TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
PULCHOWK CAMPUS

A

PROJECT REPORT

ON

KEY PHRASE DETECTION AND QUESTION GENERATION FROM TEXT USING MACHINE LEARNING

SUBMITTED BY:

AAYUSH LAMICHHANE (075BCT005)
BISHAL KATUWAL (075BCT028)
BISHANT BANIYA (075BCT030)
GOBIND PRASAD SAH (075BCT038)

SUBMITTED TO:

DEPARTMENT OF ELECTRONICS & COMPUTER ENGINEERING

May, 2022

Abstract

The end goal of any educational institution is the growth and development of students. This growth and development is marked by good results. A good result requires good answer but at the same time, required answers are obtained only if the right questions are asked. Generating a question paper is a task of great importance for any educational institute. Traditionally, the question making process has been manual, tedious and time consuming. This paper thus proposes a multi-stage neural model to ease question generation process. This system can be used to prepare subjective as well as objective questions automatically from the given documents. This model also proposes an automated scoring for objective answers.

Keywords: *Automation, Keyword Extraction, Question Generation, Evaluation, Randomization*

Contents

Abstract **ii**

Contents **iv**

List of Figures **v**

List of Tables **vi**

List of Abbreviations **vii**

1 Introduction **1**

1.1 Background 1
1.2 Problem statements 1
1.3 Objective 1
1.4 Scope 1
1.5 Limitations 2

2 Literature Review **3**

2.1 Related work 3
2.2 Related theory 4

3 Methodology **7**

3.1 Feasibility study 7
3.2 Requirement analysis 8
3.3 Data collection 8
3.4 System design 9
3.5 Testing 10

4 System design **11**

5 Results & Discussion **15**

6 Conclusions **16**

7 Limitations and Future enhancement **17**

References 17

List of Figures

4.1 Sequence to sequence model 11
4.2 Key phrase generation system 12
4.3 High Level Architecture of the system 13
4.4 Implementation of the system 14

List of Tables

3.1 Gantt Chart for work breakdown 8

List of Abbreviations

LSTM	Long Short Term Memory
biLSTM	Bidirectional Long Short Term Memory
AI	Artificial Intelligence
URL	Uniform Resource Locator
MCQ	Multiple Choice Questions
QoL	Quality of Life
QA	Question Answer
QTI	Question and Test Interoperability
seq2seq	Sequence to sequence
GRU	Gated Recurrent Unit
RNN	Recurrent Neural Networks
NLP	Natural Language Processing
SQuAD	Stanford Question Answering Dataset
BLEU	Bilingual Evaluation Understudy
ROUGE	Recall-Oriented Understudy for Gisting Evaluation

1. Introduction

Questions are the hallmark of quality learning. The key to a successful learning experience lies in asking the right questions. However, multiple cases have come to light where university questions aren't prepared with care resulting in straight up duplicate question sets (to previous years) or out of syllabus, incomprehensible questions. In order to solve such problems, AI solutions such as PrepAI and Quillionz exist which allow question setters to feed the Neural Network syllabus content, and thus generating a question set; however customization and personalization of the prepared questions isn't addressed. Our project fills these gaps. A neural network based pointer network is trained which extracts possible keywords from text, generates questions around it and allows full customization and personalization of question in accordance to the pre-set question paper standards. Hence, with this product, time is saved and human bias is deterred all the while retaining personalization and customization.

1.1 Background

Making question sets is a recurring task which costs recurring operational costs and time. And the question set might be prone to errors such as duplication from previous year, spelling and grammar error, human biases, out of syllabus question setting etc. In particular, the recent scandal of the BBA exam question set being duplicated resulted in inspiring a solution to this widespread problem faced by a lot of educational institutions.

1.2 Problem statements

- Lack of customizable AI question generator
- Lack of a weighted generation system
- Lack of fully randomized and automated system

1.3 Objective

- To implement an AI based question set generator with a couple ways of inputting content(syllabus content or question bank) which generates both subjective and objective question sets along with answers.

1.4 Scope

The scope of this project extends to:

- It provdies an automatic, reliable and unbiased questions for evaluation
- It helps students in self-evaluation
- It separates the key theme of any excerpt
- It aids in note keeping

1.5 Limitations

The project doesn't extend to:

- Languages other than English language, Nepali and other languages aren't supported.
- Implied answers, only explicit answer and key phrases are supported.

2. Literature Review

Although, some AI based question generation exist, Nepal and Nepali institution still heavily rely on manual question generation process. Manual question generation is tedious and complex. Thus, a simple and efficient way for generating question papers is automatic question paper generation system. Automatic system not only automates generation but also recognizes patterns and avoids repetition. The question paper generation system can use different mathematical models. One of the most common method is template based algorithm which has predefined template and just plugs the important phrases in the said template. This method cannot handle the complexity of natural language. Thus, an AI based smart question generation system is required.

2.1 Related work

There are currently two large projects similar to this project.They are described briefly below.

A **PrepAI:**

PrepAI is a question generator that generates tests based on AI. It accepts custom texts, PDFs, docs, youtube URLs and Wikipedia content as inputs. From the provided input, it can generate MCQs (single correct/ multiple correct), descriptive questions, statement based questions, fillups and true/false questions which are divided into three difficulty levels i.e. easy, medium and hard. The questions are then exported in three possible output formats i.r. PDF, docs, xlsx. PrepAI also provides some QoL options such as:

- Option swapping
- QA editing
- QA Rating
- Merging papers

It simply provides enough tools such that no additional programs need to be used in the question making process.

B **Quillionz:**

Quillionz is another program that generate tests based on AI. It can generate MCQs, recall, true/false questions, short answers, and fill-in type questions. In addition it also allows the user to create notes. Input content can be provided through text or pdf format. It also uses express mode to generate questions rapidly. The most impressive part is the varieties of formats the questions can be exported to. The questions can be exported to popular formats like .pdf, .txt, .doc and QTI. It has added support for viewing question context.[1]

2.2 Related theory

A **Sequence to sequence model:**

Sequence to Sequence (seq2seq) models is a special class of Recurrent Neural Network architectures that are used to solve complex Language problems like Machine Translation, Question Answering, creating Chatbots, Text Summarization, etc. The most common architecture used to build Seq2Seq models is Encoder-Decoder architecture. Both the encoder and the decoder are LSTM models (or sometimes GRU models). Encoder reads the input sequence and summarizes the information in something called the context vector. The decoder is an LSTM whose initial states are initialized to the final states of the Encoder LSTM.

B **Pointer Network:**

Pointer network is a neural architecture that learns the conditional probability of an output sequence with elements that are discrete tokens corresponding to positions in an input sequence. Pointer networks can be used to learn approximate solutions to three challenging geometric problems – finding planar convex hulls, computing Delaunay triangulations, and the planar Travelling Salesman Problem – using training examples alone. Ptr-Nets not only improve over sequence-to-sequence with input attention, but also allow us to generalize to variable size output dictionaries.

C **LSTM:**

Long short-term memory (LSTM) is an artificial neural network which, unlike standard feedforward neural networks, has feedback connections. It can process not only single data points (such as images), but also entire sequences of data (such as speech or video). LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. LSTMs were developed to deal with the vanishing gradient problem that can be encountered when training traditional RNNs. Relative insensitivity to gap length is an advantage of LSTM over RNNs, hidden Markov models and other sequence learning methods in numerous applications.

D **biLSTM:**

A Bidirectional LSTM, or biLSTM, is a sequence processing model that consists of two LSTMs: one taking the input in a forward direction, and the other in a backwards direction. BiLSTMs effectively increase the amount of information available to the network, improving the context available to the algorithm (e.g. knowing what words immediately follow and precede a word in a sentence). BiLSTM adds one more LSTM layer, which reverses the direction of information flow. Briefly, it means that the input sequence flows backward in the additional LSTM layer. Then we combine the outputs from both LSTM layers in several ways, such as average, sum, multiplication, or concatenation.

E **word2vec**

Word2vec is a technique for NLP which uses a neural network model to learn word associations from a large corpus of text. Once trained, such a model can detect synonymous words or suggest additional words for a partial sentence. As the name implies, word2vec represents each distinct word with a particular list of numbers called a vector. The vectors are chosen carefully such that a simple mathematical function (the cosine similarity between the vectors) indicates the level of semantic similarity between the words represented by those vectors. Word2vec is a group of related models that are used to produce word embeddings. These models are shallow, two-layer neural networks that are trained to reconstruct linguistic contexts of words. Word2vec takes as its input a large corpus of text and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located close to one another in the space.[2]

F **Attention Mechanism:**

The attention mechanism was introduced to improve the performance of the encoder-decoder model for machine translation. The idea behind the attention mechanism was to permit the decoder to utilize the most relevant parts of the input sequence in a flexible manner, by a weighted combination of all of the encoded input vectors, with the most relevant vectors being attributed the highest weights.[3]

G **Sequence Models**

Sequence models are the machine learning models that input or output sequences of data. Sequential data includes text streams, audio clips, video clips, time-series data and etc. Recurrent Neural Networks (RNNs) is a popular algorithm used in sequence models. Applications of Sequence models: speech recognition, sentiment classification, video activity recognition.

3. Methodology

This proposal proposes an NLP-based question generator solution that assists teachers and educators in creating test papers in less time. For this, following feasibility study has been done. Also given system is to be created.

3.1 Feasibility study

- Technical:
  Training of the neural network is to be done in local PC and in google collab. The final application is lightweight enough to be run in a standard server. The software used are free and mostly open source. Similar systems have been developed. Task is divided between members as such: 2 members for backend, 1 member for ai, 1 member for frontend.

- Financial:
  This project isn't funded. However, this project could be launched as a SaaS product if mentioned objectives are met.

- Operational:
  This project could be widespread in education institutions where question setting practices are in disarray. It can also be used by teachers who want to save time and recurring operational costs. It can also be used by those who want to remove human bias while preparing questions.

- Schedule:

		DETAILS											
PHASE			Phase I		Phase II			Phase III				Phase IV	
	PROJECT MONTH		MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	JAN	FEB	MAR
1	Project Conception and Initiation	- Project Charter											
		- Plan Review											
2	Project Definition and Planning	- Scope and Goal Setting											
		- Work Breakdown Schedule											
		- Gantt Chart											
		- Communication Plan											
3	Project Execution	- Coding											
		- Debugging											
		- Integration											
		- Testing											
4	Project Performance & Control	- Objective Execution											
		- Quality Deliverables											
		- Effort and Cost Tracking											
		- Performance											
5	Project Close	- Deploy											
		- Project Punchlist											
		- Report											

Table 3.1: Gantt Chart for work breakdown

3.2 Requirement analysis

The requirements of the project were selected based on the limitations of the existing systems and the brainstorming based on the first hand experience of unsystematic handling of this field (for instance: BBA sixth semester examination mess). As per the requirement for project goes, following things are required:

- Basic understanding of NLPs
- Python
- Deployment
- Git
- Webapp development

3.3 Data collection

For key phrase detection, pretrained word embeddings generated using a word2vec trained on the English Gigaword 5 corpus is to be used. The model is to be trained on SQuAD. SquAD has over 100k question-answer pairs based on 536 Wikipedia articles. However the test split of SQuAD is hidden from the public. Thus we only train on fraction of the QAs.

3.4 System design

This system can be divided into three distinct sections:

1. Key Phrase Generation
2. Question Generation
3. Question Customization

1. Key Phrase Generation:

   The first step in question generation is to extract important points or key phrases from an excerpt. Studies by SQuAD show that over 50% of answers are entities. [4] Thus, it is tempting to use entity tagging (spaCy2) followed by neural network based entity selection . However not all key phrases are entities also entity tagger may miss some entities. Thus a neural network from scratch is needed to extract all key phrases. This model is based on sequence to sequence model[5]. This model is trained from start to finish of a document and it retrieves the starting and end points of all key phrases. Document is first encoded into a sequence of annotation vectors . A decoder(LSTM)is then trained to point to all of the start and end locations of answers in the document. The decoder is conditioned on the annotation vectors, via an attention mechanism[3]. Basically, this network selects all important answers(greedy) and then processes for redundancy in post-processing When all key phrases are extracted, the model then reaches termination.

   After entering a relevant input, the highly important aspect of question generation is to identify which part of the given input is more important from the view point of questioning. According to existing studies key phrase extraction can be formulated as a two step process, first, using lexical features to extract key phrase candidate list.[6] In the second step, ranking models are often used to select a key phrase. For key phrase detection, this model can be pretrained using word embedding generated using word2vec extension. LSTMs as encoders and decoders in the pointer network model.

2. Question Generation:

   The second phase is to generate questions from answer vector. It is again based on sequence to sequence framework and attention mechanism. It takes the document and answer as input and returns questions as output. In question generation, the decoder vocabulary is used. The dimensionality of the character representation, the number of hidden units for both encoder/decoder is set. Since each key phrase is itself a multi -word unit, we believe that a word-level that considers an entire key phrase as a single unit is not well suited to evaluate these models.[7]. While key phrase extraction has a fairly well defined quantitative evaluation metric, evaluating generated text as in question generation is a harder problem. We can use automatic evaluation metric such as BLEU, ROUGE, METEOR or CIDEr or perform human evaluation on generated questions with the answer key phrases[7]

3. Question Customization:
   The third step is simply question customization. This involves the personalization of questions. In addition to question generated by Network pointers, it allows manual insertion, deletion and alteration of questions. It is a simple manual step that requires no machine learning or training.

3.5 Testing

Testing is to be done on SQuAD. SquAD has over 100k question-answer pairs based on 536 Wikipedia articles However the test split of SQuAD is hidden from the public. Thus we only test on fraction of the QAs.

4. System design

This system can be divided into three distinct sections:

1. Key Phrase Generation
2. Question Generation
3. Question Customization

1. Key Phrase Generation:
   The first step in question generation is to extract important points or key phrases from an excerpt. Studies by SQuAD show that over 50% of answers are entities. [4] Thus, it is tempting to use entity tagging (spaCy2) followed by neural network based entity selection . However not all key phrases are entities also entity tagger may miss some entities. Thus a neural network from scratch is needed to extract all key phrases. This model is based on sequence to sequence model[5]. This model is trained from start to

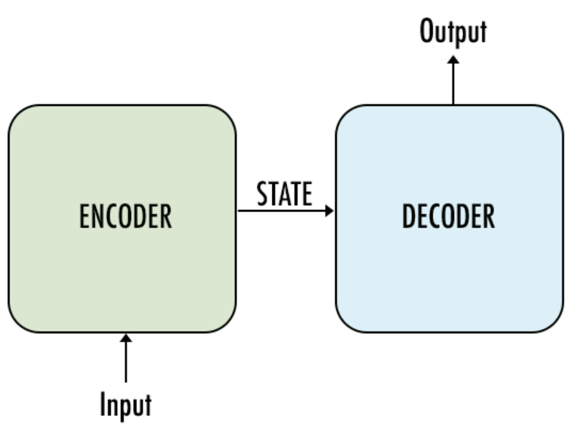

Figure 4.1: Sequence to sequence model

finish of a document and it retrieves the starting and end points of all key phrases. Document is first encoded into a sequence of annotation vectors . A decoder(LSTM)is then trained to point to all of the start and end locations of answers in the document. The decoder is conditioned on the annotation vectors, via an attention mechanism[3]. Basically, this network selects all important answers(greedy) and then processes for redundancy in post-processing When all key phrases are extracted, the model then reaches termination.

After entering a relevant input, the highly important aspect of question generation is to identify which part of the given input is more important from the view point of questioning. According to existing studies key phrase extraction can be formulated as a two step process, first, using lexical features to extract key phrase candidate list.[6] In the second step, ranking models are often used to select a key phrase. For key phrase detection, this model can be pretrained using word embedding generated using word2vec extension. LSTMs as encoders and decoders in the pointer network model. [8]

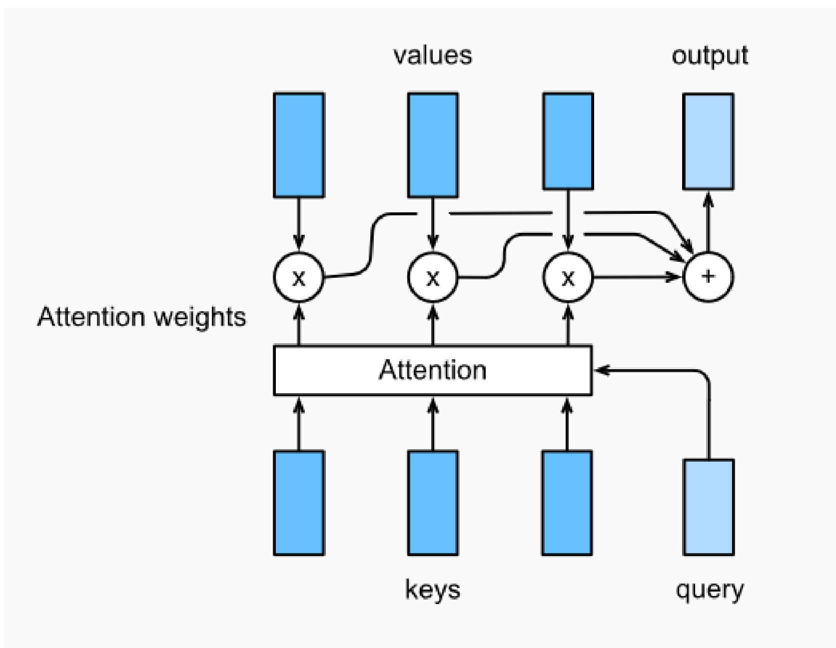

Figure 4.2: Key phrase generation system

2. Question Generation:
   The second phase is to generate questions from answer vector. It is again based on sequence to sequence framework and attention mechanism. It takes the document and answer as input and returns questions as output. In question generation, the decoder vocabulary is used. The dimensionality of the character representation, the number of hidden units for both encoder/decoder is set. Since each key phrase is itself a multi -word unit, we believe that a word-level that considers an entire key phrase as a single unit is not well suited to evaluate these models.[7]. While key phrase extraction has a fairly well defined quantitative evaluation metric, evaluating generated text as in

question generation is a harder problem. We can use automatic evaluation metric such as BLEU, ROUGE, METEOR or CIDEr or perform human evaluation on generated questions with the answer key phrases[7]

3. Question Customization:
   The third step is simply question customization. This involves the personalization of questions. In addition to question generated by Network pointers, it allows manual insertion, deletion and alteration of questions. It is a simple manual step that requires no machine learning or training.

Following figures represent the different system diagrams.

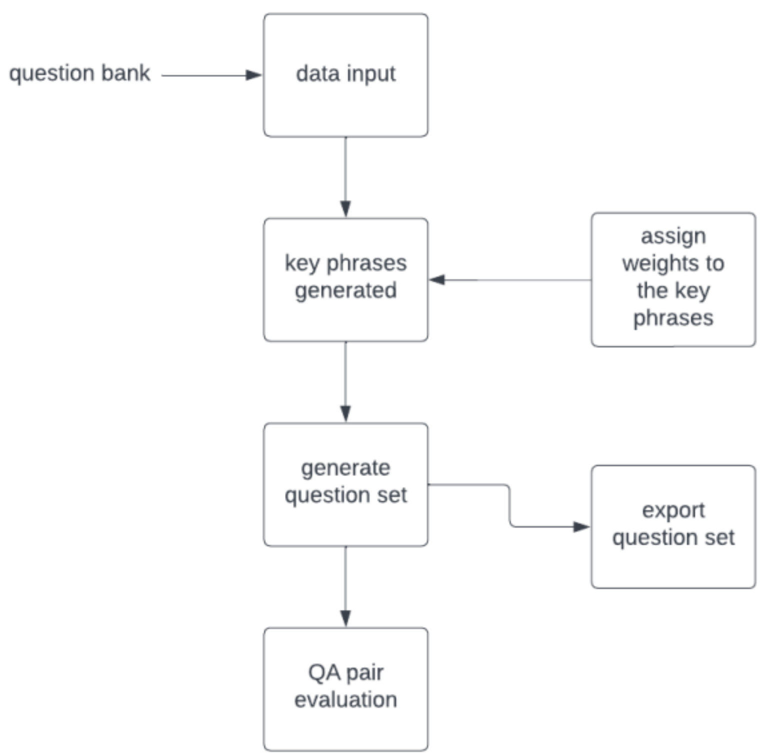

Figure 4.3: High Level Architecture of the system

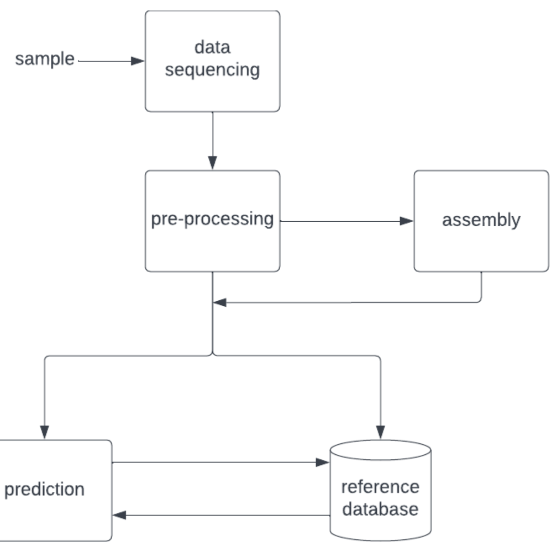

Figure 4.4: Implementation of the system

5. Results & Discussion

A neural network capable of generating questions(and answers) based on a given document. The questions can be either subjective or objective. System that generates customizable questions (and answers) which: allows for weights(for question generation) to be given to key phrases allows selection of number of questions from each chapters of the book allows for editing and addition of questions and answers after the neural net generates them A system that generates new questions based on question bank. This system is also customizable as weightage to particular key phrase can be given for influencing the selection of the question from the question bank.

Existing systems lacked customization and personalization for the local needs, and this project intends to fill those gaps. Initially we wanted to train the model using Nepali dataset of questions and answers to make the project even more intended to solve the problems of the locality. However, due to lack of relevant dataset on the topic, the idea of the project had to be pivoted. We managed to fulfill the user requirement of the locality by adding customizable questions (with answers) generation and highly customized question generation from past question bank. This project can be the next revolution in the Edtech industry providing a Test Preparation platform where tests are prepared not by humans but by AI.

6. Conclusions

Making questions by hand is time consuming, incurs recurrent operational costs and is prone to human bias. This project expands the features of existing systems making them more personalizable and customizable based on the needs of the locality.

7. Limitations and Future enhancement

This project will be useful to teachers who want to generate quality question sets in minutes. We however, only support the English language. This is due to the fact that finding relevant training dataset for Nepali language is nigh impossible. The project doesn't extend to:

- Languages other than English language, Nepali and other languages aren't supported.
- Implied answers, only explicit answer and key phrases are supported.
- Poems and phrases with hidden meanings aren't supported.

For future, the project can extend to:

- Nepali language support
- Cross-language compatibility
- Support for figurative speech

References

[1] Kavika Roy. Prepai vs. quillionz: Which is the best ai-based question generation platform? PrepAI vs. Quillionz: Which is the Best AI-Based Question Generation Platform? — by Kavika Roy — DataToBiz — Medium url = https://medium.com/datatobiz/prepai-vs-quillionz-which-is-the-best-ai-based-question-generation-platform-93495b034e39, october 2021.

[2] Xin Rong. word2vec parameter learning explained, 2014.

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[4] Chidozie Nwafor and Onyenwe Ikechukwu E. An automated mulitple-choice question generation using natural language processing techniques, 2021.

[5] Jay Alamar. Visualizing a neural machine translation model (mechanics of seq2seq models with attention. https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/, May 2018.

[6] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. Enhanced LSTM for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2017.

[7] Sandeep Subramanian, Tong Wang, Xingdi Yuan, Saizheng Zhang, Yoshua Bengio, and Adam Trischler. Neural models for key phrase detection and question generation, 2017.

[8] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks, 2015.