Cuisine In and Out: A Distributed Search Engine with Attention Machine Translation and Doc2Vec

GUANGYU ZHANG and RIHAN YANG, New York University

The project aims to give convenience for searching best matched cuisine in Yelp review dataset, and also translate English reviews to German by attention neural machine translation. It’s a great fun to explore Natural Language Processing applications in search engine, besides improving the distributed performance for large dataset. Our project and demo focus on the high performance of retrieval valuable Yelp reviews and direct real-time hyperlinks to business’ homepage with query terms of cuisine; our report will focus on illustrating zigzag discovery path on “the-state-of-the-art” neural machine translation. We have spent much time on acquiring new NLP knowledge, learnt tf-seq2seq by TensorFlow, trained translation models on GPU servers, and translated the reviewing dataset from English to German.

ACM Reference format:

1 INTRODUCTION

Real-life domains of datasets are not that ideal as WikiPedia articles, which are highly structured, terminology-identified, neutral, etc. In fact, more advanced methodologies than counting term frequencies are appealed to build a robust search engine for real-life scenarios. We used Yelp data (e.g. reviews, ratings) for information retrieval adopting Natural Language Processing (RNN with attention mechanism, as in figure 1) to translate English reviews into German, and document representation method (doc2vec) to work with traditional TF-IDF.

In section 2, we give a detailed literature summarization of attention machine translation, which is the theoretical foundation of tf-seq2seq by TensorFlow, and also the model we used to translate reviews. Section 3 summarizes what we have realized in the system. After giving references of datasets in section 4, we will show the system performance in section 5. Finally, we will make conclusions and indicate further possible improvements.

Additional Key Words and Phrases: Recurrent neural networks, attention mechanism, doc2vec

ACM Reference format:

Fig. 1. RNN Encoder-Decoder with attention mechanism overview [Britz et al. 2017].

Our project experience is a journey of discovering new knowledge and making it into reality. We were introduced much knowledge of NLP in search engine during course lectures, e.g. word embeddings, machine translation, etc., while we hope to dive deeper and chase for “the-state-of-the-art” technologies. During the first half of this semester, we self-learnt the Deep Natural Language Processing course offered by Oxford University and DeepMind [Blunsom 2017] to seek for challenging and fancy applications to search engine. Based on the introductory of this course, we further read papers to understand core principles of attention mechanism, and compare it with traditional machine translations, which we discussed in details later. In fact, it hasn’t always been favorably when we explored possible applications of attention in search engine. At the meantime, we got started with TensorFlow. Amazingly in April 2017, Google open-sourced tf-seq2seq as a general-purpose encoder-decoder framework in TensorFlow [Goldie and Britz 2017]. By the convenience of tf-seq2seq supporting attention mechanism, we were able to train and test the dataset modified from the open-sourced codes.

We have come across some "state-of-the-art" NLP research and applications, but eventually found out they are either impossible to implement without an enterprise-level platform, or internal training datasets are unavailable to the public. For example, Rush et al. applied attention models to news article summarization, which inspired us to do fuzzy query search with summarizing the cuisine reviews. However, during careful going through the paper, we realized the summarization performance highly relied on the uniqueness of corpus domain and correlation of training and testing datasets. Otherwise, informal contexts like Yelp reviews may yield trivial summarized words. It’s also amazing to find Jaech et al. implemented...
attention mechanism in learning-to-rank, while their Facebook internal click-on dataset wasn’t make public. Even though these earlier approaches haven’t been completed realized in our system, their theoretical foundations are identical and critical to our final prototype.

2 PROBLEM FORMULATION

2.1 Recurrent Neural Networks (RNN)

Recurrent Neural Networks are one of the most popular deep learning methods working with sequences of data like text, audio, and video [Olah and Carter 2016]. They can gain a high-level understanding of the input sequences, and even be used to generate new sequences of data as the output. RNN can learn a probability distribution $p(x_t|x_{t-1},\ldots,x_1)$ at time $t$, and being trained to predict the next symbol in the sequence. In general, RNN is a statistical language model based on the conditional probability distribution with the following features [Goodfellow et al. 2016]:

- Recurrent networks that produce an output at each time step and have recurrent connections between hidden units
- Recurrent networks that produce an output at each time step and have recurrent connections only from the output at one time step to the hidden units at the next time step
- Recurrent networks with recurrent connections between hidden units, that read an entire sequence and then produce a single output

Figure 2 illustrates a simple RNN language model, where $h^{(t)}$ are the hidden layers, calculated as $h^{(t)} = g(V[x^{(t)};h^{(t-1)}] + c)$. $V$ is the weight matrix, and $c$ is the bias vector of the current layer. In most literatures, the activation function $g$ is represented as tanh, while current trend prefers to Rectified linear unit (ReLU), which is more computational efficiency and easier to train. More computational details are unfold in figure 2 that maps an input sequence $x$ to another output sequence $o$, and the loss $L$ measures between $o$ and training target $y$. Weighted matrices $U, W, V$ are used to parametrize recurrent connections among inputs, hidden and outputs.

2.2 Attention RNN for Neural Machine Translation

Traditional phrase-based machine translation (PBMT) introduced by Google a decade ago will first parse an input sentence into separate phrases or words, and translate the phrases independently. In contrast, neural machine translation (NMT) will take the sentence as a whole, and treat the sentence as a base unit (rather than a word or a phrase). NMT can yield lower loss, higher BLEU score with less effort of training according to Google’s results [Goldie and Britz 2017].

Cho et al. proposed a RNN Encoder-Decoder with two RNN in 2014. In their model, one RNN encodes a sequence of symbols (e.g. text, image) of arbitrary length into a fixed length of representation, and the other RNN decodes the representation into another sequence of symbols. Both RNN in the model will be trained simultaneously so as to maximize the conditional probability of giving out a target sequence with the given source sequence. Statistical NLP assumes vocabularies with similar semantics are likely to have closing distributing features under the same contexts. Thus, this RNN Encoder-Decoder model can learn a semantically and syntactically meaningful representation of linguistic phrases [Cho et al. 2014]. A softmax activation function is used to compute a multinomial distribution (1-of-K coding)

$$p(x_t, j = 1|x_{t-1},\ldots,x_1) = \frac{\exp(w_j h_t)}{\sum^K_{j'=1} \exp(w_{j'} h_t)}$$

Based on above statistical computation, the goal of neural machine translation is under the condition of given a source sequence $e$, to find a translation $f$, which maximizes

$$p(f|e) = \Theta [p(e|f)p(f)]$$

where Koehn call $p(e|f)$ as translation model, and $p(f)$ as language model.

As great breakthrough coming by standing on the shoulders of giants (including itself), Google astonished the world again by announcing the Google Neural Machine System (GNMS) in 2016 yielding "the-state-of-the-art" results best mimicking human experts' translation. Besides the RNN Encoder-Decoder mechanism, GNMS also has brought in the Attention Mechanism, by which the decoder pays attention to a weighted distribution over the encoded input sentence vectors most relevant to generate the target word [Wu et al. 2016]. Attention reduces computation costs for NMT by connecting the bottom layer of the decoder (8 decoder layers) to the top layer of the encoder (8 encoder layers).

Attention for translation from matrix-encoded sentences is one of the most important developments in recent RNN researches, and was first proposed in [Bahdanau et al. 2014]. From a high-level view in figure 3, RNN receives two fixed-size vectors as additional inputs at each output position $t$ - embedding of the previously generated output symbol $e_{t-1}$, and encoding a “view” of the input matrix [Blunsom 2017]. To generate the fixed-size vectors at $t$, Bahdanau et al. proposed to compute a context vector $c_t$ as a weighted sum of annotations $h_t$ "containing information about the whole input sequence with a strong focus on the parts surrounding the t-th word.
The weights $\alpha_{ij}$ are called **Attention**, reflecting the importance of the annotation $h_j$ with respect to the previous hidden state $s_{i-1}$ in deciding the next state $s_i$ and generating $y_i$. Also, $e_{ij} = a(s_{i-1}, h_j)$ is an alignment model scoring "how well the inputs around position $j$ and the output at position $i$ match" [Bahdanau et al. 2014].

Both Bahdanau et al. and Google’s latest release of tf-seq2seq use bidirectional RNN (BiRNN) for annotating sequences in the encoder. Via concatenating the forward and backward hidden states in BiRNN, the annotation $h_j$ can summarize both preceding and following words, so that it will be focused on the words around $x_j$. This structure is also shown in figure 3. The overall structure published by Google is introduced in figure 1.

In addition, we used the **Gated Recurrent Unit (GRU)** to train the translation model in the project. tanh is a simple and conventional unit of RNN. After researchers’ efforts, RNN has many variations of gates, like LSTM, GRU, Depth Gates [Yao et al. 2015], which yield more promising results than conventional ones. Cho et al. also introduced GRU in this paper, whose "update gates" combine input and forget gates, and merge cell and hidden state as in figure 4. GRU has more parameters to train than conventional RNN, yet is less expensive than LSTM, so GRU is getting more and more popular.

### 2.3 BLEU Score for Translation Performance Evaluation

BLEU (bilingual evaluation understudy) score is a common metric to quantitatively evaluate the performance of machine translation, first proposed by [Papineni et al. 2002]. The closer the tokenized output of the MT to human experts’ translation, the higher the score. TF-seq2seq takes scripts from [Moses 2017] to calculate BLEU scores. "State-of-the-arts" machine translation techniques mentioned in above literatures can typically reach scores of 20 ~ 30.

$\alpha_{ij} = \exp(e_{ij}) \sum_k \exp(e_{ij})$

### 3 SYSTEM ARCHITECTURE

Improved from "Search Engine Architecture" assignments, we have mainly realized the following features:

- Used MapReduce framework to generate index files for dataset (Yelp reviews and business details) in size of GB;
- Loaded trained translation model from disk to translate Yelp business reviews, and output translated texts into separate files for the same number of partitions and hash values as document server;
- Computed 300-dimension vector representations for business reviews via doc2vec, and used to get highest vector similarity with query’s vector representation;
- Compared retrieval results between TF-IDF and doc2vec;
- Built an independent front-end API in JavaScript showing business details (like related reviews, cities), and direct click-on links to business’ homepages on Yelp.

An overview of our system is shown in figure 5.

### 4 DATASETS

We used the English-to-German WMT’16 translation task dataset, following the tutorial of tf-seq2seq. The dataset has already been preprocessed into corresponding English-German sentence pairs, and learnt subwords by Byte Pair Encoding (BPE). The vocabulary size of tokenized subword units is of ~32,000, and the bilingual training set is of ~GB in total. WMT’16 also provides extra bilingual sentence pairs taken from news for validation and tests.

Cuisine data for the search engine is taken from Yelp dataset challenge round9. The reviewing data for building TF-IDF and doc2vec is of ~3.5GB, and the business details data for illustrating in frontend is of ~100MB.

### 5 PERFORMANCE EVALUATION

#### 5.1 System Latency

It takes around 20 minutes to distributely get indexing results. Real-time retrieval latency is of ms magnitude. The translation process can take days even weeks...

---

5.2 Translation Model Training & Testing Results
We took the whole WMT’16 dataset to train with Attention BiRNN, 128-unit GRU cell and Adam optimizer following the tf-seq2seq tutorial. Detailed configurations can be modified in YAML files, and easily passed the filename to the --config_paths flag when calling the training function. We have consecutively trained for around one week on CIMS Cuda (GPU) servers for more than 560.0k steps. The trained model’s loss has begun to converge, while the BLEU score is relatively low, which indicates the model still needs to be trained. Besides, the number of GRU unit can also be increased (such as to 256) to get high score within the same training step. Figure 6 illustrates the training result given by TensorBoard.

In fact, due to the complexity of the translation model, it’s very expensive to train to gain a satisfied result. Google has spent more than 250,000 GPU hours in total to look insight into the model, and eventually get the leading BLEU score [Britz et al. 2017].

We then used the trained model to translate all Yelp reviews. Within our expectation, the translation isn’t very satisfactory since it doesn’t have a high BLEU score.

6 CONCLUSIONS
We have finished a prototype of searching for cuisine in Yelp reviews, and direct click-on links to Yelp online homepage. “The-state-of-the-art” attention machine translation models yields very close translated results as human experts, but are very expensive to train. The translation result isn’t satisfactory due to lack of training time (compared with Google’s 250,000 CPU hours). We hope to further trained the model with more iteration steps, and even larger number of RNN cells.

Furthermore, image caption can also be used for cuisine search, which is also a RNN Encoder-Decoder mechanism, with an extra convolutional neural network before RNN encoder to represent images. After getting images’ caption, we can search queries among captions, and include images as retrieval results.

ACKNOWLEDGMENTS
The project wouldn’t be accomplished without Prof. Matt Doherty’s advices and guidances. The authors also hope to thank NYU CIMS servers to provide high-performance GPU servers to run TensorFlow programs.

REFERENCES


Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner,
Cuisine In and Out: A Distributed Search Engine with Attention Machine Translation and Doc2Vec

Fig. 6. Model features given out by TensorBoard after 560.0k steps of training (with Attention BiRNN, 128-unit GRU cell and Adam optimizer). a: Loss begins to converge; b: BLEU score is low; c: Word embeddings reduced to 3D by PCA from 128D (37007 points). We have consecutively spent one week on CIMS Cuda GPU servers to reach this step, while Google has spent more than 250,000 GPU hours in total to get the leading score [Britz et al. 2017].

<table>
<thead>
<tr>
<th>Source</th>
<th>Translated</th>
</tr>
</thead>
<tbody>
<tr>
<td>I’m looking forward to my next visit.</td>
<td>Ich freue mich auf meine Zukunft, die ich für meine Zukunft habe.</td>
</tr>
<tr>
<td>the cook was really entertaining! food was delicious and well seasoned and cooked. great for small groups!</td>
<td>Die Zeit war wirklich alles, um die Lebensmittel, die sich sehr gut war, aber auch sehr gut und viele kleine Gruppen!</td>
</tr>
<tr>
<td>I hit them for $100 on the machines so i treated myself to an ice cream at danielles next door (separate review)</td>
<td>Ich habe sie für die Maschinen für die Maschinen, so wie ich selbst eine spont@@ forderte Überprüfung der Zukunft der Welt hat.</td>
</tr>
</tbody>
</table>

Table 1. Comparison between the source Yelp reviews and the translated ones by the trained model. Translated sentences don’t make sense if reverse back to English by Google online translation. Also, the translation fails when there’s a number.

Submitted May 2017
