Shahjalal University of Science and Technology, Sylhet

Department of Computer Science and Engineering

Query Suggestion For Bangla Search Engine Pipilika

A.F.M.Kamruzzaman
Reg. No.: 2014331029
4^{th} year, 2^{nd} Semester

Md. Rezaul Islam
Reg. No.: 2014331044
4^{th} year, 2^{nd} Semester

Department of Computer Science and Engineering

Supervised by:

Dr. Farida Chowdhury
Associate Professor
Department of Computer Science and Engineering

14th February, 2019

Query Suggestion For Bangla Search Engine Pipilika

A Thesis Submitted to the

Department of Computing Science and Engineering

Shahjalal University of Science and Technology

Sylhet - 3114, Bangladesh

in partial fulfillment of the requirements for the degree of

B.Sc.(Engg.) in Computer Science and Engineering

By

A.F.M.Kamruzzaman

Reg. No.: 2014331029

4^{th} year, 2^{nd} Semester

Md. Rezaul Islam

Reg. No.: 2014331044

4^{th} year, 2^{nd} Semester

Department of Computer Science and Engineering

Supervised by:

Dr. Farida Chowdhury

Associate Professor

Department of Computer Science and Engineering

14th February, 2019

Recommendation Letter from Supervisor

The thesis

entitled "Query Suggestion For Bangla Search Engine Pipilika"

submitted by the students

1. A.F.M.Kamruzzaman
2. Md. Rezaul Islam

is a record of research work and then development of an interface carried out under my supervision.

I, hereby, agree that the thesis can be submitted for examination.

Signature of the Supervisor:

Name of the Supervisor: Dr. Farida Chowdhury

Date: 14th February, 2019

Certificate of Acceptance of the Thesis/Project

The thesis
entitled " Query Suggestion For Bangla Search Engine Pipilika"
submitted by the students

1. A.F.M.Kamruzzaman
2. Md. Rezaul Islam

on 14th February, 2019
as part of the requirements of the course CSE-452, is being approved by the Department of Computer Science and Engineering as a partial fulfillment of the B.Sc.(Engg.) degree of the above students.

Head of the Dept.	Chairman, Exam. Committee	Supervisor
Dr Mohammed Jahirul Islam	Dr. Farida Chowdhury	Dr. Farida Chowdhury
Professor	Associate Professor	Associate Professor
Department of Computer	Department of Computer	Department of Computer
Science and Engineering	Science and Engineering	Science and Engineering

Abstract

Query Suggestion is a search engine feature whereby the system suggests completed queries as the user types a starting text. Queries are typically selected based on specific attributes.If a users text does not match any queries in the source queries cannot be suggested . we propose a neural language model which is called Recurrent Neural Network (RNN) that learns how to generate a query from a starting text and and we evaluate the proposed methods with a public data set. Our applied approach has no dependency on query log and this method has lower time complexity.

Keywords: Query Suggestion, RNN, LSTM etc.

Acknowledgements

We would like to thank the Department of Computer Science and Engineering, Shahjalal University of Science and Technology for supporting in this research. We are very thankful to our honorable supervisor Dr. Farida Chowdhury and Pipilika R&D Team for supporting us.

Dedication

We would like to dedicate our research to our parents and to our younger brothers.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

When a user types text into a search box, a list of queries are suggested to complete the users pretended text. every time a user types a word, the most relevant queries are attached to the users prefix and this process is called query suggestion. Query suggestion is an important step of information retrieval system that enriches users search process. Query Suggestion has much influence on search results.

From Our background study we have found two approaches for Query Suggestion process :

1. **Statistical Approach:**
   It is traditional approaches that uses a query log for query suggestion. Query log is a list of queries that user use these keywords as search queries. By applying most popular compilation and most relative compilation, process a list of suggested queries can be suggested, but there are some major limitations in these approaches. Those are given below:
   (a) If the users query does not exist in the query log then this process will fail.
   (b) The next thing is there is no fulfilled query log for Bangla search engine.
   (c) This approach is very slow approach because it has much time complexity.

(d) The traditional method does not support the contributions of the semantically structured relationship of queries.

The limitation of the traditional systems are no longer efficient for search engines nowadays. For this, it is not used by popular search engine nowadays.

2. **Recurrent Neural Network (RNN) Based Approach:**
   The neural network-based approach has no dependency and limitations like the statistical approaches. In order to overcome the traditional systems limitation, we propose a neural language model that can generate queries for a initial query. Our recurrent neural network-based language model encodes variable length prefix in a vector, and it performs the most relevant queries. Our model can be applied for a very large corpus and will be able to suggest reasonable queries.

As Pipilika Search Engine is the first Bengali search engine in Bangladesh, it is highly expected that the search engine should provide an efficient searching service to its user. A better Query Suggestion method can provide a better Searching experience in Pipilika search engine.

1.2 Reasons to Motivation in Query Suggestion

Query Suggestion is a popular User interface experience. It has two advantages:

1. By providing suggestions, it helps the user to type longer queries.
2. Its far more powerful - it helps to reformulate the best search.

Good suggestions are relevant queries that ensure the best search results. The quality of the suggestions has so much importance. But our pipilika search engine do not have any types of Query Suggestions.
So, we are motivated to apply a method which are more effective for Pipilika search engine.

1.3 Research Questions

Here are some questions that we are investigating in our research:

1. What are the limitations of statistical approach?

2. What are the techniques to be applied for the Query Suggestion in this search engine?

3. Can the language modeling approach perform better than the traditional approaches ?

4. How does improved Query Suggestion impact in performing efficient search operation?

Chapter 2

Query Suggestion

2.0.1 What is Query Suggestion?

Query Suggestion is a system that suggest queries that complete a users text as the user types. It is the first step of information retrieval, which helps users to reformulate the entire query after inputting starting texts of any length.

2.0.2 Why we Need Query Suggestion?

Query Suggestion help users find information quickly by showing queries that might be similar to the queries they are typing. Eg: If any user types "bangladesh", they may be able to pick all the next query texts related to the country bangladesh,the general or popular incidents of bangladesh. For example: bangladesh cricket, bangladesh area, bangladesh police etc.

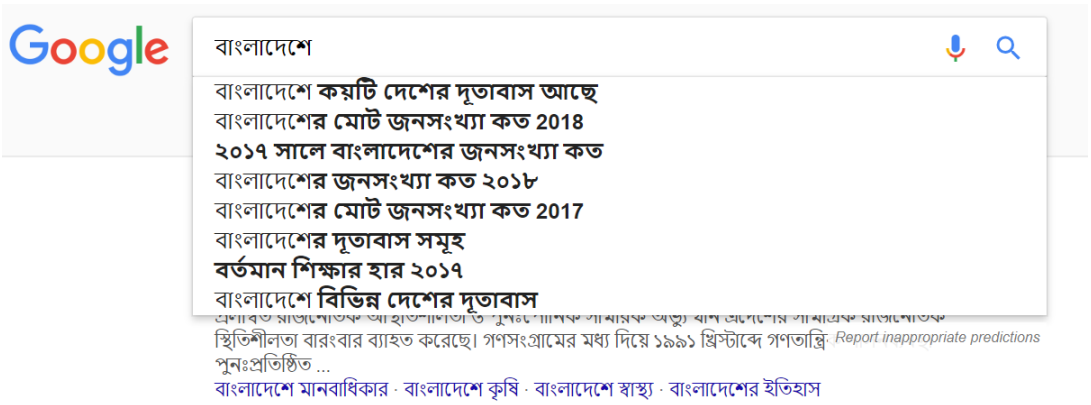


Figure 2.1: Example of Query Suggestion for Bangla text

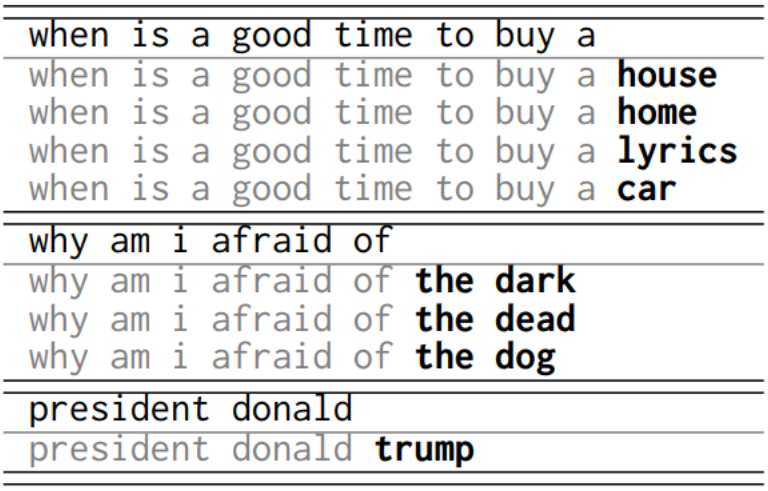


Figure 2.2: Example of Query Suggestion for English text

This search system automatically creates suggestions for a query.The automatic query suggestions can be different for different result sources and site collections.

Chapter 3

Background Study

Before we jump into discuss our own method we should take a look at Neural Network Language Model, Recurrent Neural Networks (RNN), LSTM.

3.1 Neural Network Language Model:

3.1.1 What is Neural Network Language Model?

A **neural network language model** is a language model based on Neural Networks , this model has the ability to learn distributed representations and reduces dimensional impact.

3.2 Recurrent Neural Networks (RNN):

3.2.1 What is Recurrent Neural Networks (RNN):

Recurrent Neural Network is a recurrence relation over time steps that is given by:

$$S_t = f\left(S_{t-1} * W_{rec} + X_t * W_x\right)$$

Where S_t is time step t, X_t an input at time t, W_{rec} and W_x are weighted values.

3.2.2 What is the Benefits of Recurrent Neural Networks (RNN)?

We have chosen RNN based approach rather than other approaches, Because RNN models provide a way to not only examine the current input but also provides one step back.

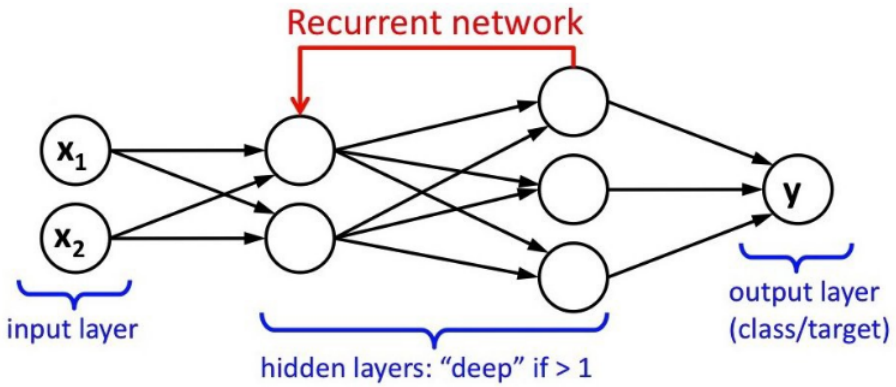


Figure 3.1: Structure of RNN

3.2.3 The Problem of Recurrent Neural Networks (RNN)?

RNN has three major problems:

1. **Vanishing Gradients:**

   In RNN model the gradient signal can be multiplied a large number of times by the weight matrix. So,Repeated use of weighted matrices causes vanishing gradients problem.

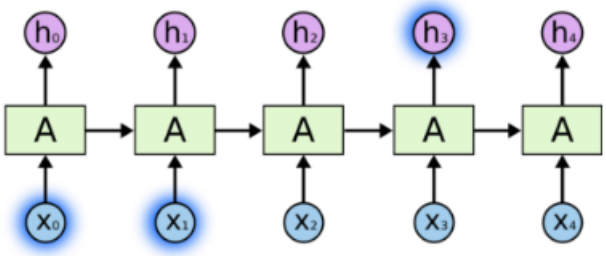
2. **Exploding Gradients:**

   It refers to the weights in this matrix are large that it can cause problems while training process.

3. **Failed To Connect With Previous More Information:**

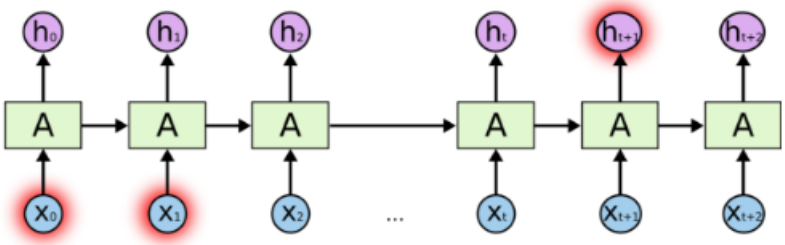
   RNN might be able to connect previous information to the present task. But RNN cannot do this all time. It depends.

   (a) Sometimes, If we are trying to predict the last word in **"the birds are flying in the sky"** its pretty easy that the next word is going to be **"sky"**. In these cases, the gap between the relevant data and the place needed to be small.



(b) There are also cases where we need more context. If our aim is to predict the last word in the text **I grew up in india I speak fluent urdu**. Then the next word may be the name of a language. But if we want to find out which language, we need the data related to **"india"**.

In these type of conditions, RNN's become unable to learn to connect the information.



Though RNN has some advantages but for these faults we haven't chosen RNN for auto-completion. We have chosen LSTM.

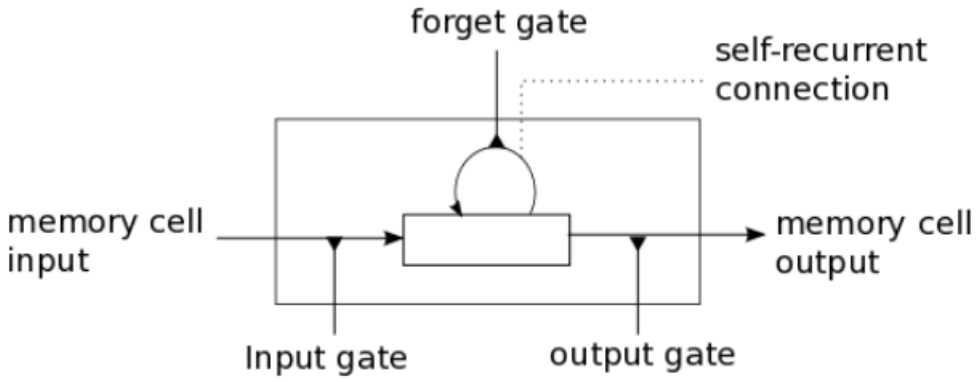
3.3 LSTM:

3.3.1 What is LSTM?

Long Short Term Memory network **LSTM** are a special kind of **RNN**, can learn long-term as well as short term dependencies. Hochreiter Schmidhuber(1997) constructed this model. and were popularized by many people for machine learning works. The work is now widely used because the results are very good.

3.3.2 Structure of LSTM:

LSTM has the structure which followsthe memory cell that consists of four elements: input, forget, output gates and a neuron is connected to itself:



All RNN models construct the form of a chain system of repeating modules of the network.

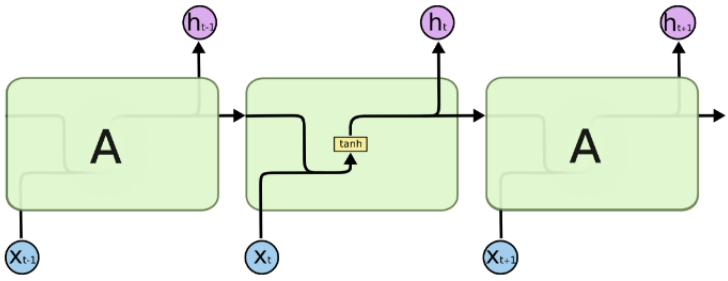


Figure 3.2: The repeating module in a standard RNN contains a single layer

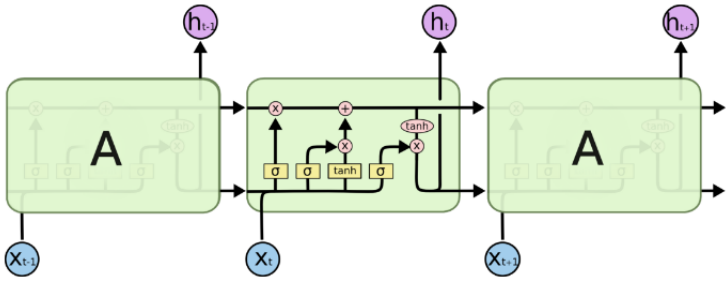


Figure 3.3: The repeating module in an LSTM contains four interacting layers

3.3.3 What are the Benefits of LSTM?

LSTM has overcome the pitfalls of RNN. These are the benefits of LSTM given below:

1. LSTMs removes the gradient vanishing problem by preserving the error.
2. LSTM avoid the long-term dependency problem.LSTM can Remember information for long periods of time.

Chapter 4

Methodology

4.1 Method

A language model that uses neural networks to resolve the dimensionality problem with a distributed representation system of text data is called a neural language model[1].

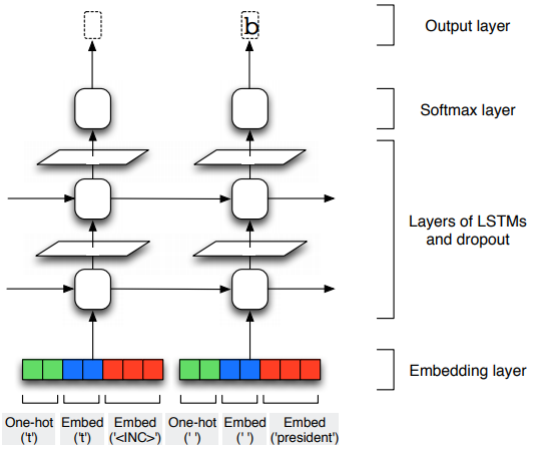


Figure 4.1: Architecture of our language model

In the figure:4.1 Green cells mean encoded vectors of characters, blue cells mean character embedded vectors, and red cells mean word embedded vectors. For a query q with texts t_1, ..., t_i, t_{i+1}, ..., t_m where t_1, ..., t_i is the prefix is r, we set query likelihood p(q|r) by a query language model defined as

[1]https://github.com/ZARIFREZAUL/QUERY-SUGGESTION-PIPILIKA/

$$p(q|r) = p\Big(t_{i+1}, ..., t_m | t_1, ..., t_i\Big) = \prod_{j=1}^{m} p(t_j | t_1, ..., t_{j-1})$$

where, each text t_k is a word or a character.

Before the Training process,

1. The full text corpus is preprocessed by removing unnecessary characters like punctuation and other characters.

2. All the words of the full corpus are tokenized as word level and stored in a vocabulary file for training process.

In Neural network architecture, we map each word to the concatenated vector. The vector is passed to two layers of LSTMs, and the output characters probability is estimated by applying a softmax function to the output of the final LSTM layer:

$$p(t_{j+1} = c | t_1, ..., t_j) = \frac{\exp(h_j * w_c^{out} + b_{out}^c)}{\sum_{c' \in V} \exp(h_j * w_{c'}^{out} + b_{out}^{c'})}$$

where h_j is a hidden vector from every LSTM cell, $w_c{}^{out}$ is an output weight vectors column for c, $b_c{}^{out}$ is an output bias vectors value for c, and V is a set of all unique characters. To prevent over-fitting, we employ a dropout layer to the output of each LSTM cell.

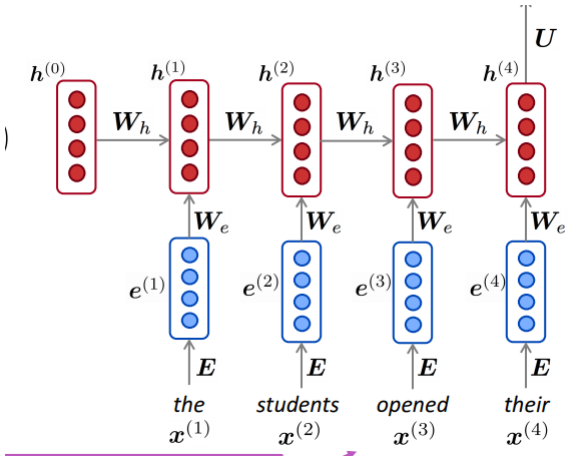


Figure 4.2: RNN Visualization

After training process complete, the weights of the corresponding vocab are saved on a file.

The model outputs the top n highest probability words for the user to choose from the model. After the process we have applied word embedding model onto our model to get more efficient suggestion of users query. For each suggested word my our model we fetched the n most similar word from the Word2Vec model . It is used to ensure that if a synonym or equivalent query is absent in our corpus and hence it will be absent in our RNN model then user will also get all the equivalent choices and so that will fulfill users wish and maximize users searching experience in our search engine

4.2 Experimental Description:

4.2.1 Experimental Setup:

This is our configuration of our model:

- **rnn_layers:** 2
- **word_level:** true
- **rnn_size:** 128
- **dim_embeddings:** 100
- **max_length:** 50
- **max_words:** 100000000

4.2.2 Experimental Tools:

We have worked mainly with the following python libraries:

- Keras
- Tensorflow
- Gensim
- Sklearn
- Numpy

Chapter 5

Result and Analysis

5.1 Result

Figure 5.1: Snapshots of the practical implementation (1)

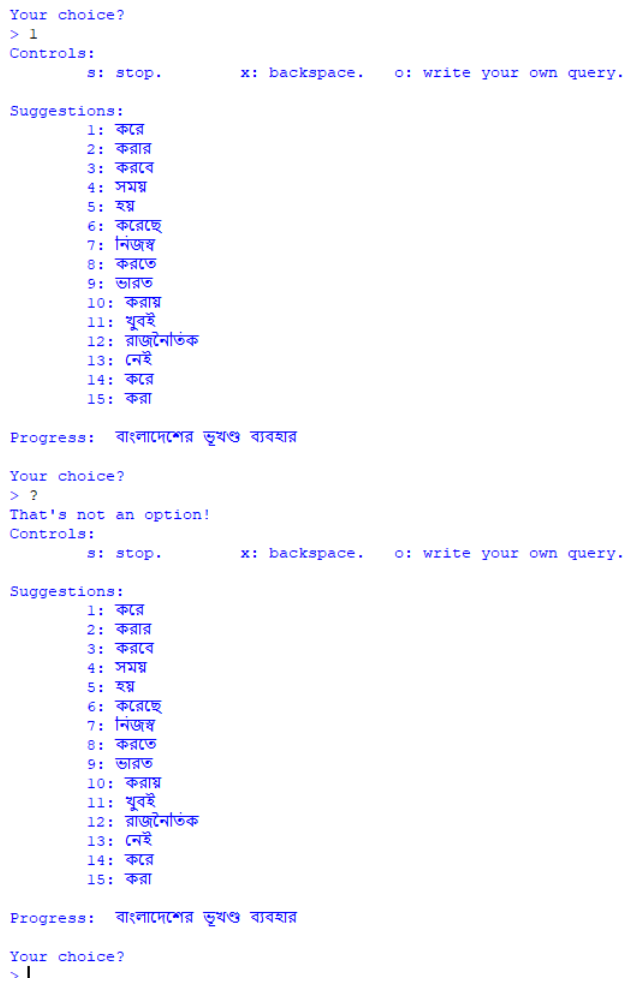


Figure 5.2: Snapshots of the practical implementation (2)

In results, we have the expected results, our model is able to predict the top most relevant suggested queries.

5.2 Analysis:

We have trained a corpus consists of 63000 word sequence by our model.

- Number Of Epochs: 10
- Accuracy rate: 81%

After completing the training process Results are absolutely good, The suggestion results have been revised as expected.

Chapter 6

Conclusion

We proposed a neural network model that can fulfill users search experience by suggesting the users intended query dynamically. We have proposed this model in such a way that any size of data can be trained by this model and would be able to get significant results.

References

[1] Kim, Y., Jernite, Y., Sontag, D. and Rush, A.M., 2016, February. [Character-Aware Neural Language Models.] In AAAI (pp. 2741-2749).

[2] Merity, S., Keskar, N.S. and Socher, R., 2017. [Regularizing and optimizing LSTM language models.] arXiv preprint arXiv:1708.02182.

[3] Mikolov, T. and Zweig, G., 2012. [Context dependent recurrent neural network language model.] SLT, 12(234-239), p.8.

[4] Ziv Bar-Yossef and Naama Kraus. 2011. [Context-sensitive query auto-completion.] In Proceedings of the 20th international conference on World wide web. ACM, 107116.

[5] Yoshua Bengio, Rejean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. [A neural probabilistic language model.] Journal of machine learning research 3, Feb (2003), 11371155.

[6] Piotr Bojanowski, Armand Joulin, and Tomas Mikolov. 2015. [Alternative structures for character-level RNNs. arXiv preprint arXiv:1511.06303 (2015).]

[7] Jan A Botha and Phil Blunsom. 2014. Compositional Morphology for Word Representations and Language Modelling.. In ICML. 18991907.

[8] Fei Cai, Maarten de Rijke, and others. 2016. [A survey of query auto completion in information retrieval.] Foundations and Trendső in Information Retrieval 10, 4 (2016), 273363.

[9] Georey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. [Improving neural networks by preventing coadaptation of feature detectors.] arXiv preprint arXiv:1207.0580 (2012).

[10] Sepp Hochreiter and Jurgen Schmidhuber. 1997. [Long short-term memory.] Í Neural computation 9, 8 (1997), 17351780.

[11] Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2015. "Characteraware neural language models". arXiv preprint arXiv:1508.06615 (2015).

[12] Mai Lankinen, Hannes Heikinheimo, Pyry Takala, Tapani Raiko, and Juha Karhunen. 2016. "A Character-Word Compositional Neural Language Model for Finnish". arXiv preprint arXiv:1612.03266 (2016).

[13] Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Cernocky, and Sanjeev Khu-danpur. 2010. "Recurrent neural network based language model". In Interspeech, Vol. 2. 3.

[14] Toma's Mikolov, Stefan Kombrink, Luka's Burget, Jan Cernocky, and Sanjeev Khudanpur. 2011. "Extensions of recurrent neural network language model". In Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on. IEEE, 55285531.

[15] Bhaskar Mitra and Nick Craswell. 2015. "Query auto-completion for rare prefixes". In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. ACM, 1755-1758.

[16] Greg Pass, Abdur Chowdhury, and Cayley Torgeson. 2006. [A picture of search.] In InfoScale, Vol. 152. 1.

[17] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Gregoire Mesnil. 2014. "Learning semantic representations using convolutional neural networks for web search". In Proceedings of the 23rd International Conference on World Wide Web. ACM, 373-374.

[18] Milad Shokouhi. 2013. "Learning to personalize query auto-completion". In Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval. ACM, 103-112.

[19] Milad Shokouhi and Kira Radinsky. 2012. "Time-sensitive query auto-completion". In Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval. ACM, 601-610.

[20] Ilya Sutskever, James Martens, and E Hinton. 2011. "Generating text with recurrent neural networks". In Proceedings of the 28th International Conference on Machine Learning (ICML-11). 1017-1024.

[21] Idan Szpektor, Aristides Gionis, and Yoelle Maarek. 2011. "Improving recommendation for long-tail queries via templates". In Proceedings of the 20th international conference on World wide web. ACM, 47-56.

[22] Saul Vargas, Roi Blanco, and Peter Mika. 2016. "Term-by-term query auto-completion for mobile search". In Proceedings of the Ninth ACM International Conference on Web Search and Data Mining. ACM, 143-152.