Automated methods for Question-Answering in Icelandic

Vésteinn Snæbjarnarson

Faculty of Industrial Engineering, Mechanical Engineering and Computer Science
University of Iceland
2021

AUTOMATED METHODS FOR QUESTION-ANSWERING IN ICELANDIC

Vésteinn Snæbjarnarson

60 ECTS thesis submitted in partial fulfillment of a
Magister Scientiarum degree in Computer Science

Advisor
Hafsteinn Einarssson

Faculty Representative
Hrafn Loftsson

M.Sc. Committee
Hafsteinn Einarsson
Hjálmtýr Hafsteinsson

Faculty of Industrial Engineering, Mechanical
Engineering and Computer Science
School of Engineering and Natural Sciences
University of Iceland
Reykjavik, September 2021

Automated methods for Question-Answering in Icelandic
Question Answering in Icelandic
60 ECTS thesis submitted in partial fulfillment of a M.Sc. degree in Computer Science

Faculty of Industrial Engineering, Mechanical
Engineering and Computer Science
School of Engineering and Natural Sciences
University of Iceland
Tæknigarður - Dunhagi 5
107, Reykjavik, Reykjavik
Iceland

Telephone: 525 4700

Bibliographic information:
Vésteinn Snæbjarnarson, 2021, Automated methods for Question-Answering in Icelandic,
M.Sc. thesis, Faculty of Industrial Engineering, Mechanical
Engineering and Computer Science, University of Iceland.

ISBN 978-9935-25-031-5
ORCID 0000-0001-9995-6181

Printing: Háskólaprent, Fálkagata 2, 107 Reykjavík
Reykjavik, Iceland, September 2021

Dedication

To Helen

Abstract

Question Answering (QA) is the automated task of providing an answer to a question posed in human language. Whether through search engines or speech controlled home assistants it has become a tightly integrated part of many peoples' daily routine at work or home. In recent years, these methods have improved greatly for commonly spoken languages such as English. This can almost wholly be attributed to advances in sequence modeling using deep neural networks, an increase in computing power, and the creation of large data sets suitable for training.

In this thesis, such QA methods are described, implemented and evaluated for Icelandic. The methods applied are a statistical approach based on term frequency, a current standard practices approach using a neural language model for Icelandic and a modern variant using pre-encoded phrase lookup. A new QA corpus and Icelandic language models are also presented.

The result is a baseline for extractive QA in Icelandic, where an answer is highlighted in a single document or larger corpora. Finally, a cross-lingual extension of the phrase lookup method is investigated and adapted for Icelandic QA. In this system, questions can be asked in Icelandic and are answered with segments from the English Wikipedia. This system is then adapted to answer Icelandic questions in Icelandic using segments from the Icelandic Wikipedia, taking advantage of a bilingual language model.

Útdráttur

Verkefni spurningasvörunar felst í því að svara spurning settum fram á mannlegu máli með sjálfvirkum hætti. Notkun slíkra kerfa er orðin hluti af daglegu lífi margra sem reiða sig á leitarvélar og raddstýringu. Á undanförnum árum hefur þessum aðferðum fleygt fram fyrir algeng tungumál á borð við ensku. Því er að mestu að þakka byltingu í notkun djúpra tauganeta, aukins reikniafls og tilkomu stórra málheilda sem henta til þjálfunar á líkönum.

Í þessu verkefni eru slík spurningarsvörunarkerfi útfærð og metin fyrir íslensku. Kerfin byggja á tölfræðiupplýsingum, hefðbundnum tauganetaaðferðum og nýstárlegri aðferðum með forgreyptum textarunum til uppflettingar. Jafnframt erný málheild fyrir spurningasvörun kynnt ásamt mállíkönum fyrir íslensku.

Með þessu fæst grunnlína til viðmiðunar í frammistöðu á spurningasvörun fyrir íslensku þar sem svar er merkt inn í texta, bæði þegar leitað er að svari í einu tilteknu skjali og í „opinni“ leit í mörgum skjölum. Að lokum er kynnt aðferð til að útbúa þvermála spurningarsvörunarkefi. Hún er sannreynd með því að útbúa slíkt kerfi sem tekur við spurningum á íslensku en veitir svör fengin upp úr enska hluta Wikipedia alfræðiorðabókarinnar. Það kerfi er svo aðlagað svo unnt sé að svara spurningum á íslensku upp úr íslenska Wikipedia. Þetta er gert mögulegt með notkun á tvímála mállíkani.

Preface

During my final year of undergraduate mathematics studies, I worked on assembling and analyzing genomic sequencing data from odd little symbionts called lichens. This required some programming and know-how in learning to operate Linux servers, though it mostly involved a lot of data processing and introduced me to running jobs on large computing clusters. After graduation, I joined a startup developing software for language learning where I got acquainted with natural language processing.

In retrospect it has become clear to me that this prepared me quite well for the work described in this thesis and I like to think that there has been a, though perhaps somewhat tangled, thread in my career leading up to this point. It certainly did not hurt either that as I started to get interested in question answering, almost two years ago, that I joined an Icelandic language technology startup.

I enjoyed doing the work described in this thesis and I hope it shows.

Contents

List of Figures xiii

List of Tables xv

Acronyms xvii

Glossary xix

Acknowledgments xxi

1. Introduction 1
- 1.1. Thesis objective . . . 2
- 1.2. Software, datasets and models described in this thesis . . . 3

2. Datasets 5
- 2.1. The Icelandic Common Crawl Corpus . . . 6
  - 2.1.1. Common Crawl . . . 6
  - 2.1.2. Extraction . . . 7
  - 2.1.3. Biases in the data . . . 8
  - 2.1.4. Comparison to other methods and datasets . . . 8
- 2.2. Natural questions in Icelandic . . . 9
- 2.3. Synthetic questions and answers in Icelandic . . . 12
  - 2.3.1. Question generation using methods from machine translation . 12
  - 2.3.2. Using generative language models to create questions . . . 13
  - 2.3.3. Translating questions . . . 13
- 2.4. Trivia-style datasets . . . 16

3. IceBERT - An Icelandic Language model 19
- 3.1. Training data . . . 19
- 3.2. Model architecture . . . 20
  - 3.2.1. Neural networks . . . 21
  - 3.2.2. Language models and vocabulary . . . 22
  - 3.2.3. Attention . . . 23
  - 3.2.4. The Transformer . . . 24
  - 3.2.5. Masked Language Modeling . . . 25
- 3.3. Training IceBERT . . . 26

4. Extractive Document Level QA **27**
4.1. Fine tuning for QA . . . 27
4.1.1. Training objective . . . 27
4.1.2. QA performance metrics . . . 28
4.2. Extractive document level QA using SQuAD style NQiI . . . 28
4.2.1. Models trained on only one dataset without negatives . . . 29
4.2.2. Further fine-tuning on NQiI with warm models . . . 31

5. Open QA using a retriever **33**
5.1. Open QA using term frequencies . . . 33
5.1.1. Effectively using BM25 for Open QA in Icelandic . . . 34
5.1.2. BM25 Open QA results . . . 35
5.2. A retriever-reader Open QA system for Icelandic . . . 36
5.2.1. Setting up the retriever . . . 36
5.2.2. Retriever-reader results . . . 36

6. Dense Open QA **39**
6.1. Introduction to dense retrieval . . . 39
6.2. Cross-lingual QA between Icelandic and English . . . 40
6.2.1. Data . . . 41
6.2.2. Multilingual language model . . . 41
6.2.3. Cross-lingual extractive QA-model . . . 42
6.2.4. Training modifications . . . 43
6.2.5. Icelandic Questions and English Results . . . 43
6.3. End-to-end Open QA for Icelandic . . . 44

7. Conclusions and future work **47**
7.1. Conclusions and summary . . . 47
7.1.1. Datasets . . . 47
7.1.2. Language models . . . 47
7.1.3. Summary of QA methods . . . 48
7.2. Future work . . . 49

References **51**

A. Appendix **57**
A.1. Examples of translated questions . . . 57
A.2. Reading comprehension hyper parameters . . . 58
A.3. XLMR-ENIS hyperparameters . . . 58
A.4. Stop words excluded from BM25 . . . 59

List of Figures

1.1. Two kinds of extractive QA systems. 2

3.1. Simple neural network with one hidden layer 21

3.2. Attention in IceBERT 24

3.3. The Transformer architecture 25

List of Tables

2.1. IC3: Filtering steps and kept data 8

2.2. IC3 and IGC unique token comparison 8

2.3. Summary statistics for NQiI 10

2.4. Comparison of question word distribution in NQiI, the English portion of TyDi QA and SQuAD 11

2.5. Summary of successfully translated SQuAD and NewsQA 16

3.1. Texts used to train IceBERT 20

3.2. IceBERT masked token perplexity and loss over development sets . . 26

4.1. Accuracy for models adapted from IceBERT and RoBERTa-base without negatives 29

4.2. Accuracy for models trained on IceBERT with negatives 30

4.3. Continued training from warm QA-models on NQiI for 1 epoch 32

4.4. Continued training from warm QA-models on NQiI for 2 epochs . . . 32

5.1. BM25 only QA 36

5.2. BM25 + IceBERT-QA question answering 37

6.1. Performance for English, Icelandic and bilingual models adapted for QA using SQuAD and SQuAD-IS 42

6.2. Performance for cross-lingual reading comprehension models 42

6.3. Semi-open cross-lingual QA between Icelandic and English 43

6.4. Open cross-lingual QA between Icelandic and English 44

6.5. Semi-open QA for Icelandic 44

6.6. Open QA for Icelandic 44

7.1. Comparison of all QA system considered 48

A.1. Reading comprehension QA hyperparameters 58

A.2. XLMR-ENIS hyperparameters 58

Acronyms

BPE byte pair encoding

DRP Dense Representations of Phrases at Scale

EM exact match

IC3 The Icelandic Common Crawl Corpus
IGC The Icelandic Gigaword Corpus

MIPS maximum inner product search
MLM masked language model

NER named entity recognition
NLP natural language processing
NQiI Natural Questions in Icelandic

POS part of speech

RoBERTa A Robustly Optimized BERT Pretraining Approach

SQuAD Stanford Question Answering Dataset

TF-IDF term frequency—inverse document frequency
TLD top level domain

QA question answering

XLM-RoBERTa cross-lingual RoBERTA

Glossary

abstractive question answering system A question answering system that returns a generated answer to a question

answer span A contiguous sequence of characters within a document defined by a start and end location, used to label answers in extractive question answering.

BM25 An optimized TF-IDF method that mitigates the effect of very frequent words on scoring.

dense vector A dense vector contains mostly non-zero values.

downstream task When an already trained neural network is trained with a different learning objective.

exact match A metric used to evaluate question answering models, a character-wise perfect match between the found answer and the answer in the evaluation dataset is considered exact.

extractive question answering system A question answering system that returns a span in a document as an answer to a question.

FAISS Faiss is a library for efficient similarity search and clustering of dense vectors.

F1 score Metric used to evaluate question answering models. It rates a partial overlap between the found answer and the answer in the evaluation dataset.

Gettu betur corpus A collection of Icelandic questions and answers used in preparation for a national quiz competition.

greynirseq A sequence modeling library built around IceBERT and down stream tasks for Icelandic.

IS-NewsQA Icelandic machine translation of the NewsQA dataset.

IS-SQuAD Icelandic machine translation of the SQuAD dataset.

knowledge distillation When training a neural network, the same data can be fed through an already trained network and the similarity in prediction included in the training objective. This is used to teach a model to mimic another model's behavior.

mBART A BART model trained on 25 different languages suitable for adapting to machine translation models.

NewsQA NewsQA is a high quality question answer dataset created using articles from CNN.

NQiI Natural Questions in Icelandic, the first Icelandic Question answer dataset for extractive question answering.

language model A model that predicts words or word segments based on prior or surrounding content.

open question answering Open question answering, sometimes referred to as open domain question answering, is the task of answering a question without providing a certain document in which to search for the answer. In the extractive case, a large corpus of files such as Wikipedia may be searched.

reading comprehension An extractive question answering task where there is only a single provided document which is searched for the answer.

retriever-reader system A kind of extractive open question answering system where there is a retriever component that searches for relevant documents and a reader module that given a document, searches for the answer within it.

spanbert SpanBERT is a language model trained to predict spans of tokens, unlike e.g. BERT that predicts a single token.

SQuAD Classic question answering dataset from Stanford.

token An atomic segment that can be embedded in a language model, the token can represent one or more characters.

Transformer Neural network architecture that uses attention

Trivia corpus A collection of Icelandic questions and answers used in an online quiz game.

Acknowledgments

I am most thankful to my colleagues at Miðeind for their feedback and our discussions on all things concerning natural language processing, including the contents of this thesis. I am in particular grateful to Haukur Barri Símonarson for patiently introducing me to language models and neural networks in general. Haukur Barri was a collaborator when training and preparing data for the IceBERT model. I also extend my gratitude to Vilhjálmur Þorsteinsson for supporting us in chasing down various side projects while working on our main task of machine translation.

I am also grateful to the Icelandic Student Innovation fund for funding the creation of the NQiI dataset, without which this project would be lacking a lot, and the five students, Bergur Tareq Tamimi Einarsson, Hildur Bjarnadóttir, Ingibjörg Iða Auðunardóttir, Unnar Ingi Sæmundsson, and Helgi Valur Gunnarsson that in summer 2020 created questions and annotated answers. I also thank Jonathan Clark at Google and Akari Asai at the University of Washington for providing the code used to label answers.

I thank Prof. Dr.-Ing. Morris Riedel and his team for providing access to the DEEP super-computer at Forschungszentrum Jülich which has been of immense importance.

I would also like to thank the University of Iceland for awarding the University's Science and Innovation motivational award for the cross-lingual part of the thesis, recognizing the importance of keeping Icelandic relevant when natural language is increasingly used to control computerized systems.

I thank Ólafur Páll Geirsson for sharing the *Gettu betur* question answer corpus with me. I thank Jinhyuk Lee for his correspondence on my questions regarding his views on applying the DensePhrases approach to a cross-lingual setting.

I thank Hafsteinn Einarsson my advisor and friend for the collaboration and his valuable input into the work over the last year. I thank Hrafn Loftsson for carefully proofreading the thesis.

Finally, I would like to thank my family and friends for supporting me, in this as in other aspects of life.

1. Introduction

Question answering (QA) systems are computerized systems that, when given an input (a question) in natural language, provide an answer of some sort, and preferably a correct one. There are two types of QA system: *extractive* and *abstractive* (or *generative*) QA systems [1]. The abstractive systems may generate an answer which may not be found in any underlying text. The extractive systems, which are the focus of this thesis, locate a segment of text that hopefully contains an answer to the question posed.

QA systems are also categorized into *open domain* (sometimes referred to as simply *open* [2] or open-book systems) and *closed* QA systems. The open systems target many documents at once or use large databases or neural networks with embedded information. The *closed* methods target a single document at once and resemble traditional *reading comprehension* [3] tasks. This difference is shown graphically in Figure 1.1. For the reading comprehension systems, some specific part of the text is targeted, e.g. a paragraph or article which is provided as input along with the question. The system is then tasked with finding the answer or possibly indicating that no such answer can be found within the context provided. A labelled answer or *answer span* within a piece of text is a contiguous sequence of characters defined by its start and end locations.

In recent years, large neural networks trained on vast amounts of text, see e.g. [4], have been adapted for various downstream purposes using knowledge transfer. These models are trained on tasks such as predicting missing words. Such models are generally referred to as *pre-trained language models* or *foundation models* [5]. These serve as a common starting point for a variety of tasks in Natural Language Processing (NLP) and artificial intelligence, such as machine translation, summarization, classification, and QA. As part of this thesis, such language models are trained and then fine-tuned for QA.

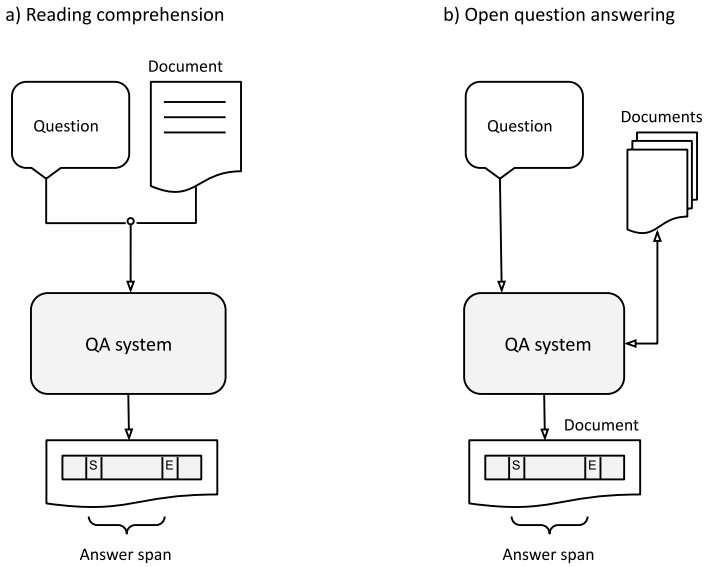

Figure 1.1: Two kinds of extractive QA systems.

1.1. Thesis objective

The aim of the thesis is to build, compare and set a baseline for Icelandic QA systems in order to answer the following questions:

(a) How can existing QA resources for English be used to develop QA systems for lower resource languages such as Icelandic?

(b) What are feasible approaches to augmenting existing QA data sets, and how does their usage affect the accuracy of QA systems?

(c) How do neural QA systems compare to classical term frequency systems in this data-scarce setup?

(d) Are cross-lingual open-domain QA systems feasible for low resource languages such as Icelandic?

(e) Can such systems be adapted to also answer questions in Icelandic?

1.2. Software, datasets and models described in this thesis

Along with this thesis, software, datasets and models are released[1]. These are published, and listed below, to encourage further work on the topic and enable reproducibility. The datasets and language models are also suited for other tasks in NLP. In particular, the bilingual language model should be of interest for those looking to take advantage of English datasets when developing or researching language technology for Icelandic.

(i) NQiI — Natural Questions in Icelandic, an Icelandic QA dataset, including train, test and development splits. Section 2.2.[2]

(ii) QAiI — a QA library for Icelandic, including a method to translate span-based datasets. Section 2.3.3.[3]

(iii) IceBERT — an Icelandic language model. Chapter 3.[4]

(iv) IceBERT-QA — an extractive QA model for Icelandic. Section 4.2.[5]

(v) Icelandic machine translations of QA datasets SQuAD 2.0 and NewsQA. Section 2.3.3.[6]

(vi) IC3 — The Icelandic Common Crawl corpus. Section 2.1.[7]

(vii) XLMR-EnIs — A cross-lingual English-Icelandic language model. Section 6.2.2.[8]

(iix) A cross-lingual DensePhrases model where questions can be asked in Icelandic and answered in English. Section 6.2.5.[9]

(x) An Icelandic DensePhrases model where Icelandic questions are answered in Icelandic. Section 6.3.[10]

[1] Supporting documents and links can be found at `https://vesteinn.is/qa/`.
[2] `https://repository.clarin.is/repository/xmlui/handle/20.500.12537/143`
[3] `https://github.com/vesteinn/qai`
[4] `https://huggingface.co/vesteinn/IceBERT`
[5] `https://huggingface.co/vesteinn/IceBERT-QA`
[6] `https://vesteinn.is/qa`
[7] `https://vesteinn.is/qa`
[8] `https://huggingface.co/vesteinn/XLMR-ENIS`
[9] `https://huggingface.co/vesteinn/open-qa-icelandic-english-densephrases`
[10] `https://huggingface.co/vesteinn/open-qa-icelandic-densephrases`

2. Datasets

At the core of any machine learning project lies a set of data that can be used for training. In NLP, this data is human language. Most of the available data is unlabeled (or "self labeled", since there is a lot that can be learned from it!) and in text form but only a negligible fraction of the total amount available has been labeled.

For some languages, there exists a lot of labeled data, but for others, less so. Icelandic is one of these languages where, until now, there existed no QA datasets with labeled answer spans. This kind of data is essential to evaluate extractive QA systems.

The following data, which is used in the next chapters to train and evaluate the language models and systems for QA, is presented in this chapter.

(i) Monolingual data in the form of the Icelandic Common Crawl Corpus (IC3) which is used to train the language models that are later converted into QA models.

(ii) QA data with labeled spans named Natural Questions in Icelandic (NQiI), the first such dataset for Icelandic.

(iii) Translations of English QA datasets, along with necessary post-processing methods.

(iv) External trivia-style datasets suitable for the evaluation of open domain QA in Icelandic.

2.1. The Icelandic Common Crawl Corpus

While large text corpora exist for English, for most of the world's languages they are hard to come by.[1] For Icelandic, there currently exists the Icelandic Gigagword Corpus (containing approximately 1.6 billion words) that contains texts collected with permission from news outlets and public institutions like courts and Parliament [6]. Other large swaths of texts exist in published books, but these are tightly held on to by the publishing industry. In this chapter, a different sort of text is collected, which is found on publicly available websites online. Models have been trained successfully on such text for English and other languages [7] using considerable resources. Here, Icelandic is targeted specifically in an efficient manner. The resulting dataset is highly heterogeneous in both style and origin.

Monolingual text corpora are key resources in language technology applications using statistical models and deep neural networks. The process described here should be applicable to other languages where datasets are hard to come by. The source of the data is the open internet, made accessible to those with relatively modest computing resources and disk storage through the targeted use of the Common Crawl datasets that comprise petabytes of data. Prior work has focused on the Common Crawl at large, e.g. [8, 7]. A key difference in the efforts described here is that the extraction is efficiently targeted and requires significantly lesser resources.

2.1.1. Common Crawl

The Common Crawl Foundation is a non-profit organization that scrapes large semi-random subsets of the internet regularly and hosts timestamped and compressed dumps of the web on Amazon Web Services. Each dump contains billions of web pages occupying hundreds of terabytes. Parsing these files directly requires storage and computing power not directly available to most and can come at a significant financial cost. The foundation also hosts indexes of URI's and their locations within the large zip files. While these indexes are also large, their processing is feasible with a few terabytes of storage. From 2008 until 2020, the foundation scraped the internet 79 times. Together, these dumps represent most of the text that has been gathered from the publicly facing internet in the last decade.

[1] Huggingface hosts a large collection of datasets at `https://huggingface.co/datasets`. At the time of writing there are 488 datasets available for English, almost ten times as many for the second most common language, Spanish.

2.1.2. Extraction

The Common Crawl indexes, which contain URI and byte offsets within the compressed dumps, are used to limit the data that needs to be searched for Icelandic texts. The Common Crawl Index Server has a public API where URIs can be queried based on attributes such as date, MIME-type and substring. To extract Icelandic, the `.is` pattern is targeted to match the Icelandic top level domain (TLD), resulting in 63.5 million retrieved pages with URIs and byte locations within the compressed Common Crawl dumps. Using the API also prevents one from having to fetch the massive index files. The main save on computational resources required to extract the data can be attributed to these steps.

By targeting only byte-offsets corresponding to the Icelandic TLD, it is possible to extract candidate websites that are high in Icelandic content. In total, the compressed content is 687 GiB on disk. All dumps since the start of the Common Crawl in 2008 until March 2020 were included.

The collected data is then sent through a process where plaintext is extracted from the WARC (Web Archive format) files using jusText [9][2] for removing boilerplate content and HTML tags.

Processing pipeline

The extracted WARC files are parsed to extract plaintext where Icelandic text is taken aside and duplicates removed. Since the `.is` TLD contains text in numerous languages the program fastText [10] is used to detect Icelandic. After processing the extracted files, 29GiB (4.2%) of text remains. Out of the 63.5 million pages retrieved, 18.6 million of them are marked as containing Icelandic.

Since the web is abundant with duplicate or near duplicate content the data is first deduplicated at the document level and then at the inter-sentence level by sliding a three-line window over the text. If any three consecutive lines have appeared together before they are discarded. The latter removes a fair amount of unwanted content such as cookie notifications and thumbnail text. After document deduplication, 8.6GB of text remains (1.3% of the original). After windowed deduplication 4.9GB of text remains in 2.2 million documents. A summary of the filtering steps taken is shown in Table 2.1.

[2]The implementation at `https://github.com/miso-belica/jusText` is used.

Table 2.1: IC3: Filtering steps and kept data.

Filtering step	Size	%
No filter	687 GB	100 %
IS and boilerplate	29 GB	4.2 %
Dedup. document	8.6 GB	1.3 %
Dedup. window	4.9 GB	0.71 %

2.1.3. Biases in the data

Since the Icelandic Common Crawl Corpus contains a wide range of texts from the open internet, it is bound to contain ethically questionable and factually wrong content. Any undertaking building on the corpus should take this into consideration. More subtle human biases present in the training data will also be introduced as has been shown for English [11].

2.1.4. Comparison to other methods and datasets

The methods outlined require significantly less computational resources than those used for e.g. the *Colossal Clean Crawled Corpus* (C4) [7] since only a fragment of the entire Common Crawl is processed. Since the Common Crawl dataset can not be re-hosted for licensing reasons, a cheap and effective extraction method is of high value. The extraction approach yields a comparable amount of Icelandic text as that found in the C4 corpus or approximately 10% more web pages at 2.2 million documents.

With the Icelandic Giga Corpus (IGC) of editorial text [6] containing around 9 GB of text and the new Icelandic Common Crawl Corpus (IC3) at 5 GB text, there is a large amount of Icelandic textual data available. The texts come from a broad spectrum of sources and are suitable for a wide range of language technology tasks.

Table 2.2: IC3 and IGC unique token comparison. Tokens are limited to those that occur at least four times and only contain characters from the Icelandic alphabet or hyphens.

IC3	IGC	IC3 ∩ IGC
1,155k	1,434k	818k

The total number of tokens is compared across IC3 and IGC in Table 2.2, after

removing those with less than five occurrences or containing characters not occurring in the Icelandic alphabet.[3] Almost one-third of the unique tokens (337k) in the IC3 are not present in IGC, and almost half of the IGC tokens (616k) are not present in IC3, confirming that there is a significant difference between the corpora.

2.2. Natural questions in Icelandic

Recognizing the necessity of developing a high quality dataset for Icelandic QA, we[4] applied for a grant to the Icelandic Center for Research which was later received. The dataset we created, with the help of five undergraduate students at Icelandic Universities, is called *Natural Questions in Iceland* (NQiI). We follow best practices [12] on creating questions that are phrased naturally and that belong in the same distribution as those that actual people might want answers to. This is to maximize performance in real world use cases for models trained on the data.

To create the QA corpus, we used custom software developed at the University of Washington which we received permission to use and modify [12]. The dataset was created by priming the students with the first hundred characters of Wikipedia articles and asking them to write down any questions that came to mind. At a later point the created questions were shown to a new group of students which were tasked with labelling potential answers in the top Wikipedia hit of a Google search for the given question. As of spring 2020, 18,378 questions have been made of which 5,405 have a labeled answer.

Methods for creating natural questions and answers

The creation of the dataset is based on the methods presented in work by Clark et al. [13] on typologically diverse languages that are recapped here for the sake of completeness. For more details and justifications, see section 3 in [13].

Question elicitation: Human annotators received the first hundred characters from an Icelandic Wikipedia article as a prompt. Based on the prompt, the annotator should write a question that they want to know the answer to and that the prompt does not answer. The prompt serves as an inspiration, and the questions do not need to have a strong connection to the prompt.

To create the prompts, we used a dump of the Icelandic Wikipedia from the 20th

[3]Without occurrence filtering there are 6.5M unique tokens in IGC and 6M unique tokens in IC3.
[4]Vésteinn Snæbjarnarson and Hafsteinn Einarsson.

of May 2020. We only selected articles with at least 250 characters, and we presented the prompts ordered by the length of the corresponding Wikipedia article in descending order.

Article retrieval: We perform a Google search for each question and select the top-ranked Icelandic Wikipedia page as a candidate that could contain the answer. We refer to these articles as passages.

Answer labeling: In a separate task, the annotators received question-passage pairs. The annotators could either label the passage as not containing an answer or select a paragraph containing the answer. If a paragraph was chosen, the annotator could decide if it was a yes/no question or provide a short minimal answer.

Software: We received permission to use and modify the software interface used in [12] to collect the data.

Summary statistics for the dataset

Five undergraduate students, all native speakers of Icelandic, were employed to create the dataset and their contribution summed up to nine months of work. The dataset contains 18k labeled question-answer pairs where the answers come from 1.4k unique Wikipedia articles. Summary statistics can be found in Table 2.3. In total, 13,740 questions were written. However, for 4,680 questions (34%), no article was found in the article retrieval step.

Table 2.3: Summary statistics for NQiI. Labeled pairs exceed those with an associated passage since different passages were targeted. N-way annotated means that N annotators all saw the same question and labelled an answer for it.

No. of questions written:	13,740
With an associated passage:	9,060
No. of labeled pairs:	18,378
With answer found:	5,405
With no answer found:	12,973
1-way annotated:	3,153
2-way annotated:	2,721
3-way annotated:	2,817
4-way annotated:	333

In Table 2.4, question types are compared between NQiI and the English development sets of TyDi QA (Typologically Diverse QA) and SQuAD (Stanford Question

Answering Dataset). It should be noted that question types are somewhat more evenly distributed than in the TyDi QA dataset. For each question type, we list the Icelandic wh-word at the beginning of the sentence. The *Other* category is composed of two sets of questions. One set contains questions that start with: "hve", "hvort", "hverjum", and "hverja", i.e. question forms not directly corresponding to the English ones listed. That set accounts for 0.6% of the total. The remaining questions did not start with any of the words listed and account for 2% of the total. Generally, these start with a verb (a typological difference, compared to English), and most of them are yes/no questions.

Table 2.4: Comparison of question word distribution in NQiI, the English portion of TyDi QA and SQuAD.

Question words	NQiI	TyDi QA	SQuAD
What	27%	30%	51%
Hvað	27%		
How	11%	19%	12%
Hvernig	5%		
Hversu	6%		
When	10%	14%	8%
Hvenær	10%		
Where	7%	14%	5%
Hvar	3%		
Hvert	3%		
Hvaðan	1%		
(Yes/No)	7%	10%	<1%
Er	5%		
Eru	1%		
Var	1%		
Who	20%	9%	11%
Hver	18%		
Hverjir	1%		
Hverjar	1%		
Which – Hvaða	12%	3%	5%
Why	3%	1%	2%
Af hverju	2%		
Hvers vegna	1%		
Other	3%		

2.3. Synthetic questions and answers in Icelandic

Since handcrafting and manually annotating datasets is time-consuming and expensive, several approaches have been taken to automate the process. Some rely on generative neural networks that, given some textual context, generate questions based on the context, possibly with the aid of pre-labeled named entities or otherwise words of high interest. Alternative modern QA generators are built using an unsupervised translation method where the model is trained to translate a text segment into a question. Finally, noisy text can be created using machine translation where existing datasets in one language are simply translated into the required language. Machine translation is not a magic bullet on its own, and some processing is required.

No direct question generation beyond translation was performed as part of this work. However, for the sake of completeness, other methods are summarized. These could be taken and may be used to further advance QA for Icelandic beyond the scope of this thesis.

(i) Questions extracted from the Icelandic Common Crawl corpus paired with cloze exercises (gap-filling) based on high interest filtered locations (similar targeting is done in [14]). These would then be used in an unsupervised translation task to generate questions.

(ii) A generative language model can be used to generate questions from content. See e.g. [15].

(iii) English questions and answers are translated to Icelandic using a neural machine translation system, as is demonstrated in this work.

2.3.1. Question generation using methods from machine translation

Lewis et al. [16] collected questions from large dumps of data collected from the open internet; around 100M questions were collected. Cloze exercises (a word is missing and the task is to find a good candidate word to fill in the blank space) are generated and an unsupervised translation method is used to translate them into questions. For high-interest cloze tasks, a named entity recognizer (NER) is used to target words of interest. A similar approach could be taken for Icelandic.

To investigate the feasibility of this approach, the Icelandic Common Crawl Corpus

was queried for questions. A simple regular expression pattern compatible with `grep` was created and tested to ensure that it would fit all trivia-style questions. The regular expression is given as follows

```
Hv|[A-ZÁÍÚÓÉÆÞ][^\s]*\ hv)[^?]*\?
```

The expression allows for Icelandic "wh-words" ("hv-words" in Icelandic) in the first or second word of a sentence and requires a question mark at the end. This approach only yields 253,870 unique questions. It is unclear whether that would be inhibitory to this matching approach. A NER-tagger has been trained on IceBERT and is publicly available through the `greynirseq`[5] package which can be used for labelling high-interest sections.

Furthermore, a method not explored by [16] from unsupervised summarization could be applied. In the training of the summarization model Pegasus [14], focus is given to locations that are of high importance to the overall text. ROUGE score and NER labeling are used to mask words of high importance, and the system is trained in generating questions based on these masked sequences. This approach could also be used for generating high interest questions.

2.3.2. Using generative language models to create questions

Large generative language models, such as GPT-3 [17], and T5 [7], can be used to generate questions from context without fine-tuning. Multilingual models can be used in some cases, but their performance in Icelandic is still lacking.[6] A new model would need training on Icelandic data of higher quality. This approach is of high importance for, e.g. DensePhrases [18] where a question is generated for all possible spans of interest for training. It is only a matter of time until models fit for this purpose have been trained for Icelandic.

2.3.3. Translating questions

Finally, a viable option for semi-synthetic question generation is to translate existing datasets using machine translation. Two things, besides the data, are required for such a solution to work

[5] See `https://github.com/mideind/GreynirSeq`.

[6] Some experiments were made to this end but the bad grammar of the output ruled out their use.

(i) A sufficiently good translation system

(ii) A method of mapping answer span annotations after translation

It just so happens that a large English-Icelandic translation model has recently been trained that shows good performance on translations from English to Icelandic.[7] The model is trained using a pretrained model *mBART* [20], a multilingual denoising autoencoder. Training data for the model, including backtranslations (monolingual data that is translated, with an earlier system, to create synthetic parallel data) was mostly prepared by my colleague Haukur Barri Símonarson at Icelandic language technology startup Miðeind ehf.[8] While formal evaluation of the translation quality has not yet been finished, the bilingual reader is invited to look at some translation examples in the appendix section A.1. This covers item **(i)** above.

What is left then is devising a method for mapping answer spans. There is no guarantee that a translated answer is found or even exist in the translated context. When the answer is translated on its own, it may be translated differently than when in context or the meaning may be corrupted completely. One would think this should be even more of an issue for Icelandic, where the context of the answer span will change its form, which does not happen when the answer is translated without the context.

To this end the Algorithm 1 is implemented and put to use to match answer spans in the translated text. Word alignments could also have been used, using e.g. [21] or [22], but the yields are so high that this heuristic method suffices. The algorithm uses direct matching of the translated answer, then the original answer and finally, if no hit has been found, fuzzy matching between the translated answer string and the answer context. The fuzzy matching is implemented using Levenshtein-distance.

Since the language model which is adapted for extractive QA is trained (see next section) on tokenized text, the translation output is tokenized to match this assumption. See appendix A.1 for some randomly selected examples of translated questions and answers.

Two datasets were chosen for translation, SQuAD which is a classic dataset for reading comprehension QA and NewsQA [23] a large high quality QA dataset based on news articles. Only 6,893 questions, 4.8% of the total data, were discarded from the SQuAD dataset using the translation method since an answer span could not be labeled. 11,478 questions, 9.6% of the total, were discarded from the NewsQA dataset. The amount of successfully translated questions, with NQiI numbers for

[7] This model was trained as part of the authors work at Miðeind ehf for the Icelandic Language Technology Programme. An improved version of the model used is made available for download at [19] .

[8] See `https://mideind.is`.

Algorithm 1 Answer-span finding in translated QA data

```
1:  procedure fuzzyMatch(answer, context, offset=0)
2:      words ← split(answer)
3:      n_words ← #(words) + offset
4:      n_context ← #(context)
5:      choices ← []
6:      for i ∈ [0, 1, ..., n_context − n_words] do
7:          choices.append(words[i : i + n_words])
8:      max ← argmax(LevenshteinScore(choices, answer))
9:      if LevenshteinScore(max, answer) > 0.9 then
10:         return context.index(max)
11:     return -1                                          ▷ No hit
12:
13: procedure answerSpan(context, answer, original_answer)
14:     if answer in context then
15:         start ← context.index(answer)
16:         return start
17:
18:     if original_answer in context then
19:         start ← context.index(original_answer)
20:         return start
21:
22:     for answ ∈ [answer, original_answer] do
23:         for offset ∈ [0, −1, 1] do
24:             start ← fuzzyMatch(answ, context, offset)
25:             if start then
26:                 return start
27:
28:     return -1                                          ▷ No hit
```

Table 2.5: Summary of successfully translated SQuAD and NewsQA questions with the processed NQiI for comparison.

Name	Subset	Questions	With answer
IS-SQuAD	Train	112,993	46,907
IS-SQuAD	Development	11,102	3,810
IS-SQuAD	Test	11,204	4,666
IS-NewsQA	Train	97,290	49,409
IS-NewsQA	Development	5,456	2,853
IS-NewsQA	Test	5,409	2,789
NQiI	Train	4,552	2,234
NQiI	Development	513	259
NQiI	Test	503	244

comparison, are shown in Table 2.5. It might be of interest to dissect the remaining data to further improve the algorithm used and gain a better understanding into what kind of questions are excluded, but this is left out for future work. Since the SQuAD dataset does not come with a released test set, a subset of the training data was taken aside from the training data for good measure.

2.4. Trivia-style datasets

QA datasets without answer contexts are also valuable resources, even if they can not be used for training in the same way as those with labeled answer spans. Their core value for this project lies in their usefulness for the evaluation of open QA systems. Two such resources have been collected for Icelandic and been made available.

Gettu betur corpus

In 2013, Ólafur Páll Geirsson, then a student at Reykjavik University, did a summer research project on QA for Icelandic [24] using a set of trivia style questions used in training for a competitive quiz show team. The system leverages term frequencies and implements modules based on three question types: persons, locations and years. He was kind enough to give access to the dataset used, a set of 4,569 questions with answers.

The dataset contains practice questions used by a team competing in "Gettu betur", an Icelandic quiz competition between junior colleges. They cover a wide variety of topics and are grouped into several categories that are discarded for this thesis's

purpose.

Icelandic trivia questions

A collection of community collected trivia-style questions is available on github[9], it contains 11,610 questions. The questions were used for an online quiz game and have various sources. The dataset is labeled both by categories and difficulty.

[9]The data is available at `https://github.com/sveinn-steinarsson/is-trivia-questions`.

3. IceBERT - An Icelandic Language model

General-purpose language models have in recent years proven incredibly powerful for fine-tuning on downstream tasks such as QA. Such a model, *IceBERT*, is trained on the Icelandic Common Crawl Corpus, the Icelandic Giga Word Corpus and a collection of other small corpora using the *RoBERTa* [25] implementation from `fairseq` [26]. IceBERT was trained by the author in collaboration with Haukur Barri Símonarson. As with all neural networks, it is trained by modifying its parameters incrementally, based on how well it can predict missing words or parts of words given a surrounding context. This simple yet powerful objective makes the model suitable for a wide array of downstream tasks where the model parameters are updated to solve more interesting problems.

These general-purpose language models are trained using a fairly simple objective such as predicting partially masked inputs using a vast amount of self labeling data (i.e. raw text). In doing so, some general knowledge of the language becomes embedded in the model. These models are then fine-tuned on downstream tasks such as QA, summarization and various kinds of classification tasks. During the pre-training phase, a general model of language structure seems to be learned. The performance on downstream tasks is greatly improved compared to randomly initialized models [4].

3.1. Training data

For a model to capture the variation found in natural language, it needs to be trained on a diverse and representative set of documents. To this extent, the datasets listed in Table 3.1 were cleaned up, split into validation, test and training sets and then tokenized and used for training. Having validation sets from each source proved to be particularly beneficial in monitoring performance by domain when training.

While the IGC [6] is the most extensive collection available of Icelandic text, it is somewhat homogeneous in style and genre. The IGC is mostly made up of news and

Table 3.1: Texts used to train IceBERT. Sports news, which are highly repetitive and homogenous, were removed from the IGC.

Dataset	Size	Tokens
Icelandic Gigaword Corpus v20.05 (IGC)	8.2 GB	1,388M
Icelandic Common Crawl Corpus (IC3)	4.9 GB	824M
Greynir News articles	456 MB	76M
Icelandic Sagas	9 MB	1.7M
Open Icelandic e-books (Rafbókavefurinn)	14 MB	2.6M
Data from the medical library of Landspitali	33 MB	5.2M
Student theses from Icelandic universities (Skemman)	2.2 GB	367M
Total	15.8 GB	2,664M

legal documents. Another drawback is its lack of text that has not been proofread, lack of informal texts and scant amount of literature and academic texts. To mitigate these drawbacks, several other sources were collected for pretraining. Unfortunately, a large collection of Icelandic literature is not available through legal means, but hopefully, the publishing industry will one day see the benefit of making enough data available for the training of models such as those discussed here.

The IC3 contains large amounts of text of many varieties, much of which is not proofread and serves well as a complement to the IGC. Academic texts found in the student thesis and data from the medical library of the University Hospital of Iceland were collected by Haukur Barri Símonarson and Pétur Orri Ragnarsson at Miðeind ehf. The academic texts were passed through a filter reminiscent of the one used for the IC3.

3.2. Model architecture

IceBERT is trained using the RoBERTa [25] implementation available in `fairseq` [26]. RoBERTa is a "robustly optimized BERT pretraining approach" where slight modifications have been made to the original BERT method as described in [4]. The most notable difference is that RoBERTa drops the next sentence prediction (NSP) objective and solely relies on the masked language model (MLM) objective. BERT is, in turn, based on the Transformer model architecture introduced in [27] with the addition of the training objectives. These model architectures are described briefly in the next section for completeness.

3.2.1. Neural networks

Artificial neural networks can be described with graphs such as the example shown in Figure 3.1. This particular example of a neural network[1] takes three values as *input*. The weighted sum of the input values is then passed through a nonlinear function called an activation function at each node in the *hidden layer*. Finally, there are two *output nodes* where a weighted sum is again passed through a nonlinear function. For each node and each leg in the network, there are learned parameters which control the parameters of the nonlinear functions.

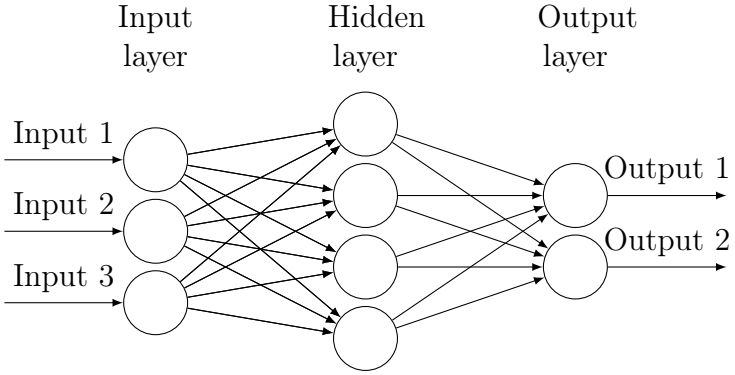

Figure 3.1: Simple neural network

The input is $x = (x_1, x_2, x_3)$, and the nodes in the hidden layer have parameters $h = (h_1, h_2, h_3, h_4)$. Between the nodes in the input layers and the hidden layers there are weights $w_{ij}, i \in [1, 2, 3], j \in [1, 2, 3, 4]$. Let $o = (o_1, o_2)$ be parameters of the output nodes, and between the nodes in the hidden layer and the output layer $z_{ij}, i \in [1, 2, 3, 4], j \in [1, 2]$. The activation function is denoted with σ, which takes as argument the input along with the node's parameters. To evaluate the output, the following calculation is then performed.

$$f(x) = \sigma\Big(z \cdot \sigma(w \cdot x, h), o\Big)$$

The network's parameters are updated using *back-propagation* after each forward pass of the model. This process is usually referred to as training the network. Back-propagation computes the gradient for each of the parameters in the network with

[1] In particular, this is a perceptron with a single hidden layer. There are many variants of neural networks, but these are the classical feed-forward networks usually thought of when artificial neural networks are mentioned. They were described as early as 1958 in [28] long before the computational power needed to apply them properly was available.

respect to the loss at training time. The loss is a measurement of the network's performance on the last set of training examples seen. For back-propagation to work, it is important to have a mostly differentiable activation function. Although the neural network used in IceBERT is much more complex than the one shown here, it is trained using the same principles.

3.2.2. Language models and vocabulary

A language model is a system, e.g. a trained neural network, that can assign probabilities to words or word segments in a sequence of text.[2] The models we are interested in here are neural and the input and output of these models are not only whole words or single characters but often parts of words. We refer to the atomic items of the input as *tokens*.

An auto-regressive language model is trained to maximize the probability of predicting the next token based on the earlier tokens, simply stated. Given a vocabulary Σ, the set of allowed tokens, we want to train a method that maximizes

$$p(x_t|x_{t-1}, x_{t-2}, ..., x_0) \text{ where } x_i \in \Sigma \text{ for } i \in [0, ..., t-1]$$

This kind of model is useful for generating text based on a sequence of prior tokens. Another kind of model is more suitable for determining the properties of tokens within a segment of text when given the entire context.

$$p(x_t|x_n, ..., x_{t+1}, x_{t-1}, x_{t-2}, ..., x_0) \text{ where } x_i \in \Sigma \text{ for } i \in [0, ..., n], i \neq t$$

It should be noted that the probability of the token is based on all the surrounding tokens in the sequence. This approach is referred to as masked language modeling, and we can intuitively think of it as hiding one token at a time and training the system to predict the missing token. Pretraining IceBERT is based on such an approach.

[2] This can be extended to images, audio, or anything that is otherwise encoded meaningfully into a sequence of symbols.

Byte Pair Encoding

The vocabulary used for IceBERT was created by feeding training data through an algorithm known as *byte pair encoding* (BPE). The method segments words into smaller constituents called subwords where more frequent sequences become their own tokens. For instance, the word "surrounding" could end up being broken up into "sur", "round" and "ing". Since the trained models use a fixed vocabulary with an embedding vector corresponding to each token, a smaller vocabulary means there are fewer parameters and thus a smaller model. Using subwords also means there is a higher coverage of the tokens in the training data, than for whole words on their own. It might also be beneficial that parts of words overlap in their embedding vectors. Finally, this enables predicting words that are composed of subwords, even if they were never seen during training.

The BPE algorithm can be described as follow: Each byte that occurs in a word in the training corpus is considered part of the vocabulary and thus considered a token. The two most frequent adjacent bytes are then merged into a token which is added to the vocabulary. The last step is repeated until the set vocabulary size is achieved. When the data is later prepared for training or inference, it is segmented until no tokens can be split into two adjacent tokens.

When the vocabulary is used at training time, an embedding matrix is generated where each token in the vocabulary is represented by a vector, most often of fixed width corresponding to the input dimensionality of a neural network. The initial values are randomly initialized in the beginning.

3.2.3. Attention

Attention is a mechanism by which a neural network is allowed to use its surrounding context as a reference point. It is different from the multi-layer perceptron shown in Section 3.1. When applying attention, *query*, *key* and *value* matrices are used to calculate contextual information. The names are justified by thinking of a) the query matrix as containing information about the current token, b) the key containing information about the token we are attending to with respect to the earlier token, and c) the value is the information associated with the relation between the two tokens. In matrix representation, this can be represented with the following formulas.

$$X \times W^Q = Q$$
$$X \times W^K = K$$

$$X \times W^V = V$$

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{Q \times K^T}{\sqrt{d_k}}) \cdot V$$

The parameters in the matrices W^Q, W^K and W^V are learned, and X is the embedding matrix. Note that we are doing matrix multiplication, with every row in X corresponding to one token in the input sequence. d_k is the dimension of the key vectors, and the division helps in stabilizing the gradients.

In Figure 3.2 we can see how attention for the word "kindur" (e. *sheep*) is mostly attending to itself and "bóndi" (e. *farmer*) in the sentence "Jón bóndi átti margar kindur." (e. *Jón farmer had many sheep.*). Each colored column corresponds to a given attention head and the strength of the color corresponds to a high query and key multiplier.

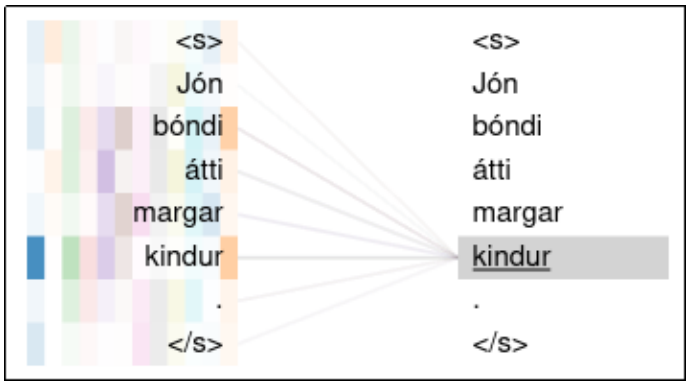

Figure 3.2: Example of attention in IceBERT. Each colored column corresponds to a given attention head and the strength of the color corresponds to a high query and key multiplier. The figure is created using the package `bertviz` *[29] .*

3.2.4. The Transformer

In the paper "Attention is all you need" [27], the transformer architecture is introduced. The paper introduces multiple attention heads allowing different query, key and value matrices to be learned to attend to different patterns in the input. It also introduces a sinusoidal positional encoding that is fed into the attention mechanism.

The typical Transformer architecture shown in Figure 3.3 is an encoder-decoder model of N transformer layers. The variable N is a hyper-parameter tuned to set the size of the model. The layers contain, besides the attention step, a residual connection and layer normalization step. The residual connection passes the non

attended input past the attention mitigating vanishing gradients, and the layer normalization step normalizes the activity to speed up training and prevent exploding gradients.

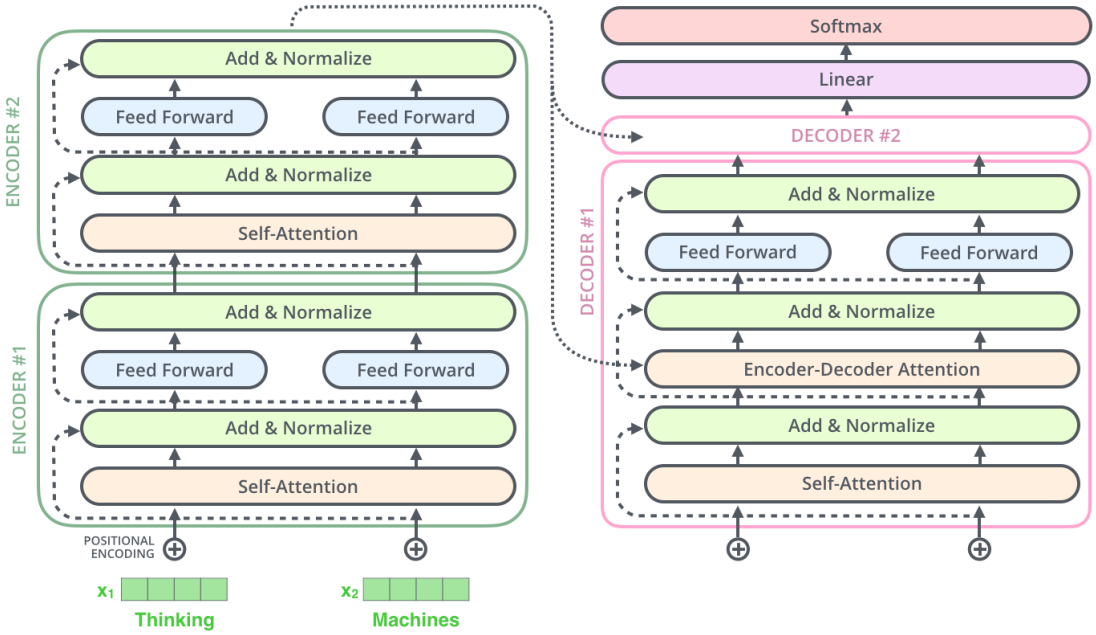

Figure 3.3: The Transformer architecture. Image taken from [30].

3.2.5. Masked Language Modeling

Masked language modeling is an objective used for training neural networks. One or more tokens are masked out in the input sequence and the difference between the true token and the predicted token embeddings are used to calculate the loss. As described earlier the loss is then used to calculate the gradients for each parameter's contribution to the loss and their values are adjusted accordingly.

In the original BERT paper [4], it was shown that models trained on this objective are well suited for fine-tuning on a wide variety of downstream tasks in NLP. More recent papers such as [31] have shown that the traditional NLP pipeline[3] is in many ways accounted for in the internals of these models.

[3]The tasks considered are labeling of part-of-speech, constituents, dependencies, entities, coreferences, semantic proto-roles, semantic roles and relations.

3.3. Training IceBERT

As mentioned earlier, the RoBERTa implementation used to train IceBERT mainly differs from the original BERT implementation in that it makes away with the next sentence prediction task and only uses the mask filling objective. It also tunes hyperparameters such as increasing batch size and, instead of using a character-level BPE vocabulary, uses the byte-level BPE vocabulary.

A base IceBERT-model with 12 layers was trained using 24 Nvidia V100 GPUs with 32GB RAM for about a week or 160k updates (57 epochs). An effective batch size of 2000 sentences was used following the RoBERTa hyperparameter setup.

Table 3.2: IceBERT masked token perplexity and loss over development sets.

Dataset	Best loss	PPL
Student theses from Icelandic universities (Skemman)	2.102	4.40
Data from medical library of Landspitali (Hirslan)	2.102	5.34
Icelandic Gigaword Corpus (IGC)	1.955	3.88
Icelandic Common Crawl Corpus (IC3)	2.102	4.74
Greynir News articles	1.975	3.93
Icelandic Sagas	2.102	8.20
Open Icelandic e-books (Rafbókavefurinn)	2.102	10.24

When fine-tuning IceBERT for part-of-speech (POS) tagging or named entity recognition (NER) using `greynirseq` it reaches state of the art performance. The trained models achieve 98.2% correct POS-labelling on the MIM-GOLD [32] dataset (excluding "x" and "e" labels) and 92.4 F1 on the MIM-GOLD-NER [33] dataset using ten-fold cross validation. These are almost 10% higher results than for a randomly initialized model[4] that is then fine tuned. This performance is a good indication of the quality of the model and is encouraging with regards to further training for QA.

[4]This was discovered when the path to IceBERT had been misspelled leading to the model being randomly initialized.

4. Extractive Document Level QA

Extractive document level question answering or *reading comprehension* is the task of returning a span or location in a document as an answer to a question. This is the more classical approach to QA systems as compared to open systems, where finding the correct source document is also part of the task. For English, the most studied (and recently showed to be highly flawed [34]) datasets for this kind of task are the Stanford Question Answering Datasets (SQuAD v1 and v2 [3]) where a paragraph, a question and an answer location are given, as well as whether the question has an answer within the paragraph.

This extractive kind of QA system can be trained on top of the language models described in the earlier chapter or built using straightforward statistical methods such as using term frequencies. In this chapter, such models are trained on top of the IceBERT model to evaluate performance, and the feasibility of using translated data to improve quality assessed.

4.1. Fine tuning for QA

4.1.1. Training objective

To adapt an existing language model and fine-tune it for QA the *language model head*[1] is removed from the model. The majority of the neural network is still there, but the remaining weights only output a dense vector referred to as an embedding. It is easy enough to construct a new classifier head that takes these embeddings as input and returns not one but two predicted locations that mark the predicted answer span. At first, their output is random, but by further training the model with a good enough dataset, they will quickly start returning sensible values.

[1] This is the final classification layer of the neural network that assigns probabilities to each token in the vocabulary.

4.1.2. QA performance metrics

The two most common metrics for QA are exact match (EM) and F1-score. These are calculated per question/answer pair and then averaged to give a total metric.

Exact Match (EM)

Exact match is a binary score, a 100% character match gives a score of 1, and any difference gives a score of 0. This performance metric is very strict and can result in low results for datasets where there are inconsistencies in the labeling of answer spans.

F1 score

The F1 score is more lenient than the exact match score and is commonly used for NLP tasks. It is calculated based on common words and calculated as follows:

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Where precision is the ratio of shared words against the prediction, and the recall the ratio of shared words against the correct answer span.

4.2. Extractive document level QA using SQuAD style NQiI

A part of the NQiI dataset can be mapped to the same format as SQuAD. Span locations within Wikipedia articles can be mapped to locations in paragraphs. Yes/no-questions and questions without answers were removed. Around 4000 questions remain and those were split into train, development and test sets. The base IceBERT model was fine-tuned on this dataset for QA using the Transformers library from Huggingface [35]. Results can be seen in Table 4.1.

The EM score is quite low and can be attributed to the small dataset size as well as the unfinished state of the early NQiI dataset. The F1 score is promising with a score around 31.8.

Since the curated datasets NQiI is quite small, it should be beneficial to warm up the QA-model using synthetic data. Using a warm models refers to continuing training using a model that has already seen some data. The translated questions from SQuAD and NewsQA could serve this purpose well. To ensure a valid comparison, between the performance over the original questions in English and the translated questions in Icelandic, only those successfully translated are included when the performance score is computed for RoBERTa-base.

Several models are trained using the datasets, the results are presented in Tables 4.1 and 4.2.

4.2.1. Models trained on only one dataset without negatives

A good QA system should be able to report that no answer can be found in a given paragraph. For completeness and to get further insight into the difference in the behavior of models trained over different datasets results where negative examples have been excluded are shown in Table 4.1. No warm-up of the models is done for these models. The translated NewsQA dataset seems to be of higher quality, which is perhaps not surprising given its original high quality and that the training data of the neural machine translation model contained lots of news text. The performance drop after translation is comparable for both SQuAD and NewsQA with a drop in F1-score of 5.1 and 6.5 respectively.

Table 4.1: Accuracy for models adapted from IceBERT and RoBERTa-base without negatives.

Model	Dataset	F1	Exact match
IceBERT-base	IS-SQuAD	24.0	20.0
IceBERT-base	IS-NewsQA	30.8	22.9
IceBERT-base	NQiI	31.8	10.5
RoBERTa-base	SQuAD	29.1	25.5
RoBERTa-base	NewsQA	37.3	30.8

In Table 4.2, negative examples are included. Models are trained for each of the three training sets and a combination of them all, they are then evaluated on all three datasets.

Training results for the English datasets adapted from RoBERTa-base are included for comparison. For SQuAD there is a drop in exact matches from 74.3 % to 70.6 %. For NewsQA there is a drop from 59.3 % to 55.7% exact match. Unsurprisingly, some performance is lost due to machine translation, but overall, it is not catastrophic in any way.

Merging the training datasets leads to some drop in performance for NQiI (69.5 to 67.7 F1), a small drop for IS-SQuAD (72.1 to 71.2 F1) but the IS-NewsQA datasets benefits (59.7 to 60.1 F1). Overall the model trained on all the data should have the best generalization performance, assuming the training sets are not detrimental to each others' performance. The model trained on all data does indeed show good performance on all datasets.

It is worth noting that the performance of the model trained on NQiI, when measured over the other datasets is much lower than for the other models. The F1 score over IS-SQuAD is only 28.2 as compared to 72.1 for the model trained on IS-SQuAD and 24.2 for IS-NewsQA compared to 63.4 for the model trained on IS-NewsQA, approximately a 60% lower score in both cases. This low performance can probably be attributed to the training set being much smaller than the other ones (see Table 2.5 for size comparison).

Table 4.2: Accuracy for models trained on IceBERT with negatives. Results for models adapted from RoBERTa-base using the original SQuAD and NewsQA datasets are included for comparison.

Training set	Evaluation set	With answer		No answer		All	
		F1	Exact	F1	Exact	F1	Exact
SQuAD (EN)	SQuAD (EN)	58.2	53.6	85.1	85.1	75.9	74.3
IS-SQuAD	IS-SQuAD	45.4	41.2	86.0	86.0	**72.1**	70.6
	IS-NewsQA	34.3	25.9	67.1	67.1	49.9	45.5
	NQiI	39.7	25.7	95.0	95.0	63.1	54.9
NewsQA (EN)	NewsQA (EN)	53.9	47.3	72.5	72.5	62.8	59.3
IS-NewsQA	IS-SQuAD	36.1	31.8	77.7	77.7	63.4	61.9
	IS-NewsQA	43.0	35.2	78.1	78.1	59.7	55.7
	NQiI	29.0	19.7	94.1	94.1	56.5	51.1
NQiI	IS-SQuAD	29.6	5.2	27.5	27.5	28.2	19.9
	IS-NewsQA	20.7	4.4	27.9	27.9	24.2	15.6
	NQiI	52.2	15.1	88.7	88.7	**69.5**	67.6
All	IS-SQuAD	44.7	40.1	85.0	85.0	71.2	69.6
	IS-NewsQA	43.4	35.1	78.4	78.4	**60.1**	55.8
	NQiI	44.8	37.1	83.5	83.5	67.7	64.6

These models were all trained for 4 epochs, and the hyper-parameters used can be found in the appendix under A.2.

The performance drop from translation is around 5% which is lower than one might expect it to be. This may be in some part due to the way the answer span finding algorithm filters the questions (5% for SQuAD and 10% lost for NewsQA).

It would be interesting to look at the set of questions which are answered correctly in English but incorrectly when translated to Icelandic. This might tell us something about the quality of the translations and what kind of questions are not as easily translated. Similarly, it would be interesting to investigate whether the "hard" questions in English are difficult when translated to Icelandic. A first step might be to train a binary classifier over the English data to predict performance post-translation. This digression is left out as future work.

4.2.2. Further fine-tuning on NQiI with warm models

Next, the benefits of further training the models exclusively on the NQiI dataset is considered. Given the translated origin of the IS-SQuAD and IS-NewsQA datasets, it would be natural to think that using original Icelandic text would improve performance. All models trained with negatives are further trained solely on the NQiI dataset for two more epochs, results are shown in Table 4.3. To assess the benefit of training even further, results after four epochs are shown in Table 4.4.

The performance gains are limited to the F1 score for NQiI (figures marked bold in the tables, deltas increase 5 to 15 points for the models not trained on NQiI) while the exact match drops, indicating that there may be some labeling noise due to annotator disagreement. The small size and unpolished state of the NQiI dataset are likely to blame here. The high quality of the translation model may also be narrowing the gap between using real and synthetic data.

There is a clear benefit to NQiI F1 performance when fine-tuning from the warm models, but only when evaluated for NQiI. Fine-tuning of the model trained on all training datasets improves F1 by 2.5 percentage points (5% relative gain) while still retaining decent performance on the translated datasets. Even so, the drop in EM in the range of 2%-16% makes it hard to conclude otherwise than that the overall performance has dropped. With the aforementioned labeling noise likely to blame to some degree.

Table 4.3: Continued training from warm QA-models on NQiI for 1 epoch. Datasets are still with negatives. The $\Delta F1$ and ΔEM values are comparisons to the performance of the original warm models.

Pretr.	Eval.	With answer		No answer		All		All	
		F1	EM	F1	EM	F1	EM	Δ F1	Δ EM
IS-SQuAD	IS-SQuAD	47.2	12.1	42.8	42.8	44.3	32.3	-27.8	-38.3
	IS-News	36.2	12.4	31.6	31.6	34.0	21.5	-15.9	-24.0
	NQiI	57.2	19.1	93.2	93.2	72.4	50.4	**9.3**	-4.5
IS-News	IS-SQuAD	38.6	10.7	47.8	47.8	44.6	35.0	-18.8	-26.9
	IS-News	43.9	19.0	43.0	43.0	43.5	30.5	-16.2	-25.2
	NQiI	56.0	19.4	91.9	91.9	71.2	50.0	**14.7**	-1.1
NQiI	IS-SQuAD	29.8	5.5	28.2	28.2	28.8	20.4	0.6	0.5
	IS-News	19.9	4.1	28.9	28.9	24.2	15.9	0.0	0.3
	NQiI	53.9	15.5	89.2	89.2	68.8	46.6	**-0.7**	-21.0
All	IS-SQuAD	50.7	39.6	74.2	74.2	66.2	62.3	-5.0	-7.3
	IS-News	46.1	35.7	72.3	72.3	58.6	53.2	-1.5	-2.6
	NQiI	59.3	16.8	91.9	91.9	73.0	48.5	**5.3**	-16.1

Table 4.4: Continued training from warm QA-models on NQiI for 2 epochs. Datasets are still with negatives. The $\Delta F1$ and ΔEM values are comparisons to the performance of the original warm models.

Pretr.	Eval.	With answer		No answer		All		All	
		F1	EM	F1	EM	F1	EM	Δ F1	Δ EM
IS-SQuAD	IS-SQuAD	46.3	10.6	39.5	39.5	41.9	29.6	-30.2	-41
	IS-News	34.4	10.3	33.8	33.8	34.1	21.5	-15.8	-24
	NQiI	58.2	17.4	93.7	93.7	73.2	49.6	**10.1**	-5.3
IS-News	IS-SQuAD	40.1	8.9	36.3	36.3	37.6	26.9	-25.8	-35.0
	IS-News	44.0	16.0	36.7	36.7	40.5	25.9	-19.2	-29.8
	NQiI	57.3	16.1	91.0	91.0	71.5	47.7	**15.0**	-3.4
NQiI	IS-SQuAD	29.2	5.2	30.7	30.7	30.2	21.9	2.0	2.0
	IS-News	20.7	4.2	34.1	34.1	27.1	18.5	2.9	2.9
	NQiI	53.1	16.4	92.3	92.3	69.7	48.5	**0.2**	-19.1
All	IS-SQuAD	50.9	37.0	72.2	72.2	64.9	60.1	-6.3	-9.5
	IS-News	45.9	34.9	71.9	71.9	58.3	52.6	-1.8	-3.2
	NQiI	58.9	19.7	92.8	92.8	73.2	50.6	**5.5**	-14.0

5. Open QA using a retriever

Open-domain QA, or simply open QA, is in many ways much more useful than the document level QA. In most real-life situations where an answer is sought, it is unknown in what paragraph or document it can be found. In Open QA, finding the *correct* paragraph or document is considered part of the task.[1]

These kind of QA systems require something more than an extractive QA-model for fine-tuning. It is infeasible to attend to all the text available. Open QA system (or information retrieval) system have traditionally been developed using a two-step process utilizing, e.g. *term frequency-inverse document frequency* (TF-IDF) that looks up relevant paragraphs based on term frequencies, a method at least considered since the early 70's [36]. This relatively simple system has proven quite useful and will be explored further in this chapter.

More recent systems use neural networks similar to the language models introduced in the earlier chapters to embed documents to match against a query or question. These can be cumbersome in usage but have been shown to improve performance over that of TF-IDF. In particular, ORQA [37], Dense Passage Retrieval [38], REALM [39], and DensePhrases [18] are worth mentioning. The main drawback has been that the very large search space needs to be re-embedded often at training time. This is a major issue, particularly during end-to-end training in models like ORQA [37] where the extractive model and retriever model are trained simultaneously and the indexed embeddings of the corpus become stale at (almost) every step. These approaches are explored further in the next chapter, while the following sections rely on TF-IDF.

5.1. Open QA using term frequencies

QA systems have traditionally been built without the use of neural networks. A common choice for pairing sentences and questions with documents for retrieval

[1] It should be noted that generative language models such as GPT-3 [17] can be thought of as open QA systems if primed for QA. They can answer questions directly without there being any clear source of the answer.

is TF-IDF (term-frequency inverse-document-frequency) and a modified version, BM25 [40]. BM25 is better because too frequent usage of words does not skew the lookup as much. This method is a classic "bag of words" retrieval function that is commonly used.

The Icelandic Wikipedia, downloaded on the 20th of may 2020, is used as a document source, and the three datasets, NQiI, Gettu betur corpus and the Icelandic trivia questions are used to test this approach. For each question an answer is retrieved from the corpus if one is found in spans averaging 100 words (sentences are not split) to see how many of the questions can be answered using only BM25. If the correct answer span is contained within the retrieved segment it is marked correct. The results are shown in Table 5.1.

5.1.1. Effectively using BM25 for Open QA in Icelandic

Prior to calculating similarity scores between questions and text segments, the text is normalized by first tokenizing[2] then lemmatizing the text before lowercasing it and stripping all punctuation. This step is particularly important for morphologically rich languages such as Icelandic, where the same word can have many different word forms based on its grammatical features. For context creation, the Wikipedia articles were split into segments averaging 100 words.

In the lemmatization step, the Icelandic NLP package `greynir` [42] was used. It is based on constituency parsing, which ensures lemma quality. The parser in `greynir` returns multiple lemma candidates when it can not determine which variant is the correct one. Ambiguous candidates with differing part-of-speech (POS) tags were resolved using the POS-tagging functionality in the `greynirseq` package, the POS-tagger was trained on top of IceBERT.

Common words, often called stop words, and question words were removed from the lemmas to prevent them from polluting the comparison space. See appendix A.4 for a complete list of words used.

TF-IDF is calculated as follows, where $\mathrm{tf}(t, d)$ is the frequency of the term t in document d, df(t) is the overall count of documents where the the term t occurs and N is the total number of documents.

$$\text{tf-idf}(t, d) = \mathrm{tf}(t, d) \cdot \log\left(\frac{N}{\mathrm{df}(t) + 1}\right)$$

[2]Using the `tokenizer` [41] package.

The score ranks documents for a given word based on its occurrences in all documents considered. BM25 differs from basic TF-IDF in that it dampens the effects of very high term frequency and normalizes for document length.

$$\mathrm{BM25}(t,d) = \frac{(k+1)\cdot \mathrm{tf}(t,d)}{k\cdot(1-b+b\cdot(|d|/\lambda))+\mathrm{tf}(t,d)} \log\left(\frac{N}{\mathrm{df}(t)+1}\right)$$

Where b and k are tunable constants, λ is the average document length and $|d|$ is the length of document d. The BM25 implementation used is imported from the package `gensim` [43].

5.1.2. BM25 Open QA results

The results for BM25 seen in Table 5.1 are promising. Question-paragraph matches are marked as containing the answer if the answer is found anywhere within the paragraph. Due to the long contexts used (paragraphs averaging around 100 words) and the multiple candidates, the F1 score is calculated as the maximum of all possible spans within the selected contexts.

$$F1_{\max}(\text{answer}, \text{candidates}) = \text{argmax}_{i,s,e} F_1(\text{answer}, (\text{candidate}_i)_s^e)$$

Where $(\text{candidate}_i)_s^e$ is the sequence of words starting at index s and ending at e within candidate_i. It is worth pointing out that this kind of score will only work for questions that overlap in their words with candidate contexts, using different words that have the same meaning may well lead to no answer being found when different phrasing might have sufficed.

As the candidates considered for an answer match are increased the accuracy goes up as seen in Table 5.1. The performance for NQiI is best when it comes to F1 and when the most candidates are considered, as that dataset was created using the Icelandic Wikipedia. The other datasets are not far behind and are comparable in performance, though the Trivia questions are a bit further ahead when it comes to finding answers.

Table 5.1: Results for BM25 only QA. The answers are searched for in the Icelandic Wikipedia. The articles are split up into segments of 100 words.

Dataset	Candidates	Contains answer	Max F1
Gettu betur	1	18.4 %	22.6
Trivia	1	21.8 %	22.4
NQil	1	18.6 %	36.4
Gettu betur	5	30.9 %	36.9
Trivia	5	35.9 %	36.5
NQil	5	34.0 %	61.8
Gettu betur	10	35.8 %	42.1
Trivia	10	41.3 %	42.0
NQil	10	44.0 %	71.5

5.2. A retriever-reader Open QA system for Icelandic

The methods developed for QA using term frequencies and extractive QA are now combined to create a retriever-reader open QA system for Icelandic where the answer is given as a short span of text from within a Wikipedia article. First, the question is paired with articles from the Icelandic Wikipedia, then an answer is searched for among the top-hits using the reading comprehension model trained in the prior chapter.

5.2.1. Setting up the retriever

In contrast to the 100 word segments used for the BM25 only approach, this time around the entire Wikipedia article is considered when extracting the answer. Preprocessing with lemmatization, lowercasing, punctuation and stop word stripping was done as described in the earlier chapter on BM25.

5.2.2. Retriever-reader results

Using the best model from the section on extractive QA as the reader component it is time to use the open datasets for evaluation. Results for using the warm model trained on all translated data as well a model with further finetuning on NQiI are shown in table 5.2. As before, all of the Icelandic Wikipedia is searched.

Table 5.2: BM25 + IceBERT-QA. BM25 is used as a retriever, selecting relevant documents from the Icelandic Wikipedia. IceBERT-QA labels answer spans from within those documents.

Dataset	Candidates	All		Ft. on NQiI	
		Exact	F1	Exact	F1
Gettu betur	1	7.1	15.9	5.5	15.6
Trivia	1	10.2	16.5	7.1	15.5
NQiI	1	2.7	16.8	2.4	17.9
Gettu betur	3	7.1	16.1	5.7	15.9
Trivia	3	10.2	16.5	7.2	15.7
NQiI	3	2.7	16.9	2.4	18.1
Gettu betur	5	7.1	16.0	5.6	15.8
Trivia	5	10.3	16.6	7.2	15.7
NQiI	5	2.7	17.3	2.4	18.1

The numbers are unsurprisingly lower than when the evaluations were calculated over multiple spans as when BM25 was only applied. While the exact match scores are mildly disappointing the F1 scores are encouraging. Using more candidates does not improve scores, which can probably be attributed to the fact that the candidate segments mostly come from the same articles.

More high quality data would probably not have hurt, in particular data intended for use with the underlying Wikipedia corpus. The NQiI dataset is small and the Gettu betur and Trivia datasets may not be particularly well suited for searching in the Icelandic Wikipedia. The BM25 approach on its own seems to do quite well though, and the extractive reading comprehension model does well on its own. Something about their combination clearly falls short. It might be wise to look into if the use of lemmatization in the retrieval stage is causing some of the morphological information to be lost that is important during the reader phrase, i.e. if the articles suitable for the reader are different from those found using the retriever. While interesting, this further investigation is left out for future work.

6. Dense Open QA

To further improve and speed up the results achieved with the BM25 retriever a new approach using pre-encoded segments and maximum inner product search (MIPS) is taken in this chapter.

6.1. Introduction to dense retrieval

Current state-of-the-art methods for retrieval-based QA use end-to-end training of neural networks. This is different from the method in the earlier chapter where retrieval was performed with term frequencies and BM25. The use of neural networks that parametrize the input sequences into a *"dense"* space enables the training of such systems. The term dense means that most of the vector values are non-zero and thus information rich.

In recent papers such as ORQA [37] and DPR [38], two neural networks are trained where one acts as a retriever, and another extracts the answers. The problem with these approaches is the incredibly expensive retrieval and training, where indexed embeddings need to be refreshed regularly.

In the paper *Dense Representations of Phrases at Scale (DRP)* [18] the authors describe a moderately cheap, highly efficient way of creating a QA system for English referred to as *DensePhrases*. The method goes as follows:

(i) Split the source text into all possible substrings within some length range, generate questions using a generative language model for all of the substrings as described in Section 2.3.2.

(ii) Use an existing language model to encode all substrings and the generated questions, train the substring encoder to maximize the inner product of generated questions and their respective substrings.

(iii) Train the question encoder to align with the substring encoder to predict start and end positions of answer spans.

Steps (i) to (iii) are then repeated with human-created datasets such as Natural questions [44] and SQuAD. During training time, an extractive QA model is also distilled [45]. The distillation works by sending the same data through a different "teacher" model and awarding the original model for similar predictions to the teacher while punishing it for different predictions. The resulting model consists of three transformer encoders, one for encoding phrases and one for each end of the answer spans locations. The encoders all come from finetuning large language models.

In the DRP paper, the authors use SpanBERT [46] as a pre-trained language model and the English Wikipedia as a source of answer contexts. By using FAISS [47] for storing embeddings, it is feasible to run real-time QA on a computer with 100GB of RAM and an 11GB GPU. At inference time MIPS is performed over the FAISS embeddings to search for the highest scoring span embedding.

6.2. Cross-lingual QA between Icelandic and English

Inspired by the DRP method and the moderate success of the BM25 + IceBERT method, one might wonder whether this approach would not be suitable with text in differing languages. Where one language is embedded with the query encoder and another with the phrase encoder. Since answers can not always be found in Icelandic and people often resort to searching in English, it would be useful to be able to search in Icelandic and get an answer in English.

For a single pre-trained model to be used, it needs some notion of both phrase and query content, though nothing intuitively objects to the use of different models for the phrase and query embeddings.[1] To this end, a bilingual language model was trained following XLM-RoBERTa [48] suitable for both Icelandic and English encoding to maximize the chance of a shared representation space, see e.g. [49] for an analysis of this phenomenon. The Icelandic training data is the same as the one used for IceBERT. The Books 3 corpus[2] is used as source for English data, it contains around 100GB of data.

[1] This was tried using SpanBERT for English and IceBERT for Icelandic, but the results were disappointing, some alignments of models or a common vocabulary might help out, though.

[2] This is similar to [50] and was made available in the issue section of the GitHub repository `https://github.com/soskek/bookcorpus/issues/27`

6.2.1. Data

As the DRP pre-print available at the time of writing focused on using Natural questions [44] and SQuAD, these datasets were partially translated into Icelandic to form SQuAD-ISQ and NQ-ISQ with questions in Icelandic and answers in English. In addition, English questions were generated for all spans using a large generative language model by the authors (to enable training of question embeddings that could be aligned to any answer span). All of those questions were also translated into Icelandic. This resulted in a large a cross-lingual Icelandic-English QA dataset, with the questions in Icelandic and the answers in English.

It is worth explaining why the spans in Wikipedia were not translated for a fully translated Icelandic corpus. Since the translation model used is highly context dependant it is not feasible to translate all sub-strings on their own, their meaning would be lost far too often and their concatenation would not form a coherent text. If entire articles were to be translated then the generated questions would loose their context as mapping of spans across translations is only viable for high interest locations such as entities.

6.2.2. Multilingual language model

To train the bilingual model a sentencepiece [51] vocabulary of size 50k was created using the training data. The XLMR-ENIS model was then trained using *fairseq* and ported along with the vocabulary for compatibility with the *transformers* library from Huggingface.

The model took a while longer to train than the monolingual IceBERT model. Since the English training data came from all kinds of books including text books on all topics, including literature and science, it can be thought to complement the text used to train IceBERT. It is also worth noting that the English data was an order of magnitude larger than the Icelandic. At training time, the Icelandic data was up-sampled[3] to ensure the languages were equally represented. Training took 18 days on 24 32GB v100 cards with infiniband and completed 203k updates with an effective batch size of 8.4k. See Appendix A.3 for hyperparameters used.

To evaluate the model's performance a document level QA-model was trained using the SQuAD dataset and its translation, one using the RoBERTA-base model and the others on XLMR-ENIS. The results are shown in Table 6.1 and as expected there is a slight drop in performance for the bilingual model since it has to incorporate

[3]On average each batch contained the same amount of Icelandic and English text, resulting in the Icelandic text being seen more often throughout the training.

information about both English and Icelandic.

Table 6.1: Performance for English, Icelandic and bilingual models adapted for QA using SQuAD and SQuAD-IS.

Model	Dataset	With answer		No answer		All	
		F1	Exact	F1	Exact	F1	Exact
RoBERTa (EN)	SQuAD	58.2	53.6	85.1	85.1	75.9	74.3
XLMR-ENIS	SQuAD	53.6	50.0	85.0	85.0	74.2	73.0
XLMR-ENIS	IS-SQuAD	41.9	37.9	86.9	86.9	71.5	70.1
IceBERT (IS)	IS-SQuAD	45.4	41.2	86.0	86.0	72.1	70.6

Results for the translated SQuAD set are included in Table 6.1 along with the IceBERT + IS-SQuAD numbers from Table 4.2. There is only a slight drop (0.5) in exact match performance and F1 score (0.6) when using XLMR-ENIS as compared to using IceBERT as the original model. This confirms that the model is a strong language model both for Icelandic and English and is well suited for further adaption.

6.2.3. Cross-lingual extractive QA-model

A cross-lingual extractive QA model was then trained using the datasets containing translated Icelandic questions and original English answers and paragraphs. The output of the teacher model is then compared to the DensePhrases model output at training and the difference contributes to the loss calculations. Performance is measured on the development set of natural questions with translated questions. Note that this model does not predict missing answers to be compatible with the training of the DensePhrases model.

Table 6.2: Performance for cross-lingual reading comprehension models. Questions have been translated into Icelandic while the context and answers are in English.

Model	Training dataset	F1	Exact
XLMR-ENIS	NQ-ISQ	74.9	67.1
XLMR-ENIS	SQuAD-ISQ	59.9	50.6
XLMR-ENIS	NQ-ISQ + SQuAD-ISQ	75.8	67.9

The datasets NQ-ISQ and SQuAD-ISQ used in Table 6.2 refer to the Natural questions and SQuAD datasets with only the questions machine translated into Icelandic. With 2/3 questions answered exactly the model trained on both datasets servers well as a teacher model for the DensePhrases training.

6.2.4. Training modifications

Unfortunately, the code made available by the DPR authors did not support the XLMR-model using sentencepiece vocabulary, it had to be modified to be made to work.[4]

The model is trained in the same way as described in the DRP paper, first on generated data, then on the "real" data, though in the cross-lingual case, it is with translated questions.

6.2.5. Icelandic Questions and English Results

The cross-lingual DensePhrases model is first trained using the translated generated data for two epochs, then fine tuned on the real data with translated questions. The cross-lingual reading comprehension model is used for distillation.

The performance of the resulting system as measured using the Natural questions dataset with translated questions is a success. It only differs from the English results by about 10% in the *semi-open* case. Semi-open means that instead of all of Wikipedia being queried for answers, only the paragraphs originating in the question-answer data in the development set are searched.

Table 6.3: Performance for semi-open cross-lingual QA between Icelandic and English. The performance for the English model as trained by [18] is included for reference.

Model	Languages	Exact	F1	Exact top 10	F1 top 10
Is - En	Is - En	36.5	41.8	67.5	73.7
En - En*	En - En	40.3	47.2	63.6	72.2

The En-En model data shown in table 6.3 is from the DRP authors.[5] Moving from the semi-open case to using all of the underlying Wikipedia, table 6.4, leads to a drop in performance as expected. Results for the non-translated data are included. It is worth pointing out that the system can still answer English questions surprisingly well with higher performance than in the cross-lingual case.

[4]The modified code is available at `https://github.com/vesteinn/DensePhrases`

[5]The "Xx - Yy" notation refers to the query encoder being trained to use language Xx and the phrase encoder the language Yy.

Table 6.4: Performance for open cross-lingual QA between Icelandic and English. The performance for the non translated dataset is included for reference.

Model	Languages	Exact	F1	Exact top 10	F1 top 10
Is - En	Is - En	11.3	15.2	29.6	38.5
Is - En	En - En	14.0	18.9	34.7	45.0

6.3. End-to-end Open QA for Icelandic

Finally, building on the success of the Icelandic-English case, it would be interesting to see if it is possible to train a system that can answer Icelandic questions in Icelandic.

It is not yet feasible to train a DensePhrases model for Icelandic from scratch since there still does not exist a generative model suitable for creating Icelandic questions. This generative step was done for each phrase or substring in the English case for DensePhrases using methods described in e.g. [52].

In light of this, another approach is taken where the Icelandic-English DensePhrases system is modified to also handle asking questions in Icelandic. To this end, an Icelandic extractive QA-model is adapted from XLMR-ENIS for use as a teacher model. The resulting system is evaluated using the NQiI dataset. The resulting performance is shown in table 6.5.

Table 6.5: Semi-open QA for Icelandic. The model is adapted from the cross-lingual Icelandic-English model.

Dataset	Exact	F1	Exact top 10	F1 top 10
NQiI	21.8	38.7	46.0	70.7

The semi-open QA task is only performed over the paragraphs contained in the development data. To fully test the performance, all of the Icelandic Wikipedia is embedded and evaluated using NQiI, Gettu betur and the Trivia corpus.

Table 6.6: Open QA for Icelandic. The model is adapted from the cross-lingual Icelandic-English model.

Dataset	Exact	F1	Exact top 10	F1 top 10
NQiI	9.7	18.8	26.8	44.6
Gettu betur	6.0	8.3	14.8	20.6
Trivia	5.4	6.9	14.6	18.4

The results in 6.6 are promising. They are higher (3.6x) than for the BM25+IceBERT-

QA setup on NQiI and slightly lower for the Gettu Betur and the Trivia questions as seen in **??**. This setup is also much faster in performance.

It is safe to conclude that it is well feasible to train an open cross-lingual QA system on partially translated English data and then fine tune it on real original data. This kind of approach is particularly useful for low resource languages such as Icelandic and. The NQiI evaluation performance is even surprisingly good when extended to multiple candidates, reaching an exact match of 27% and F1-score of 45.

7. Conclusions and future work

QA is an interesting subject and has some very useful applications. It is used daily by most computer users via search engines or smart assistants. In this thesis the goal was set out to narrow the QA performance gap between Icelandic, a language spoken by only hundreds of thousands of people, and English. All in all the experiments made confirm that it is well possible to use English resources to develop QA systems for Icelandic, both for querying a single document and a larger corpus of documents. In particular, it proved fruitful to use translated datasets, a cross-lingual language model and recent solutions for English that take advantage of pre-encoded phrase lookup.

7.1. Conclusions and summary

7.1.1. Datasets

A new, yet still very imperfect dataset, Natural Questions in Icelandic, was created for training and validation across a variety of experiments. However, it is clear from the experiments performed as part of this thesis that a larger and higher quality version of NQiI is sorely needed. Luckily, the machine translations of existing English datasets SQuAD and NewsQA worked out and with their help performance could be improved. This answers the first two questions posed in Section 1.1 positively, both regarding the feasibility and efficacy of using translated English data for QA in Icelandic.

7.1.2. Language models

The Icelandic language model **IceBERT** and Icelandic-English language model **XLMR-ENIS** were trained and fine-tuned numerous times over almost one and a half year and may serve as a foundation for a multitude of adapted tasks in years to come. Both will be released for use by the wider community along with the

Icelandic Common Crawl Corpus used to train them.

7.1.3. Summary of QA methods

Six different approaches to QA were considered and the results are briefly summarized in Table 7.1. The numbers are not directly comparable due to differences in models and metrics.

Table 7.1: QA systems considered in this thesis. Note that performance is not comparable between most of the entries. The evaluation for only BM25 is based on the answer being in the found segment and the maximum F1 metric found is used.

Open	System	Evaluation set	EM	F1-score
X	IceBERT-QA	NQiI	51	73
X	XLMR-ENIS-QA	NQ-ISQ / SQuAD-ISQ	68	76
✓*	BM25 only	NQiI	19	36
✓	BM25 + IceBERT-QA	NQiI	2	18
✓	DensePhrases IS-EN	NQ-ISQ	11	15
✓	DensePhrases IS-IS	NQiI	10	19

The first reading comprehension models can successfully identify spans with answers to many questions. The model trained on the translated SQuAD and Natural questions datasets along with NQiI is available at `https://huggingface.co/vesteinn/IceBERT-QA` for live inference.

The second system is akin to the first but trained over the bilingual XLMR-ENIS model using English data with questions translated in Icelandic.

The third approach using only BM25 is surprisingly good at finding relevant passages for questions, utilizing term frequencies, but note that it does not label a precise location within the 100 words found.

The fourth approach combines the first and third into a retriever-reader approach.

The fifth system is something I have not come across before, and I believe to be a novel method, where questions can be asked in Icelandic and answered in English. This system can be tested at `https://vesteinn.is/qa/`.

Finally, the sixth system is an open QA system for Icelandic. The performance is fairly good and this was something I did not expect would be possible at the start of the project. This system is also available for testing at `https://vesteinn.is/qa/`.

The EM results are much better than for the retriever-reader based approach using BM25 and IceBERT.

All in all, the systems show convincing performance in the tasks they have been trained at. The last questions in Section 1.1 regarding the benefits of using neural networks for QA in Icelandic and the feasibility of a cross-lingual open-domain QA system can thus be answered. It is not only possible to create such systems, but they can be used to bootstrap an Icelandic only open-domain QA system. While the work presented constitutes a foundation for Icelandic QA, it also has relevance for other low resource languages that can take advantage of machine translation when data is scarce.

7.2. Future work

Some approaches can be taken to further the work described in this thesis. The most obvious of which is to improve on the datasets, increase their size, review them better and improve inter-annotator consistency. This could be done for the NQiI dataset or by creating new datasets. Additionally, the steps described in Chapter 2.3. on question generation could be taken, in particular when a large generative language model can demonstrate sufficiently high performance. Being able to automatically generate questions would remove the need for the translation step and allow training of DensePhrases from scratch.

An improved language model should also yield better results. IceBERT and XLMR-ENIS saw very little literature or books written in Icelandic. Increasing the depth of the model should also give higher performance. As should using a language model trained to mask a span of tokens such as SpanBERT, this is shown to improve results in [18].

Finally, it would be interesting to do a more thorough investigation into what kind of questions the systems presented here have trouble answering. The annotation information from the Trivia and the Gettu betur corpus might come in handy there. Perhaps some different underlying corpora, other than Wikipedia, might prove more suitable for these datasets. It would also be interesting to see to what extent the systems are robust to modifications in the wording of questions.

References

[1] Angela Fan et al. "ELI5: Long Form Question Answering". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 3558–3567. DOI: 10.18653/v1/P19-1346. URL: https://aclanthology.org/P19-1346.

[2] Jonathan Herzig et al. "Open Domain Question Answering over Tables via Dense Retrieval". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, June 2021, pp. 512–519. DOI: 10.18653/v1/2021.naacl-main.43. URL: https://aclanthology.org/2021.naacl-main.43.

[3] Pranav Rajpurkar et al. "SQuAD: 100,000+ Questions for Machine Comprehension of Text". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 2383–2392. DOI: 10.18653/v1/D16-1264. URL: https://aclanthology.org/D16-1264.

[4] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423.

[5] Rishi Bommasani et al. *On the Opportunities and Risks of Foundation Models*. 2021. arXiv: 2108.07258 [cs.LG].

[6] Steinþór Steingrímsson et al. "Risamálheild: A Very Large Icelandic Text Corpus". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA), May 2018. URL: https://www.aclweb.org/anthology/L18-1690.

[7] Colin Raffel et al. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". In: *Journal of Machine Learning Research* 21.140 (2020), pp. 1–67. URL: http://jmlr.org/papers/v21/20-074.html.

[8] Guillaume Wenzek et al. "CCNet: Extracting High Quality Monolingual Datasets from Web Crawl Data". English. In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, May 2020, pp. 4003–4012. ISBN: 979-10-95546-34-4. URL: https://aclanthology.org/2020.lrec-1.494.

[9] Jan POMIKÁLEK. "Removing Boilerplate and Duplicate Content from Web Corpora [online]". Doctoral theses, Dissertations. Masaryk University, Faculty of InformaticsBrno, 2011. URL: https://theses.cz/id/nqo9nn/.

[10] Piotr Bojanowski et al. "Enriching Word Vectors with Subword Information". In: *Transactions of the Association for Computational Linguistics* 5 (2017), pp. 135–146. DOI: 10.1162/tacl_a_00051. URL: https://aclanthology.org/Q17-1010.

[11] Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. "Semantics derived automatically from language corpora contain human-like biases". In: *Science* 356.6334 (2017), pp. 183–186. ISSN: 0036-8075. DOI: 10.1126/science.aal4230. eprint: https://science.sciencemag.org/content/356/6334/183.full.pdf. URL: https://science.sciencemag.org/content/356/6334/183.

[12] Akari Asai et al. "XOR QA: Cross-lingual Open-Retrieval Question Answering". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, June 2021, pp. 547–564. DOI: 10.18653/v1/2021.naacl-main.46. URL: https://aclanthology.org/2021.naacl-main.46.

[13] Jonathan H. Clark et al. "TyDi QA: A Benchmark for Information-Seeking Question Answering in Typologically Diverse Languages". In: *Transactions of the Association for Computational Linguistics* 8 (2020), pp. 454–470. DOI: 10.1162/tacl_a_00317. URL: https://aclanthology.org/2020.tacl-1.30.

[14] Jingqing Zhang et al. "PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization". In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 13–18 Jul 2020, pp. 11328–11339. URL: https://proceedings.mlr.press/v119/zhang20ae.html.

[15] Chris Alberti et al. "Synthetic QA Corpora Generation with Roundtrip Consistency". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 6168–6173. DOI: 10.18653/v1/P19-1620. URL: https://aclanthology.org/P19-1620.

[16] Patrick Lewis, Ludovic Denoyer, and Sebastian Riedel. "Unsupervised Question Answering by Cloze Translation". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 4896–4910. DOI: 10.18653/v1/P19-1484. URL: https://aclanthology.org/P19-1484.

[17] Tom Brown et al. "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901. URL: https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[18] Jinhyuk Lee et al. "Learning Dense Representations of Phrases at Scale". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 6634–6647. DOI: 10.18653/v1/2021.acl-long.518. URL: https://aclanthology.org/2021.acl-long.518.

[19] Vésteinn Snæbjarnarson et al. *GreynirTranslate - mBART25 NMT models for Translations between Icelandic and English*. CLARIN-IS. 2021. URL: http://hdl.handle.net/20.500.12537/125.

[20] Yinhan Liu et al. "Multilingual Denoising Pre-training for Neural Machine Translation". In: *Transactions of the Association for Computational Linguistics* 8 (2020), pp. 726–742. DOI: 10.1162/tacl_a_00343. URL: https://aclanthology.org/2020.tacl-1.47.

[21] Yun Chen et al. "Accurate Word Alignment Induction from Neural Machine Translation". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 566–576. DOI: 10.18653/v1/2020.emnlp-main.42. URL: https://aclanthology.org/2020.emnlp-main.42.

[22] Chris Dyer, Victor Chahuneau, and Noah A. Smith. "A Simple, Fast, and Effective Reparameterization of IBM Model 2". In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Atlanta, Georgia: Association for Computational Linguistics, June 2013, pp. 644–648. URL: https://aclanthology.org/N13-1073.

[23] Adam Trischler et al. "NewsQA: A Machine Comprehension Dataset". In: *Proceedings of the 2nd Workshop on Representation Learning for NLP*. Vancouver, Canada: Association for Computational Linguistics, Aug. 2017, pp. 191–200. DOI: 10.18653/v1/W17-2623. URL: https://aclanthology.org/W17-2623.

[24] Ólafur Páll Geirsson. *IceQA: Developing a question answering system for Icelandic*. 2013. URL: http://www.ru.is/faculty/hrafn/students/IceQA.pdf.

[25] Yinhan Liu et al. "RoBERTa: A Robustly Optimized BERT Pretraining Approach". In: *CoRR* abs/1907.11692 (2019). arXiv: 1907.11692. URL: http://arxiv.org/abs/1907.11692.

[26] Myle Ott et al. "fairseq: A Fast, Extensible Toolkit for Sequence Modeling". In: *Proceedings of NAACL-HLT 2019: Demonstrations*. 2019.

[27] Ashish Vaswani et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[28] F. Rosenblatt. "The perceptron: A probabilistic model for information storage and organization in the brain." In: *Psychological Review* 65.6 (1958), pp. 386–408. ISSN: 0033-295X. DOI: 10.1037/h0042519. URL: http://dx.doi.org/10.1037/h0042519.

[29] Jesse Vig. "A Multiscale Visualization of Attention in the Transformer Model". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 37–42. DOI: 10.18653/v1/P19-3007. URL: https://www.aclweb.org/anthology/P19-3007.

[30] Jay Alammar. *The Illustrated Transformer*. 2018. URL: http://jalammar.github.io/illustrated-transformer/.

[31] Ian Tenney, Dipanjan Das, and Ellie Pavlick. "BERT Rediscovers the Classical NLP Pipeline". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 4593–4601. DOI: 10.18653/v1/P19-1452. URL: https://aclanthology.org/P19-1452.

[32] Starkaður Barkarson et al. *MIM-GOLD 20.05*. CLARIN-IS. 2020. URL: http://hdl.handle.net/20.500.12537/39.

[33] Svanhvıt Ingólfsdóttir, Ásmundur Alma Guðjónsson, and Hrafn Loftsson. *MIM-GOLD-NER – named entity recognition corpus*. CLARIN-IS. 2020. URL: http://hdl.handle.net/20.500.12537/42.

[34] Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. "Question and Answer Test-Train Overlap in Open-Domain Question Answering Datasets". In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, Apr. 2021, pp. 1000–1008. URL: https://aclanthology.org/2021.eacl-main.86.

[35] Thomas Wolf et al. "Transformers: State-of-the-Art Natural Language Processing". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. DOI: 10.18653/v1/2020.emnlp-demos.6. URL: https://aclanthology.org/2020.emnlp-demos.6.

[36] Karen Spärck Jones. "A statistical interpretation of term specificity and its application in retrieval". In: *Journal of Documentation* 28 (1972), pp. 11–21.

[37] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. "Latent Retrieval for Weakly Supervised Open Domain Question Answering". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 6086–6096. DOI: 10.18653/v1/P19-1612. URL: https://aclanthology.org/P19-1612.

[38] Vladimir Karpukhin et al. "Dense Passage Retrieval for Open-Domain Question Answering". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 6769–6781. DOI: 10.18653/v1/2020.emnlp-main.550. URL: https://aclanthology.org/2020.emnlp-main.550.

[39] Kelvin Guu et al. "REALM: Retrieval-Augmented Language Model Pre-Training". In: *CoRR* abs/2002.08909 (2020). arXiv: 2002.08909. URL: https://arxiv.org/abs/2002.08909.

[40] Stephen Robertson et al. "Okapi at TREC-3." In: Jan. 1994.

[41] Vilhjálmur Þorsteinsson, Hulda Óladóttir, and Sveinbjörn Þórðarson. *Tokenizer for Icelandic text*. CLARIN-IS. 2020. URL: http://hdl.handle.net/20.500.12537/65.

[42] Vilhjálmur Þorsteinsson, Hulda Óladóttir, and Hrafn Loftsson. "A Wide-Coverage Context-Free Grammar for Icelandic and an Accompanying Parsing System". In: *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*. Varna, Bulgaria: INCOMA Ltd., Sept. 2019, pp. 1397–1404. DOI: 10.26615/978-954-452-056-4_160. URL: https://aclanthology.org/R19-1160.

[43] Radim Rehurek and Petr Sojka. "Gensim–python framework for vector space modelling". In: *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic* 3.2 (2011).

[44] Tom Kwiatkowski et al. "Natural Questions: A Benchmark for Question Answering Research". In: *Transactions of the Association for Computational Linguistics* 7 (Mar. 2019), pp. 452–466. DOI: 10.1162/tacl_a_00276. URL: https://aclanthology.org/Q19-1026.

[45] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. *Distilling the Knowledge in a Neural Network*. 2015. arXiv: 1503.02531 [stat.ML].

[46] Mandar Joshi et al. "SpanBERT: Improving Pre-training by Representing and Predicting Spans". In: *Transactions of the Association for Computational Linguistics* 8 (2020), pp. 64–77. DOI: 10.1162/tacl_a_00300. URL: https://aclanthology.org/2020.tacl-1.5.

[47] Jeff Johnson, Matthijs Douze, and Hervé Jégou. "Billion-scale similarity search with GPUs". In: *arXiv preprint arXiv:1702.08734* (2017).

[48] Alexis Conneau et al. "Unsupervised Cross-lingual Representation Learning at Scale". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 8440–8451. DOI: 10.18653/v1/2020.acl-main.747. URL: https://aclanthology.org/2020.acl-main.747.

[49] Steven Cao, Nikita Kitaev, and Dan Klein. "Multilingual Alignment of Contextual Word Representations". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL: https://openreview.net/forum?id=r1xCMyBtPS.

[50] Sosuke Kobayashi. *Homemade BookCorpus*. https://github.com/BIGBALLON/cifar-10-cnn. 2018.

[51] Taku Kudo and John Richardson. "SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 66–71. DOI: 10.18653/v1/D18-2012. URL: https://aclanthology.org/D18-2012.

[52] Luis Enrico Lopez et al. "Transformer-based End-to-End Question Generation". In: *CoRR* abs/2005.01107 (2020). arXiv: 2005.01107. URL: https://arxiv.org/abs/2005.01107.

A. Appendix

A.1. Examples of translated questions

Question	Answer
Hversu mörg Grammy-verðlaun vann Beyoncé fyrir sína fyrstu sólóplötu?	fimm
Beyoncé Giselle Knowles-Carter (fædd 4. september 1981) er bandarísk söngkona, lagahöfundur, plötuframleiðandi og leikkona. Fædd og uppalin í Houston í Texas efndi hún til ýmissa söng- og danskeppni sem barn og öðlaðist frægð síðla árs 1990 sem aðalsöngkona R & B-stúlknahópsins Destiny's Child. Stýrt af föður sínum, Mathew Knowles, varð hópurinn einn af metsölustúlknahópum allra tíma í heiminum. Hiatus þeirra sá um útgáfu fyrstu plötu Beyoncé, Dangerously in Love (2003), sem kom henni á laggirnar sem einleikara á heimsvísu, vann til fimm Grammy-verðlauna og kom fram á Billboard Hot 100 númer eitt, "Crazy in Love" og "Baby Boy".	
Spenna sem er á móti rafmagnsspennu mótorsins kallast hvað?	back electromotive force
Þar sem armavafningar jafnstraums- eða altæks hreyfils fara í gegnum segulsvið hafa þeir framkallað spennu í þeim. Þessi spenna hefur tilhneigingu til að vera á móti rafspennu hreyfilsins og er svokallaður "back electromotive force (emf)". Spennan er í réttu hlutfalli við ganghraða mótorsins. Bakspenna mótorsins, að viðbættu spennufalli yfir innra slitmóti og burstum, skal vera jöfn spennunni við burstana. Þetta er grundvallaraðferðin við að stjórna hraða í jafnstraumshreyfli. Ef vélrænt álag eykst þá hægir á mótornum; þá myndast emf-álag af neðra baki og meiri straumur er dreginn af framboðinu. Þessi aukni straumur gefur viðbótarsnúningsvægið til að jafna nýja álagið.	
Hver varð fyrsti lýðræðislega kjörni forseti?	Ange-Félix Patassé
Þegar önnur umferð kosninganna var loks haldin 1993, aftur með fulltingi alþjóðasamfélagsins samræmt af GIBAFOR, vann Ange-Félix Patassé í annarri umferð atkvæðagreiðslunnar með 53% atkvæða en Goumba vann 45,6%. Flokkur Patassés, flokkurinn Mouvement pour la Libération du Peuple Centrtrack (MLPC) eða Hreyfing fyrir frelsi Mið-Afríkulýðveldisins, öðlaðist einfaldan en ekki hreinan meirihluta sæta á þingi, sem þýddi að flokkurinn Patassé þurfti á bandalagi að halda.	

A.2. Reading comprehension hyper parameters

Hyperparameters for running the QA tasks are mostly kept the same as for the English examples from Huggingface.

Table A.1: Reading comprehension QA hyperparameters

Param	Value
Batch size (sequences)	12
Learning rate	3e-5
Epochs	4
Max seq. length	512
Document stride	64

A.3. XLMR-ENIS hyperparameters

Table A.2: XLMR-ENIS hyperparameters

Param	Value
Fp16 init scale	128
Fp16 scale tolerance	0.0
Min loss scale	0.0001
Criterion	masked lm
Optimizer	adam
Lr scheduler	polynomial decay
Task	multilingual masked lm
Num workers	2
Max tokens	8192
Batch size	16
Required batch size multiple	8
Validate interval	5000
Max tokens valid	8192
Curriculum	0
Bucket cap mb	25
Arch	roberta base
Clip norm	1.0
Update freq	22
Lr	0.0006
Min lr	1
Adam betas	0.9-0.98
Adam eps	1e-06
Weight decay	0.01
Warmup updates	15000
End learning rate	0.0
Power	1.0
Total num update	1500000
Sample break mode	complete
Tokens per sample	512
Mask prob	0.15
Leave unmasked prob	0.1
Random token prob	0.1
Multilang sampling alpha	0.7
Dropout	0.1
Attention dropout	0.1
Encoder layers	12
Encoder embed dim	768
Encoder ffn embed dim	3072
Encoder attention heads	12
Activation fn	gelu
Pooler activation fn	tanh
Activation dropout	0.0
Pooler dropout	0.0
Max positions	512

A.4. Stop words excluded from BM25

The following common words[1] and question words were excluded from the BM-25 lookup to focus on words of higher interest.

að af afhverju allur annaðhvort annar á eða ef eins en enda enginn ég frá hafa hann hinn hjá hún hvað hvaða hvaðan hvenær hver hverju hvernig hvert hví hvor hvorki hvort hvorugur í minn munu nálægt neinn nema né nokkur og ó sá sem sinn sjálfur svo til undir vegna vera verða yfir ýmis það þar þegar þess þessi þinn þó þótt þú æ

[1]Based on the list here `https://github.com/ViktorMS/stoppord`