Learning Semantic Representations Using Convolutional Neural Networks for Web Search

Yelong Shen
Kent State University
Kent, OH, USA
yshen@cs.kent.edu

Xiaodong He
Microsoft Research
Redmond, WA, USA
xiaohe@microsoft.com

Jianfeng Gao
Microsoft Research
Redmond, WA, USA
jfgao@microsoft.com

Li Deng
Microsoft Research
Redmond, WA, USA
deng@microsoft.com

Grégoire Mesnil
University of Montréal
Montréal, Canada
gregoire.mesnil@umontreal.ca

ABSTRACT

This paper presents a series of new latent semantic models based on a *convolutional* neural network (CNN) to learn low-dimensional semantic vectors for search queries and Web documents. ① By using the convolution-max pooling operation, local contextual information at the word n-gram level is modeled first. ② Then, salient 显著的 local features in a word sequence are combined to form a global feature vector. Finally, ③ the high-level semantic information of the word sequence is extracted to form a global vector representation. The proposed models are trained on click-through data by maximizing the conditional likelihood of clicked documents given a query, using stochastic gradient ascent. The new models are evaluated on a Web document ranking task using a large-scale, real-world data set. Results show that our model significantly outperforms other semantic models, which were state-of-the-art in retrieval performance prior to this work.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; I.2.6 [**Artificial Intelligence**]: *Learning*

Keywords

Semantic Representation, Convolutional Neural Network

1. INTRODUCTION

Latent semantic models, such as latent semantic analysis (LSA) and its extensions, are able to map a query to its relevant documents at the semantic level (e.g.,[2]). However, most latent semantic models still view a query (or a document) as a bag of words. Therefore, they are not effective in capturing fine-grained contextual structures for information retrieval.

目前的模型是使用词袋模型，无法捕捉细粒度的上下文结构信息.

Modeling contextual information in search queries and documents is a long-standing research topic in information retrieval (IR) [2][4][8]. Usually, the contextual information captured by models such as TF-IDF, BM25, and topic models, is often too coarse-grained to be effective. As an alternative, there are retrieval methods such as the phrase-based translation model [5] that directly model phrases (or word n-grams), but they often suffer from the data sparseness problem. In a separate line of research, deep learning based techniques have been proposed for semantic understanding[3][6][9][10]. Salakhutdinov and Hinton [9] demonstrated that the semantic structures can be extracted via a semantic hashing approach using a deep auto-encoder. Most recently, a Deep Structured Semantic Models (DSSM) for Web search was proposed in [6], which is reported to outperform significantly semantic hashing and other conventional semantic models.

传统方法捕捉的上下文信息的粒度太大了.

In this study, based on a *convolutional* neural network [1], we present a new Convolutional Deep Structured Semantic Models (C-DSSM). Compared with DSSM, C-DSSM has a convolutional layer that projects each word within a context window to a local contextual feature vector. Semantically similar words-within-context are projected to vectors that are close to each other in the contextual feature space. Further, since the overall semantic meaning of a sentence is often determined by a few *key* words in the sentence, thus, simply mixing all words together (e.g., by summing over all local feature vectors) may introduce unnecessary divergence and hurt the effectiveness of the overall semantic representation. Therefore, C-DSSM uses a max pooling layer to extract the most salient local features to form a fixed-length global feature vector. The global feature vector can be then fed to feed-forward neural network layers, which perform affine transformations followed by non-linear functions applied element-wise over their inputs to extract highly non-linear and effective features.

WWW'14 Companion, April 7–11, 2014, Seoul, Korea.
ACM 978-1-4503-2745-9/14/04.
http://dx.doi.org/10.1145/2567948.2577348

2. C-DSSM FOR EXTRACTING CONTEXTUAL FEATURES FOR IR

The architecture of the C-DSSM, is illustrated in Figure 1. The C-DSSM contains a word hashing layer that transforms each word into a letter-tri-gram input representation, a convolutional layer to extract local contextual features, a max-pooling layer to form a global feature vector, and a final semantic layer to represent the high-level semantic feature vector of the input word sequence.

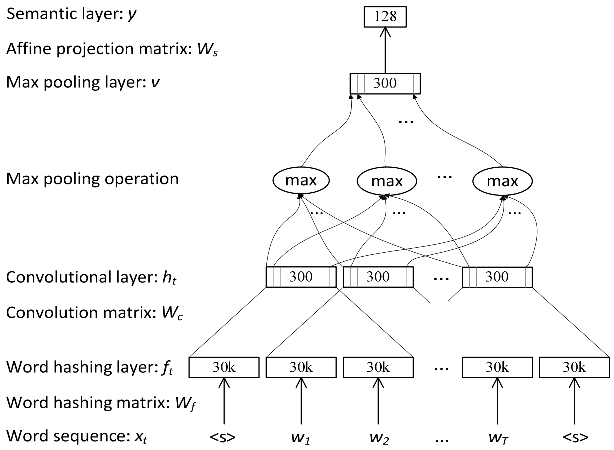

Figure 1: Illustration of the C-DSSM. A convolutional layer with the window size of three is illustrated.

In what follows, we describe each layer of the C-DSSM in detail, using the annotation illustrated in Figure 1.

The word hashing layer transforms each word in an input word sequence into a feature vector using the technique called *word hashing* proposed in [6]. For example, the word is represented by a count vector of its letter-tri-grams.

The convolution operation can be viewed as sliding window based feature extraction. It is designed to capture the contextual features for a word. Consider a word at the t-th position in a word sequence. The word hashing feature vectors of all the context words within a window around w_t are firstly concatenated to form a context window vector, and then projected to a local contextual feature vector h_t, as shown in Figure 1.

The contextual feature vectors extracted at the convolutional layer are local features, one for each word. They need to be combined to obtain a global feature vector with a fixed size. For the IR task, we want to suppress the non-significant local features and only retain the salient features that are useful for IR in the global feature vector. Therefore, we use a max operation, known as *max pooling*, which forces the network to retain only the most useful local features produced by the convolutional layers.

On top of the global feature vector, a semantic layer is applied to extract the high-level semantic representation, denoted by y.

In this model, both the convolutional layer and the semantic layer use the $tanh$ function as the non-linear activation function.

We further compute the relevance score between the query and each document by measuring the cosine similarity between their semantic vectors. Formally, the semantic relevance score between a query Q and a document D is measured as:

$$R(Q,D) = \text{cosine}(y_Q, y_D) = \frac{y_Q{}^T y_D}{\|y_Q\|\|y_D\|}$$

where y_Q and y_D are the semantic vectors of the query and the document, respectively. In Web search, given the query, the documents are ranked by their semantic relevance scores.

The parameters of the C-DSSM to be learned include convolution matrix W_c and semantic projection matrix W_s, as illustrated in Figure 1. Note that the word hashing matrix W_s is fixed without need of learning. The C-DSSM is trained on clickthrough data by maximizing the conditional likelihood of the clicked documents given a query, using stochastic gradient ascent. Learning of the C-DSSM is similar to that of learning the DSSM described in [6].

3. EXPERIMENTS

实验

We have evaluated the retrieval models on a large-scale real world data set, called the evaluation data set henceforth. The evaluation data set contains 12,071 English queries sampled from one-year query log files of a commercial search engine. On average, each query is associated with 65 Web documents (URLs). Following [5], we only used the title field of a Web document for ranking. The results are reported by mean Normalized Discounted Cumulative Gain (NDCG) [7]. In our experiments, the clickthrough data used for model training include 30 million of query/clicked-title pairs sampled from one year query log files. We then tested the models in ranking the documents in the evaluation data set. The main results of our experiments are summarized in Table 1, where we compared the proposed C-DSSM (Row 6) with a set of baseline models, including BM25, the unigram language model (ULM), phrase-based translation model (PTM), word-based translation model (WTM), and the DSSM. The proposed C-DSSM (Row 6) has a convolutional layer and a max-pooling layer, both having 300 neurons, and a final output layer using 128 neurons. The results show that the proposed C-DSSM outperforms all the competing methods with a significant margin. All models, except BM25 and ULM, use the same clickthrough data for learning. Superscripts α, β, and γ indicate statistically significant improvements ($p < 0.05$) over BM25, PTM, and DSSM, respectively. The proposed C-DSSM outperforms all the competing methods with a significant margin.

3000万个
q-d pair for training.

Table 1: Comparative results with the previous approaches.

#	Models	NDCG@1	NDCG@3	NDCG@10
1	BM25	0.305	0.328	0.388
2	ULM	0.304	0.327	0.385
3	WTM	0.315^{α}	0.342^{α}	0.411^{α}
4	PTM (len ≤ 3)	0.319^{α}	0.347^{α}	0.413^{α}
5	DSSM	0.320^{α}	$0.355^{\alpha\beta}$	$0.431^{\alpha\beta}$
6	**C-DSSM win =3**	$\mathbf{0.342}^{\alpha\beta\gamma}$	$\mathbf{0.374}^{\alpha\beta\gamma}$	$\mathbf{0.447}^{\alpha\beta\gamma}$

4. CONCLUSION

The work presented in this paper developed a novel learnable deep learning architecture based on the use of a CNN to extract both local contextual features (via the convolution layer) and global contextual features (via the max-pooling layer) from text. Then the higher layer(s) in the overall deep architecture makes effective use of the extracted context-sensitive features to perform semantic matching between documents and queries, both in the form of text, for Web search applications.

5. REFERENCES

[1] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P., 2011. Natural language processing (almost) from scratch. In Journal of Machine Learning Research, vol. 12

[2] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T., and Harshman, R. 1990. Indexing by latent semantic analysis. J. Amer. Soc. Information Science, 41(6): 391-407

[3] Mesnil, G., He, X., Deng, L., and Bengio, Y., 2013. "Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding." In Interspeech.

[4] Gao, J., Nie, J-Y., Wu, G. and Cao, G. 2004. Dependence language model for information retrieval. In SIGIR.

[5] Gao, J., He, X., and Nie, J-Y. 2010. Clickthrough-based translation models for web search: from word models to phrase models. In CIKM, pp. 1139-1148.

[6] Huang, P., He, X., Gao, J., Deng, L., Acero, A., and Heck, L. 2013. Learning deep structured semantic models for web search using clickthrough data. In CIKM

[7] Jarvelin, K. and Kekalainen, J. 2000. IR evaluation methods for retrieving highly relevant documents. SIGIR, pp. 41-48.

[8] Metzler, D. and Croft, B. 2005. A Markov random field model for term dependencies. In SIGIR.

[9] Salakhutdinov R., and Hinton, G., 2007. Semantic hashing. in Proc. SIGIR Workshop Information Retrieval and Applications of Graphical Models.

[10] Tur, G., Deng, L., Hakkani-Tur, D., and He, X., 2012. Towards Deeper Understanding Deep Convex Networks for Semantic Utterance Classification, In ICASSP