0833	0.0200	0.0323
Quelo	2.13	0.2917	0.3281	0.3088
DBlexipedia$_e$(k=10)	5.4	0.2500	0.1104	0.1532
Patty	936	0.5694	0.0015	0.0029

Micro-averaged Precision, Recall and F1 with respect to GOLD.

Results

System/goldLemonDBPPatterns	Avg.NB	Recall	Precision	F1
L-LEX(k=10)	9.9	0.3611	**0.0875**	**0.1409**
L-LEX(FreqQ3Limit(7,25))	21.8	0.4583	0.0505	0.0909
L-LEX(All)	687.4	**0.8194**	0.0029	0.0057
E-LEX(k=10)	10	0.3333	0.0800	**0.1290**
E-LEX(FreqQ3Limit(7,25))	23.3	0.5000	0.0514	0.0933
E-LEX(All)	1557	**0.8056**	0.0012	0.0025
union(k=10)	10	0.3889	0.0933	0.1505
union(FreqQ3Limit(7,25))	10.8	0.4861	**0.1080**	**0.1768**
union(All)	2162.5	0.9444	0.0010	0.0021
L-LEXRandom(k=10)	9.9	0.2083	0.0505	0.0813
E-LEXRandom(k=10)	10	0.0833	0.0200	0.0323
Quelo	2.13	0.2917	0.3281	0.3088
DBlexipedia$_e$(k=10)	5.4	0.2500	0.1104	0.1532
Patty	936	0.5694	0.0015	0.0029

(Quelo) RECORDEDIN $\xrightarrow{lex}$ { *"recorded in"* }

Results

System/goldLemonDBPPatterns	Avg.NB	Recall	Precision	F1
L-LEX(k=10)	9.9	0.3611	**0.0875**	**0.1409**
L-LEX(FreqQ3Limit(7,25))	21.8	0.4583	0.0505	0.0909
L-LEX(All)	687.4	**0.8194**	0.0029	0.0057
E-LEX(k=10)	10	0.3333	0.0800	**0.1290**
E-LEX(FreqQ3Limit(7,25))	23.3	0.5000	0.0514	0.0933
E-LEX(All)	1557	**0.8056**	0.0012	0.0025
union(k=10)	10	0.3889	0.0933	0.1505
union(FreqQ3Limit(7,25))	10.8	0.4861	**0.1080**	**0.1768**
union(All)	2162.5	0.9444	0.0010	0.0021
L-LEXRandom(k=10)	9.9	0.2083	0.0505	0.0813
E-LEXRandom(k=10)	10	0.0833	0.0200	0.0323
Quelo	2.13	0.2917	0.3281	0.3088
DBlexipedia$_e$(k=10)	5.4	0.2500	0.1104	0.1532
Patty	936	0.5694	0.0015	0.0029

Example output

PROGRAMMING LANGUAGE	*written in*, **uses**, include, **based on**, supports, is a part of, **programming language for** (4/1)
AFFILIATION	member of, associated with, *affiliated with*, **affiliated to**, **affiliate of**, **accredited by**, tied to, founded in, president of, associate member of (4/1)
COUNTRY	village in, **part of**, one of, *located in*, commune in, town in, born in, refer to, county in, country in, city in (2/1)
MOUNTAINRANGE	**mountain in**, **located in**, **include**, range from, **mountain of**, **mountain range in**, *part of*, **lies in**, reach, peak in, **find in**, highest mountain in (8/1)
DISTRIBUTOR	sell, appear in, allocate to, air on, **release**, **make**, star in, appear on (2/2)
LEADER	lead to, **leader of**, **led by**, **is a leader in**, visit, become, **lead**, lead producer of, *president of*, **elected leader of**, left (6/3)

system= Union.FreqQ3Limit7-25
italics= items in the gold
bold= items found by our system not in the gold
(N/G) N= nb. items found by our system G= nb. of items in the gold

Conclusion

- Learn embeddings of word representations and RDF triples to identify plausible lexicalisations
- When applied to DBPedia we obtain competitive results with existing approaches

Future work

- Conduct a larger scale evaluation
  larger number of properties, data-type properties
- Extend the gold lexicon (+ crowd-sourcing validation)
- Explore a more complex representation of natural language phrases (currently a bag-of-words)

Thank you !

Questions ?

We would like to thank Sebastian Walter for kindly providing us with the MATOLL corpus [Walter et al., 2014b]

References I

Bannard, C. and Callison-Burch, C. (2005).
Paraphrasing with bilingual parallel corpora.
In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 597–604. Association for Computational Linguistics.

Bordes, A., Weston, J., and Usunier, N. (2014).
Open question answering with weakly supervised embedding models.
CoRR, abs/1404.4326.

Fader, A., Zettlemoyer, L. S., and Etzioni, O. (2013).
Paraphrase-driven learning for open question answering.
In *ACL (1)*, pages 1608–1618. Citeseer.

McCrae, J., Spohr, D., and Cimiano, P. (2011).
Linking lexical resources and ontologies on the semantic web with lemon.
In *The semantic web: research and applications*, pages 245–259. Springer.

Nakashole, N., Weikum, G., and Suchanek, F. (2012).
Discovering and exploring relations on the web.
Proceedings of the VLDB Endowment, 5(12):1982–1985.

Trevisan, M. (2010).
A portable menuguided natural language interface to knowledge bases for querytool.
Master's thesis, Free University of Bozen-Bolzano (Italy) and University of Groningen (Netherlands).

Walter, S., Unger, C., and Cimiano, P. (2014a).
Atoll—a framework for the automatic induction of ontology lexica.
Data & Knowledge Engineering, 94:148–162.

References II

Walter, S., Unger, C., and Cimiano, P. (2014b).
M-atoll: a framework for the lexicalization of ontologies in multiple languages.
In *The Semantic Web–ISWC 2014*, pages 472–486. Springer.