An Ontology-based Latent Semantic Indexing Approach Using Long Short-Term Memory Networks

Ningning Ma, Hai-Tao Zheng (✉), and Xi Xiao

Tsinghua-Southampton Web Science Laboratory,
Graduate School at Shenzhen, Tsinghua University, China
mnn15@mails.tsinghua.edu.cn
zheng.haitao@sz.tsinghua.edu.cn
xiaox@sz.tsinghua.edu.cn

Abstract. Nowadays, online data shows an astonishing increase and the issue of semantic indexing remains an open question. Ontologies and knowledge bases have been widely used to optimize performance. However, researchers are placing increased emphasis on internal relations of ontologies but neglect latent semantic relations between ontologies and documents. They generally annotate instances mentioned in documents, which are related to concepts in ontologies. In this paper, we propose an Ontology-based Latent Semantic Indexing approach utilizing Long Short-Term Memory networks (LSTM-OLSI). We utilize an importance-aware topic model to extract document-level semantic features and leverage ontologies to extract word-level contextual features. Then we encode the above two levels of features and match their embedding vectors utilizing LSTM networks. Finally, the experimental results reveal that LSTM-OLSI outperforms existing techniques and demonstrates deep comprehension of instances and articles.

1 Introduction

With the rapid development of Internet, the information is growing explosively on the web and document indexing technology is becoming increasingly more crucial. The core task of existing search engine is to understand the real intention of user and semantic meanings of web content, aiming to obtain more semantically expressive resources. Generally, the semantic indexing methods in the literature can be distinguished according to the following two categories: (1) ontology-based approaches utilizing ontology and knowledge base as background knowledge and (2) statistical approaches, to identify groups of words that commonly appear together and therefore may jointly describe a particular reality [16].

The first category which utilizes ontologies and knowledge bases [3, 11, 14] to understand semantic context has recently been an area of considerable interest in semantic indexing. An *ontology* is a formal knowledge description of concepts and their relationships, an ontology together with a set of individual *instances*

of classes constitutes a *knowledge base*. There are two main challenges: utilizing relations among concepts in ontologies and mapping information in documents into knowledge bases. However, researchers are placing increased emphasis on internal relations of ontologies but neglect latent semantic relations between ontologies and documents. They generally annotate all instances mentioned in documents, which are related to concepts in ontologies.

Meanwhile, as for the second category, probabilistic topic models view each document as a mixture of various topics and each topic as a mixture of words [4, 13], which possess fully generative semantics of documents. The sense of a word is a hidden random variable that is inferred from data. However, they do not carry the explicit notion of sense that is necessary for word sense disambiguation. Fortunately, background ontologies and knowledge bases are generally exploited to determine the meaning and the contextual information of an ambiguous word.

Thus, we propose an Ontology-based Latent Semantic Indexing model utilizing topic models and Long Short-Term Memory networks (LSTM-OLSI). An instance's contextual information is extracted utilizing semantic relations in ontology knowledge base; a document's *general topic* (a sequence of words) is extracted by an importance-aware topic model. The similarity between an instance and a document is measured by the distance between their corresponding sequences (an instance's contextual information and a general topic information) embedding vectors computed by the LSTM networks. The results indexed by the instances with profound meanings intuitively depict that LSTM-OLSI outperforms existing techniques and demonstrates deep comprehension of instances and articles.

Our main contribution is outlined as follows:

- We take into account both word-level contextual information to resolve word sense ambiguity and document-level semantics to clarify the semantics of the documents.
- We propose an importance-aware topic model to generate a general topic, which is a comprehensive topic composed of all the subtopics.
- Further, we present a strategy for explicitly encoding semantic relationships between the documents and the knowledge base using LSTM networks.

In Section 2, we discuss related work in ontology-based and machine learning based indexing approaches. We present the LSTM-OLSI model in Section 3. Further, we discuss the experimental methods and compare the LSTM-OLSI with some state-of-the-art methods in Section 4. Finally, Section 5 concludes our work.

2 Related Work

Several research approaches for indexing documents have been proposed. We classify them as ontology-based indexing approaches and machine learning based indexing approaches, discuss these two types of related work.

Ontology-based indexing approaches For a variety of text analysis problems, the knowledge representation is useful, such as document similarity computation, search result re-ranking and semantic indexing [15, 17]. There are a variety of semantic search approaches [3, 14, 19] utilizing ontology knowledge base which span the four main processes of an IR system: indexing, querying, searching and ranking [9]. The ideal indexing is to choose a set of features to represent documents. Posch [19] enriched domain-specific ontologies with encyclopedic background knowledge, leveraged textual and structural information to implement automatic classification and subject indexing of documents. Winfried [11] proposed an index structure which is based on the concept of ontology. Lee, Min, Oh, and Chung [14] presented effective semantic indexing and search techniques considering the semantic relationships in ontologies and proposed a weighting measure for the semantic relationships. Concretely, they considered the number of meaningful semantic relationships, the coverage of the keywords, and the discriminating power of the keywords. Hahm [10] proposed an indexing method dealing with semantic path in ontologies, which computed semantic scores among different instances in ontologies for each term. Those semantic instances with shorter paths had larger score.

Machine learning based indexing approaches Several studies have developed indexing techniques leveraging supervised and unsupervised machine learning methods. A standardized supervised learning approach is the *bag-of-words* approach, which represents documents by the words they contain and neglectes the order of the words. Moreover, sequences of words (*n-grams*) are utilized to represent a document and the words are weighted by different schemes such as TF-IDF. However, these approaches cannot depict the semantic meaning of the documents. To address this shortcoming, some novel methods are proposed. Latent Semantic Indexing (LSI) [8] maps documents to low-dimensional concept vectors utilizing a document-word matrix called singular value decomposition (SVD). The relevance of a document to one or several keywords is assumed to be proportional to the cosine similarity between their concept vectors. A significant step forward in this regard was Probabilistic Latent Semantic Indexing (PLSI) [13] model, which views documents as mixtures of topics and ranks the documents by the probability of the query given the document distribution over topics. Utilizing topic models is one of the state-of-art unsupervised learning approaches to index documents. Latent Dirichlet Allocation (LDA) [4] extends PLSI and assumes that topic distributions have a Dirichlet prior. Newman, Koilada, Lau, and Baldwin [18] exploited an unsupervised Bayesian model and applied the method Dirichlet process segmentation for extracting key phrases from a document. Ma and Zhang [16] proposed a semantic search method based on LDA and view each theme generated from topic models as the basic message block that waits constantly to be searched. Chebil, Soualmia, Omr, and Darmoni [6] exploited a possibilistic network that carries out partial matching between documents and a biomedical vocabulary to index the biomedical documents.

3 Ontology-based Latent Semantic Indexing Model

We design a novel semantic indexing framework by taking into account both the word-level contextual information and the document-level semantics, to resolve word sense ambiguity and to clarify the semantics of the documents, respectively.

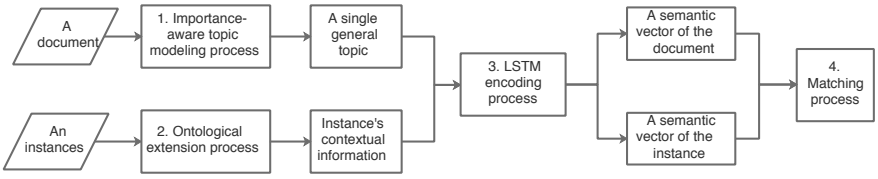

Fig. 1. The architecture of LSTM-OLSI.

The LSTM-OLSI is performed as follows: Firstly, we extract a single *general topic* (a sequence of words) for each document utilizing an importance-aware topic model inspired by Latent Dirichlet allocation (LDA) [4] (Step 1). After that, we extract an instance's contextual information utilizing semantic relations in ontology knowledge base (Step 2). Finally, we encode a semantic vector of a general topic and a semantic vector of an instance using LSTM networks, respectively (Step 3). And we measure the similarity between an instance and a document utilizing the cosine similarity between the semantic vectors of two sequences (Step 4).

In general, the overall LSTM-OLSI model consists of the following three essential components (see Figure 1), which are described in more detailed in Section 3.1, Section 3.2 and Section 3.3:

(1) The probabilistic topic modeling process (Step 1);
(2) The ontological extension process (Step 2); and
(3) The LSTM sequence encoding and matching process (Step 3 and Step 4).

3.1 Probabilistic topic modeling process

The importance-aware topic model in LSTM-OLSI is inspired by Latent Dirichlet allocation (LDA) [4]. We start with a brief introduction of LDA. LDA models each of D documents as a mixture over K latent topics, each of which describes a multinomial distribution over a W word vocabulary. The original space of vocabulary is mapped to several topics, which can better depict the semantic meaning of the documents. After conducting the Gibbs sampling process to train the model, given the parameters α and β, the joint distribution of a topic mixture θ, a set of N topics z, and a set of N words w, it is easy to obtain the following useful results related to the words, documents and topics in the documents set:

- Latent topics with the most likely words in each topic and a topic-to-word probability distribution (i.e., $p(w \mid z, \beta)$);
- A document-to-topic probability distribution (i.e., $p(z \mid \theta)$).

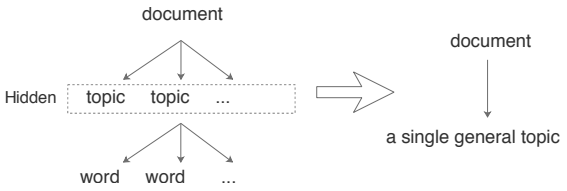

Fig. 2. The generation process of the *general topic*.

The LDA model is somewhat more elaborate than the three-level models often studied in the classical hierarchical Bayesian literature. However, to index the documents with high accuracy, we target to take a sequence of the most semantically expressive words to represent each document. Certainly, it is inappropriate to take one or more topics with higher probabilities to represent a document, for the reason that the words with higher probabilities in the topics with lower probabilities may be ignored. Therefore, we propose an approach to directly compute the semantic relations between the documents and the words. We generate a single *general topic* for each document which differs from conventional topic models (Figure 2). By marginalizing over the hidden topic variable z, however, we can understand LDA as a two-level model and capture direct semantic relations between the documents and the words. To capture relations between the documents and the words, we derive the document-specific word probability distribution $p(w \mid \theta, \beta)$ as:

$$p(w \mid \theta, \beta) = \sum_{z} p(w \mid z, \beta) p(z \mid \theta) \tag{1}$$

However, the semantically expressive capacities are different between the two levels of probabilities, namely, the document-specific topic probabilities and the topic-specific word probabilities. In other words, without considering the different importance between the above two levels of probabilities, some words with low first-level probabilities but high second-level probabilities may be computed with a high correlation score. For example, we assume a document has a *health* related topic with a probability of 0.6 and has a *technology* related topic with a probability of 0.2. So the theme of the document is more likely about *health*. We also assume that the *health* related topic has a word *disease* with a probability of 0.2, but the *technology* related topic has a word *tech* with a probability of 0.8.

So the relevant score of *tech* is calculated as $0.2 \times 0.8 = 0.16$, and the relevant score of *disease* is computed as $0.6 \times 0.2 = 0.12$, which is the smaller than *tech*. As we described above, the word *disease* is more relevant to this document than *tech*, since the theme of this document is more likely about *health*.

Therefore, we propose a self-adaptive approach utilizing an importance-aware manner applied to various realistic scenarios. Specifically, we define a correlation score $Correl(w_n, d)$ of a word w_n for a document d is computed as:

$$Correl(w_n, d) = \sum_z (f(p(w_n \mid z, \beta))g(p(z \mid \theta_d))) \tag{2}$$

where $f(p(w_n \mid z, \beta))$ represents the importance score of the probability of word w_n occurring in topic z, $g(p(z \mid \theta_d)$ represents the importance score of the probability of topic z occurring in document d for any given word.

The $f(p)$ score function and the $g(p)$ score function are given as follows:

$$f(p) = p^r \tag{3}$$

$$g(p) = p^q \tag{4}$$

where r is the penalty factor acting on the topic specific word probability and q is the penalty factor acting on the document specific topic probability.

The penalty factor q greater than 1 can increase the gap between those topics with larger probabilities and those with smaller probabilities. If q is less than 1, it can reduce the gap. The same applies to the penalty factor r which acts on the topic-to-word probability. As we have described above, in this indexing system, the document-to-topic level is more semantically expressive than the topic-to-word level, so we fix $r = 1$ and tune q. Finally, we select the first several W words in the whole vocabulary with the highest correlation scores for each document, thus deriving the *general topic*, which is a comprehensive topic composed of all the sub-topics.

It is a key issue in the topic model to determine the value of parameter K, the number of latent topics. Inappropriate setting of K customarily imposes an adverse impact on the modeling results. In particular, we computed the perplexity of a held-out test set to evaluate the models. The perplexity, used by convention in language modeling, is monotonically decreasing in the likelihood of the test data, and is algebraically equivalent to the inverse of the geometric mean per-word likelihood [4]. More formally, for a test set of M documents, the perplexity is:

$$perplexity(D_{test}) = exp(-\frac{\sum_{d=1}^{M} log(p(\mathbf{w}_d))}{\sum_{d=1}^{M} N_d}) \tag{5}$$

where $\mathbf{w}_d$ denotes text document d and N_d represents the number of words in document d. Moreover, a lower perplexity score indicates a better generalization performance [4].

3.2 Ontological extension process

The DBpedia ontology is leveraged to depict the word-level semantics in the process of LSTM-OLSI, collaborating with the probabilistic topic model which depicts the document-level semantics. The main tasks performed by this ontological extension process (Step 2 in Figure 1) are discussed in detail below (Figure 3).

For each instance, it has a verbal description in the ontology and we extend its semantic representations utilizing the ontology and knowledge base to depict its semantic meaning. The semantic representation of an instance is illustrated in its verbal description and its contextual instances' verbal descriptions. Intuitively, instances are stored in RDF statements which are triples of the form $(subject, property, object)$. The form of *subject* or *object* typically indicates an instance in the knowledge base. A semantic path is composed of one or more *properties*. As the length of a semantic path gets longer, the relevance between the source and the destination decreases [14]. Therefore, for the single instance, we regard the set of its adjacent instances directly connected with a 1-length path in the knowledge base as its contextual instances.

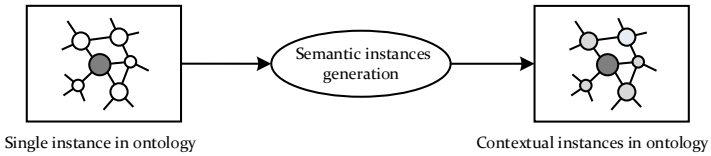

Fig. 3. The ontological extension process.

The verbal descriptions of the instance and its contextual instances are regarded as the instance's ontological contextual information, which essentially are a sequence of words and depict the word-level semantics.

3.3 LSTM sequence encoding and matching process

In Section 3.1, a document is encoded into a *general topic* (a sequence of words), and in Section 3.2, an instance is encoded with its ontological contextual information (also a sequence of words). We treat the above two levels of information as two sequences of words with internal structures, i.e., word dependencies. And the two sequences are about the same length.

The recurrent neural networks (RNN), a type of deep neural networks, have been widely used in time sequence modeling. However, it is generally difficult to learn the long term dependency within a sequence due to vanishing gradients problem. One of the effective solutions is using memory cells named Long Short Term Memory (LSTM). Therefore, we use LSTM recurrent networks to sequentially take each word in a sentence, extract its information, and embed it into

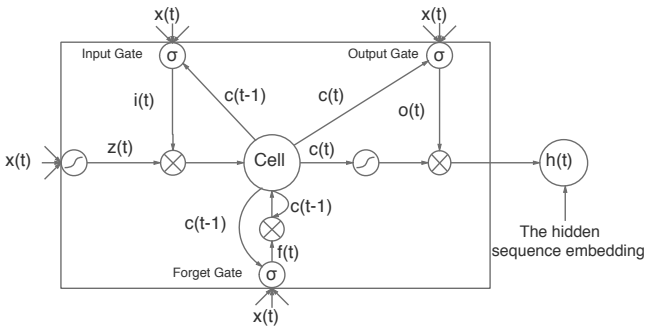

Fig. 4. The LSTM architecture used for sequence embedding.

a semantic vector. Due to its ability to capture long term memory, the LSTM accumulates increasingly richer information as it goes through the sequence. The encoding process is performed word-by-word sequentially. At each time step, a new word in the sequence is encoded into the semantic vector, and the word dependencies embedded in the vector are updated. So when it reaches the last word, the semantic vector has embedded all the words and their dependencies, the hidden layer of the network provides a semantic representation of the whole sequence [12].

We use the architecture of LSTM illustrated in Figure 4 for the proposed sequence embedding method. LSTM has three gates: input gate ($i(t)$), forget gate ($f(t)$) and output gate ($o(t)$) which slows down the disappearance of past information and makes Backpropagation through Time (BPTT) easier. In this figure, $\sigma()$ is the sigmoid function, $c(t)$ is the cell state vector and $h(t)$ is the hidden activation vector, which can be used as a semantic representation of the t-th word. We utilize this architecture to find an embedded vector for each word, then use the $h(last)$ corresponding to the last word in the sentence as the semantic vector for the entire sequence.

However, the core task is the learning of the embedding vector for a sentence, that is, to train a model that can automatically transform a sequence of words to a vector that encodes the semantic meaning of the sequence. Our approach of sequence encoding is inspired by the work in sequence to sequence learning (seq2seq) [21]. They utilize a recurrent network as an encoder to read in an input sequence into a hidden state, which is the input to a decoder recurrent network that predicts the output sequence. To derive the sequence embedding, we encode a sequence and reconstruct the original sequence to train the seq2seq model. Thus we can automatically transform a sequence of words to a vector that encodes the semantic meaning of the sequence. The weights for the decoder network and the encoder network are the same.

The method developed in this paper trains the model so that sequences that are paraphrase of each other are close in their semantic embedding vectors. We

adopt the cosine similarity $C(document, instance)$ between the semantic vectors of two sentences as a measure for their similarity:

$$C(D, I) = \frac{h_D(L_D)^T h_I(L_I)}{\|h_D(L_D)\| \cdot \|h_I(L_I)\|} \tag{6}$$

where L_D and L_I are the lengths of the document's semantic sequence D and the instance's semantic sequence I, respectively, $h_D()$ and $h_I()$ are the hidden activation vectors of D and I, respectively.

4 Experiments

To demonstrate the reliability and stability of our approach, we conduct further experiments. The ontology-based processing part is fully-implemented in Java and the topic modeling part is in Scala. The LSTM sequence encoding and matching process is implemented utilizing TensorFlow [2].

We first describe the datasets, the preprocessing details, the performance measure method and the baseline methods in Section 4.1. Then we report the experimental process and the comparison with the other four baseline models in Section 4.2. Finally, we illustrate the results indexed by many instances with profound meanings, which intuitively depict that LSTM-OLSI has deeper semantic comprehension of both instances and news articles.

4.1 Experimental setup

Dataset. We exploit a real world news corpus collection, a movie sentiment dataset and a rich ontology knowledge base to conduct the experiments.

- **MSNNews**: The news articles are extracted from a large news corpus, which are news articles searched from MSN news web pages. We organize volunteers to classify these news articles manually into categories according to its article content, and we select five categories: crime, health, politics, soccer and technology. We select one million news articles and current affairs happened recently, and the average word count of news articles is about 250.
- **IMDB**: The IMDB movie sentiment dataset contains 25,000 labeled and 50,000 unlabeled documents in the training set and 25,000 in the test set [1]. The average length of each document is 241 words and the maximum length of a document is 2,526 words. We utilize this dataset to train the seq2seq model and to derive the LSTM networks' weights.
- **DBpedia**: We exploit the version of DBpedia 2016-04 as instance data base, which involves 9.5 billion pieces of information (RDF triples). The DBpedia ontology currently covers 685 concepts which form a subsumption hierarchy, described by 2,795 different properties and contains about 4,233,000 instances. The knowledge base is big enough to contain most domains of knowledge in our daily life so the knowledge in daily news can be represented.

Data preprocessing. Before our algorithm is capable to index news documents, we first perform certain preprocessing procedures. First, we extract context information from DBpedia knowledge base to establish our experimental basis for indexing. In the context of this work, we extract 6.0M instances from DBpedia that we eventually utilize in our work. Next, after extracting news corpus involving the five categories, we lowercase all characters, perform word segmentation and remove stop words. In each category, we randomly held out 10% of the data for test purposes and trained the models on the remaining 90%, to conduct 10-fold cross-validation.

Performance measure. We conduct an evaluable method to estimate accuracy of the indexing results. More concretely, if a document in the corresponding category *crime* is searched by the query keyword *crime*, we consider it as a relevant news document. We compute precision as:

$$precision = \frac{|relevant\_news| \bigcap |retrieved\_news|}{|retrieved\_news|} \tag{7}$$

and recall as:

$$recall = \frac{|relevant\_news| \bigcap |retrieved\_news|}{|relevant\_news|} \tag{8}$$

Baseline methods. We exploit a comparative evaluation with four indexing approaches to estimate our approach: a topic modeling-based approach, an ontology-based approach, a LSTM sentence matching approach and a TF-IDF approach.

LDA based approach. The first baseline model in comparison is a probabilistic topic modeling approach from Ma and Zhang [16] which generates the latent topics with the most likely words in each topic. According to a user query, the first several thematic keywords with the highest probabilities of each possible semantically related topic are recommended to the user. With the user-selected topic or topics of interest, a ranked list of documents is returned.

Ontology-based approach. The second baseline approach utilizes ontology-based method. In this method [11], indexing of documents is a statement about the aboutness of documents, that is, an indexing term is only assigned if the corresponding concepts are covered issues within the context of the document.

LSTM text matching approach. Another baseline method utilizes LSTM networks to encode and match the documents and the instances' ontological information, but it does not encode the documents into the *general topics*.

TF-IDF. The last baseline model in comparison is the standardized indexing model Term Frequency Inverse Document Frequency (TF-IDF) [20], a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus.

4.2 Experimental Results

The experiments conducted in the implementation and evaluation stages are classified as two categories of experiments. The first category aims to tune the parameters and coefficient in order to optimize the performance of the proposed LSTM-OLSI. The set of experiments belonging to the first category are: initializing α, β and the number of iterations in topic model; tuning the number of topics (K) by computing perplexity; and tuning the length of the *general topic* (W) and the penalty factor q utilizing a grid search method. The second category aims to highlight the effectiveness of LSTM-OLSI. The set of experiments belonging to the second category are: comparing the performance of LSTM-OLSI to some existing approaches, evaluating the deep semantic comprehension capacity of LSTM-OLSI. The next subsections discuss the above two categories in details.

Implementation and optimization

First, we tune the parameters and coefficient to optimize the performance of LSTM-OLSI. The probabilistic topic model is trained with 5000 iterations of Gibbs sampling using $\alpha = 50/\mid K \mid$ and $\beta = 0.01$ (the default values used, e.g., in [5, 23]).

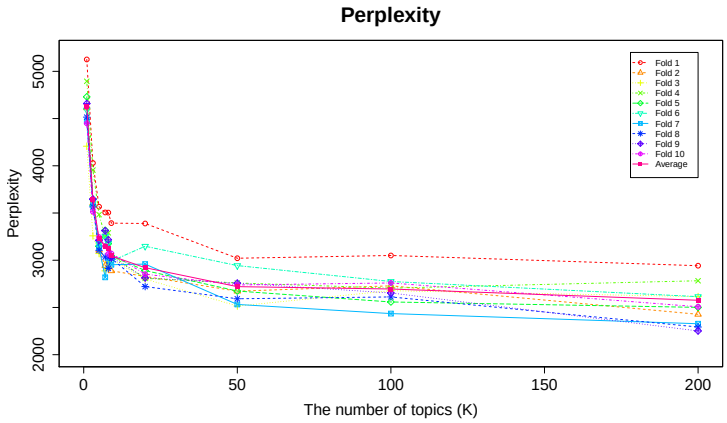

Fig. 5. Perplexity resulting curve using 10-fold cross-validation.

Afterwards, in order to determine the optimal number of topics for the news corpus (parameter K), we present a 10-fold cross-validation process to compute the curves of perplexity (Figure 5). The ten *Folds* represent the result when the corresponding 10% held out data was treated as the test data and *Average* depicts the average value of the 10 folds. It is depicted that in most cases, the values of perplexity reach relatively low scores when K is greater than 50. In

addition, the *Average* curve gets its lowest point in the case that the number of topics is 50. Therefore, the optimal number of topics of the news corpus should be 50.

Finally, we compute the length of the single *general topic* (W) and the penalty factor (q). We utilize the grid search method to tune W in the range from 100 to 300 and $q = \{1, 2, 3\}$. Figure 6 (left) depicts the average precision of the five categories during tuning parameter W and q. The curves reach their maximum values in the case that $W = 120$. Figure 6 (right) depicts when W is greater than 120, the recall curves slightly increase when W is growing. Basically, in our experiments, the setting of $W = 120$ and $q = 3$ consistently provides an optimal performance than other configurations of W. To keep the balance of the precision and recall effectively, we fix $W = 120$ for these data sets. As for the distinct data set, the performance of LSTM-OLSI is also optimal when $q = 3$, that is, LSTM-OLSI is fairly robust to the change of data sets. So we fix $W = 120$ and $q = 3$ for the news corpus.

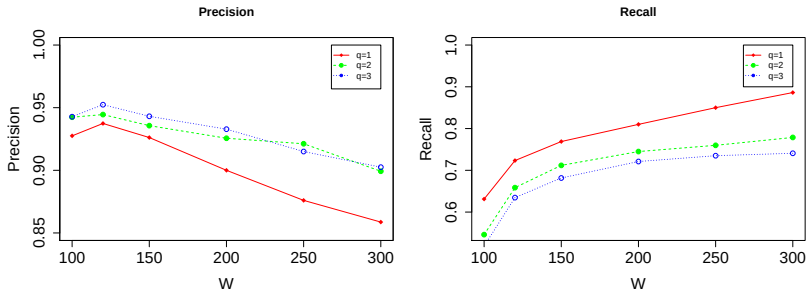

Fig. 6. Grid search of W and q.

In this way, we derive a *general topic* (a sequence of words) of each document and each instance's ontological contextual information (also a sequence of words). We utilize both the word2vec embeddings and the seq2seq encoder weights to initialize the LSTM model for the sequence embedding task. After training the sequence encoding model for roughly 500K steps with a batch size of 128 [7], we obtain the embedding vectors of the news articles and the instances. Finally, the cosine similarity is utilized to match and index the news articles and the instances.

Evaluation

To highlight the effectiveness of our indexing approach, we compare the performance of LSTM-OLSI with other approaches, namely, the LDA based method, ontology-based indexing model (OIM in Figure 7), the LSTM text matching method and the TF-IDF method (see Figure 7). For each approach, we computed the precision on the five categories of news corpus respectively and the

average precision of them. We consider TF-IDF as the baseline against the other comparing approaches to evaluate the stability of the news corpus since TF-IDF is the standardized indexing model. Figure 7 indicates LSTM-OLSI (95.4%) has statistically significant improvements over LSTM (92.8%), LDA (91.3%), OIM (90.2%) and TF-IDF (82.3%), respectively. The proposed LSTM-OLSI outperforms all the other methods with a significant margin.

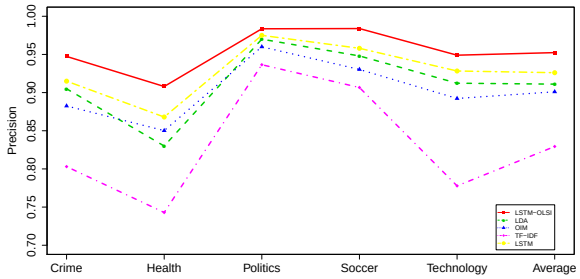

Fig. 7. Comparison with baseline models.

To intuitively evaluate the performance of LSTM-OLSI, we show many intuitive examples. Take a semantic instance *Heart and Crime* as an example, the word *heart* is semantically relevant to news articles about heart and love, and the word *crime* is relevant to the news articles about violence and crime. It is interesting to see that the theme of the indexed news articles is an ingenious combination of *heart* related topics and *crime* related topics, which is maternal and parent-child related crime (see Table 1).

Table 1. Intuitive LSTM-OLSI results of profound instances.

Instance	Sample indexed news titles
Heart and Crime	Mom who killed kids in reincarnation case gets life Knife with blood found in home where children killed Dad of Ohio girl found dead in crib gets 3 years in prison
Knife Play	Pokémon Go: armed robbers use game to lure players into trap Vigil held for teen shot dead after basketball game
Politics of Love	Seen and not heard: homeless people absent from election even as ranks grow For Clinton, sisterhood is powerful and Trump helps Barkley: 'More power to Draymond for slapping the hell out of that kid'
Music Technology	IBM is making a music app that can create entirely new songs just for you CloudPlayer now lets you take your playlists everywhere Apple links with NASA to make music from space
Sports Nutrition	Top 10 Rules You Must Follow Every Day to Lose 10 Pounds Make These 3 Changes, Burn More Calories
Happy Nation	Defender Riise announces retirement at 35 The incredible career of Zlatan Ibrahimović

Moreover, we exploit more intuitive examples to estimate the performance of LSTM-OLSI. Table 1 reports some sample news results indexed by several instances with profound meanings. LSTM-OLSI demonstrates deep comprehension besides literal comprehension of the instances. For example, the instance *Knife Play* indexes robbery or murder (interpretation of $Knife$) news relevant to the Pokémon Go mobile game or basketball games (interpretation of $Play$). Since music has the meanings of art, the instance $Music\ Technology$ can index technology news about music apps. And the instance *Politics of Love* indexes politic news about homeless people and current events about Clinton's sisterhood (interpretation of *Love*). More detailed information is located in Table 1. The results indicate that utilizing LSTM-OLSI model, we can retrieve more semantically expressive news articles with complex queries containing multiple topics.

5 Conclusions and Future Work

The work presented in this paper develops a novel ontology-based latent semantic indexing approach to extract both word-level contextual features (via ontology and knowledge base) and document-level contextual features (via the importance-aware topic model) from text. Then the LSTM networks make effective use of the extracted two levels of features to encode and match the documents and the instances. We summarized a method of selecting adjustable parameters to achieve higher accuracy according to the complexity and the diversity of the news documents. Moreover, the indexing results indicate that LSTM-OLSI outperforms the state-of-the-art and has deep semantic comprehension of both instances and news articles. Our future work will further extend the methods to include 1) Using the proposed semantic indexing method for other important information retrieval tasks for which semantic indexing has a key role, 2) Developing more general version of the proposed model exploiting more knowledge bases.

Acknowledgments This research is supported by National Natural Science Foundation of China (Grant No. 61375054), Natural Science Foundation of Guangdong Province Grant No. 2014A030313745Basic Scientific Research Program of Shenzhen City Grant No. JCYJ20160331184440545), and Cross fund of Graduate School at Shenzhen, Tsinghua University (Grant No. JC20140001).

References

1. A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. Learning word vectors for sentiment analysis. In ACL, 2011
2. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Ghemawat, S. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467.

3. Alec, C., Reynaud-Delatre, C., & Safar, B. An ontology-driven approach for semantic annotation of documents with specific concepts. In ISWC 2016.
4. Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. Journal of machine Learning research, 3(Jan), 993-1022.
5. Borisov, A., Serdyukov, P., & de Rijke, M. (2016, April). Using Metafeatures to Increase the Effectiveness of Latent Semantic Models in Web Search. In WWW.
6. Chebil, W., Soualmia, L. F., Omri, M. N., & Darmoni, S. J. (2015). Indexing biomedical documents with a possibilistic network. Journal of the Association for Information Science and Technology.
7. Dai, A. M., & Le, Q. V. (2015). Semi-supervised sequence learning. In NIPS (pp. 3079-3087).
8. Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. Journal of the American society for information science, 41(6), 391.
9. Fernández, M., Cantador, I., López, V., Vallet, D., Castells, P., & Motta, E. (2011). Semantically enhanced information retrieval: An ontology-based approach. Web semantics: Science, services and agents on the world wide web, 9(4), 434-452.
10. Hahm, G. J., Yi, M. Y., Lee, J. H., & Suh, H. W. (2014). A personalized query expansion approach for engineering document retrieval. Advanced Engineering Informatics, 28(4), 344-359.
11. Gödert, W. (2016). An ontologybased model for indexing and retrieval. Journal of the Association for Information Science and Technology, 67(3), 594-609.
12. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.
13. Hofmann, T. (1999, August). Probabilistic latent semantic indexing. In Proceedings of SIGIR (pp. 50-57). ACM.
14. Lee, J., Min, J. K., Oh, A., & Chung, C. W. (2014). Effective ranking and search techniques for web resources considering semantic relationships. IPM, 50(1), 132-155.
15. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., ... & Bizer, C. (2015). DBpediaa large-scale, multilingual knowledge base extracted from Wikipedia. Semantic Web, 6(2), 167-195.
16. Ma, B., Zhang, N., Liu, G., Li, L., & Yuan, H. (2016). Semantic search for public opinions on urban affairs: A probabilistic topic modeling-based approach. IPM, 52(3), 430-445.
17. Mukherjee, S., Ajmera, J., & Joshi, S. (2014, April). Unsupervised approach for shallow domain ontology construction from corpus. In Proceedings of WWW (pp. 349-350). ACM.
18. Newman, D., Koilada, N., Lau, J. H., & Baldwin, T. (2012, December). Bayesian Text Segmentation for Index Term Identification and Keyphrase Extraction. In COLING (pp. 2077-2092).
19. Posch, L. (2014, October). Enriching ontologies with encyclopedic background knowledge for document indexing. In ISWC (pp. 537-544).
20. Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. Journal of documentation, 28(1), 11-21.
21. Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In NIPS (pp. 3104-3112).
22. Wang, Q., Xu, J., Li, H., & Craswell, N. (2011, July). Regularized latent semantic indexing. In Proceedings of SIGIR (pp. 685-694). ACM.
23. Wei, X., & Croft, W. B. (2006, August). LDA-based document models for ad-hoc retrieval. In Proceedings of SIGIR (pp. 178-185). ACM.