Encoding Distributional Semantics into Triple-Based Knowledge Ranking for Document Enrichment

Muyu Zhang, Bing Qin, Mao Zheng, Graeme Hirst, and Ting Liu

Research Center for Social Computing and Information Retrieval Harbin Institute of Technology, Harbin, China

Department of Computer Science, University of Toronto, Toronto, ON, Canada

{myzhang,qinb,mzheng,tliu}@ir.hit.edu.cn gh@cs.toronto.edu

1. Introduction

Document enrichment focuses on retrieving relevant knowledge from external resources, which is essential because text is generally replete with gaps. We use triples of *Subject, Predicate, Object* as knowledge and incorporate distributional semantics to rank them.

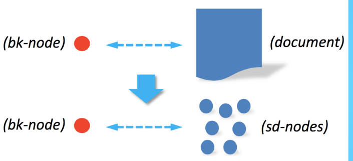
(1) Our model first extracts these triples automatically from raw text.

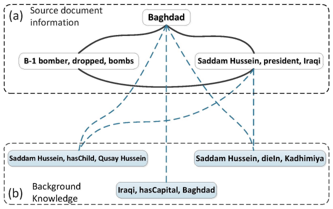
(2) After that, it converts these triples into real-valued vectors with LDA.

(3) These triples are then represented, together with the source document, as a graph of triples, and a global iterative algorithm is adopted to select the most relevant ones.

2. Triple-graph based Model

We use triple graph as a document representation, where the triples of "Subject, Predicate, Object" serve as nodes, and the edges indicate their relatedness. So the relevance between document and knowledge is converted into that between document nodes and the knowledge.




(a) Source document information
Baghdad
B-1 bomber, dropped, bombs
Saddam Hussein, president, Iraqi
Saddam Hussein, hasChild, Qusay Hussein
Saddam Hussein, dieIn, Kadhimiya
(b) Background Knowledge
Iraqi, hasCapital, Baghdad

There are two kinds of nodes extracted automatically:
(1) source document nodes;
(2) knowledge nodes.
Then we propagate the relevance weight from source nodes to the knowledge

3. Encoding Distributional Semantics

We employ the publicly available implementation of LDA, JGibbLDA2 (Phan et al., 2008). For every word w_n, we get k distributional probabilities over k topics. Then we combine k possibilities together as a real-valued vector to represent w_n

$$\vec{v}_{w_n} = (p_{w_n z_1}, p_{w_n z_2}, \ldots, p_{w_n z_k})$$

$$p_{tz_i} = \sum_{w_n \in t} p_{w_n z_i}$$

$$\vec{v}_t = \frac{(p_{tz_1}, p_{tz_2}, \ldots, p_{tz_k})}{\sum_{i=1}^{k} p_{tz_i}}$$

We compute the semantic relatedness between triples as their cosine-similarity and use it as the probability of them propagating to each other.

4. Knowledge for Document Classification

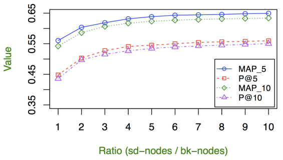
Intuitively, the recovery of this information will be helpful to real applications. In this work, we select the top *N* most-relevant knowledge and use them as extra features, together with the features extracted from the source document.

5. Experiment

(1) Evaluation as a ranking problem

Setup: We use 600 documents as source documents to be enriched. Then we use 16,599 documents as the source of background knowledge which are extracted automatically. After ranking, we annotate the result manually.

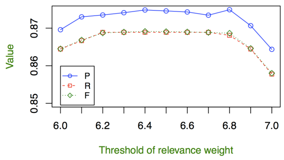
Model	MAP_5	P@5	MAP_10	P@10
VSM	0.4968	0.3435	0.4752	0.3841
WE	0.4356	0.3354	0.4624	0.3841
LDA	0.6134	0.4775	0.6071	0.5295
Ours-S	0.5325	0.3762	0.5012	0.4054
Ours	**0.6494**	**0.5597**	**0.6338**	**0.5502**



(2) Task-based evaluation

Setup: 17,199 documents over 9 topics. Our model gets the top-N relevant knowledge for every document as enrichment which serves as extra features for document classification.

Model	P	R	F
VSM+one-hot	0.8214	0.8146	0.8168
VSM+tf-idf	0.8381	0.8333	0.8336
LDA+SVM	0.8512	0.8422	0.8436
LDA+SVM+Ours-S	0.8584	0.8489	0.8501
LDA+SVM+Ours	**0.8748**	**0.8689**	**0.8691**



6. Future Work

(1) Explore a better way to encode distributional semantics.

(2) Explore the effect of introducing background knowledge in other NLP tasks, especially discourse parsing.