Deep Learning for Query Semantic Domains Classification

I-Ting Fang
Bing
Microsoft Corporation
Bellevue, WA 98004
itingfan@microsoft.com

Abstract

Long Short Term Memory (LSTM), a type of recurrent neural network, has been widely used for Language Model. One of the application is speech query domain classification where LSTM is shown to be more effective than traditional statistic models and feedforward neural networks. Different from speech queries, text queries to search engines are usually shorter and lack of correct grammar, while hold certain patterns. In this project, we demonstrate the models that are capable of carrying information both forward and backward, bi-direction LSTM, outperform forward only LSTM. Convolutional Neural Network which encapsulates words regardless of the order also achieves a comparable result as bi-directional LSTM.

1 Introduction

Understanding the query intent is a crucial task for a commercial search engine. Especially for queries where it provides more value to show the answer directly on the search results than showing web results. For example, unit converter is useful for query {kg to lb}, a clock showing time is preferable for query {time in paris}. Here we call the element that fulfill the query without clicking on web results *domain* answers. Previous work on query domain classification or text categorization [10] often use n-grams features, do feature selection and apply classification models, like SVM or boosted decision trees. Not much work had been published in this area until the emergence of Deep Learning, when researchers started to employ various deep neural nets to tackle the task for speech queries [11]. The research on query intent covers a wide range of topics. Among them, using past user engagement for prediction [13] is the most common direction which is beyond the focus of this project.

Deep neural network has gained much attention in recent years. Different advanced network structure has been developed to accomplish computer vision and natural language processing (NLP) tasks. Among various architecture of neural network, Recurrent Neural Network (RNN) has been widely applied to language model [4]. The property of recurrence helps break the limitation of words dependency boundary that often suffered by feedforward deep neural network. However, the RNN is hard to train due to the vanishing point behaviors. To address the issue, gated memory unit like long short term memory (LSTM) [9], and gated recurrent unit (GRU) [3] were introduced and shown their robustness that further improve the performance. Another breakthrough that makes Deep Learning NLP gains its popularity is the development of word2vec [7] and GloVe [6]. They make the best use of computation power on NLP tasks and yet preserve the semantic distance property.

A comparison of recurrent neural network for lexical domain classification was done in [1]. The authors show the comparison of n-grams and word vectors models and demonstrate better performance of recurrent neural network with LSTM/GRU using word vectors for domain classification task on Cortana speech query dataset. Based on the observation, this project will use word vectors and start building up experiments from basic LSTM units. Different from the Cortana dataset, the query dataset used in this project (Bing dataset) has wider variety and tends to be shorter and noisier. Given the nature of search engine queries, the order of the words is not as organized as speech, so the model that can carry information both forward and backward could be beneficial to the Bing dataset. Thus, bi-directional LSTM models will be the main focus in the experiments and compared to the performance of forward LSTM models.

Given the property of search engine queries, semi-random word orders and limited word length per classification task, the mechanism of Convolutional neural network seems to be a good fit. CNNs are widely used in computer vision tasks and become state of art models, but it's rarely used in NLP applications. However, a recent paper [5] show its potential for sentence classification. Similar to the proposed system, we treat each query as one sentence, and apply convolution and max-pooling layers before prediction.

In the end of this report, we will evaluate the performance of different architectures of RNN and CNN models with their accuracy and F1 scores. As an extension, we also run the experiments for queries in German, compare the results of using English/German fixed training data and German along training data.

2 Methods

Pre-trained GloVe word vectors are known for retaining the semantic distance in the word vector representation space. This is a desired quality for the semantic domain classification tasks for this project since we want to maximize the margins between classes. So we use vectors from GloVe as initial values for the matched query words. For query words that do not exist in GloVe vocabulary, the vectors will be initialized with random values using Xavier initializer.

2.1 Recurrent Neural Network

A conventional RNN state's update involves the current inputs as well as the previous states. Given a query composed by words x = $(x^1 x^2 \dots x^T)$, it computes the hidden states h = $(h^1 h^2 \dots h^T)$ and outputs $\hat{y} = (\hat{y}^1 \hat{y}^2 \dots \hat{y}^T)$ from t = 1 to T as follows:

$$h^t = H(W^{xh} x^t + W^{hh} h^{t-1} + b^h)$$

$$\hat{y}^t = softmax(W^{hy} h^t + b^{y)}$$

here W denotes weight matrices, b denotes bias vectors and H(·) is the hidden layer activation function. Through backward propagation, the weight updating values tend to be very small as the time step gets larger. One of the solutions is to employ Gated Recurrent Unit (GRU). It adds update (z^t) and reset gate (r^t) to retain the information over long time steps.

$$z^t = \sigma(W^z x^t + U^z h^{t-1} + b^z)$$

$$r^t = \sigma(W^r x^t + U^r h^{t-1} + b^r)$$

$$\hat{h}^t = tanh(W x^t + r^t \circ U h^{t-1} + b)$$

$$h^t = z^t \circ h^{t-1} + (1 - z^t) \circ \hat{h}^t$$

here $\sigma(\cdot)$ is sigmoid function. If update gate z is close to 1, then the information in that unit can be propagated through many time steps.

Another popular solution is the use of Long Short Term Memory. LSTM in [9] separates the memory cell and the outputs. A later version [12] added forget gates (f) to the memory cell which provide a way to reset memory. Altogether, the cell allows error message to flow at different strength based on the input. The following equations illustrate the update procedure.

$$i^t = \sigma(W^i x^t + U^i h^{t-1} + b^i)$$

$$f^t = \sigma(W^f x^t + U^f h^{t-1} + b^f)$$

$$o^t = \sigma(W^o x^t + U^o h^{t-1} + b^o)$$

$$m^t = i^t \circ \; tanh(W^c x^t + U^c h^{t-1} + b^c) + f^t \circ m^{t-1}$$

$$h^t = o^t \circ \tanh(m^t)$$

$$\hat{y} = softmax(W^{pred} h^T + b^{pred})$$

here i^t, f^t, o^t, m^t, h^t denote the input, forget, output gates, and memory units, hidden state respectively. At the final classification layer, softmax is applied to the linear transformation of the output at the final time step. We have experimented with the average of outputs at each time step and then apply softmax, but it didn't lead to better performance. So we will use only final step outputs throughout this report. To compare different combination of LTSM, we trained the models listed in Table 1. If not mentioned, the hidden layer has 100 dimensions.

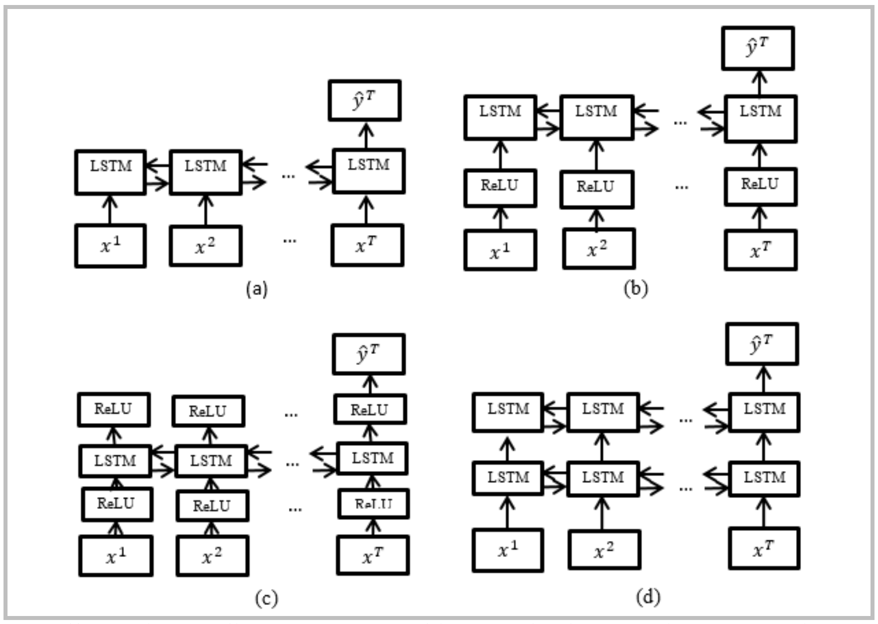

Figure 1: Illustration of different RNN architecture in the experiment. (a) BiLSTM. (b) ReLU-BiLSTM. (c) BiLSTM- 2ReLU. (d) BiLSTM2-50.

Table 1: Models

Model	***Description***
LSTM	1-layer LSTM with word vectors updated
BiLSTM-100/50	Bi-directional LSTM-Embed
BiGRU	Replace LSTM with GRU.
BiLSTM-Fixed	BiLSTM without updating word vector.
ReLU-BiLSTM	Hidden layer before BiLSTM
BiLSTM-2ReLU	ReLU-BiLSTM plus a hidden projection layer.
BiLSTM2-50	2-layer BiLSTM. 50 dimensions per layer.
CNN	Convolutional neural network

A 1-layer forward direction LSTM will be used as the baseline. Then we train the bi-directional LSTM. With word vector length 300, hidden layer dimension 100 and 15 final domains along with embedding word vectors, the bi-directional LSTM model almost hits the resource limit of our experiment environment, so other model variations in this report will have less or at most in the same magnitude of parameters. By default, the embedding is trainable, but in order to see its impact, we will train a model with fixed word vectors (BiLSTM-Fixed). We also add one 100 dimension hidden layer between embedded word vectors and LSTM inputs and a 30 dimension hidden layer between LSTM output and the final prediction in separate models, as shown in Figure 1. Note that these additional hidden layers are just “feedforward” but not “recurrent. And since we first reduce the input dimensions to LSTM, ReLU-BiLSTM has less parameters than BiLSTM. Finally, we stack an additional layer of BiLSTM with each layer having 50 dimensions’ states and memory.

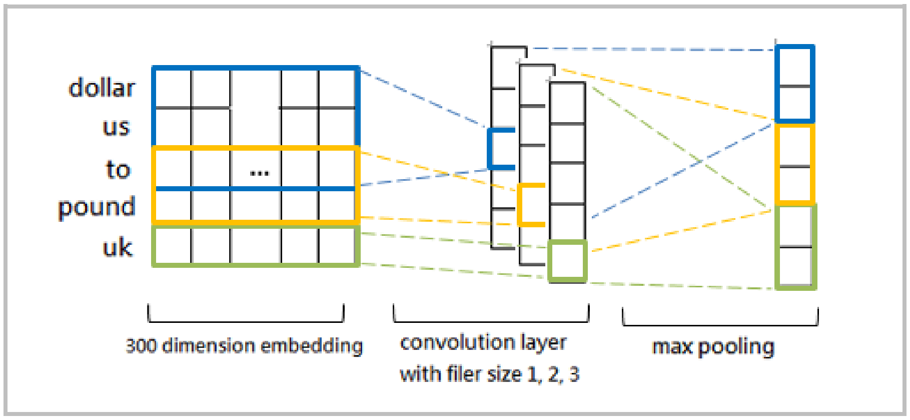

Figure 2: Convolutional Neural Network.

2.2 Convolutional Neural Network

Adopting the system proposed for sentence classification in [8], we use only 1 trainable embedding channel, apply convolution over uni-gram, bi-gram and tri-gram with 128 numbers of filters each. And then apply non-linearity followed by max pooling layer. Before we do final softmax prediction, dropout is done after max pooling step.

3 Experiment

There are 14 domains included in the experiment, namely Math Calculation, Currency, Dictionary, Direction, Flight Status, Stock, Soccer, MLB NBA, Translation, Weather, Time, Map, and Merchants/Institutions. We added 15th category as out of the above domains. Each domain has around 100 to 800 thousands samples based on the real domain coverage from traffic to bing.com. Out of domain queries include both queries that do not belong to these 14 domains, plus domain queries that have poor user engagement.

There are 8,683,485 queries from train, dev, and test data with ratio of 80%, 10%, 10% correspondingly. In total, 1,278,450 distinct words exist in the query set. But due to the memory limitation, only 40% of them are used in training. We take 372,602 matched words from pre-trained GloVe, and add words that have appearance more than 10 times to the final vocabulary for training. This leads to 493,620 words in the final vocabulary.

To make training more efficient, the queries with the same word length are put in the same mini-batch. We also pad the queries to have the same length, and drop the empty parts during back-propagation.

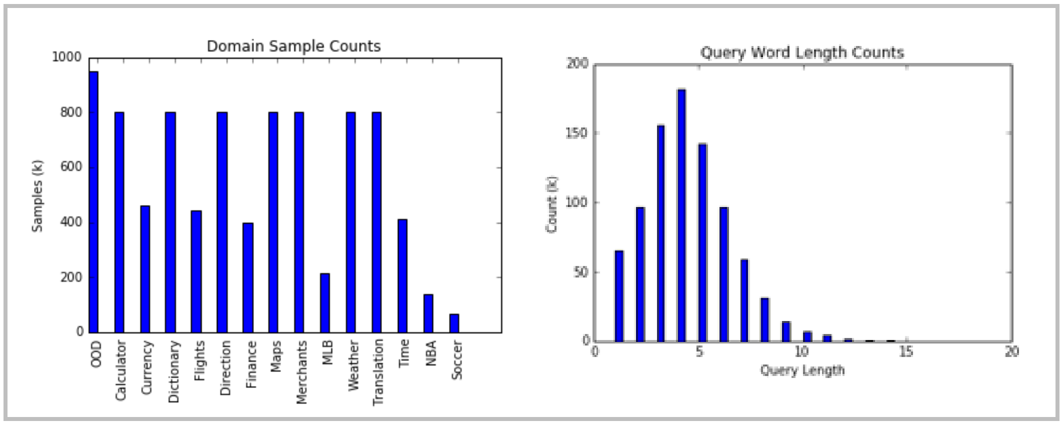

Figure 3: (a) Domain data. (b) Length of query histogram.

3.1 Extension to German

A similar experiment can be done for non-English query domain classification; German is tested in this experiment. Instead of comparing different structure of neural networks, we will use BiLSTM and compare the result of using different training data, one half English, half German, and another German only. The intuition behind the idea is that since we don't have German pre-trained word vectors, we want to leverage the knowledge embedded in the English pre-trained vectors to guide the training of word vectors for German.

4 Results

Results for all models are shown in Table 2. The baseline forward direction only LSTM performs far worse than the bi-direction models. By comparing the performance of BiLSTM-Fixed and BiLSTM, we can also find that updating embedding word vectors helps the classification performance. More details on updated word vectors will be covered in 4.1.

Table 2: Results

Model	**Accuracy (%)**	***F1 Score***
LSTM-100	82.72%	0.8510
BiLSTM-50	92.12%	0.9218
BiLSTM	**93.35%**	**0.9345**
BiGRU	93.18%	0.9321
BiLSTM-Fixed	91.86%.	0.9167
ReLU-BiLSTM	93.15%	0.9327
BiLSTM-2ReLU	93.22%	0.9316
BiLSTM2-50	**93.32%**	**0.9342**
CNN	**93.33%**	0.9334

Among all RNN variations, BiLSTM with 100 hidden states and memory cells has the highest Accuracy and F1 scores. It is also the model which has the most parameters and takes the longest time to train. BiGRU which has simpler architecture in the memory cell has slightly worse accuracy than BiLSTM. Stacking two 50-dimension bi-directional LSTM (BiLSTM2-50) outperforms BiLSTM-50, and has comparable results with 100-dimension bi-directional

LSTM (BiLSTM). The observation shows us that splitting the dimensionality in 1 hidden layer into 2 layers reduces the number of parameters, and yet keep the high performance for our task. Finally, CNN shows as good results as the best of BiLSTM models, but it also took twice amount of time to train.

4.1 Word Vectors

The training objectives of GloVe and domain classification are different. The first is trained on the occurrence of words, while the latter is trying to move the words around to increase the margin between different classes. So we can expect word vectors which belong to the same domain to be transformed as close as possible. Figure 2 demonstrates the results meet the intuition. In (a), both Major League Baseball teams, Yankees, and National Basketball Association teams, Lakers, are very close to each other. After training, as shown in (b), all the MLB teams overlapped and are separated out from NBA teams. Measurement unit words, “lb”, “”kg”, and currency names words. “jpy”, “euro”, get closer to their own kinds as well. On the other hands, since city name can appear in multiple domains like Weather, Time, MLB, they are more scattered than in pre-trained vectors space.

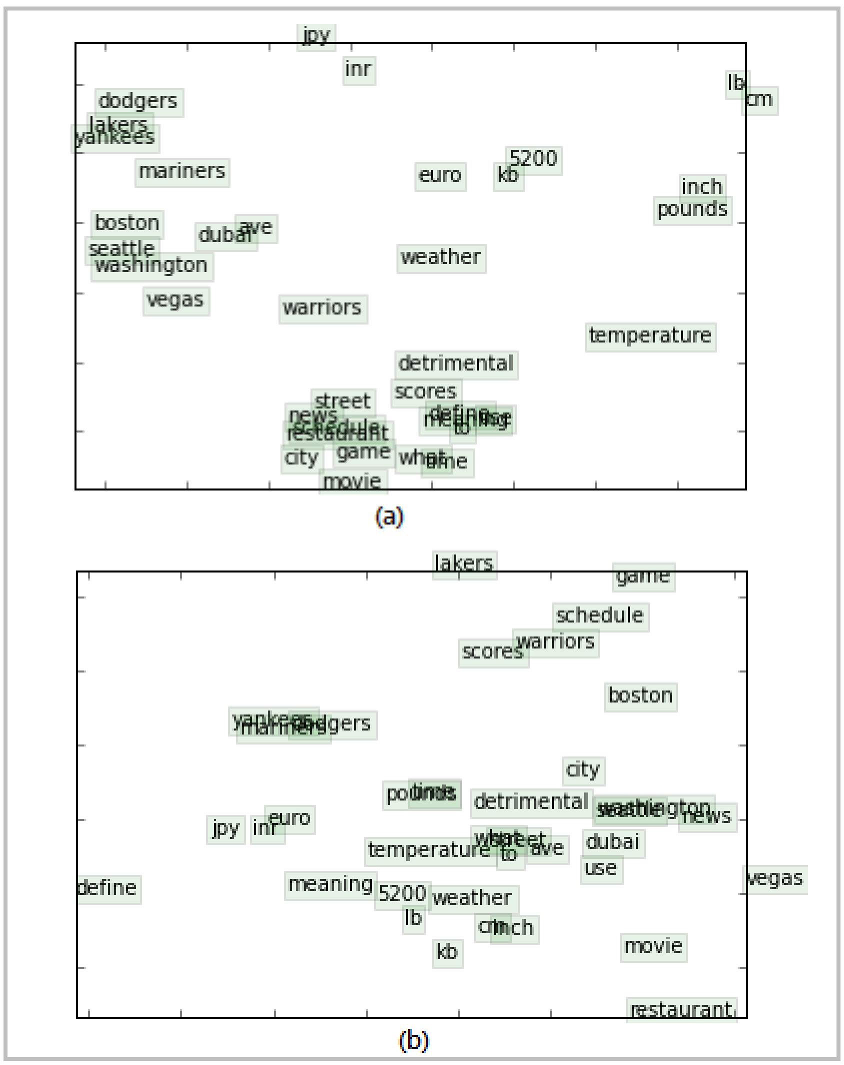

Figure 2: (a) shows words projection using GloVe pre-trained word vectors. (b) shows the projection of updated word vectors after domain classifier training.

4.1 Domain Classification for German

The model trained with half English and half German is shown to have slightly higher accuracy 93.04% than 92.88% by the model trained on German only. While the gain in accuracy is not drastic, the benefit on model training time is more noticeable. The model leverage English pre-trained vector achieve the reported accuracy in just 2 epochs while the pure German model converged after 5 epochs.

5 Conclusion

The goal of this project is to explore the architecture of neural network models that best suits the semantic domain classification task, for search engine queries specifically. The results show descent accuracy, 93.35% for 15 domains task, with bi-directional LSTM models. One model tends to surpass another by having more parameters with the price of training time. The biggest bottleneck of the system is the handling of words not in the vocabulary. Merely 1-word queries with unseen words account for 35% of total false prediction. To further improve the performance, we either simply increase the memory for larger vocabulary or implement the idea proposed in [14] to have a hybrid word-character model that switches to character-based classification when face unknown words.

References

[1] Ravuri, Suman, and Andreas Stolcke. "A Comparative Study of Recurrent Neural Network Models for Lexical Domain Classification."

[2] Li, Shoushan, and Chengqing Zong. "Multi-domain sentiment classification." Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers. Association for Computational Linguistics, 2008.

[3] Junyoung Chung, C¸ aglar Gulc¸ehre, KyungHyun Cho, and ¨ Yoshua Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” CoRR, vol. abs/1412.3555, 2014.

[4] Toma´s Mikolov, Martin Karafi ˇ at, Luka ´ s Burget, Jan “Honza” ˇ Cernock ˇ y, and Sanjeev Khudanpur, “Recurrent neural network ´ based language model,” in Proc. Interspeech, Makuhari, Japan, Sept. 2010, pp. 1045–1048

[5] Kalchbrenner, Nal, Edward Grefenstette, and Phil Blunsom. "A convolutional neural network for modelling sentences." *arXiv preprint arXiv:1404.2188*(2014).

[6] Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global Vectors for Word Representation." *EMNLP*. Vol. 14. 2014.

[7] Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." *Advances in neural information processing systems*. 2013.

[8] Kim, Yoon. "Convolutional neural networks for sentence classification." *arXiv preprint arXiv:1408.5882* (2014).

[9] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory."*Neural computation* 9.8 (1997): 1735-1780.

[10] Schapire, Robert E., and Yoram Singer. "BoosTexter: A boosting-based system for text categorization." *Machine learning* 39.2 (2000): 135-168.

[11] Ravuri, Suman V., and Andreas Stolcke. "Neural network models for lexical addressee detection." *INTERSPEECH*. 2014.

[12] F. Gers, J. Schmidhuber, and F. Cummins, “Learning to forget: Continual prediction with lstm,” *Neural Computation*, vol. 12, pp. 2451–2471, 2000.

[13] Li, Xiao, Ye-Yi Wang, and Alex Acero. "Learning query intent from regularized click graphs." *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2008.

[14] Luong, Minh-Thang, and Christopher D. Manning. "Achieving Open Vocabulary Neural Machine Translation with Hybrid Word-Character Models."*arXiv preprint arXiv:1604.00788* (2016).