Query Segmentation using LSTMs

Siddhartha Mishra
Indian Institute of Technology
Hyderabad, Telengana
es15btech11018@iith.ac.in

Divya Spoorthy
Indian Institute of Technology
Hyderabad, Telengana
es15btech11001@iith.ac.in

Raaghav R.
Indian Institute of Technology
Hyderabad, Telengana
es15btech11021@iith.ac.in

ABSTRACT

Here, we propose a two models for the problem of Query Segmentation one using Bi-Directional LSTMs and Bi-Directional LSTMs using CRF(Conditional Random Fields) layer by concerting the problem of Query Segmentation into that of Sequence Tagging. We show that in our model, Bi- Directional LSTMs help in capturing the time sequence relationship of words in a phrase/sentence. Our model can produce the state of the art (or close to) accuracy on Query Segmentation problem.

CCS CONCEPTS

• **Information systems** → **Query reformulation**; • **Computing methodologies** → *Supervised learning by classification*;

KEYWORDS

Query Segmentation, Sequence Tagging, bidirectional LSTMs, CRF

ACM Reference Format:
Siddhartha Mishra, Divya Spoorthy, and Raaghav R.. 2019. Query Segmentation using LSTMs. In *Proceedings of ACM Conference (Conference'17),* Jennifer B. Sartor, Theo D'Hondt, and Wolfgang De Meuter (Eds.). ACM, New York, NY, USA, Article 4, 4 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

Query Segmentation is the task of breaking(Segment) the web queries into multiple adjacent phrases so that each segment can refer to something meaningful and helps in highlighting the important aspects of the query and therefore increasing the relevance of the retrieved information. Sequence tagging is a pattern recognition task which assigns labels to each of the words present in the given sequence of query, POS tagging is an example. Query Segmentation and Sequence tagging pose two different problems and here we show that Query Segmentation can be modified so that it becomes a problem of Sequence Tagging, where we assign a label to each word in the query, whether there is a break after the end of the word or not. This basically converts this into a simple binomial classification problem.

Hence we turn to Long Short Term Memory units(LSTMs) and Conditional Random Fields(CRFs) for the purpose of classification. LSTMs are a unique kind of RNNs which are capable of long-term dependencies which were explicitly designed to avoid the long-term dependency problem. CRFs are a probabilistic framework for labeling and segmenting structured data. The underlying idea is that of defining a conditional probability distribution over label sequences given a particular observation sequence.

Conference'17, July 2017, Washington, DC, USA
2019. ACM ISBN 123-4567-24-567/08/06…$15.00
https://doi.org/10.475/123_4

2 RELATED WORK

2.1 Wikipedia Based Normalization

In this method, the n-gram frequencies of all potential segment s with at least two words are retrieved. For each segment s we check whether it is present in the Wikipedia title dictionary. if the segment s is found we replace its frequency with the maximal Google n-gram frequency. Finally the segmentation with the highest score is chosen. [**?**].

2.2 Naive Query Segmentation

Here it is assumed the phrases contained in web actually exist in the web. Here they use Goog;e n-gram corpus ans regard query q as a sequence $(w_1, w_2, ..., w_n)$ of n key words. A valid segmentation S for q is a sequence of disjunct segments s, each a contiguous subsequence of q, whose concatenation equals q. There are 2 nâĹŠ1 valid segmentations for q, and $(n^2 - n)/2$ potential segments that contain at least two keywords from q. Our algorithm derives a score for a valid segmentation as follows. First,the n-gram frequency count (s) of each potential segment s is retrieved.

2.3 Unsupervised Query Segmentation using query logs

Here, given a collection of search queries we consider an n-gram M = (w1, w2, .. wn). Assuming there are k queries q1, q2, ... qn, which denote the subset of queries in the log that contain the words of M, though not necessarily occurring together as n-gram.

The premise is that search queries can be viewed as bags of Multi-Word Expressions, which implies permutation of the MWEs constituting a particular search query will effectively represent the same query. To tesbt if an observed n - gram is an MWE we could ask if the constituents of an MWE appear together more frequently than they would under a bag-of- words null model. [**?**].

2.4 Conditional Random Fields

Conditional Random Fields model the conditional distribution P(y|x) where y = $y_1, y_2....y_n$ is the label sequence and x = $x_1, x_2....x_n$ is the input sequence. Begin, Inside, Outside (BIO) tags are introduced to map query into a sequence of tags.

2.5 RNN Encoder - Decoder Framework

The problem is mapped to a language conversion RNN task. The original query is treated as an input sequence of one language and the segmented query as the output sequence from another language. Vocabulary of the target includes a special break token other than those present in the source vocabulary. The queries and the segmentations are combined for training and the target segmented query is generated during test phase.

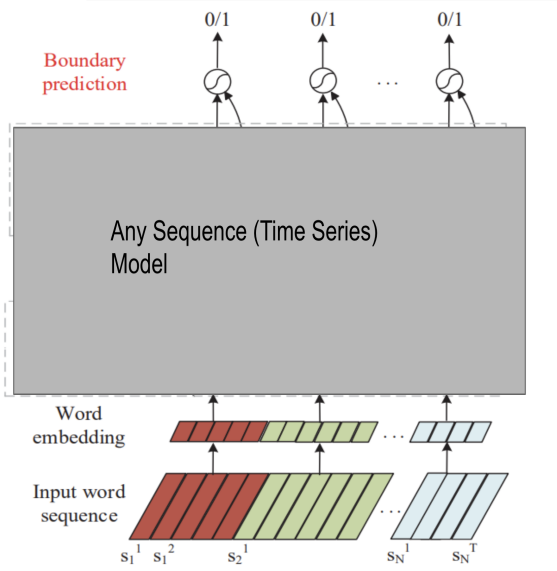

Figure 1: Generic architecture for query segmentation modeled as a sequence tagging problem

3 MODELS

The first step of Query Segmentation is to convert every word in the input sequence into its word embedding vector. Word embedding vector converts the vocabulary indices into a smaller space were similar word have high similarity among vectors. Here we used two different types of word embeddings, word2vec and fasttext, where we could see that the embeddings generated using fasttext were better able to capture the sequential form of the input in comaprison to word2vec.

Feeding these embeddings sequentially to a sequence model for example in the time series models like Hidden markov models, Conditional Random Fields, Many-to-Many models of RNN, LSTM or GRU cells, we get their corresponding outputs and their probabilities. Here each node corresponds to the output of the word and its respective loss.

$$T1 = P[break\ after\ word] \tag{1}$$

f(s1) = Probability of break predicted by network

Binary Cross Entropy loss function:

$CE = \sum_{n=1}^{C'=2} t_i log(f(s_i)) = -t_1 log(f(s_1)) + (1 - t_1) log(1 - f(s_1))$

3.1 Model units

3.1.1 RNN Cell. All the recurrent neural networks have the form of a chain repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer. Here RNN cell with many-to-one expansion with softmax output layer is used.

3.1.2 LSTM Cell. Just like RNNs, LSTMs also have the chain like structure, but the repeating model has a different structure. Instead of having a single neural network layer, there are four interacting in a special way.

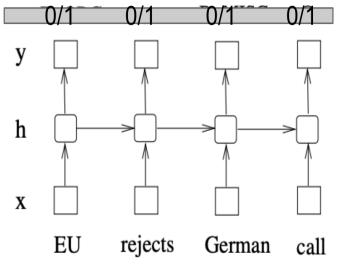

Figure 2: RNN cell and many-to-one expansion with softmax output

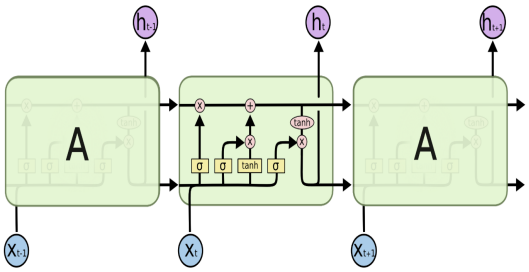

Figure 3: The repeating module in an LSTM contains four interacting layers.

3.1.3 CRF Layer. CRFs are used to capture the sequential data and take the previous context into account. Feature functions are used which have multiple input values:

(1) The set of input vectors X
(2) The position i of the data point we are predicting
(3) The label of data point i - 1 in X
(4) The label of data point i in X

The neighbour tag information is used in predicting tags. Next, input and output layers are directly connected and gradient descent is used.

3.2 Model1 : Bidirectional LSTM

In query segmentation task, we have access to both past and future input features for a given time, we can thus utilize a bidirectional LSTM network. In doing so, we can efficiently make use of past features via forward states and future features via backward states for a specific time frame. We train bidirectional LSTM networks using backpropogation through time(BPTT).

3.3 Model2 : Bi-Directional LSTM with CRF layer

CRF models give high tagging accuracy and captures neighbour label relationship, where as bidirectional LSTM helps capturing the time sequence relationship of words in phrase/sentence. Together the hybrid model is an intelligent way to capture which words should be clustered together to form a segment.

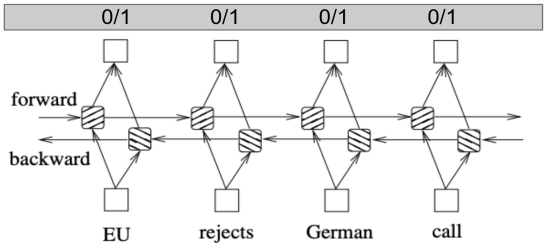

Figure 4: The repeating module in an LSTM contains four interacting layers.

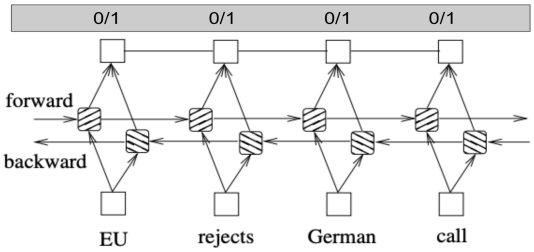

Figure 5: The repeating module in an LSTM contains four interacting layers.

Table 1: Model Parameters

Bi-Directional LSTM
N_epochs : 10000
Embedding dimension : 32
Vocabulary : indices for set of words in training data
Hidden layer dimension: 15
Output Layer : Softmax
Optimizer : Adam
Initialization : Xavier
Learning rate (Adaptive) - init : 0.001
Bi-Directional LSTM with CRF layer
N_epochs : 10000
Embedding dimension : 32
Vocabulary : indices for set of words in training data
Hidden layer dimension: 18
Output Layer : Softmax
Optimizer : Adam
Initialization : Xavier
Learning rate (Adaptive) - init : 0.0005

A EXPERIMENTS

A.1 Data

Here the dataset we used is the webis-qsec data. This dataset consists of three human annotators provided RJs mentioning usefulness

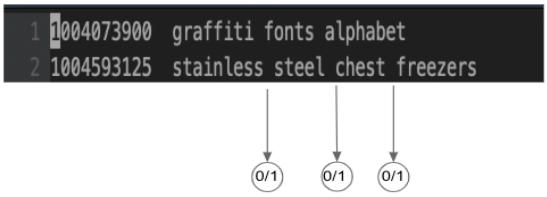

Figure 6: Converting to sequence tagging problem

Figure 7: Calculating target labels

of URL given query for all links obtained for all combinations of quoted version of queries. The queries were a randomly sampled query log from Bing Australia deduplicated and randomized to remove top bias. Another dataset webis-qsec provided by Amazon mechanical turk. The data is in the following form:

$$ID - Querystringsegmentedby \mid (separator) \tag{2}$$

A.2 Calculating target labels

Here Our approach of converting the query segmentation to sequence tagging is that after every word, there is a tag with value:

(1) 0 for continuing the phrase
(2) 1 for break (segmenting)

Hence the problem is converted such that, given a query sentence(sequence of words, tag the end of each word with a tag value 0 / 1.

There are 10 judges for the webis-qsec dataset. We use the normalized weight using the following way:

P[break after w_i] = (nvotes for break after word) / total number of votes

As shown in the Figure 7, in the first training label:

(1) P[graffiti, |] = (0 + 3 + 0) / 10 = 3 / 10
(2) P[graffiti, space] = (5 + 0 + 2) / 10 = 7 /10

A.3 Evaluation Metric

Supervised Framework In Supervised framework it is difficult to obtain bigger dataset. Hence we define two different kind of accuracies:

(1) **Query accuracy** Query Accuracy is defined as fraction of query segments which match with resultant
(2) **Break Accuracy** Break Accuracy is the fraction of breaks correctly captured

Both of the above measures can have a weighted version if there are multiple judges or voters

Table 2: Results

Model	Query Accuracy	Break Accuracy
WT[7]	0.431	0.769
WT + SNP[7]	0.585	0.837
CRF	0.465	0.814
Query segmentation revisited	0.082	0.189
Mishra et al	0.58	0.206
clickthrough	0.056	0.176
BiLSTM	0.485	0.775
BiLSTM with CRF	0.525	0.875

A.4 Results

Results for various existing methods and our method are mentioned in the Table2.

B CONCLUSION

From the above results it can be observed that, this hybrid model is comparable to the state of the art algorithms. The algorithms are implemented in the GitHub link https://github.com/Siddhartha1234/Query-Segmention-LSTM For future work to improve performance, running this model on an unsupervised Evaluation framework to get more data can be done. Random queries can be streamed from search engine, users and unsupervised data can be constructed / evaluated to get a bigger data to train the model dynamically. Feedback layers can be added (Idea used in clickthrough model) for the backward states for the dynamic training approach.

C REFERENCES

[1] N. Mishra, R. Saha Roy, N. Ganguly, S. Laxman, and M. Choudhury. Unsupervised query segmentation using only query logs. In WWW âĂŹ11, pages 91âĂŞ92. ACM, 2011.
[2] Query Segmentation via RNNs Encoder-Decoder Framework YC Lin 2017
[3] Bidirectional LSTM-CRF Models for Sequence Tagging by Zhiheng Huang, Wei Xu, Kai Yu
[4] Unsupervised Query Segmentation Using Clickthrough for Information Retrieval by Yanen LiâĹŮ 1, Bo-June (Paul) Hsu2, ChengXiang Zhai1, Kuansan Wang2
[5] Query Segmentation Revisited Qir Woo Hagen et al. PSB , 2011