Applications of Natural Language Processing for Low-Resource Languages in the Healthcare Domain

by

Jeanne Elizabeth Daniel

Thesis presented in partial fulfilment of the requirements for the degree of Master of Science (Applied Mathematics) in the Faculty of Science at Stellenbosch University

Supervisor: Prof. W. Brink

March 2020

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: March 2020 ....

Abstract

Since 2014 MomConnect has provided healthcare information and emotional support in all 11 official languages of South Africa to over 2.6 million pregnant and breastfeeding women, via SMS and WhatsApp. However, the service has struggled to scale efficiently with the growing user base and increase in incoming questions, resulting in a current median response time of 20 hours. The aim of our study is to investigate the feasibility of automating the manual answering process. This study consists of two parts: i) answer selection, a form of information retrieval, and ii) natural language processing (NLP), where computers are taught to interpret human language. Our problem is unique in the NLP space, as we work with a closed-domain question-answering dataset, with questions in 11 languages, many of which are low-resource, with English template answers, unreliable language labels, code-mixing, shorthand, typos, spelling errors and inconsistencies in the answering process. The shared English template answers and code-mixing in the questions can be used as cross-lingual signals to learn cross-lingual embedding spaces. We combine these embeddings with various machine learning models to perform answer selection, and find that the Transformer architecture performs best, achieving a top-1 test accuracy of 61.75% and a top-5 test accuracy of 91.16%. It also exhibits improved performance on low-resource languages when compared to the long short-term memory (LSTM) networks investigated. Additionally, we evaluate the quality of the cross-lingual embeddings using parallel English-Zulu question pairs, obtained using Google Translate. Here we show that the Transformer model produces embeddings of parallel questions that are very close to one another, as measured using cosine distance. This indicates that the shared template answer serves as an effective cross-lingual signal, and demonstrates that our method is capable of producing high quality cross-lingual embeddings for low-resource languages like Zulu. Further, the experimental results demonstrate that automation using a top-5 recommendation system is feasible.

Uittreksel

Sedert 2014 bied MomConnect vir meer as 2.6 miljoen swanger vrouens en jong moeders gesondheidsinligting en emosionele ondersteuning. Die platform maak gebruik van selfoondienste soos SMS en WhatsApp, en is beskikbaar in die 11 amptelike tale van Suid-Afrika, maar sukkel om doeltreffend by te hou met die groeiende gebruikersbasis en aantal inkomende vrae. Weens die volumes is die mediaan reaksietyd van die platform tans 20 ure. Die doel van hierdie studie is om die vatbaarheid van 'n outomatiese antwoordstelsel te ondersoek. Die studie is tweedelig: i) vir 'n gegewe vraag, kies die mees toepaslike antwoord, en ii) natuurlike taalverwerking van die inkomende vrae. Hierdie probleem is uniek in die veld van natuurlike taalverwerking, omdat ons werk met 'n vraag-en-antwoord datastel waar die vrae beperk is tot die gebied van swangerskap en borsvoeding. Verder is die antwoorde gestandardiseerd en in Engels, terwyl die vrae in al 11 tale kan wees en meeste van die tale kan as lae-hulpbron tale geklassifiseer word. Boonop is inligting oor die taal van die vrae onbetroubaar, tale word gemeng, daar is spelfoute, tikfoute, korthand (SMS-taal), en die beantwoording van die antwoorde is nie altyd konsekwent nie. Die gestandardiseerde Engelse antwoorde, wat gedeel word deur meertalige vrae, asook die gemende taal in die vrae, kan gebruik word om kruistalige vektorruimtes aan te leer. 'n Kombinasie van kruistalige vektorruimtes en masjienleer-modelle word afgerig om nuwe vrae te beantwoord. Resultate toon dat die beste masjienleer-model die Transformator-model is, met 'n top-1 akkuraatheid van 61.75% en 'n top-5 akkuraatheid van 91.16%. Die Transformator toon ook 'n verbeterde prestasie op die lae-hulpbron tale, in vergelyking met die lang-korttermyn-geheue (LSTM) netwerke wat ook ondersoek is. Die kwaliteit van die kruistalige vektorruimtes word met parallelle Engels-Zulu vertalings geëvalueer, met die hulp van Google Translate. So wys ons dat die Transformator vektore vir die parallelle vertalings produseer wat baie na aan mekaar in die kruistalige vektorruimte, volgens die kosinus-afstand. Hierdie resultate demonstreer dat ons metode die vermoë besit om hoë-kwaliteit kruistalige vektorruimtes vir lae-hulpbron tale soos Zulu te leer. Verder demonstreer die resultate van die eksperimente dat 'n top-5 aanbevelingstelsel vir outomatiese beantwoording haalbaar is.

Acknowledgements

I would like to express my sincere gratitude to the following organisations and people, without whom the success of this thesis would not have been possible:

- My supervisor, Prof. Willie Brink, who continuously provided valuable feedback, encouraged me to explore creative avenues and work independently, and provided tremendous support during the data acquisition process.

- Praekelt Foundation for providing me with computational resources, and financial assistance in my travels to present our paper at the ACL conference in Florence, Italy.

- Charles Copley, who represented Praekelt Foundation in our research partnership, for his assistance in acquiring the MomConnect data, his contribution to our paper, and general support throughout this thesis.

- Monika Obracka, who represented the engineering team from Praekelt Foundation and joined me at the ACL conference, for the valuable feedback on the engineering aspects of our research and our insightful and inspiring conversations while exploring all the trattorias in Florence!

- Google for assisting me in their travel grant, which enabled me to present our paper at the ACL conference.

- The Centre for Scientific and Industrial Research DST Inter-bursary Support Programme for their financial assistance during my Masters.

- My family, for emotional and financial support throughout my studies.

- My mentor, Stuart Reid, for believing in me and encouraging me to follow my dreams.

- Ryan Eloff, for your unwavering support, love, excellent sense of humour, and companionship, without which I would never have had the courage to complete this thesis.

Dedication

This thesis is dedicated to my father, Jurgens Johannes Daniel, who, from a very young age, cultivated a love for mathematics and science in me. He supported and encouraged me to pursue my passions, no matter how outrageously big or insignificantly small the world might deem them.

Contents

Nomenclature

This section provides an overview of the notation used throughout the thesis, unless stated otherwise.

a	A scalar
$\boldsymbol{a}$	A vector
$\boldsymbol{A}$	A matrix
$\mathcal{M}$	A set
a_i	An element i of vector $\boldsymbol{a}$, with the indexing starting at 1
A_{ij}	An element i, j of matrix $\boldsymbol{A}$
$\boldsymbol{a}^{(t)}$	A vector at time step t
$a_i^{(t)}$	An element i of vector $\boldsymbol{a}$ at time step t
ϵ	An arbitrarily small positive quantity, called an epsilon
ρ	A learning rate, always positive
α	An activation function
$\mathbb{R}$	The set of real numbers
$\mathbb{N}$	The set of natural numbers
$\{0, 1, \dots, n\}$	A set of all the integers between 0 and n
$\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \dots, \boldsymbol{x}^{(n)}$	An ordered sequence of n vectors
$f : \mathbb{A} \rightarrow \mathbb{B}$	A function f with domain $\mathbb{A}$ and range $\mathbb{B}$
$f(\boldsymbol{x}; \theta)$	A function f of $\boldsymbol{x}$, with a set of parameters θ
$\lVert\boldsymbol{x}\rVert$	L^2 norm of $\boldsymbol{x}$
$\log(x)$	The natural logarithm of x
$\exp(x)$	The exponential function of x
$\sigma(x)$	The logistic sigmoid of x, $\frac{1}{1+\exp(-x)}$
$\mathrm{ReLU}(x)$	The rectified linear unit function of x, $\max(0, x)$
$\tanh(x)$	The hyperbolic tangent function of x, $\frac{e^x - e^{-x}}{e^x + e^{-x}}$
$\mathrm{softmax}(\boldsymbol{z})$	The softmax function of a vector $\boldsymbol{z}$, with ith element as $\frac{\exp(z_i)}{\sum_j \exp(z_j)}$

Chapter 1

Introduction

1.1 Motivation

We are entering the fourth industrial revolution, brought forth by globalization, ubiquitous access to information via the World Wide Web, and renewed interest in the field of artificial intelligence. Natural language processing (NLP) is a sub-field of artificial intelligence; where computers are taught to interpret and understand human language. The field has grown significantly, spawning various sub-fields with real-world applications such as machine translation, sentiment analysis, and automatic question-answering. This was driven in part by advances in distributional representations of words and phrases: fixed-length, real-valued vectors that capture semantic information, and in some cases, context. The digitization of virtually every aspect of society has resulted in a wealth of digital resources available for further advancing the field of NLP.

However, the availability of digital resources is a double-edged sword and reflect the prevailing inequalities of modern society. English comprises an estimated 55% of the top 10 million websites on the World Wide Web[1], even though native English speakers make up only about 22% of the world's population. There are at least 7102 spoken languages world wide, with 2138 in Africa. The distribution of native speakers of different languages across the world remains very unbalanced, with nearly two-thirds speaking one of the following 12 languages: Mandarin, Hindi-Urdu, English, Arabic, Spanish, Bengali, Russian, Portuguese, German, Japanese, French, and Italian. In contrast, approximately 3% of the population world wide speak 96% of all the languages in the world, with many of these languages running the risk of dying out in the next century (Noack and Gamio, 2015). Some languages can be characterized as low-resource languages for which there exists little to no (publicly-available) digital resources, relative to their number of speakers (Cieri *et al.*, 2016). This could be due to a variety of factors: a lack of a documented grammar, a lack of digital platforms supporting this language, or a lack of political will.

[1] `https://w3techs.com/technologies/history_overview/content_language`

NLP research requires massive amounts of annotated texts and powerful computational resources. Dataset construction involves a high cost. For example, hand-crafting a parallel dataset suitable for machine translation between two languages is a protracted process that relies on expert knowledge. Because of the high cost involved, most annotated datasets exist only for majority languages like English, Mandarin, and Spanish. Low-resource languages are thus doubly neglected: in dataset creation (keeping them low-resource), and in novel NLP research on these languages. The result is that NLP research is heavily skewed towards majority languages and has limited opportunities for researching low-resource languages. This presents a significant challenge for developing economies with speakers of low-resource languages. New language technologies can improve current market sectors or create entirely new ones, and increase access to administrative services, healthcare and education via digital platforms in mother-tongue languages. Therefore it is necessary and important that we dedicate research to building datasets and developing natural language processing tools for low-resource languages.

We focus our efforts on a particular language modelling challenge in the healthcare domain, called MomConnect. In short, MomConnect is an initiative of the South African National Department of Health that aims to improve the health and well-being of pregnant women, breastfeeding mothers, and their infants. The service connects users to the healthcare platform via SMS and WhatsApp and has registered more than 2.6 million users since 2014, with 95% of clinics in South Africa participating in the registration process. Users can pose questions to the platform in all 11 official languages of South Africa (many of which are low-resource languages) and receive expert-crafted template responses in English which are manually selected by the MomConnect helpdesk staff. The platform receives about 1200 messages daily. The recent introduction of WhatsApp as an additional communication channel has resulted in an increase in the number of incoming questions, and currently the median response time per question is 20 hours. Until this bottleneck is addressed, the service cannot scale to a larger user base.

The template-based nature of the answers enables the use of NLP techniques to automate the response pipeline. The dataset of recorded questions and answers is quite unique to question-answering datasets as it contains multilingualism combined with a lack of reliable language labels, low-resource languages, inconsistencies in the answering process, and noise in the data. All these factors make it a particularly challenging problem and allows us the opportunity to test theories regarding cross-lingual embedding spaces by using the shared template response as a cross-lingual signal. This has been shown to facilitate knowledge transfer between languages, and strengthen embeddings of low-resource languages (Ruder, 2017). We provide a proof-of-concept for automating the answer selection process, by training machine learning models in combination with cross-lingual language models on previously-answered questions to predict the most appropriate answer.

1.2 Problem Statement

In this thesis, we aim to address the scalability factor of MomConnect by investigating the feasibility of automation in the helpdesk question-answering pipeline. This allows us to apply computational linguistic theories to a real-world problem for social impact, while simultaneously addressing the lack of language diversity in NLP research. During 2018 we, together with Praekelt Foundation, worked with the National Department of Health to gain access to a dataset of recorded MomConnect questions and answers for research purposes. Our study presents a unique opportunity for applying computational linguistics in the healthcare space due to the following properties of the dataset:

- 230,000 multilingual – Afrikaans, English, Ndebele, Northern Sotho, Southern Sotho, Swati, Tsonga, Tswana, Venda, Xhosa, and Zulu – question-answer pairs in the maternal healthcare domain,
- the multilingual questions are paired with template English responses,
- a lack of reliable languages labels and multiple low-resource languages,
- and questions with code-mixing, typos, misspellings, and inconsistencies in the answering process.

Historically, limited research has been dedicated to the low-resource languages that form part of this study, which have differing properties to majority languages. Even less (if any) research has been dedicated to multilingual, low-resource question-answering without language labels and with a high level of noise. This is also the first time any research has been applied to MomConnect to specifically automate the question-answering pipeline.

1.3 Research Objectives

Our goal is to investigate question answering and language modelling techniques that can be used to automate the question-answering pipeline. More specifically, our research objectives can be summarized as:

- addressing each of the challenges encountered in the dataset,
- experimenting with cross-lingual embedding spaces using the shared template answers as cross-lingual signals,
- investigating different question-answering techniques,
- providing a proof-of-concept for automating the answering pipeline using a top-5 recommender system,
- and performing an analysis of the cross-lingual sentence embeddings.

1.4 Related Work

Barron *et al.* (2018) discuss MomConnect as a case study for using mobile technology to promote universal healthcare access for pregnant and breastfeeding women in challenging socio-economic settings. Unlike many other developing nations, South Africa has a universal mobile phone penetration and high female literacy rates, and mobile ownership is on par for men and women. It is due to these characteristics that MomConnect is almost universally accessible in South Africa. Today MomConnect serves as a platform for real-time data and feedback collection and is integrated with the public healthcare system. Its success has been enabled through strong support and governance by the South African National Department of Health, technical assistance provided by non-government organizations, and funding by generous donors.

Using a small dataset of text exchanges recorded by the MomConnect helpdesk between 2015 and 2016, Engelhard *et al.* (2018) explore the feasibility of automatically identifying high-priority messages as well as assessing the quality of the helpdesk responses. To investigate the feasibility of automated triage, they scanned the dataset for messsages with keywords relating to abuse or mistreatment. Using keyword matching, they were able to flag 71 messages with complaints of domestic violence, discrimination, verbal abuse, violations of privacy, and poor service at the hands of healthcare facilities. Engelhard *et al.* also trained a multinomial naive Bayes classifier with bag-of-words to learn the associated labels assigned by the helpdesk to incoming messages, such as "Question", "Message Switch", "Compliment", "Complaint", "PMTCT" (prevention of mother-to-child transmission of HIV and AIDS), "Language Switch", "Opt Out", "Channel Switch", "Spam", and "Unable to Assist", and achieved a test accuracy of 85%. To assess the quality of the helpdesk responses, they took a random sample of 481 English questions and evaluated the appropriateness of the responses as well as the response time. They found that almost 19% of the responses sent by the helpdesk was either suboptimal or incorrect, and the median response time was 4 hours. These results show that automated triage and labelling are feasible, while the quality and response time of the helpdesk can be improved.

On a much larger dataset, with significant overlap to the one we discuss in this thesis, Obrocka *et al.* (2019) explore the level of code-mixing and the feasibility of single language identification. After cleaning the questions and spltting them into chunks, they apply the language tagging tool Polyglot to identify the three most common languages found in the MomConnect dataset – English, Xhosa, and Zulu – and evaluate the performance on hand-labelled data. While they achieve an accuracy of almost 78% in the single language tagging task, they note that this is not significantly better than simply classifying to the majority language, English, which has a prevalence of nearly 76% in the evaluation set. Using their findings from Polyglot, they estimate the level of code-mixing in the dataset to be approximately 10%.

1.5 Background Information

This section provides a brief overview of language modelling and question-answering techniques. This is only meant to be a brief introduction to these topics, which are discussed in more depth in Chapters 3 and 6. Modelling the MomConnect data is a unique challenge in itself, as it combines multilingual, low-resource questions with English template answers. The absence of reliable language labels and informal text means that we cannot take traditional language modelling approaches. In this thesis we discuss the following topics:

- machine learning,
- feedforward neural networks,
- recurrent neural networks,
- sequence modelling,
- metric learning,
- distributional language modelling,
- and question answering.

In our research study, all these disciplines intersect. Feedforward and recurrent neural networks as well as sequence modelling techniques form the basis of many state-of-the-art machine translation and sentence embedding techniques.

Distributional Language Models Distributional language models assume that languages have a distributional structure, i.e. that tokens found in a particular language can be represented as a function of other tokens found around them. The distributional hypothesis (Harris, 1954) describes the hypothetical system of the members of a distributional language structure and the rules dictating how they interact. The hypothetical system can be extended to mathematical models called distributional vector representations or embedding spaces of words or sentences. The vector representations of words and sentences should be constructed in such a way that semantically-related words and similar sentences are within close proximity of each other in the distributional space. Most approaches have been developed for monolingual training data, but we also consider approaches to cross-lingual word and sentence embedding spaces. While most cross-lingual embedding techniques were initially developed for a specific language or set of languages, the techniques can be applied to other languages with varying degrees of success, provided enough suitable training data exist.

Word Embeddings Word embeddings are dense representations of words that capture semantically related information, and in some cases the surrounding context and polysemy. They can be used in downstream NLP tasks such as question-answering, sentiment classification, and machine translation. Count-based word embedding techniques capture co-occurrence statistics through dimensionality reduction techniques, while prediction-based techniques try to predict the vector representations of words using nonlinear models such as neural networks, which results in vector representations that capture semantic similarities and exhibit additive compositionality behaviour. However, word embeddings are not without limitations. Take the continuous bag-of-words and the skip-gram models, two prediction-based embedding techniques introduced by (Mikolov *et al.*, 2013*a*). They can only produce embeddings for words that were included in their training vocabulary and fail on unseen words. The embeddings also suffer due to their inability to capture context-sensitive vectors. FastText, introduced by Bojanowski *et al.* (2017), improves on the classical approaches of Mikolov *et al.* by taking into account subword and character-level information. This produces improved word embeddings for morphologically rich languages, as well as enabling the model to produce vectors for words it did not encounter during training. Both ELMo and BERT, introduced by (Peters *et al.*, 2018) and (Devlin *et al.*, 2018), respectively, are models that can produce deep contextualized word embeddings by training on sequences of characters. By encoding on a character level, the models are able to deal with unseen words. Both models employ bi-directional architectures which capture syntax and semantics, as well as polysemy.

Sentence Embeddings Sentence embeddings can be powerful as they allow comparison between sentences that have similar meanings or intent but little word overlap. A simple but effective approach to sentence embedding construction is simply averaging the embeddings of the words found in the sentence (Wieting *et al.*, 2015). More complex sentence embeddings employ recurrent neural network encoder-decoder models (Kiros *et al.*, 2015), bi-directional long short-term memory networks with max-pooling (Conneau *et al.*, 2017) or Transformers (Cer *et al.*, 2018). Sentence embeddings have been shown to outperform word embeddings on a number of downstream evaluation tasks.

Cross-lingual Word and Sentence Embeddings Typically, word embeddings and sentence embeddings are trained on monolingual corpora, but the application can be expanded to include bilingual or multilingual training. In these cases, the embedding space includes words and sentences from multiple languages, and typically translations of the same words, e.g. "house" (English) and "maison" (French), are mapped to similar vector representations. This allows for translation across languages by simply retrieving the nearest neighbour of a word or sentence in the embedding space. Training such an embedding space requires some form of cross-lingual signal to allow the model to learn where the overlap between languages are, such as a dataset of parallel sentences. Monolingual embedding spaces can be aligned with the help of a bilingual lexicon dictionary, by learning a transformation matrix that can linearly map words from the source language to the target language (Mikolov *et al.*, 2013*b*). Instead of aligning monolingual embedding spaces, Gouws and Søgaard (2015) train bilingual word embedding spaces directly using psuedo-bilingual data. To construct their pseudo-bilingual dataset, they concatenate the source language and target language corpora and then randomly shuffle the corpus. To further introduce cross-lingual signals, the authors randomly replace source words with the target language equivalent words, with respect to some task like part-of-speech tagging or super-sense tagging. Pseudo-bilingual corpora can even be created without any parallel signals by concatenating and shuffled bilingual documents with shared topics, e.g. Wikipedia pages of the same topics in multiple languages (Vulic and Moens, 2016). Bilingual word embeddings can be extended to multilingual sentence embeddings and even language-agnostic sentence embeddings (Artetxe and Schwenk, 2018) by encoding multilingual sentences to byte-pairs and training an encoder-decoder model that takes in a source language sentence and translates it to a target language sentence. The language-agnostic sentence embeddings produced by the encoder can be used in downstream tasks and can even be extrapolated to unseen but related languages.

Question Answering using Answer Selection Answer selection can be formulated as an information retrieval problem, where the aim is to retrieve the most appropriate answer from a finite set of candidate answers for a given query. A frequently-asked-questions approach economically reuses previously answered questions to answer new questions. Thus, we can treat answer selection as a classification task, where the two varying factors are the method of encoding the question, and the model used to classify the encoded question. We can use a sentence embedding technique to encode the question, and then do nearest neighbour classification to find the most appropriate answer. Alternatively, we can train the sentence encoder and a classification model end-to-end, such that the log-likelihood of the ground truth answers is maximized by the learnt embeddings.

1.6 Thesis Overview

This thesis is the result of a partnership between Praekelt Foundation, a non-profit organization, and researchers from Stellenbosch University. We describe a real-world scenario where one works with imperfect data, and where each data point represents a user with concerns about the welfare of their child. The applications of multilingual question-answering in a low-resource setting are sparse and thus we explore the various fields that intersect with this topic. Chapter 2 gives an in-depth study of machine learning with the necessary foundations in feedforward and recurrent neural networks, as well as metric learning. Chapter 3 introduces the topic of language modelling, with a review of word embedding, sentence embedding, and cross-lingual embedding techniques. Our methodology spans over three chapters. We describe our data collection process in Chapter 4 where we provide an overview of the necessary data protection, anonymization, and ethical clearance processes for working with sensitive data. We describe our exploratory data analysis in Chapter 5. We introduce the question-answering problem, our experimental design, results and discussion in Chapter 6. Chapter 7 presents a conclusion to the work conducted during the course of this thesis and briefly discusses the findings and contributions, as well as suggestions for future research. In addition we provide a proof of concept for automating the MomConnect helpdesk question-answering pipeline using a top-5 recommendation system.

Chapter 2

Machine Learning

2.1 Introduction

In this chapter, we provide an in-depth review of the literature related to machine learning as background information for the reader. First we outline the different types of machine learning in Section 2.2. Then we delve into the broad field of feedforward neural networks in Section 2.3. In Section 2.4 we introduce recurrent neural networks, their inherent shortcomings, and the architectures designed to address these shortcomings. In Section 2.5, we introduce metrics and discuss the topic of metric learning with deep learning models.

2.2 Basics of Machine Learning

Machine learning is a field of study where a computer aims to automatically learn to perform tasks from sampled data, without the task being defined explicitly. This is done by exploiting statistical patterns within the sampled data, also known as the training data, represent as a matrix $\boldsymbol{X}$. The columns of $\boldsymbol{X}$ represent the features, and the rows of $\boldsymbol{X}$ represent each of the data points. We also define f, our machine learning model, and $\boldsymbol{\epsilon}$, our error vector which captures the difference between the approximation and the real values. Our machine learning model f can be of varying complexity and interpretability.

During the training phase, a machine learning model f is fitted to the training data $\boldsymbol{X}$. Following the training phase, the trained model f can be applied to new, as yet unseen data, commonly known as the test data or the leave-out set (Hastie *et al.*, 2017). In general, machine learning can be divided into three main categories: unsupervised learning, supervised learning, and reinforcement learning. There are also subcategories that overlap between these categories, such as semi-supervised learning and transfer learning.

In unsupervised learning, we only observe $\boldsymbol{X}$. Here we are not interested in prediction, rather we want to interpret the data by discovering previously-unknown patterns. Topics within unsupervised learning include, but are not limited to: clustering, dimensionality reduction, density estimation, data summarization, and outlier detection. While supervised learning aims to learning a conditional probability distribution of $\boldsymbol{X}$ given $\boldsymbol{y}$, unsupervised learning tries to infer a probability distribution from $\boldsymbol{X}$.

With supervised learning, we also have a vector of targets $\boldsymbol{y}$, where each entry in $\boldsymbol{y}$ corresponds to (supervises) a row in $\boldsymbol{X}$, such that

$$\boldsymbol{y} = f(\boldsymbol{X}) + \boldsymbol{\epsilon}. \tag{2.1}$$

The primary goal of any supervised machine learning model is to predict $\boldsymbol{y}$ in such a way that it minimizes the errors in $\boldsymbol{\epsilon}$. Supervised learning can further be divided into two sections: classification and regression. A regression model outputs a vector of quantitative (continuous) values. On the other hand, a classification model outputs a qualitative (discrete) value.

Semi-supervised techniques learn from a combination of labelled and unlabelled data. This type of learning is typically applied in a setting where there is a small amount of labelled data, and a comparatively large amount of unlabelled data. There is a high cost associated with manually labelling data, while acquiring unlabelled data might be relatively cheap in comparison. This form of learning can be used as an alternative to discarding the unlabelled data and only learning from the small set of labelled data points, or using only the unlabelled data in an unsupervised learning setting. We can also train a model to learn from the small set of labelled data, and infer labels for the unlabelled data. This form of learning is called transductive learning.

Transfer learning makes use of (often unsupervised) pre-trained models to transfer learned knowledge from one task or domain to another. This can be done by using the learned parameters of the pre-trained model as an initializer for the new machine learning model, which is then fine-tuned using new data. The pre-trained model can also be used as a feature extraction tool for downstream tasks. Transfer learning is popular in image classification and natural language processing, where training models from scratch is computationally expensive. This type of learning is also useful when there is an abundance of general-domain data, while limited resources exist for the target domain or downstream task.

2.3 Feedforward Neural Networks

Feedforward neural networks (FFNNs) are networks of artificial neurons with a parameter set $\boldsymbol{\Theta}$ that forward-propagates information, from the input $\boldsymbol{x}$, to the output $\boldsymbol{y}$. The goal of a feedforward neural network is to approximate some function $\boldsymbol{y} = f(\boldsymbol{x})$.

The Universal Approximation Theorem states that, under certain conditions, any continuous function f can be arbitrarily well approximated by a continuous feedforward neural network with only a single hidden layer and any continuous nonlinear activation function. Formally, the Universal Approximation Theorem can be defined as follows (Hornik, 1991):

Let $\alpha(\cdot)$ be an arbitrary, nonlinear activation function and let $\boldsymbol{x} \in \mathbb{R}^m$. Let $C(\boldsymbol{x})$ denote the space of continuous functions on $\boldsymbol{x}$. Then, for all $f \in C(\boldsymbol{x})$, and for all $\epsilon > 0$, there exists constants $n, m \in \mathbb{N}$, where $i \in \{1, \ldots, n\}$, $j \in \{1, \ldots, m\}$, and $W_{ij}, b_j, a_i \in \mathbb{R}$, such that

$$(A_n f)(x_1, \ldots, x_m) = \sum_{i=1}^{n} a_i \alpha\big(\sum_{j=1}^{m} W_{ij} x_j + b_j\big), \tag{2.2}$$

serves as an approximation of $f(\cdot)$, with

$$|f - A_n f| < \epsilon, \tag{2.3}$$

where n denotes to the size of the hidden layer, W_{ij} denotes the weight of the input x_i received by the jth hidden neuron, b_j is the bias term and a_j the associated weight term for the jth hidden neuron. There are caveats to this theorem (Nielsen, 2015):

- the quality of the approximation if not a guarantee, but there does exist an n for which the condition $|f - A_n f| < \epsilon$ is satisfied,
- the function f to approximate must be continuous and real-valued.

The versatility of feedforward neural networks might seem to contradict the No Free Lunch Theorem of Wolpert and Macready (1997), which informally states that "if a learning algorithm performs well on some datasets, it will perform poorly on some other datasets." However, the No Free Lunch Theorem merely implies that there is no algorithm that is generalizable for and performs well on all types of problems. The Universal Approximation Theorem states that a function can be approximated within an epsilon, but finding the exact weights that allow for this approximation is challenging, and there the approximation is not guaranteed to perform well on rare or unseen data.

2.3.1 The Biological Neuron and The Artificial Neuron

The common biological neuron comprises of dendrites, a cell body, and an axon, as displayed in Figure 2.1. Stimuli are received from adjacent neurons by the dendrites, which are then handled by the cell body. The incoming signals are combined and processed, and if the magnitude of the resulting signal is above some threshold, the neuron activates and consequently fires an output signal (called an action potential) to neighbouring neurons via the axon branches. There are an estimated 100 billion such neurons in the human brain. In all animals brains, memories and habits are formed by repeated activations of groups of neurons, a process called synaptic plasticity. The more a group of neurons fire together, the stronger the synaptic connections, a process called long-term potentiation. Over time, if the connections between the group of neurons are not activated enough or at all, the connection weakens, resulting in a decrease in synaptic strength, called long-term depression.

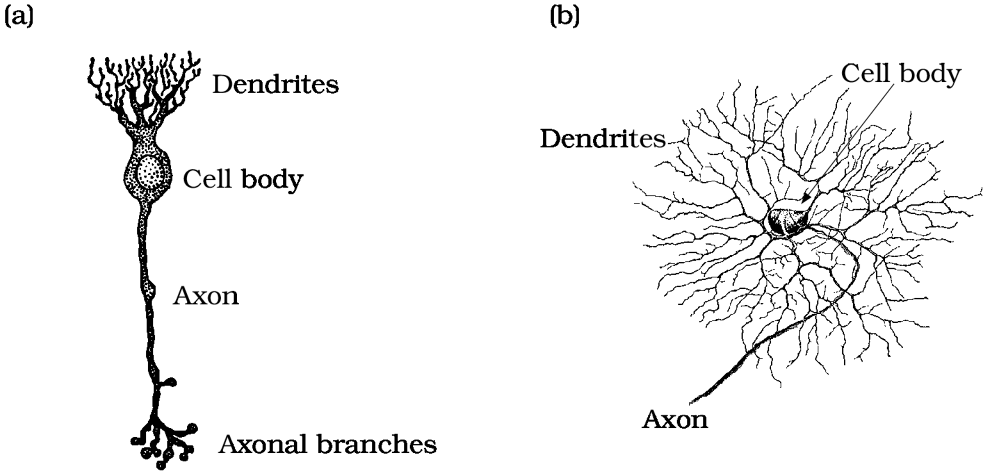

Figure 2.1: Two types of retinol neurons: (a) a midget bipolar, and (b) a parasol-type ganglion cell (Dacey and Petersen, 1992)

The functioning of the biological neuron is an inspiration for the mathematical model called the perceptron or the artificial neuron, first defined by Rosenblatt (1958). For a given a training dataset with binary class labels $\{\boldsymbol{x}_i, y_i\}_{i=1,\ldots,N}$, the perceptron learning algorithm searches for an optimally separating hyperplane that minimizes the number of misclassified points in the training dataset. The n-dimensional hyperplane in $\mathbb{R}^n$ is defined as all $\boldsymbol{x}$ for which:

$$f(\boldsymbol{x}) = \beta + \boldsymbol{w}^\top \boldsymbol{x} = 0. \tag{2.4}$$

If a response is incorrectly classified, then $\beta + \boldsymbol{w}^\top \boldsymbol{x}_i$ is negative for $y_i = 1$, and positive for $y_i = -1$. $\mathcal{M}$ is a set containing all misclassified points for the current set of weights. After each time step and subsequent parameter update, the set may change to include new or remove misclassified points. The perceptron algorithm aims to find the optimal parameters $\boldsymbol{w}$ and β that minimize

$$D(\boldsymbol{w}, \beta) = -\sum_{i \in \mathcal{M}} y_i(\beta + \boldsymbol{x}_i^\top \boldsymbol{w}), \tag{2.5}$$

where $\boldsymbol{x}_i$ is the input vector and y_i is the associated response of the ith data point in $\mathcal{M}$, β is the bias term, and $\boldsymbol{w}$ is the weight vector. Assuming $\mathcal{M}$ is fixed for this step, the gradients are calculated as

$$\frac{\partial D(\boldsymbol{w}, \beta)}{\partial \boldsymbol{w}} = -\sum_{i \in \mathcal{M}} y_i \boldsymbol{x}_i, \tag{2.6}$$

$$\frac{\partial D(\boldsymbol{w}, \beta)}{\partial \beta} = -\sum_{i \in \mathcal{M}} y_i. \tag{2.7}$$

At the start $\boldsymbol{w}$ and β are randomly initialized. Then, at each step $\boldsymbol{w}$ and β are updated as

$$\begin{pmatrix} \boldsymbol{w} \\ \beta \end{pmatrix} \leftarrow \begin{pmatrix} \boldsymbol{w} \\ \beta \end{pmatrix} + \rho \begin{pmatrix} y_i \boldsymbol{x}_i \\ y_i \end{pmatrix}. \tag{2.8}$$

The learning rate, $\rho > 0$, controls the step size of each parameter update per time step. The updated parameters should result in a separating hyperplane with fewer misclassified points. The algorithm will converge if the training data can be separated by a linear hyperplane. However, as Ripley (1996) noted, due to the random initialization there exists more than one solution, and each particular solution depends on the initial values of β and $\boldsymbol{w}$. Further, the finite number of steps can be very large, and if the data cannot be separated by a linear hyperplane, the algorithm will never converge, rather it will form cycles. The perceptron is the predecessor of modern deep learning architectures such as the feedforward neural network. Individual perceptrons within networks take as input the outputs of preceding perceptrons, which are then linearly-combined. The concept of synaptic plasticity is mirrored in the increase or decrease in the weights of the neuron, and the firing of the action potential to succeeding perceptrons is a nonlinear activation of the linearly-combined inputs. Throughout our review of deep learning, the terms perceptron and artificial neuron are equivalent. The human brain, in contrast to the conventional Von Neumann machine, can process information in a complex, nonlinear, and parallel fashion. Artificial neural networks, and particularly feedforward neural networks have been shown to successfully learn the mapping of input data to output data for a variety of complex problems. The architecture is inspired by the biological information processing unit, more commonly known as the brain.

2.3.2 Nonlinear Activation Functions

One key aspect of the Universal Approximation Theorem is the nonlinear activation function α. This represents a mathematical abstraction of the action potential firing in a biological neuron. Nonlinear functions such as the sigmoid functions, applied on top of a perceptron, allow for modelling training data with nonlinear properties. Sigmoid activation functions are characterized by their S-shaped curves. This class of functions are real-valued, differentiable, and monotonically increasing. The logistic sigmoid function is given as

$$\sigma(x) = \frac{1}{1 + \exp^{-x}}. \tag{2.9}$$

This function has a range between 0 and 1. One caveat is that values much larger than 1 are forced to almost 1, and values much less then 0 are forced to almost 0, resulting in a saturation of values.

Other nonlinear activation functions have begun to replace the popular sigmoid logistic function over the years, most notably the rectified linear unit (ReLU), which, combined with modern deep learning models, has achieved state-of-the-art results (Nwankpa *et al.*, 2018). ReLU is defined simply as

$$\text{ReLU}(x) = \max(0, x). \tag{2.10}$$

This function rectifies negative values to 0, and maintains the magnitude of values above 0. One drawback of the ReLU function is that it is not differentiable for $x = 0$. Such a value is rare, but the result is that learning does not take place for values around 0 (Goodfellow *et al.*, 2016). Another activation function that has gained popularity is the hyberbolic tangent, defined as

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \tag{2.11}$$

The hyperbolic tangent is useful as it has a range of $(-1, 1)$ and negative instances are mapped strongly negative and not suppressed. It is also differentiable and monotonically increasing. The sigmoid and hyperbolic tangent are both functions that are only really sensitive to values around the middle of their respective ranges. The gradients of both functions also saturate which is particularly problematic for the weight updates. As the number of layers in the network increases, the multiplication of the gradients through back-propagation results in a phenomenon called the vanishing or exploding gradient.

2.3.3 Feedforward Neural Networks

A vanilla feedforward neural network consists of at least three layers of densely-connected neurons. Networks capable of representing nonlinear functions make use nonlinear activation functions, like the logistic sigmoid. Note that different activation functions can be used in different layers. All artificial neural network architectures have the same basic structure: the zeroth layer is the input layer, which feeds into a sequence of one or more layers (called the hidden layers as they are not exposed directly to the inputs or outputs), to finally reach the last layer, also known as the output layer. Consider an arbitrary layer l with N_l neurons and a differentiable nonlinear activation function $\alpha^{(l)}$ associated with the layer. In a fully-connected feedforward neural network, the neurons in layer l receives as inputs the outputs of neurons in layer $l-1$. The linear combination of all the inputs of neuron j in layer l, denoted by $n_j^{(l)}$, can be stated as a recursive function

$$n_j^{(l)} = \sum_{i=1}^{N_{l-1}} W_{ij}^{(l)} \alpha^{(l-1)}(n_i^{(l-1)}) + b_j^{(l)}. \tag{2.12}$$

Here $b_j^{(l)}$ is the bias of neuron j in layer l, and $\alpha^{(l-1)}$ is the activation function applied to the net inputs of the neuron i in layer $l-1$, and then weighted by a factor $W_{ij}^{(l)}$. We substitute $\alpha^{(l-1)}(n_i^{(l-1)})$ with $a_i^{(l-1)}$ as the activated output of neuron i in layer l, and can thus simplify Equation 2.12 as

$$n_j^{(l)} = \sum_{i=1}^{N_{l-1}} W_{ij}^{(l)} a_i^{(l-1)} + b_j^{(l)}. \tag{2.13}$$

For layer $l = 1$, the inputs from the previous layer (the input layer) would be the features of the input vector $\boldsymbol{x}$. The signals are propagated forward until they reach the final (output) layer L. Layer L will then output a vector which will have the same shape as the supervising target vector.

2.3.4 Classification using Feedforward Neural Networks

The goal of a feedforward neural network is to model complex distributions, and sometimes to perform classification. In the case of classification (where our target label is a discrete variable), we ultimately want to obtain a target vector $\hat{\boldsymbol{y}}$ with positive values summing to 1. We can train a model for classification using maximum likelihood estimation. This can be done by minimizing the loss between the ground truth labels and the model predictions. The loss function for a single training example is given as

$$\mathcal{L} = -\sum_i y_i \log(\hat{y}_i). \tag{2.14}$$

To avoid confusion with notations for the final layer L, we denote all loss and error functions with $\mathcal{L}$. The final layer of the neural network will produce a vector of unnormalized probabilities $\boldsymbol{z}$, with $z_i \propto \log P(y = i|\boldsymbol{x})$ and $\boldsymbol{x}$ is the input vector. To use negative log-likelihood, we need to normalize the output vector such that the probabilities sum to 1. We can do this by applying the softmax function to each element of $\boldsymbol{z}$:

$$\text{softmax}(\boldsymbol{z})_i = \frac{\exp(z_i)}{\sum_j \exp(z_j)}. \tag{2.15}$$

Now we can obtain a target label as argmax($\boldsymbol{z}$). The loss function to be optimized during training depends on the task at hand. For classification tasks, probability-based functions such as categorical cross-entropy are more useful, whereas in regression tasks, distance functions such as mean squared error are more applicable.

2.3.5 Updating the Parameters of the FFNN

The parameters of any feedforward neural network can be updated with the help of back-propagation and a gradient-based optimization algorithm. Back-propagation is a learning algorithm that calculates the gradients in the network to minimize some error function $\mathcal{L}$ and can be attributed to multiple authors who independently discovered it (Bryson and Ho, 1969; Werbos and John, 1974; Parker, 1985; Rumelhart *et al.*, 1986) but we will describe the algorithm as outlined by Demuth *et al.* (2014). The algorithm prescribes how the parameters of any feedforward neural network can be optimized to learn an approximation function from a set of training examples. Training a neural network requires some form of supervision that accompanies the input vector $\boldsymbol{x}$, provided by the target vector $\boldsymbol{y}$. For a given time step t in the optimization procedure, a given input vector $\boldsymbol{x}$ with target vector $\boldsymbol{y}$ is passed to the input layer of the neural network. The signal propagates forward through the network to produce the final output vector, $\hat{\boldsymbol{y}}$, with the same dimensions as the target vector. The loss function $\mathcal{L}$ for the single training example, in this case squared error, is then computed as

$$\mathcal{L} = ||\hat{\boldsymbol{y}} - \boldsymbol{y}||^2. \tag{2.16}$$

The objective of back-propagation is to calculate the gradients that will minimize the loss for that time step. The calculated gradients are then used to update the parameters in the network with the help of a gradient-based optimization algorithm. One such algorithm is gradient descent, first introduced by Cauchy (1847). Gradient descent tries to find the global minimum of the loss function by updating the parameters in the opposite direction of the calculated gradients for that time step. The loss function $\mathcal{L}$ encapsulates all the weights and biases from which $\hat{\boldsymbol{y}}$ is computed, thus we can write our updating rule at time step $t+1$ as:

$$W_{ij}^{(l)}(t+1) = W_{ij}^{(l)}(t) - \rho \frac{\partial \mathcal{L}}{\partial W_{ij}^{(l)}(t)}, \tag{2.17}$$

$$b_j^{(l)}(t+1) = b_j^{(l)}(t) - \rho \frac{\partial \mathcal{L}}{\partial b_j^{(l)}(t)}. \tag{2.18}$$

The learning rate, $\rho > 0$, controls the step size of each parameter update per time step. Here $W_{ij}^{(l)}(t)$ refers to the weight factor of the signal received by neuron j in layer l from neuron i in layer $l-1$ at time step t, and $b_j^{(l)}(t)$ refers to the bias term of neuron j in layer l at time step t. Working backwards from the final layer L, we can model the relationship between $\mathcal{L}$ and the weights and biases of the final layer using Equation 2.13 as

$$\mathcal{L} = \sum_{j=1}^{N_L} \left(\alpha^{(L)} \left(\sum_{i=1}^{N_{L-1}} W_{ij}^{(L)} a_i^{(L-1)} + b_j^{(L)} \right) - y_j \right)^2, \tag{2.19}$$

while omitting the dependence on time step t and with $\alpha^{(L)}$ denoting to the activation function of the final layer. The partial derivatives of $\mathcal{L}$ with respect to parameters in the final layer L are calculated using the chain rule:

$$\frac{\partial \mathcal{L}}{\partial W_{ij}^{(L)}} = \sum_{n=1}^{N_L} \frac{\partial \mathcal{L}}{\partial n_n^{(L)}} \frac{\partial n_n^{(L)}}{\partial W_{ij}^{(L)}}, \tag{2.20}$$

$$\frac{\partial \mathcal{L}}{\partial b_{j}^{(L)}} = \sum_{n=1}^{N_L} \frac{\partial \mathcal{L}}{\partial n_n^{(L)}} \frac{\partial n_n^{(L)}}{\partial b_{j}^{(L)}}. \tag{2.21}$$

We define the sensitivity $s_n^{(l)}$ as the partial derivative of $\mathcal{L}$ with respect to the net input of the individual neuron n, and thus the sensitivity is given as:

$$s_n^{(l)} = \frac{\partial \mathcal{L}}{\partial n_n^{(l)}}. \tag{2.22}$$

Thus we can rewrite Equations 2.20 and 2.21 for the final layer L as

$$\frac{\partial \mathcal{L}}{\partial W_{ij}^{(L)}} = \sum_{n=1}^{N_L} s_n^{(L)} \frac{\partial n_n^{(L)}}{\partial W_{ij}^{(L)}}, \tag{2.23}$$

and

$$\frac{\partial \mathcal{L}}{\partial b_{j}^{(L)}} = \sum_{n=1}^{N_L} s_n^{(L)} \frac{\partial n_n^{(L)}}{\partial b_{j}^{(L)}}. \tag{2.24}$$

This is also true for any l. As previously stated, the activation function α should be differentiable, and thus we can use the chain rule directly to obtain the sensitivity for the final layer L:

$$s_n^{(L)} = 2\left(a_n^{(L)} - y_n\right) \dot{\alpha}^{(L)}(n_n^{(L)}), \qquad n = 1, 2, \ldots, N_L, \tag{2.25}$$

where $\dot{\alpha}$ is the first derivative of α. The net input of neuron n in the final layer L is given by Equation 2.13. Thus we can rewrite the remaining part of the partial derivatives in Equations 2.23 and 2.24 as

$$\frac{\partial n_n^{(L)}}{\partial W_{ij}^{(L)}} = \frac{\partial}{\partial W_{ij}^{(L)}} \left(\sum_{k=1}^{N_{L-1}} W_{kn}^{(L)} a_k^{(L-1)} + b_n^{(L)} \right) = a_i^{(L-1)}, \tag{2.26}$$

and

$$\frac{\partial n_n^{(L)}}{\partial b_{j}^{(L)}} = \frac{\partial}{\partial b_{j}^{(L)}} \left(\sum_{k=1}^{N_{L-1}} W_{kn}^{(L)} a_k^{(L-1)} + b_n^{(L)} \right) = 1. \tag{2.27}$$

Thus, the partial derivatives of $\mathcal{L}$ with respect to the weights and the biases in the final layer L can be rewritten simply as

$$\frac{\partial \mathcal{L}}{\partial W_{ij}^{(L)}} = s_j^{(L)} a_i^{(L-1)}, \tag{2.28}$$

$$\frac{\partial \mathcal{L}}{\partial b_{j}^{(L)}} = s_j^{(L)}. \tag{2.29}$$

The updating rules, with time step dependency included, are

$$\begin{aligned} W_{ij}^{(l)}(t+1) &= W_{ij}^{(l)}(t) - \rho\, s_j^{(l)}(t) a_i^{(l-1)}(t)\,, && (2.30) \\ b_j^{(l)}(t+1) &= b_j^{(l)}(t) - \rho\, s_j^{(l)}(t). && (2.31) \end{aligned}$$

To make use of these rules, we rely on the sensitivity vector $\boldsymbol{s}^{(l)}$, with $l = 1, \ldots, L-1$. The components of $\boldsymbol{s}^{(l)}$ are defined in Equation 2.22. From the equation, it is clear that we need to understand how $\mathcal{L}$ is dependent on $n_j^{(l)}$, the linear combination of all the inputs of neuron j in layer l. One thing worth noting is that $n_j^{(l)}$ depends in turn on $n_i^{(l-1)}$ for $i = 1, 2, \ldots, N_{l-1}$, since the net input propagates forward from layer $l-1$ to layer l, conditional on the activation of the neurons in the previous layer. More specifically,

$$\begin{aligned} n_j^{(l)} &= \sum_{i=1}^{N_{l-1}} W_{ij}^{(l)} a_i^{(l-1)} + b_j^{(l)} && (2.32) \\ &= \sum_{i=1}^{N_{l-1}} W_{ij}^{(l)} \alpha^{(l-1)}(n_i^{(l-1)}) + b_j^{(l)}. \end{aligned}$$

Thus, the sensitivity of layer $l-1$ can be written as

$$\begin{aligned} s_j^{(l-1)} &= \frac{\partial \mathcal{L}}{\partial n_j^{(l-1)}} = \sum_{i=1}^{N_l} \frac{\partial \mathcal{L}}{\partial n_i^{(l)}} \frac{\partial n_i^{(l)}}{\partial n_j^{(l-1)}} && (2.33) \\ &= \sum_{i=1}^{N_l} s_i^{(l)} \frac{\partial}{\partial n_j^{(l-1)}} \left(\sum_{k=1}^{N_{l-1}} W_{ki}^{(l)} \alpha^{(l-1)}(n_k^{(l-1)}) + b_i^{(l)} \right) \\ &= \sum_{i=1}^{N_l} W_{ji}^{(l)} s_i^{(l)} \dot{\alpha}^{(l-1)}(n_j^{(l-1)}) \\ &= \dot{\alpha}^{(l-1)}(n_j^{(l-1)}) \sum_{i=1}^{N_l} W_{ji}^{(l)} s_i^{(l)}. \end{aligned}$$

Clearly, the sensitivity vector of layer $l-1$ is dependent on the sensitivity vector of layer l, but the activations in layer l depend on the activations in layer $l-1$. We thus have two opposing flows of information, hence the name, back-propagation.

2.4 Recurrent Neural Networks

Feedforward neural networks are suitable for modelling non-sequential data. However, suppose we wish to model an ordered sequence of values $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(\tau)}$. Recurrent neural networks (RNNs) can be used for a variety of ordered sequence processing applications including financial time series forecasting, speech recognition, text and music generation, question-answering, machine translation and video activity recognition. Recurrent neural networks are able to scale to much longer sequences than feedforward neural networks without having to increase the model size. They can also process variable-length sequences, and the parameters are shared through time. We describe the definition and training process of recurrent neural networks as outlined by Goodfellow *et al.* (2016).

2.4.1 Recurrent Connections

The structure of a recurrent neural network is comparable to that of a standard feedforward neural network, with the addition of recurrent connections in the hidden units that allow for information to persist through time. In the terminology associated with recurrent neural networks, a time step t refers to the elements at time t in the ordered sequence, and not the time step associated with gradient updates (as in the previous section). The recurrent connections enable the model to recognize and recall temporal and spatial patterns. RNN layers can take on different architectures with different sets of parameters depending on the sequence modelling task at hand. For example, in Figure 2.2, the model outputs a value for every input value and the hidden units have recurring connections. Other architectures might take in an entire sequence of values and only produce a single output. For our notation of the unrolling of the recurrent connections, we drop the superscript indicating the layer, and replace it with a superscript indicating the time step within the sequence.

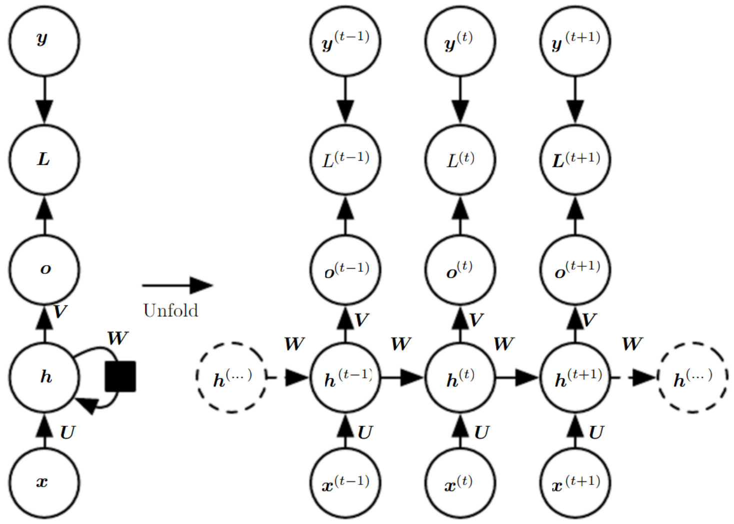

Figure 2.2: Time-unfolded computational graph of a recurrent hidden unit (Goodfellow *et al.*, 2016)

The hidden state vector for time step t is defined as

$$\boldsymbol{h}^{(t)} = f\left(\boldsymbol{h}^{(t-1)}, \boldsymbol{x}^{(t)}; \theta\right) \tag{2.34}$$

where $\boldsymbol{x}^{(t)}$ is the input of the sequence at time step t, f is the recursive function that maps the hidden state from time step $t-1$ to time step t, and θ is used to parameterize f. The hidden state $\boldsymbol{h}^{(t)}$ takes as input both the hidden state at the previous time step, $\boldsymbol{h}^{(t-1)}$, and the input of the sequence at the current time step, $\boldsymbol{x}^{(t)}$, and can thus be viewed as a fixed-length vector that encapsulates all the information of the sequence $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(t)}$. The unrolling of a single hidden unit with recurring connections can be observed in Figure 2.2.

2.4.2 Learning through Time

The forward propagation through the unrolled computational graph of the hidden unit $\boldsymbol{h}$, shown in Figure 2.2, can be demonstrated using the following update equations for each time step t and component j:

$$h_j^{(t)} = \alpha\left(b_j + \sum_i W_{ij} h_i^{(t-1)} + \sum_i U_{ij} x_i^{(t)}\right), \tag{2.35}$$

$$o_j^{(t)} = c_j + \sum_i V_{ij} h_i^{(t)}, \tag{2.36}$$

where $x_i^{(t)}$ is the ith component of the input at time step t, b_j and c_j are the bias terms for the hidden component j, α is the nonlinear activation function used for the hidden unit vector $\boldsymbol{h}^{(t)}$, U_{ij} denotes the weight of the signal passed from input $x_i^{(t)}$ to hidden component j, W_{ij} denotes the weight of the signal passed from the hidden component i at time step $t-1$ to the hidden component j at time step t, V_{ij} denotes the weight of the signal passed from the hidden component i to the output component j, and $o_j^{(t)}$ denotes the output component j at time step t. The vector $\boldsymbol{o}^{(t)}$ contains the unnormalized outputs at time step t. Applying the softmax function to $\boldsymbol{o}^{(t)}$ results in the normalized output vector $\hat{\boldsymbol{y}}^{(t)}$. A loss function $\mathcal{L}$, in this case negative log-likelihood, measures the difference between each normalized output $\hat{\boldsymbol{y}}^{(t)}$ and the corresponding target $\boldsymbol{y}^{(t)}$ for time step t. For an ordered sequence of input values paired with a sequence of target values, the total loss across the unrolled computational graph is the sum of the losses over $t = 1, \ldots, \tau$:

$$\mathcal{L}\left(\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(\tau)}\}, \{\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(\tau)}\}\right) = -\sum_t \log p_{\text{model}}\left(\hat{\boldsymbol{y}}^{(t)} | \boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(t)}\right). \tag{2.37}$$

Here $p_{\text{model}}\left(\hat{\boldsymbol{y}}^{(t)} | \boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(t)}\right)$ is the normalized output of the model for time step t. To compute the gradients the inputs have to be propagated forward through the unrolled computational graph shown in Figure 2.2. The errors are computed and then back-propagated through the unrolled graph, much like the back-propagation for feedforward neural networks. The training of recurrent neural networks is expensive, and cannot be parallelized due to its sequential nature. The memory cost and run-time complexity are both $O(\tau)$. While in theory the recurrent neural network is a simple but powerful model, in practice it can be difficult to train. As the number of recurrent neural network layers increases, we see problems arising with computing the gradient updates. Because of the unrolling of the graph per time step, computing the gradients can be slow and sometimes we may run into the vanishing/exploding gradient problem. If we have sigmoid activations throughout our network, many of the recurrent units will have very small derivatives. The sequential multiplication of these derivatives can result in a gradient too small, essentially vanishing, for effective training. Recurrent neural networks also struggle to retain information that spans over many time steps in the data. The reason for this long-term memory loss is that the magnitude of the gradients of long-term interactions are much smaller than that of the short-term interactions (Bengio *et al.*, 1994). The architecture of the vanilla recurrent neural network only allows for short-term memory retention, for all other events the gradients simply become insignificantly small. Thus, recurrent neural networks are well-suited to model short-term dependencies but not long-term dependencies (Pascanu *et al.*, 2012). Many solutions to the inherent weaknesses of recurrent neural networks have been proposed, such as using ReLU activations instead of sigmoid which does not result in such a small derivative, and adding feedback loops or forget gates between different recurrent units to allow for modelling of long-term dependencies.

2.4.3 Long Short-term Memory Networks

In an attempt to address some of the shortcomings of the recurrent neural network, Hochreiter and Schmidhuber (1997) introduced the long short-term memory (LSTM) network, a variant of the original recurrent neural network architecture with internal recurrence. It shares the architecture of the recurrent neural networks, but with more parameters, such as gating units and an internal state unit that explicitly address the long-term dependency problem of the recurrent neural network. We describe the LSTM cell as outlined by Goodfellow *et al.* (2016). Each of the gating units has its own set of parameters. The first addition is the state unit $s_j^{(t)}$ at time step t and cell j (henceforth the notation) which has a linear self-loop. Another key addition to the original recurrent neural network architecture is the forget gate unit $f_j^{(t)}$, that controls the self-loop weight, calculated as:

$$f_j^{(t)} = \sigma\Big(b_j^f + \sum_i U_{ij}^f x_i^{(t)} + \sum_i W_{ij}^f h_i^{(t-1)}\Big), \tag{2.38}$$

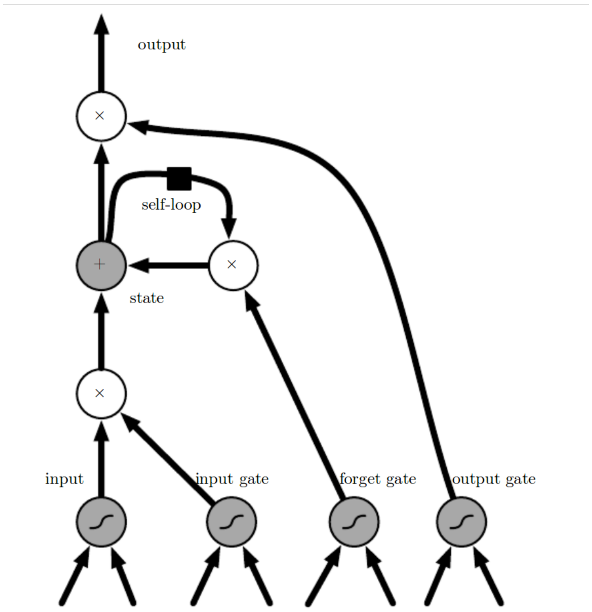

Figure 2.3: An LSTM recurrent unit (Goodfellow *et al.*, 2016)

where σ is the sigmoid function that scales the input to a range between 0 and 1, $x_i^{(t)}$ is the ith component of the input at time step t, $h_i^{(t-1)}$ contains the output of LSTM cell i at time step t, and b_j^f, U_{ij}^f, and W_{ij}^f parameterize the forget gate. The idea is that if $f_j^{(t)}$ is close to 0, the LSTM cell will "forget" what occurred in the previous state s_j^{t-1}, and if not, then it should "remember". The internal state $s_j^{(t)}$ is thus updated as:

$$s_j^{(t)} = f_j^{(t)} s_j^{(t-1)} + g_j^{(t)} \sigma\Big(b_j + \sum_i U_{ij} x_i^{(t)} + \sum_i W_{ij} h_i^{(t-1)}\Big), \tag{2.39}$$

where b_j, U_{ij}, and W_{ij} parameterize the internal state of the jth component of the LSTM cell. The external input gate unit $g_j^{(t)}$ is computed as follows:

$$g_j^{(t)} = \sigma\Big(b_j^g + \sum_i U_{ij}^g x_i^{(t)} + \sum_i W_{ij}^g h_i^{(t-1)}\Big), \tag{2.40}$$

where b_j^g, U_{ij}^g, and W_{ij}^g parameterize the external input gate. The output $h_j^{(t)}$ can be "shut off" using the output gate $q_j^{(t)}$ as follows:

$$h_j^{(t)} = \tanh(s_j^{(t)}) q_j^{(t)}, \tag{2.41}$$

$$q_j^{(t)} = \sigma\Big(b_j^o + \sum_i U_{ij}^o x_i^{(t)} + \sum_i W_{ij}^o h_i^{(t-1)}\Big), \tag{2.42}$$

where tanh is the hyperbolic tangent and b_j^o, U_{ij}^o, and W_{ij}^o parameterize the output gate. The inner workings of the LSTM cell are detailed in Figure 2.3.

2.4.4 Gated Recurrent Units (GRU)

In contrast to the LSTM unit, the gated recurrent unit (GRU) (Cho *et al.*, 2014) has a single unit that controls the forget gate and the decision to update the hidden states. We describe the functionality of the GRU as outlined by (Goodfellow *et al.*, 2016). The hidden state unit $h_j^{(t)}$ is updated using the following equation:

$$h_j^{(t)} = u_j^{(t-1)} h_j^{(t-1)} + (1 - u_j^{(t-1)}) \sigma\Big(b_j + \sum_i U_{ij} x_i^{(t)} + \sum_i W_{ij} r_i^{(t-1)} h_i^{(t-1)}\Big), \quad (2.43)$$

where $u_j^{(t)}$ represents the update gate and $r_j^{(t)}$ the reset gate for cell j at time step t. The two parameters are defined as:

$$u_j^{(t)} = \sigma\Big(b_j^u + \sum_i U_{ij}^u x_i^{(t)} + \sum_i W_{ij}^u h_i^{(t)}\Big), \quad (2.44)$$

$$r_j^{(t)} = \sigma\Big(b_j^r + \sum_i U_{ij}^r x_i^{(t)} + \sum_i W_{ij}^r h_i^{(t)}\Big), \quad (2.45)$$

where b_j^u, U_{ij}^u, and W_{ij}^u parameterize the update gate and b_j^r, U_{ij}^r, and W_{ij}^r parameterize the reset gate for cell j. The update gate $u_j^{(t)}$ can choose to copy, or ignore components of the old state, or linearly vary between these two extremes. The reset gate $r_j^{(t)}$ introduces nonlinearity by controlling which components of the previous hidden state should be used to compute the current hidden state, while relying on output of $u_j^{(t)}$.

2.5 Metric Learning

In traditional information retrieval settings, we would extract features from objects and then apply a predefined distance metric to calculate their similarity. This means that the feature representations and the metric are not learned in conjunction. This inherent weakness can be addressed with metric learning, where the objective is to take some predefined metric and adapt it to the training data (Bellet *et al.*, 2013). A metric f is a measure of similarity between two objects x and y, and must satisfy the following axioms:

- $f(x, y) \geq 0$,
- $f(x, y) = 0$, if and only if $x = y$,
- $f(x, y) = f(y, x)$,
- $f(x, z) \leq f(x, y) + f(y, z)$.

There are two types of distance metrics: predefined metrics, e.g. Euclidean distance, and learned metrics which can only be defined with knowledge of the data. An example of a learned metric is the Mahalanobis distance, which scales the Euclidean distance between two points using the covariance matrix observed from the training data.

2.5.1 Predefined Distance Metrics

A naive but effective distance metric for count-based vectors would be to measure the set overlap using Jaccard Similarity Index, originally introduced by Paul Jaccard in 1901. If A is the set of unique tokens found in object A, and B is the set of unique tokens found in object B, then the Jaccard Similarity between the two objects can be calculated as follows:

$$\text{Jaccard}(A, B) = \frac{|A \cap B|}{|A \cup B|}, \tag{2.46}$$

thus normalizing the number of overlapping tokens in both sets by the total number of unique tokens found in both sets. Other distance metrics that can be applied to vectors include Euclidean distance and cosine distance:

$$\begin{aligned} \text{Euclidean}(\boldsymbol{p}, \boldsymbol{q}) &= \sqrt{\sum_i (q_i - p_i)^2}, && (2.47) \\ \text{Cosine}(\boldsymbol{p}, \boldsymbol{q}) &= 1 - \frac{\sum_i q_i p_i}{\sqrt{\sum_i q_i^2}\sqrt{\sum_i p_i^2}}. && (2.48) \end{aligned}$$

The Euclidean distance is equivalent to the L2 distance, and the complement of the cosine distance can be interpreted as the L2-normalized dot product between two vectors.

2.5.2 Exact Nearest Neighbour Retrieval

Formally, the exact nearest neighbour problem can be defined as finding the nearest neighbour item $\boldsymbol{q}^*$ for a search query $\boldsymbol{q}$ from a set of N vectors $\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$, where $\boldsymbol{x}_i$ lies in an M-dimensional space $\mathbb{R}^M$, such that

$$\boldsymbol{q}^* = \arg\min_{\boldsymbol{x} \in \mathcal{X}} \; D(\boldsymbol{q}, \boldsymbol{x}), \tag{2.49}$$

where $D(\cdot)$ is a distance function (Wang *et al.*, 2014). A generalization of the exact nearest neighbour problem is k-nearest neighbours (k-NN). Classification with k-NN can be performed using a majority voting on the labels associated with the k items found nearest to the query item $\boldsymbol{q}$ (Cover and Hart, 1967). The vote of each item in the set of k nearest items can either be

- weighted equally (uniform weights), or
- weighted according to the inverse of their distance to the query item.

During training only an implicit boundary is learned, therefore the k-NN is a non-parametric model and computationally inexpensive to train. Naive inference (linear search), however, can become very expensive. Finding the 1-nearest neighbour for a single test query is $O(NM)$, where N is the number of training samples. As the number of dimensions grows, the data points become more sparse and nearest neighbour search becomes inefficient. This problem is known as the Curse of Dimensionality.

2.5.3 Similarity Learning with Siamese Networks

Siamese networks were independently introduced by both Bromley *et al.* (1993) and Baldi and Chauvin (1993) as a similarity-learning algorithm for signature verification and fingerprint verification, respectively. Instead of predicting a class label, these networks directly measure the similarity between samples. This is useful for scenarios where the number of classes is very large or unknown during training, or where there is a only a few training samples per class (Chopra *et al.*, 2005).

The applications of Siamese networks have since been extended to face recognition (Chopra *et al.*, 2005; Taigman *et al.*, 2014), one-shot image recognition (Koch and and, 2015), calculating semantic similarity between sentences (Mueller and Thyagarajan, 2016) and natural language inference (Conneau *et al.*, 2017).

Early approaches to training Siamese networks made use of pairs of similar and dissimilar samples. Bromley *et al.* (1993) used a modified version of the back-propagation algorithm to minimize the angle between embeddings of signatures of the same person and maximize the angle between embeddings of a real and forged signature pair. For facial verification, Chopra *et al.* (2005) learned the parameters of the Siamese network using a contrastive loss function. For a given pair of embeddings $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ produced by the Siamese network for input samples $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$, respectively, and a boolean value y that indicates whether the two samples are similar ($y = 0$) or not ($y = 1$), the contrastive loss is calculated as:

$$\mathcal{L}(\boldsymbol{x}_1, \boldsymbol{x}_2, y) = (1-y)\mathcal{L}_S(D(\boldsymbol{x_1}, \boldsymbol{x_2})) + y\mathcal{L}_D(D(\boldsymbol{x}_1, \boldsymbol{x_2})). \tag{2.50}$$

Here $D(\cdot)$ is the chosen distance function, $\mathcal{L}_s$ is the partial loss function for similar pairs and $\mathcal{L}_D$ is the partial loss function for dissimilar pairs. Chopra *et al.* (2005) showed that choosing a monotonically increasing function for $\mathcal{L}_S$ and a monotonically decreasing function for $\mathcal{L}_D$ guarantees that minimizing $\mathcal{L}$ will decrease the distance between similar pairs and vice versa. For the face verification task, Chopra *et al.* chose their contrastive loss function to be

$$\mathcal{L}(\boldsymbol{x}_1, \boldsymbol{x}_2, y) = (1-y)\frac{2}{q}(D(\boldsymbol{x_1}, \boldsymbol{x_2}))^2 + (y)(2q)\exp\left(\frac{-2.77}{q}D(\boldsymbol{x}_1, \boldsymbol{x_2})\right), \tag{2.51}$$

where the constant q is equal to the upper bound of $D(\cdot)$. One of the drawbacks of pairwise matching is there is no way to ground the distances between pairs of samples. A triplet loss function aims to minimize the relative distance rather than the absolute distance between similar pairs. A training triplet is given as $(\boldsymbol{x}_a, \boldsymbol{x}_p, \boldsymbol{x}_n)$, where $\boldsymbol{x}_a$ is the embedding of the anchor sample, $\boldsymbol{x}_p$ is the positive sample (similar to the anchor), and $\boldsymbol{x}_n$ is the negative sample (dissimilar to the anchor). Then the triplet loss function for a single training triplet is given as:

$$\mathcal{L}(\boldsymbol{x}_a, \boldsymbol{x}_p, \boldsymbol{x}_n) = \max(0, m + D(\boldsymbol{x}_a, \boldsymbol{x}_p) - D(\boldsymbol{x}_a, \boldsymbol{x}_n)), \tag{2.52}$$

where $D(\cdot)$ is the chosen distance function, and m is the margin between $D(\boldsymbol{x}_a, \boldsymbol{x}_p)$ and $D(\boldsymbol{x}_a, \boldsymbol{x}_n)$. The objective of the triplet loss is to ensure that the negative sample is embedded further from the anchor than the positive sample, by a fixed margin m.

2.6 Chapter Summary

In Section 2.3 we discuss the inner workings of artificial neural networks and how the architecture draws inspiration from the human brain. The simplest feedforward neural network architecture has an input layer, one or more hidden layers, and an output layer. The challenge is learning the best set of parameters to approximate the function f that maps the input $\boldsymbol{x}$ to the target $\boldsymbol{y}$. During training, the back-propagation algorithm computes the gradient updates necessary to reduce the difference between the real function and its approximation. Reaching the global minimum of the error function can be achieved by adjusting the weights in steps proportional to the negative of the gradients, a process called gradient descent. Feedforward neural networks can be used to model complex distributions, however they are limited in their capabilities to model time-dependent and sequential data points. In Section 2.4, we introduce recurrent neural networks that are able to deal with ordered sequences of variable length. The recurrent connections between the hidden units allow for information to persist through time. As the sequences become longer, retaining information over long periods of time becomes harder and the gradients become too small, resulting in a phenomenon called vanishing or exploding gradients. One way to deal with this is by adding feedback loops and control gates to better model the long-term dependencies found in temporal sequences. Long short-term memory networks and gated recurrent units do just that. In Section 2.5 we introduce metric learning as the task of adapting predefined metrics to the training data to better learn pairwise similarity. One such technique is the Siamese triplet loss that learns to rank similarity based on relative comparisons of objects.

Chapter 3

Language Modelling

3.1 Introduction

Natural language, or human language, is a set of rules (a grammar) that govern the composition of clauses, phrases, and words (a vocabulary). Across the world, there are more than 7000 languages spoken today, with only about half having had their grammar recorded and written down by linguists. Linguistics and computer science intersect in the field of language modelling. Language modelling is a sub-field of artificial intelligence, where one uses computers to model statistical properties which aim to capture the grammar and context found in a given language. Prior to the recent renewed interest in neural networks, computational linguistics and machine translation relied heavily on linguistic theory, hand-crafted rules, and statistical modelling. Today, the most advanced computational linguistic models are developed by combining large annotated corpora and deep learning models, in the absence of linguistic theory and hand-crafted features. Its applications are widely studied, and include machine translation, topic modelling, part-of-speech tagging, sentiment analysis, and question-answering. Feedforward neural networks have been used to create embeddings of words that capture semantic information and, in some cases, context and polysemy. Recurrent neural networks are ideal for language modelling, as they can process variable-length sequences, and the final hidden representation can be used as an encoding of the entire sequence. The notion that we can represent language in a mathematical form comes from the assumption that language takes on an inferable distributional structure.

3.2 The Distributional Hypothesis

The distributional hypothesis of Harris (1954) states that language can be structured with respect to various features, in the sense that, for example, a set of phonemes can be structured with respect to some feature in an organized system of statements. This system would describe the members of the set and their interactions with other members of the set. Language can thus, according to Harris be described in terms of a distributional structure, where the distribution is the sum of all its domains. The distributional structure of a language can be analysed using a few key concepts:

- elements,
- similarity,
- dependence,
- substitution,
- and domains.

Firstly, any form of speech or text can be divided into smaller discrete parts (elements), where we can then analyse the co-occurrence of the elements with other elements. Some elements may be similar to one another, in the sense that they may have overlapping or even identical distributions. These elements can be grouped together. When subdividing speech or text into elements, we can obtain a smallest set of elements where all the elements of the set are completely dependent within some domains, meaning that they will never not occur together, not necessarily contiguous, and always in a particular order. These elements with complete serial dependence then become a single element. There are, however, dependencies between elements that are less rigid. For example, we may find the morpheme *king* always occurs close to *ly* or *dom*, in the forms *kingly* and *kingdom*. The morpheme *dom* with which *king* occurs, may still have all of its properties when it occurs without *king*, i.e. all the predecessors of *dom* are nouns. We may also find that various elements have identical or near-identical patterns of co-occurrence, leading to opportunities for substitution within certain domains. The substitutability and serial dependence described above always occur within a domain, e.g. stems and suffixes within words, words within phrases, and phrases within sentences. Using document encoding techniques we can infer and study the relations among elements and the behaviour within domains. These techniques can roughly be divided into either count- or prediction-based approaches.

3.3 Count-based Word Vectors

Besides rule-based approaches, there were quite a few useful language modelling techniques that preceded neural network language modelling, many of which being count-based. These techniques range from simple encodings, such as a bag-of-words approach, to more complex methods that factorize the co-occurrence matrices, such as GloVe.

3.3.1 Bag-of-words

One of the earliest references to a “bag-of-words” is made by Harris (1954), where he states that “language is not merely a bag of words but a tool with particular properties which have been fashioned in the course of its use.” A simple way to represent a document is the bag-of-words technique, where we completely disregard the word order and grammatical structure. When using bag-of-words in conjunction with a classification task such as sentiment classification, each feature is the count of a word for a given document. For example, suppose our training dataset consists of two movie reviews, positive and negative, of the movie *Titanic* from the website *Metacritic*:

{*“A film that sweeps us away into a world of spectacle, beauty and excitement,”*, *“Cameron manhandles the real story, scavenging it for his own puny narrative.”*}

To create a vocabulary from all the reviews, we first remove all punctuation, cast the words to lowercase, and then we separate each document into a list of words as follows:

`review_1` = [‘a’, ‘film’, ‘that’, ‘sweeps’, ‘us’, ‘away’, ‘into’, ‘a’, ‘world’, ‘of’, ‘spectacle’, ‘beauty’, ‘and’, ‘excitement’]
`review_2` = [‘cameron’, ‘manhandles’, ‘the’, ‘real’, ‘story’, ‘scavenging’, ‘it’, ‘for’, ‘his’, ‘own’, ‘puny’, ‘narrative’]

Then we take the unique words found in the training examples and order them to produce a training vocabulary:

`vocab` = [‘a’, ‘and’, ‘away’, ‘beauty’, ‘cameron’, ‘excitement’, ‘film’, ‘for’, ‘his’, ‘into’, ‘it’, ‘manhandles’, ‘of’, ‘own’, ‘puny’, ‘purposes’, ‘real’, ‘scavenging’, ‘spectacle’, ‘story’, ‘sweeps’, ‘that’, ‘the’, ‘us’, ‘world’]

Now we can represent each training example as a bag-of-words encoding where each feature represents the count of a unique word found in the review:

“A film that sweeps us away into a world of spectacle, beauty and excitement” $\rightarrow [2, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1]$
“Cameron manhandles the real story, scavenging it for his own puny narrative.” $\rightarrow [0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0]$

The overlap between the two reviews is small, so a machine learning model may easily learn which words can be associated with positive labels and so forth. Bag-of-words encodings of documents are simple yet powerful, and serve as good baseline encodings.

3.3.2 Term Frequency-Inverse Document Frequency (TF-IDF)

This encoding captures the importance of a word relative to a document in a given corpus. This is often useful for tasks such as topic modelling, as it diminishes the importance of words that appear ubiquitously throughout the corpus, and highlights the importance of words that are rare but often the strongest indicator of a topic. The TF-IDF metric for term i and document j is the multiplication of the term frequency TF_{ij} and the inverse document frequency IDF_i. We describe the TF-IDF metric as it is defined by Rajaraman and Ullman (2011). Suppose we have a corpus containing N documents. Then we define the term frequency TF_{ij} to be

$$TF_{ij} = \frac{f_{ij}}{\max_k f_{kj}}, \tag{3.1}$$

where f_{ij} is the frequency of the term (word) i in document j. Thus, TF_{ij} measures the frequency of a term i relative to the most frequent term in document j. The inverse-document frequency (IDF) is calculated as follows:

$$IDF_i = \log_2\left(\frac{N}{n_i}\right), \tag{3.2}$$

where n_i is the number of times term i appears in the corpus containing N documents. The implication here is that a high-scoring term is likely to be more informative of the document topic than a low-scoring term.

3.3.3 Latent Semantic Analysis (LSA)

Latent semantic analysis (LSA), first introduced by Deerwester *et al.* (1990), allows for determining the similarity of words by analysing large text corpora. A term-document matrix describes the frequency of terms (words) as they occur in documents (usually sentences), where the rows correspond to the terms and the columns to the documents. The goal is to map the high-rank term-document matrix to a low-rank space while preserving the most important information. This then allows for document classification and comparing documents in a low-dimensional space. LSA involves applying a commonly-used dimensionality reduction technique to the term-document matrix, namely singular value decomposition (SVD). SVD is defined as follows:

$$\boldsymbol{M} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top, \tag{3.3}$$

where $\boldsymbol{M}$ is the $m \times n$ term-document matrix, $\boldsymbol{U}$ is an $m \times m$ orthogonal matrix, $\boldsymbol{\Sigma}$ is an $m \times n$ rectangular diagonal matrix where the entries on the diagonal, σ_i, are the non-negative singular values of $\boldsymbol{M}$, and $\boldsymbol{V}$ is an $n \times n$ orthogonal matrix. The columns of $\boldsymbol{U}$ and the columns of $\boldsymbol{V}$ are called respectively the left-singular and right-singular vectors of $\boldsymbol{M}$ (Golub and Reinsch, 1970). We can order the singular vectors according to their corresponding singular values in $\boldsymbol{\Sigma}$, such that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_m$. For a lower-rank reconstruction of the data, we choose a k such that $k < m$. We then limit the matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ to their first k column vectors, and call them $\boldsymbol{U}_k$ and $\boldsymbol{V}_k$. We also reduce the size of the diagonal matrix $\boldsymbol{\Sigma}$ to the first k singular values, and name it $\boldsymbol{\Sigma}_k$. Then we reconstruct the low-rank matrix $\boldsymbol{M}_k$ as follows:

$$\boldsymbol{M}_k = \boldsymbol{U}_k \boldsymbol{\Sigma}_k \boldsymbol{V}_k^\top. \tag{3.4}$$

We define the cumulative explained variance ratio α retained in the low-rank matrix as

$$\alpha = \frac{\sum_{i=1}^{k} \sigma_i^2}{\sum_{i=1}^{m} \sigma_i^2}. \tag{3.5}$$

Latent semantic analysis effectively reduces the original term-document matrix to a low-rank matrix that consumes less memory while still preserving useful information. The original matrix is presumed to be noisy and sparsely populated. The method of SVD denoises the matrix and reduces the sparsity in the low-rank space. The singular vectors are the most efficient way to decompose the matrix, so terms that exhibit similar behaviour across different documents may be found to have similar representations in the low-rank space. Therefore, the low-rank matrix now represents dense word embeddings composed of the first k singular vectors of the original term-document matrix.

3.3.4 Pointwise Mutual Information (PMI)

According to Fano (1961), the mutual information between two sample points x and y, with respective probabilities $P(x)$ and $P(y)$, can be described as

$$I(x, y) = \log\left(\frac{P(x, y)}{P(x)P(y)}\right). \tag{3.6}$$

This concept can be extended to words $w \in V_W$ and their context words $c \in V_C$ to become the pointwise mutual information:

$$PMI(w, c) = \log\left(\frac{P(w, c)}{P(w)P(c)}\right), \tag{3.7}$$

where $P(w)$, $P(c)$, and $P(w, c)$ are the probabilities of observing word w, word c, and both words w and c within a fixed window size, respectively (Church and Hanks, 1990). The intuition behind this metric is as follows.

- If $P(w, c) \gg P(w)P(c)$, then there must be a strong correlation between w and c, and so $PMI(w, c) \gg 0$.

- If $P(w,c) \approx P(w)P(c)$, then w and c are nearly independently distributed, and $PMI(w,c) \approx 0$.
- If $P(w,c) \ll P(w)P(c)$, then w and c are negatively correlated, and so $PMI(w,c) \ll 0$.

Further, this metric helps to address magnitude imbalances within the co-occurrence matrix. However, it is computationally expensive, as it creates a matrix of $|V_W| \cdot |V_C|$ entries. Further, many co-occurrences are never observed, such that $PMI(w,c) = \log_2(0) = -\infty$. Levy and Goldberg (2014) argue that for these reasons, the PMI matrix is ill-defined. Levy and Goldberg goes on to addresses the concept of negatively correlated words by creating a positive pointwise mutual information (PPMI) and then shifting the matrix, resulting in a shifted positive pointwise mutual information (SPPMI) matrix:

$$\begin{aligned} PPMI(w,c) &= \max(0, PMI(w,c)), && (3.8) \\ SPPMI(w,c) &= \max(0, PPMI(w,c) - \log k), && (3.9) \end{aligned}$$

where $k > 1$. They show that factorizing both the SPPMI and the PPMI matrices to create word embeddings results in good performance on several semantic similarity tasks.

3.3.5 Global Vectors (GloVe)

Pennington *et al.* (2014) introduce a global log-bilinear method, called Global Vectors (GloVe), for creating word vectors that takes into account the global frequencies found in a given corpus. Their method combines matrix factorization and context windows. GloVe models are trained using only the non-zero elements of the word-word co-occurrence matrix, instead of the entire sparse matrix. The objective to minimize is as follows:

$$J = \sum_{i=1}^{V} \sum_{j=1}^{V} f(X_{ij})(\boldsymbol{w}_i^\top \tilde{\boldsymbol{w}}_j + b_i + \tilde{b}_j - \log X_{ij})^2 \tag{3.10}$$

where X_{ij} is the number of times the word i appears together with the context of word j, $\boldsymbol{w}_i \in \mathbb{R}^d$ the word vectors and $\tilde{\boldsymbol{w}}_j \in \mathbb{R}^d$ the context word vectors. The scalars b_i and $\tilde{b}_j$, refer to the bias terms associated with the word vectors and the context word vectors, respectively, and $f(\cdot)$ is the weighting function. The weighting function ensures that all co-occurrences are not weighted equally, and especially targets those that rarely or never happen. The chosen weighting function is defined as

$$f(x) = \min\left(1, \left(\frac{x}{x_{\max}}\right)^{\alpha}\right), \tag{3.11}$$

where $x_{\text{max}} = 100$ and $\alpha = 0.75$ are fixed by Pennington *et al.*. This function is non-decreasing and is relatively small for large values of x. The authors argue that their model, and count-based word embeddings in general, are fundamentally equivalent to prediction-based methods, as both probe the underlying co-occurrence statistics. However, the count-based methods, and especially GloVe, can be considered more efficient at capturing global statistics.

3.4 Prediction-based Word Embeddings

Bengio *et al.* (2003) first proposed an alternative to frequency-based word vectors where one extracts the vector representations of words from nonlinear statistical models, specifically neural networks. Neural networks can scale on much larger corpora than the restricted class of log-linear models, but they lack explainability. Bengio *et al.* outlined that this approach should have the following characteristics:

1. associate a real-valued, fixed-length feature vector with each vocabulary word,
2. the feature vectors encompass the joint probability distribution of the word sequences,
3. and parameters of the probability function are learned in conjunction with the representations.

This is the first definition of today's widely-used word embeddings. In the same publication Bengio *et al.* also proposed an architecture for estimating distributional vector representation using a neural network language model (NNLM). In 2008, Collobert and Weston showed how a feedforward neural network can be used to learn word embeddings by taking as input the words left and right of the input, and trying to predict the center word. However, as Mikolov *et al.* (2013*a*) noted, the simplest models can outperform the more complex models if they are given enough data to learn from. It is thus important to keep in mind that different models are suited for different settings, and there is no "one size fits all" model. A few years after Collobert and Weston (2008) and a decade after Bengio *et al.* (2003), Mikolov *et al.* (2013*a*), proposed two novel architectures for efficiently estimating continuous vector representations of words from large datasets, namely the continuous bag-of-words (CBOW) and the skip-gram model. An unexpected side-effect of these models is their ability to produce word vectors that seem to possess some approximate additive compositionality, such that one can do vector addition with words. For example:

$$\text{“man”} + \text{“royal”} = \text{“king”}.$$

In Bojanowski *et al.* (2017), the authors extend the capability of skip-gram by introducing subword information that takes into account the internal structure of words. This improves the quality of word vectors for rare words, and even allows for representations of unseen words. Building on the advances of all these language models is ELMo (Peters *et al.*, 2018) and BERT (Devlin *et al.*, 2018), two bi-directional LSTM models that take in sequences of characters and produce state-of-the art word embeddings that also encode the context of the sentence surrounding the word.

3.4.1 Feedforward Neural Network Language Model

In 2003, Bengio *et al.* proposed using feedforward neural networks to extract word vectors. This model consists of an input layer, a linear projection layer with dimensions $n \times D$, a hidden layer, and an output layer of dimension $|V|$, where V is a finite-sized vocabulary of words. The input vector is constructed from the n previous words in the training sequence, which are one-hot encoded. Training of the neural network using a sequence of words $w^{(1)}, \dots, w^{(T)}$ with $w_t \in V$ aims to maximize the following objective:

$$L = \frac{1}{T} \sum_{t=1}^{T} \log f(w^{(t)}, w^{(t-1)}, \dots, w^{(t-n+1)}; \theta) + R(\theta), \tag{3.12}$$

where T is length of the training sequence, $R(\theta)$ is the regularization term, and θ is the parameter set of the network. The projection layer learns to map any element to a real-valued vector of size D.

3.4.2 Continuous Bag-of-Words (CBOW)

The first of the two architectures proposed by Mikolov *et al.* (2013*a*) is the continuous bag-of-words (CBOW) model. This model is almost identical to the NNLM described above, with the key differences being the omission of the nonlinear hidden layer and the fact that the projection is shared across the surrounding window of words (the context). This means that the words surrounding the current word, $w^{(t)}$, are projected onto the same position, creating an averaged vector for the context, hence the name "*continuous* bag-of-words". The objective here is to correctly predict the center word given the surrounding window of words (see Figure 3.1). More formally, the aim is to maximize the average log probability for a sequence of training words $w^{(1)}, w^{(2)}, \dots, w^{(T)}$ and a context window size c:

$$J = \frac{1}{T} \sum_{t=1}^{T} \log p(w^{(t)} | w^{(t-c)}, \dots, w^{(t-1)}, w^{(t+1)}, \dots, w^{(t+c)}). \tag{3.13}$$

The vector representations of words can be extracted from the projection layer and used for various down-stream tasks, such as sentiment classification.

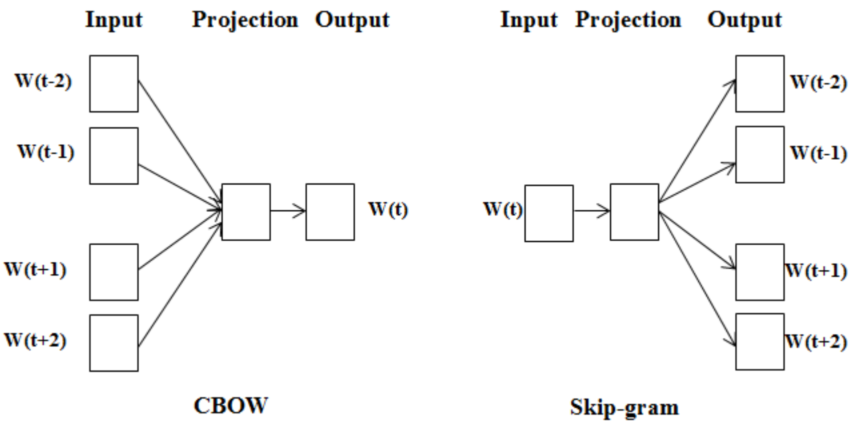

Figure 3.1: CBOW and skip-gram model (Mikolov *et al.*, 2013*a*)

3.4.3 Skip-gram Negative Sampling

In addition to the CBOW model, Mikolov *et al.* (2013*a*) also introduced the first iteration of the skip-gram model. The skip-gram model, like CBOW, omits the hidden layer of the NNLM, but differs from CBOW as it predicts the words surrounding the current word $w^{(t)}$ (see Figure 3.1). Formally stated, the skip-gram model aims to maximize the average log probability, for a given a sequence of training words $w^{(1)}, w^{(2)}, w^{(3)}, \ldots, w^{(T)}$ and a context window size c:

$$J = \frac{1}{T} \sum_{t=1}^{T} \sum_{-c \leq j \leq c, j \neq 0} \log p(w^{(t+j)} | w^{(t)}). \tag{3.14}$$

Increasing the window size c results in more training examples and possibly more informative vector representations, but at the cost of greater computation. The computational complexity of skip-gram is also much larger than that of CBOW. skip-gram incorporates down-sampling of distant words to give them less weight. While substituting $w^{(t+j)}$ with a and $w^{(t)}$ with b for simplicity, the simple skip-gram model defines $p(a|b)$ over the entire vocabulary V as:

$$p(a|b) = \frac{\exp(\boldsymbol{u}_a^\top \boldsymbol{v}_b)}{\sum_{w \in V} \exp(\boldsymbol{u}_w^\top \boldsymbol{v}_b)}, \tag{3.15}$$

where $\boldsymbol{v}_w$ and $\boldsymbol{u}_w$ are the respective input and output vector representations of the word w. Mikolov *et al.* (2013*c*) extend the simple skip-gram model by introducing negative sampling, where they replace the term $\log p(w^{(t+j)}|w^{(t)})$ in Equation 3.14 with

$$\log \sigma(\boldsymbol{u}_a^\top \boldsymbol{v}_b) + \sum_{i=1}^{k} \mathbb{E}_{w_i \sim P_n(w_i)} [\log \sigma(-\boldsymbol{u}_{w_i}^\top \boldsymbol{v}_b)]. \tag{3.16}$$

Again, we substitute $w^{(t+j)}$ with a and $w^{(t)}$ with b, and σ is the logistic sigmoid calculated as $1/(1+\exp(-\boldsymbol{u}^\top \boldsymbol{v}))$. The goal is to encourage the model to produce vector representations that are most dissimilar to the k noise vectors (negative samples) sampled from a noise distribution $P_n(w) \sim U(w)^{3/4}/Z$. Mikolov *et al.* (2013c) also address the imbalance between rare and common words by employing an aggressive sub-sampling technique, where the probability of discarding a word w is

$$P(w) = 1 - \sqrt{\frac{t}{f(w)}}, \tag{3.17}$$

with $f(w)$ defined as the frequency of word w. The authors found that subsampling resulted in a significant improvement in the quality of the vector representations of rare words, and accelerated learning.

Levy and Goldberg (2014) found that the skip-gram negative sampling method implicitly factorizes a PMI matrix. Gittens *et al.* (2017) provide a thorough theoretical explanation for the appearance of additive compositionality in the word vectors, and also show that skip-gram word vectors are maximally informative.

3.4.4 FastText

The techniques we have discussed up until now represent each word of the vocabulary with a distinct vector, thus ignoring the internal structure of words. Bojanowski *et al.* (2017) extend the skip-gram negative sampling (SGNS) model by also taking into account subword information in their technique called FastText. The authors argue that one can improve the vector representations of morphologically rich languages by learning vectors for character n-grams. Words are then represented as the sum of the vectors of their n-grams. The method extends on SGNS as follows. Each word is broken up into a set of character n-grams, with special boundary symbols at the beginning and end of each word. The original word is also retained in the set. For example, for $n = 3$ and the word *"there"*, we have the following n-grams:

$$\texttt{<th}, \texttt{the}, \texttt{her}, \texttt{ere}, \texttt{re>} \tag{3.18}$$

and the special feature `<there>`. This approach allows for a clear distinction between the feature `<the>` and `the`. In their experiments, Bojanowski *et al.* (2017) create n-grams for $3 \leq n \leq 6$. A vector representation for word w is calculated as the sum of the vector representations of its set of n-gram features, $\mathcal{G}_w$. We denote the vector of the n-gram feature $g \in \mathcal{G}_w$ as $\boldsymbol{z}_g$. The scoring function that replaces the term $\boldsymbol{u}_a^\top \boldsymbol{v_b}$ in Equation 3.15 is defined as

$$s(b,a) = \sum_{g \in \mathcal{G}_b} \boldsymbol{z}_g^\top \boldsymbol{v}_a, \tag{3.19}$$

where $\boldsymbol{v}_a$ is the vector representation of the word a. Again, we have substituted the terms $w^{(t+j)}$ with a and $w^{(t)}$ with b. Thus, we can rewrite the skip-gram negative sampling equations as

$$\log \sigma(\sum_{g \in \mathcal{G}_b} \boldsymbol{z}_g^\top \boldsymbol{v}_a) + \sum_{i=1}^{k} \mathbb{E}_{w_i \sim P_n(w_i)}[\log \sigma(-\sum_{g \in \mathcal{G}_b} \boldsymbol{z}_g^\top \boldsymbol{v}_{w_i})]. \tag{3.20}$$

This simple approach enables sharing representations across the vocabulary, can handle rare words better, and can even handle unseen words (a property the previous models lacked). It trains fast and requires no preprocessing of the words nor any prior knowledge of the language. The authors performed a qualitative analysis and showed that their technique outperforms models that do not take subword information into account.

3.4.5 Embeddings from Language Models (ELMo)

Peters *et al.* (2018) build on the advances of previous language models, by creating deep contextualized vector representations of words that can deal with unseen words, syntax and semantics, as well as polysemy (words taking on multiple meanings given the context). ELMo makes use of vectors derived from a pre-trained two-layer bi-directional LSTM model. The word vectors are derived from the whole sentence, which are fed in as a sequence of character tokens to provide the context. The bi-directional LSTM learns both a forward and backward language model, where the forward language model predicts the next token given the past sequence, and the backward language model predicts the previous token given the future sequence. For a sequence of N tokens $t^{(1)}, t^{(2)}, \ldots, t^{(N)}$, the model jointly maximizes the log-likelihood in the forward and backward directions:

$$\sum_{k=1}^{N} \left[\log p(t^{(k)} | t^{(1)}, ..., t^{(k-1)}; \overrightarrow{\Theta}_{LSTM}) + \log p(t^{(k)} | t^{(k+1)}, ..., t^{(N)}; \overleftarrow{\Theta}_{LSTM}) \right]. \tag{3.21}$$

Doing so enables the learning of shared parameters for both the token embeddings and the softmax layer in the forward and backward direction, while learning the bi-directional LSTM parameters ($\overrightarrow{\Theta}_{LSTM}$ and $\overleftarrow{\Theta}_{LSTM}$) separately for each direction. Following this is the computation of task specific linear combinations of all the bi-directional LSTM layers:

$$\textbf{ELMo}_k^{task} = \gamma^{task} \sum_{j=0}^{L} s_j^{task} \boldsymbol{h}_{k,j}^{LM}, \tag{3.22}$$

where $\boldsymbol{h}_{k,j}^{LM} = [\overrightarrow{\boldsymbol{h}}_{k,j}^{LM}; \overleftarrow{\boldsymbol{h}}_{k,j}^{LM}]$ is the context-dependent representation produced by the LSTM layer j at position k in the sequence, $\boldsymbol{s}^{task}$ is the softmax-normalized weights and γ^{task} is a task-specific scaling parameter. The pre-trained contextual representations can be used as is in downstram tasks, or further tuned on domain-specific data. ELMo has been shown to outperform state-of-the-art models in several NLP benchmark tasks, including the Stanford Question Answering Dataset (SQuAD) (Rajpurkar *et al.*, 2016), the CoNLL 2003 Named Entity Recognition (NER) task (Tjong Kim Sang and De Meulder, 2003), and the part-of-speech tagging dataset from the Wall Street Journal portion of the Penn Treebank (Marcus *et al.*, 1993).

3.4.6 Bi-directional Encoder Representations from Transformers (BERT)

Vaswani *et al.* (2017) introduce the Transformer, a new type of encoder-decoder model that relies solely on attention to draw global dependencies between the input and output sequences. Attention allows the model to focus on different parts of the input sequence at every step of the output sequence. This enables modelling dependencies without any regards for their distance in the sequences. This architecture is devoid of any recurrence or convolutions, and thus its training can be parallelizable. As usual, the encoder maps an input sequence $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n_x)}$ to a sequence of continuous representations, $\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(n_x)}$. Given these continuous representations, the decoder then generates an output sequence $\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(n_y)}$ of symbols one element at a time. At each time step t, the model makes use of the previously generated symbols, $\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(t-1)}$, to generate the next output symbol $\boldsymbol{y}^{(t)}$.

The attention used by the Transformer is the scaled dot-product attention with a set of queries in matrix $\boldsymbol{Q}$, a set of keys in matrix $\boldsymbol{K}$, and a set of values in matrix $\boldsymbol{V}$, and is computed as follows:

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d_K}}\right)\boldsymbol{V}, \tag{3.23}$$

where d_K is the dimension of the keys and acts as a scaling factor. Multi-headed attention allows for attention to be aggregated across h different, randomly-initialized representation subspaces. Thus,

$$\text{MultiHead}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)\boldsymbol{W}^O, \tag{3.24}$$

where Concat refers to concatenating each head, defined as:

$$\text{head}_i = \text{Attention}(\boldsymbol{Q}\boldsymbol{W}_i^Q, \boldsymbol{K}\boldsymbol{W}_i^K, \boldsymbol{V}\boldsymbol{W}_i^V), \tag{3.25}$$

with $\boldsymbol{W}_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_V}, \boldsymbol{W}_i^K \in \mathbb{R}^{d_{\text{model}} \times d_V}$, $\boldsymbol{W}_i^V \in \mathbb{R}^{d_{\text{model}} \times d_V}$, and $\boldsymbol{W}^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$. The scalar d_V represents the dimension of the values and d_{model} denotes the dimension of the model's embedding space. This multi-headed attention function can also be parallelized and trained across multiple computers. The authors also inject information about the relative and absolute positions of the values in the sequence using positional encoding to allow for the modelling of time-dependencies. This is done by summing the positional encodings with the input embeddings, which are defined as

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{\text{model}}}), \tag{3.26}$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{\text{model}}}). \tag{3.27}$$

Combining all these elements results in state-of-the-art embeddings that, when compared to previous models, has reduced computational complexity per layer, parallelizable computation, and better long-term dependency modelling. Vaswani *et al.* (2017) show that the Transformer outperforms previous models on machine translation tasks for English-German and English-French, with significantly faster training time.

Bi-directional encoder representations from Transformers, or BERT (Devlin *et al.*, 2018), is a pretrained language model that uses a deep bi-directional encoder-decoder model to learn vector representations of words without supervision by jointly conditioning on both the left and right context surrounding the word. Previously, the Transformer (Vaswani *et al.*, 2017) could only benefit from the previously generated tokens in the self-attention layers. This is sub-optimal for tasks that benefit from bi-directional understanding, such as question answering. BERT alleviates this constraint by optimizing for two objectives during pre-training:

- Masked Language Model (MLM): a random masking of some of the tokens in the input sequence; the goal is to predict the masked token's vocabulary ID given the surrounding context fused across both directions.

- Next Sentence Prediction (NSP): a binary classifier for predicting the correct next sentence from one of two options, given the current sentence.

The model is first trained on unlabelled data according to the two pre-training objectives (MLM and NSP). Then the model is fine-tuned using labelled data for a number of downstream tasks. Each downstream task has a separate fine-tuned model, but all of them have the same architecture. Devlin *et al.* (2018) found that during fine-tuning, most hyperparameters remained constant. Although this variation of the Transformer seems simple, it manages to set 11 new benchmark scores, most notably on the Stanford Question Answering Dataset (SQuAD) (Rajpurkar *et al.*, 2016), the General Language Understanding Evaluation (GLUE) task (Wang *et al.*, 2018), and the Multi-genre Natural Language Inference (MultiNLI) task (Williams *et al.*, 2017). The pre-trained embeddings are available online[1] and for 104 different languages.

[1] `https://github.com/google-research/bert`

3.5 Sentence Embeddings

While word embeddings have become ubiquitous and almost synonymous with natural language processing, learning dense representations of phrases and sentences has become a field of research in its own right. Varying techniques have been explored, from the most simple but powerful word vector averaging (Wieting *et al.*, 2015), to predicting the sentences surrounding a target sentence (Kiros *et al.*, 2015), to more advanced approaches using LSTMs and max-pooling layers (Conneau *et al.*, 2017). The goal is to embed sentences in a lower-dimensional space such that sentences with similar meanings or intentions are found close to one another using some similarity metric.

3.5.1 Word Averaging

Wieting *et al.* (2015) introduced averaged word embeddings to create sentence embeddings. The methodology is straightforward: given a sentence of words $w^{(1)}, w^{(2)}, \ldots, w^{(N)}$, each with corresponding vector representations $\boldsymbol{v}^{(1)}, \boldsymbol{v}^{(2)}, \ldots, \boldsymbol{v}^{(N)}$, we calculate the averaged sentence embedding as:

$$g_{\text{avg}}(w^{(1)}, w^{(2)}, \ldots, w^{(N)}) = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{v}^{(i)}. \tag{3.28}$$

Here the only parameters to be learned are those of the chosen word embedding model, which can be trained separately. Wieting *et al.* (2015) found that simple architectures, such as this one, outperformed more complex architectures in out-of-domain scenarios, as well as being the most efficient in terms of training.

3.5.2 Skip-thought

Cho *et al.* (2014) first proposed using recurrent neural network encoder-decoder models for machine translation. RNNs are ideal for sequence-to-sequence modelling, as they can take in variable-length sequences. The input sequence $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n_x)}$ is often called the context, and the goal is to encode a vector representation of this context, $\boldsymbol{c}$. Often a function of the final hidden state of the RNN encoder is used as the context vector, as seen in Figure 3.2. The RNN decoder conditions on the context vector $\boldsymbol{c}$ to generate an output sequence $\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(n_y)}$. Using a training set of N input-output sequences, the encoder-decoder model is trained to jointly maximize the average log-likelihood

$$\frac{1}{N}\sum_{i=1}^{N} \log P(y_i^{(1)}, \ldots, y_i^{(n_{i,y})} | x_i^{(1)}, \ldots, x_i^{(n_{i,x})}). \tag{3.29}$$

Kiros *et al.* (2015) introduced a new technique of learning general-purpose, distributed sentences without supervision, called skip-thought. This is done

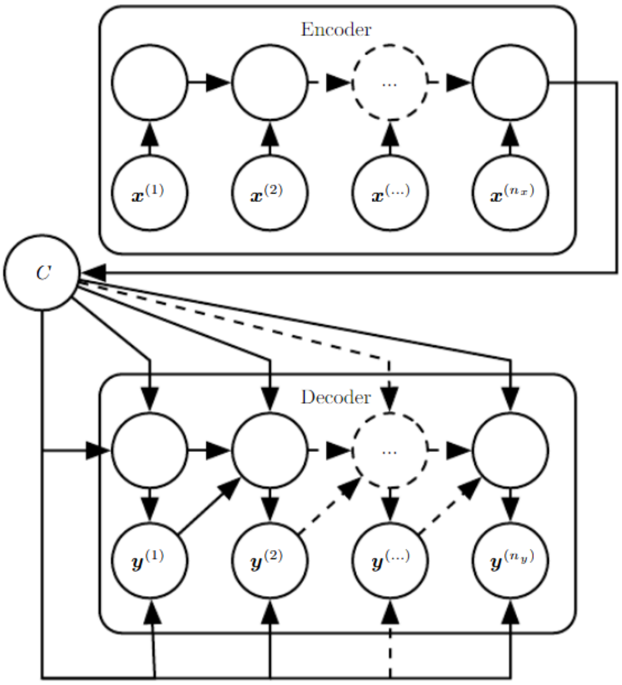

Figure 3.2: Sequence-to-sequence RNN architecture (Goodfellow *et al.*, 2016)

by training a RNN encoder-decoder model with GRU activations to reconstruct the sentences surrounding a target sentence. The result is that sentences with similar semantic and syntactic properties are mapped to a shared region in the sentence embedding space. They train on the BookCorpus set which contains books from 16 different genres (Zhu *et al.*, 2015). After training, the sentence embeddings are extracted from the encoder model.

More formally, for a sentence tuple (s_{i-1}, s_i, s_{i+1}), let $w_i^{(t)}$ denote the tth word for the sentence s_i and let $\boldsymbol{x}_i^{(t)}$ denote its word embedding. The encoder produces a hidden state $\boldsymbol{h}_i^{(t)}$ for each time step. For the last word $w_i^{(N)}$, the hidden state $\boldsymbol{h}_i^{(N)}$ represents the encoding of the full sentence. The recurrent neural network decoder then conditions on the outputs of the encoder at each time step while attempting to reconstruct sentences s_{i-1} and s_{i+1}. The objective to be optimized by the model is the sum of the log-probabilities for the previous and next sentences conditioned on the encoder representations:

$$\sum_t \log P(w_{i+1}^{(t)} | w_{i+1}^{(<t)}, \boldsymbol{h}_i) + \sum_t \log P(w_{i-1}^{(t)} | w_{i-1}^{(<t)}, \boldsymbol{h}_i). \tag{3.30}$$

which is summed over all the training tuples. Kiros *et al.* (2015) show that combining the sentence embeddings with linear classifiers achieves comparable results on a variety of benchmark tests with the same time complexity as more complex models. These benchmark tests include semantic relatedness evaluation (Marelli *et al.*, 2014), image-caption retrieval (COCO) (Lin *et al.*, 2014), movie review dataset (Pang and Lee, 2005), and text retrieval question classification (TREC) (Li and Roth, 2002).

3.5.3 InferSent

Conneau *et al.* (2017) build on previous approaches to sentence embeddings by introducing InferSent, a method for learning universal sentence representations using supervised data from the Stanford Natural Language Inference dataset (Bowman *et al.*, 2015). The authors combine a bi-directional LSTM model with pooling layers to produce high-quality sentence embeddings. Thus, for a sequence of T words, $w^{(1)}, \dots, w^{(T)}$, the bi-directional LSTM network computes a set of hidden representations, $\boldsymbol{h}^{(1)}, \dots, \boldsymbol{h}^{(T)}$, where $\boldsymbol{h}^{(t)}$ is the concatenations of the forward and backward LSTM outputs per time step t:

$$\overrightarrow{\boldsymbol{h}}^{(t)} = \overrightarrow{LSTM}(w^{(1)}, \dots, w^{(T)}) \tag{3.31}$$

$$\overleftarrow{\boldsymbol{h}}^{(t)} = \overleftarrow{LSTM}(w^{(1)}, \dots, w^{(T)}) \tag{3.32}$$

$$\boldsymbol{h}^{(t)} = [\overrightarrow{\boldsymbol{h}}^{(t)}, \overleftarrow{\boldsymbol{h}}^{(t)}]. \tag{3.33}$$

Because T can vary, the authors employ pooling techniques, including max-pooling and average-pooling, to create a fixed length vector from the set of vectors $\boldsymbol{h}^{(1)}, \dots, \boldsymbol{h}^{(T)}$. Max-pooling entails selecting the maximum value of each feature in the hidden states $\boldsymbol{h}^{(1)}, \dots, \boldsymbol{h}^{(T)}$. Average-pooling entails taking the average of each feature in the hidden states $\boldsymbol{h}^{(1)}, \dots, \boldsymbol{h}^{(T)}$. Conneau *et al.* (2017) show that max-pooling yields superior results to average-pooling, and that their technique consistently outperforms unsupervised methods such as skip-thought on a variety of transfer learning tasks, including the COCO image-caption retrieval dataset (Lin *et al.*, 2014), the multi-genre natural language inference (MultNLI) dataset (Williams *et al.*, 2017), and the semantic textual similarity evaluation (STS14) (Agirre *et al.*, 2014).

3.5.4 Universal Sentence Encoder

Cer *et al.* (2018) introduce two pre-trained sentence embedding models specifically for transfer learning tasks that offer trade-offs between accuracy and computational cost. The authors found that transfer learning using sentence embeddings outperform transfer learning with Word2Vec word embeddings (Mikolov *et al.*, 2013*c*) on a variety of downstream classification tasks, including fine-grained text retrieval question classification (TREC) (Li and Roth, 2002), the customer review dataset (Hu and Liu, 2004), the subjectivity assessment of movie reviews and plot summaries dataset (Pang and Lee, 2004), movie review dataset (Pang and Lee, 2005), the phrase-level sentiment classification dataset (Socher *et al.*, 2013), and the semantic textual similarity of sentence pairs (Cer *et al.*, 2017). These two pre-trained models are:

- Transformer (Vaswani *et al.*, 2017),
- deep averaging networks (DAN) (Iyyer *et al.*, 2015).

The Transformer is a resource-hungry model that achieves superior accuracy, while the DAN targets efficient inference for minor sacrifices in accuracy. The Transformer architecture, which we discussed in Section 3.4.6, produces sentence embeddings by using attention to compute fixed-length context-aware representations of words in the sentence, which are then summed and normalized by dividing by the square root of the length of the sentence.

The DAN architecture ignores context and sentence structure by simply averaging pre-trained GloVe word embeddings (Pennington *et al.*, 2014) and feeding this into a deep feedforward neural network where each layer learns a more abstract representation, and the final layer is a softmax layer. Both models were trained on resources drawn from various web sources. This unlabelled data was then augmented with labelled data from the Stanford Natural Language Inference (SNLI) corpus made available by Bowman *et al.* (2015). This step was found to improve the performance of the sentence embeddings on various transfer tasks.

3.6 Cross-lingual Embedding Spaces

Suppose we want to understand how the embedding spaces of different languages interact with one another. Cross-lingual word embeddings incorporate multiple languages into a single shared vector space that enables the transfer of lexical information between languages, as seen in Figure 3.3. By training multilingual language models, we can obtain robust cross-lingual word representations that are useful for NLP tasks like bilingual lexicon induction, machine translation, and cross-lingual information retrieval.

Additionally, the shared representation space may strengthen low-resource language representations when combined with high-resource languages (Ruder, 2017). An inherent property of high-dimensional vector spaces is hubness, where certain points, commonly referred to as hubs, emerge as popular nearest neighbours for many other points (Radovanović *et al.*, 2010). Bilingual lexicon induction looks for nearest neighbours of words across languages, and naturally is affected by the level of hubness of the cross-lingual word embeddings. Ideally, the translation of a word or sentence could then be retrieved in the cross-lingual embedding space using nearest neighbour retrieval techniques.

As the field of cross-lingual word embeddings expands, it becomes more apparent that the value of cross-lingual word embeddings lies in the training data and the cross-lingual signal used, rather than the model architectures (Levy *et al.*, 2017). According to Ruder (2017), the different methodologies of training data can be described in terms of the type of alignment and the comparability of the data across the languages. Exactly comparable (parallel)

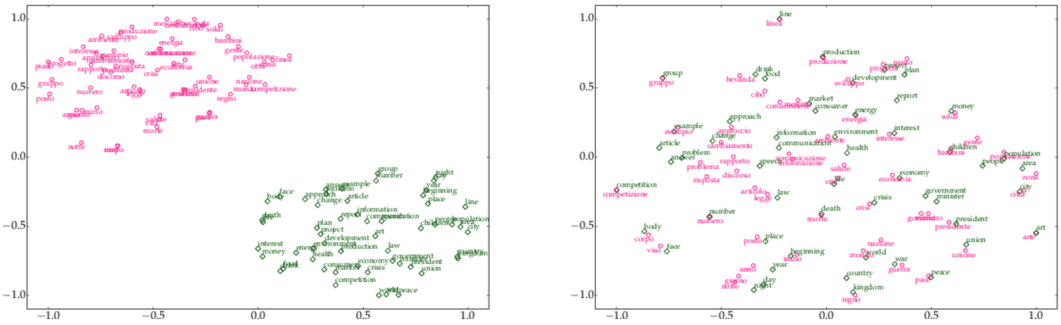

Figure 3.3: Unaligned monolingual embedding spaces (left) and aligned cross-lingual embedding space (right) (Ruder, 2017).

data can be bilingual lexicons (word-level alignment) or translated sentence pairs (sentence-level alignment). In the absence of such parallel data, one can make use of comparable data with shared class labels or shared topics or multimodal grounding such as shared images between sentences or documents where the goal could be to jointly optimize for cross-lingual language modelling and a specific NLP task (Ruder, 2017). Alternatively, cross-lingual word embeddings can be learned from only monolingual data in an unsupervised fashion through bilingual dictionary induction. However, using unsupervised cross-lingual representations is only justified when it outperforms supervised cross-lingual representations.

3.6.1 Linear Projection Between Monolingual Mappings

Following the introduction of the Word2Vec models by the same authors, (Mikolov *et al.*, 2013*b*) also showed how two monolingual word embedding spaces can be aligned using a transformation matrix $\boldsymbol{W}$ and a bilingual lexicon of commonly found words in each language $\{\boldsymbol{s}_i, \boldsymbol{t}_i\}_{i=1}^{n}$, where $\boldsymbol{s}_i$ is the vector of the word in the source language and $\boldsymbol{t}_i$ the vector of the translation to the target language. The assumption they made is that the distributional representations of the same words across languages are geometrically similar and therefore a $\boldsymbol{W}$ exists for the following optimization problem:

$$\min_{\boldsymbol{W}} \sum_{i=1}^{n} \|\boldsymbol{W}\boldsymbol{s}_i - \boldsymbol{t}_i\|^2, \tag{3.34}$$

which can be found using stochastic gradient descent. The transformation matrix $\boldsymbol{W}$ can now be used to linearly project new, as yet unseen word vectors from one language to another.

3.6.2 Task-specific Bilingual Word Embeddings

Gouws and Søgaard (2015) introduce a method for learning task-specific bilingual word embeddings using non-parallel training data and only a small bilingual dictionary as cross-lingual signal. The bilingual dictionary consists of language equivalent words with respect to some task, either part-of-speech tagging or super sense (high-level equivalence) tagging. The translation equivalents of words can be obtained using Google Translate, Wiktionary, or WordNet (Miller, 1995). The authors create a mixed corpus by concatenating and then shuffling the source language corpus and the target language corpus. They then iterate through the mixed corpus and replace each word w with its translation equivalent w' with probability $\frac{1}{2k}$, if w is in the task-specific bilingual dictionary $\mathcal{R}$ and k is the cardinality of $\mathcal{R}$. Then they train the cross-lingual word embedding space using CBOW (Mikolov *et al.*, 2013a), although any word embedding technique could be applied here. The cross-lingual embeddings can then be applied to cross-language part-of-speech and super sense tagging on target languages and in different domains.

3.6.3 Document-comparable Bilingual Word Embeddings

Vulic and Moens (2016) show how bilingual word embeddings can be learned without parallel data, or a bilingual dictionary. They build pseudo-bilingual documents by combining bilingual resources with shared topics, e.g. Wikipedia articles in multiple languages aligned through inter-wiki links or news texts discussing the same topics but in different languages. The shared topic across languages serves as a weak cross-lingual signal, which is crucial for cross-language knowledge. Their proposed method includes merging these bilingual document pairs with shared topics, and then randomly permuting words from both languages in the merged document pairs. They then train an SGNS model (Mikolov *et al.*, 2013a) with default parameters on the pseudo-bilingual documents with a vocabulary that spans across both languages. The SGNS model ignores syntax and this results in gently pushing similar words across both languages to a similar representation in the shared bilingual embedding space. Ignoring syntax also means that this method requires no knowledge of the ontology of the languages at hand or bilingual-dependent modelling assumptions. The authors train on non-parallel data from three language pairs in the Indo-European language family: Spanish-English, Italian-English, and Dutch-English. At the time of release, the bilingual embeddings achieved state-of-the-art performance on two semantic tasks: bilingual lexicon extraction (Vulić and Moens, 2013a,b), and suggesting word translations in context (Vulić and Moens, 2014). However, training with only document-comparable data is limited in its capabilities of capturing cross-lingual sentence or document representations, as syntax and word ordering is ignored.

3.6.4 Language-agnostic Sentence Embeddings

Bilingual embedding spaces can be extended to multilingual sentence embeddings spaces, or even language-agnostic sentence embeddings, where the embeddings are indifferent to the input language and the downstream NLP task. Artetxe and Schwenk (2018) propose language-agnostic sentence embeddings which supports 93 languages. A single bi-directional LSTM encoder is combined with a shared byte-pair encoding vocabulary across languages to construct sentence embeddings of the source language sentence. Byte-pair encoding is a lossless data compression algorithm that works by replacing common pairs of consecutive bytes with a byte that does not appear in that data. These encoded sentences are then fed to a decoder LSTM which constructs the sentence in the target language. Because the sentences are encoded to byte pairs, the model is unaware of the language ID, hence the language agnosticism. According to Artetxe and Schwenk, this approach can generate improved embeddings for low-resource languages and allow for zero-shot transfer learning of an NLP model from one language to another.

Instead of translating every sentence into every other language, which would require an N-way parallel corpus and be computationally expensive, the authors found that training with only two target languages per source language and separate alignments for each language pair yielded desirable results. The authors performed simple preprocessing steps to standardize text across all the languages. At the time of publication, no standardized way to evaluate multilingual sentence embeddings existed, so the authors tested the transfer capabilities of their sentence embeddings on different multilingual tasks (on languages besides English), including

- cross-lingual natural language inference (XNLI) for 14 languages, introduced by Conneau *et al.* (2018),
- cross-lingual document classification (MLDoc) for 7 languages, introduced by Schwenk and Li (2018),
- bi-text mining (BUCC) – identifying translations of sentence pairs in documents – for 4 languages, introduced by Zweigenbaum *et al.* (2018),
- and most importantly, Tatoeba, a nearest neighbour similarity search dataset for 112 languages, introduced by Artetxe and Schwenk.

On all these tasks, the language-agnostic sentence embeddings achieved strong downstream performance. The competitive results on low-resource languages indicate the benefits of jointly training across multiple languages. Language-agnostic sentence embeddings are even able to generalize to unseen languages, so long as the unseen languages are related to some of the languages it was trained on. This is one of the first successful general-purpose multilingual sentence embeddings that can be used for transfer learning in tasks and on other (unseen) languages.

3.7 Chapter Summary

The distributional hypothesis of Harris (1954) states that language can be structured with respect to some features in an organized system of statements. We can learn these distributed representations such that words and sentences that have similar meanings or intentions are mapped close to one another. Word embedding learning can be divided into two camps: count-based and prediction-based. Count-based methods aim to capture co-occurrence statistics while prediction-based methods try to predict a word vector given a context vector, or predict a context vector given a word vector. With sentence embeddings we can compare sentences that have similar meanings or intent but with little word overlap. Sentence embedding techniques range from simple word embedding averaging to the more complex method of extracting the hidden representations from sequence modelling architectures. Typically, word and sentence embeddings are learned from monolingual corpora. But these language models can be extended by training on multiple languages to create cross-lingual embeddings. Cross-lingual training relies on some form of cross-lingual signal; either a dictionary of word translation pairs, or a parallel corpus (suitable for machine translation) or some shared class label across languages. In the case of cross-lingual embeddings, the translations of words are mapped close together in the embedding space, allowing for translations of words across languages by simply retrieving their nearest neighbours. Combining high-resource languages with low-resource languages in the same embedding space allows for knowledge transfer and can even improve the embeddings of the low-resource languages. Learning high-quality embeddings usually requires very large corpora, which often is not always available for low-resource languages. For high-resource languages, large corpora can be collected with relative ease from web sources like Wikipedia and online forums. As previously discussed, the MomConnect dataset contains question-answer pairs in the healthcare domain. While the questions are multilingual, with some in low-resource languages, the answers are English templates. The lack of reliable language labels and shared English template answers present us with the opportunity to apply the cross-lingual learning techniques discussed in Section 3.6.

Chapter 4

Data Acquisition and Anonymization

4.1 Introduction

In this chapter we describe the process of acquiring and anonymizing a dataset of MomConnect questions and answers. We also give a brief overview of data protection and privacy laws in South Africa and anonymization guidelines that prevail in the European Union and the United States of America.

The MomConnect dataset, prior to being anonymized, contains identifiable information as well as information about users' health, personal information, and HIV-status. Due to its highly sensitive content, restricted access to the MomConnect dataset was conditional on anonymization protocols, adherence to the POPI Act, an ethical clearance review, and was granted with the sole purpose of research. While Praekelt Foundation has permission to work with the data as a stakeholder in the continuing project, the South African National Department of Health is the owner of the data. As a third party, gaining access to the data was a 10 month process which involved the following steps:

1. gaining a formal letter of permission from the National Department of Health to access the data,

2. formalizing a data-sharing agreement between Stellenbosch University and Praekelt Foundation,

3. seeking and gaining clearance from the Research Ethics Committee of Stellenbosch University,

4. and Praekelt Foundation anonymizing the data to sufficiently de-identify the data before sharing it with us.

While this process was lengthy, it was also necessary to ensure the rights, dignity and safety of the participants were protected at all times during this study.

4.2 Data-sharing Agreement

As part of the process to gain access to the MomConnect dataset, we entered into a data-sharing agreement with Praekelt Foundation. Access to the MomConnect dataset was granted subject to the following conditions:

- that the data undergo pseudonymization and encryption protocols to ensure its confidentiality, integrity, availability and resilience, in accordance with data protection laws,
- that all research partners take the necessary precautions to minimize the risk of re-identification through affinity,
- that access to the data is restricted to appropriate individuals,
- and that the data is to be destroyed once it is no longer necessary.

The Protection of Personal Information (POPI) Act 4 of 2013 is South Africa's data protection law which governs data processing and storage by public and private bodies. Praekelt Foundation adhered to internationally-recognized guidelines established by the GDPR regulations in the EU and the HIPAA regulations in the United States of America to anonymize the data.

Protection of Personal Information (POPI) Act 4 of 2013 The POPI Act outlines the required minimums for the collection, processing, and storing of personal information, provides for the rights of persons regarding unsolicited electronic communication and automated decision making, and regulates the flow of personal information across the borders of South Africa. Further, the POPI Act aims to balance the need for protecting personal information against the need for unimpeded access to information for economic, scientific, and social progress. More information regarding the POPI Act of 2013 can be found in Appendix A.1.

The data collected by the National Department of Health was made available to us to investigate ways of improving the MomConnect service. Praekelt Foundation ensured that the data was sufficiently de-identified and anonymized using HIPAA and GDPR guidelines. The necessary steps were taken to ensure that only designated researchers had access to the data.

Health Insurance Portability and Accountability Act (HIPAA) The HIPAA Safe Harbour Method describes steps to perform de-identification of protected health information of users (HHS.gov, 2012) in the United States of America. In the context of HIPAA, de-identification refers to the removal of specific fields of data that can be used – either alone or when combined with other fields – to uniquely identify an individual. Such fields in the MomConnect dataset are listed in Appendix B. Prior to being granted access to a static copy of the MomConnect data, all fields mentioned in Appendix B were stripped or hashed in accordance with the HIPAA Safe Harbour Method.

General Data Protection Regulations (GDPR) of 2016 The GDPR is the European Union's data protection and privacy laws and, amongst others, provides specific legislation for safeguards to protect users in the case of a data breach, including guidelines for pseudonymization or full anonymization. Pseudonymization or full anonymization should be done in such a manner that the result cannot be used to identify a subject without additional information. Datasets can also not be made publicly available without explicit, informed consent of the data subject. Data subjects also have the right to request a copy of their data collected, and reserve the right to have their data erased under certain conditions.

The standard test for sufficient anonymization in EU is called the Motivated Intruder Test. The motivated intruder is a hypothetical person assumed to not have any specialist knowledge such as computer hacking skills, or access to specialized equipment, and who would not resort to criminal acts such as burglary to gain access to data that is kept securely. This hypothetical person starts off with no prior knowledge but wishes to identify the data subject's identity from a set of anonymized and de-identified data points. Further, reasonable assumptions regarding the competency and public resources available to the motivated intruder exist. The motivated intruder can employ investigative techniques such as inquiries regarding the identity of the person or advertise for someone with knowledge to come forward. If the hypothetical motivated intruder would fail to identify the data subject, the anonymization and de-identification techniques employed are considered sufficient.

4.3 Anonymization Protocols

Guided by the GDPR and the HIPAA guidelines, and prior to any data processing or analysis, an anonymization protocol was established by Praekelt Foundation. The consensus was that simply removing identifiers such as name and telephone number may be insufficient, as identities might still be deducible from contextual information. For example, the overlap of a clinic location, date, and sign-up language could narrow the number of possibilities to a handful of individuals. As such, the anonymization protocol included ranking data fields by importance to the research study and algorithmically removing data in order of increasing priority, to ensure some lower bound on the size of any single distinguishable group. This approach is conceptually similar to the k-anonymization algorithm (Samarati and Sweeney, 1998; El Emam and Dankar, 2008). Absolute identifiers (e.g. expected delivery date) were replaced with relative quantities (e.g. days to delivery). Age data was bucketed in groups of 5 years and other identifier information (e.g. health district information, user identifiers etc.) were hashed against one-time random numbers, to prevent direct identification. The hashing method still allowed for aggregated data analyses to proceed under the restricted distribution of the dataset to academic partners.

4.4 Ethical Clearance

The purpose of ethical clearance is to contribute to the safeguarding of the rights, dignity, safety, and well-being of all actual or potential participants in research studies. At Stellenbosch University, an independent Research Ethics Committee oversees a well-defined process through which one can apply for ethical clearance. We submitted our research proposal, data-sharing agreement and formal letter of permission from the National Department of Health to the Research Ethics Committee, as per their guidelines. The Committee deemed our investigation to be of low risk to all the participants involved and in compliance with data protection laws, and granted us permission to proceed with our research study.

Chapter 5

Exploratory Data Analysis

5.1 Introduction

Upon being granted access to the anonymized MomConnect dataset, the first step was to analyse the data both quantitatively and qualitatively. Insights gained from the data analysis allowed us to identify which data fields would be most useful for predicting the most appropriate answer for an incoming question, as well as guide our experimental design. We used the Python Pandas library (McKinney, 2010) to process and aggregate the dataset. We describe the different data fields in Section 5.2. We report the results of quantitative and qualitative analysis of the different data fields in Section 5.3. Finally, we discuss various challenges encountered within the data and our proposed solutions in Section 5.4.

5.2 Data Dictionary

In the anonymized MomConnect dataset there are $229,421$ data points. Each data point consists of 18 fields, with both qualitative and quantitative values. Most importantly, we have the `helpdesk_question` and `helpdesk_response`, which contain the questions posed by the users and associated responses given by the helpdesk staff. A detailed data dictionary can be found in Appendix B, which includes examples and the level of identifiability of each field. The necessary measures to prevent re-identification have already been put in place. In brief, each question and answer pair is accompanied by the following information:

- anonymous ID for user,
- helpdesk communication channel, i.e. WhatsApp or SMS,
- the time of the helpdesk question asked relative to Expected Date of Delivery (EDD),

- helpdesk response timestamp relative to EDD,
- helpdesk response time in days,
- year of birth of the mother, rounded to closest multiple of 5,
- days from registration to EDD, rounded to closest multiple of 10,
- days since previous question asked by user (N/A if it is the first question),
- the question label, i.e. Question, Message Switch, Compliment, Complaint, PMTCT, Opt Out, Channel Switch.
- the language in which the user signed up, e.g. 'eng_ZA',
- the province, e.g. 'Eastern Cape Province',
- the district, e.g. 'Oliver Tambo District Municipality',
- the anonymized subdistrict code,
- whether the user comes from an urban, rural, or semi-rural area,
- the anonymized facility (e.g. public clinic) code,
- and the date at which the data was extracted, rounded to the nearest week.

5.3 Quantitative and Qualitative Analysis

We analyzed both the qualitative and quantitative fields found in the MomConnect dataset. Before the analysis we first had to clean fields containing mixed datatypes and missing values. We report the descriptive statistics and frequency distributions of the most relevant fields. We also demonstrate inconsistencies found in the manual labelling and answering of incoming questions.

5.3.1 User and Usage Statistics

When counting unique `helpdesk_anonymous_id`'s, we find that there are $86,246$ unique users in the dataset, with 2.66 recorded exchanges per user on average. However, less than 25% of users posed more than 3 queries to the helpdesk, meaning that there is a long tail of inactive users. Almost 60% of the unique users make use of SMS as their preferred mode of communication, and 40% use WhatsApp. However, when observing the interactions between `helpdesk_communication_channel` and `dhis2_ruralurban` in Table 5.1, an interesting pattern emerges: WhatsApp, which is a much cheaper form of communication than SMS, has a low penetration rate in rural areas, but is increasingly prevalent in peri-urban and urban areas. The cost associated with

	SMS	WhatsApp
Rural	72%	28%
Peri-Urban	62%	38%
Urban	54%	46%

Table 5.1: Interaction between communication channel and urbanisation

different communication channels is also reflected in the fact that users who use WhatsApp as their communication mode have on average 3.14 recorded exchanges, while those who use SMS have only 2.23 exchanges with the helpdesk platform on average. Rural areas often lack cell phone reception and easy access to airtime purchasing. It is also users in rural areas that may benefit more from the service, as they often have to travel much further than those in urban areas to access public healthcare. The majority of users (70%) are living in urban areas, while 23% live in rural areas, and 7% live in peri-urban areas.

We also observe interesting trends in the distribution of sign-up language labels per unique users. As previously described, each user registers their preferred language of communication, but is free to ask questions in any language of their choosing. During a random sampling, we encountered many users who asked questions in languages other than the one they registered, and also frequently used code-mixing (alternating between languages) within their questions. We expand on the language distribution in conjunction with regional data in Appendix C.1.

Language	Users
Afrikaans	2,577
English	51,243
Ndebele	97
N. Sotho	3,399
S. Sotho	749
Swati	287
Tsonga	1,360
Tswana	948
Venda	1,105
Xhosa	6,819
Zulu	17,615
Redacted	26
Total	**86,246**

Table 5.2: Unique users in dataset per sign-up language

In Table 5.2 we note that several languages have less than 1,000 registered users, namely Ndebele, Southern Sotho, Swati, and Tswana. Collectively they make up 2.4% of all the users. These languages can be considered extremely low-resource within the context of our problem statement. The top four languages, namely Afrikaans, English, Xhosa and Zulu collectively make up 90.7% of the users. These statistics do not reflect the national population statistics, where Afrikaans, English, Xhosa, and Zulu home language speakers collectively make up 62% of the population of South Africa. The number of users that register English as their preferred communication language (59.41%), is especially interesting and indicates that, in the MomConnect user base, English is the de facto language of communication both for first language and non-first language speakers. English is one of the most widely spoken languages world-wide and on the internet, and is unanimously considered a high-resource language.

5.3.2 Distribution of Questions and Answers

In the raw case-sensitive dataset, we found 182,276 unique, multilingual questions and 42,675 unique monolingual (English) answers. The frequencies of both the questions and the answers follow power law distributions, with a long tail, as seen in Figure 5.3.

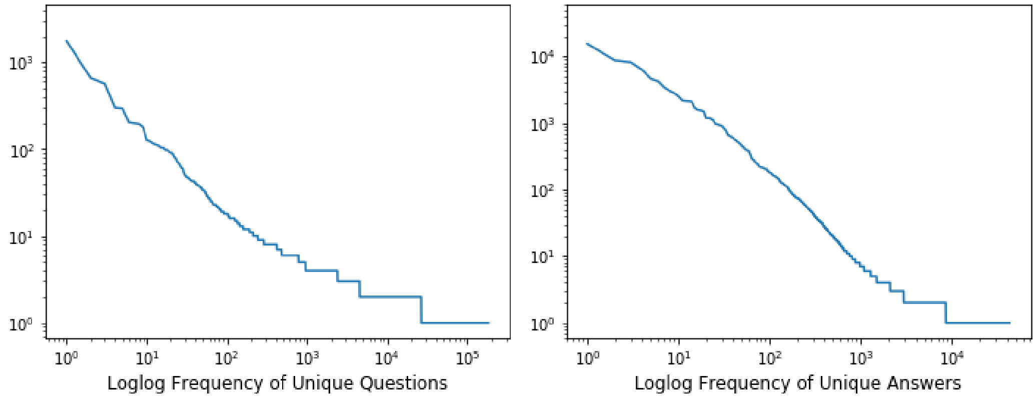

Figure 5.3: Loglog plot of the frequencies of questions (left) and answers (right)

Prior to our analysis, we made a reasonable assumption that there would be a consistent mapping of questions to answers. This assumption did not hold, as we discover that unique questions do not consistently receive the same answer. In fact, 4% of the answers in the dataset were found to be inconsistently assigned to incoming questions. This percentage is diluted by the fact that about 85% of the questions occur only once. When looking at the unique answers per question that occurred at least twice in the dataset, this percentage changes to 28%. This means that of all the unique questions that occur more than once in the dataset, almost a third did not consistently receive the same template responses. The implication of these results are that the questions are not answered in a consistent manner. The reasons for these inconsistencies are unclear.

5.3.3 Question and Response Time Distributions

After cleaning the fields and removing all the NaN values we calculated the mean, standard deviation, minimum values, 25th quantile, 50th quantile (median), 75th quantile and maximum values for the time-related fields. We report our findings in Table 5.4. Note that a positive `helpdesk_days_to_edd_*` means the question was in a period prior to the expected delivery date (EDD) of the user. All the values are reported in days.

	Mean	Std	Min	25%	50%	75%	Max
helpdesk_days_to_edd_question	38.21	132.60	-1023.44	-11.37	44.46	120.39	870.32
helpdesk_days_to_edd_reply	35.72	132.88	-1025.49	-13.59	41.76	117.65	868.66
helpdesk_response_time_days	2.50	4.35	0.000229	0.0742	0.8469	2.94	117.85
days_to_edd_registration_created_at_10	145.10	61.05	-30.00	100.00	150.00	190.00	330.00
helpdesk_time_between_questions_days	22.62	45.48	0.00	0.40	5.15	22.99	646.58

Table 5.4: Descriptive statistics of time-related fields

The first two fields are correlated as every question will have a response in a reasonable amount of time. Next is `helpdesk_response_time_days`, which measures how long the helpdesk took to respond to the query and is the difference between the first two fields. The fourth field is data captured at registration and indicates when the user registered, relative to her expected delivery date. It remains constant for every user. This field has been rounded to 10. Lastly, `helpdesk_time_between_questions_days` captures the difference in days between the current question and the previous question the user has posed. If the data point is the first recorded for the user, i.e. they have never asked any questions before, the value of this is NaN.

There are some obvious extreme values, such as a maximum of 646.58 days between questions (`helpdesk_response_time_days`), or a maximum `days_to_edd_registration_created_at_10` of 330 days. This is most likely incorrectly captured data, as these values simply do not make sense. A pregnancy is usually 9 months or about 273 days, which should be the logical maximum for the field. Because we have such extreme values, we cannot rely on the mean to be an informative metric. Rather, we discuss the fields in terms of their quantiles. The most important field is `helpdesk_response_time_days`, as it quantitatively measures the responsiveness of service. At least 25% of questions posed by users received a response within 0.0742 days (107 minutes). However, the median response time is 0.8469 days, which translates to about 20 hours. This means that the bottom 50% of questions only received a response after 20 hours, giving us some idea of the backlog the staffed helpdesk is facing.

5.3.4 Helpdesk Label Distribution

When a question is received, the helpdesk staff manually assigns a label according to the nature of the question. This value is recorded in the field `helpdesk_label_name`. We observe many inconsistencies in the labelling of incoming questions when examining the frequency distribution within certain questions and answers. The field can take on 12 possible values: "Question", "Message Switch", "Compliment", "Complaint", "PMTCT", "Language Switch", "Opt Out", "Channel Switch", "Other", "Spam", "WhatsApp", and "Unable to Assist".

Value	**Frequency**
Question	182,237
Message Switch	23,148
Compliment	12,655
PMTCT	7,225
Language Switch	1,373
Complaint	1,124
Opt Out	832
Other	376
Spam	323
Channel Switch	55
Unable to Assist	50
WhatsApp	23

Table 5.5: Frequency distributions of the helpdesk label name values

Here "Question" is the most generic value and occurs the most frequently. "Message Switch" is assigned to queries that request to switch from ante-natal (pregnancy-related) information to post-natal (infant care and breastfeeding related) information. The user usually sends this query after they gave birth and the occurrence of this value is loosely correlated with the user's expected delivery date. The "Compliment" and "Complaint" values are assigned to incoming messages related to positive or negative feedback, directed either at MomConnect or a public health clinic. The value "PMTCT" (prevention of mother-to-child transmission) is assigned to questions related to preventing HIV and AIDS transmission from mother to child, either during pregnancy or during the breastfeeding period. The value "Language Switch" is assigned to user requests for language change. The user can at any time opt out of receiving messages from MomConnect with an "Opt Out" request. The values "Other" and "Spam" are assigned to messages outside the norm. The "Unable to Assist" value is assigned to queries where the helpdesk staff was unable to assist the user in their request for information. Finally, "WhatsApp" is assigned to user requests that their preferred communication channel be switched to WhatsApp. These queries are not necessarily disjoint from the messages labelled as "Channel Switch", which indicates requests to change the communication channel. When looking only at the label distribution, it would seem that "Unable to Assist" occurs very infrequently (only 50 times). But when we examined the individual responses associated with this value, we noticed that this field was not uniformly mapped to the answers, for unknown reason. However, one of the responses associated with the value – *"Thank you for your message but we are unable to help you with this query. Please go to your clinic if you need medical assistance"* – is, in fact, the most frequently-occurring response in the raw dataset. We found that this was the exact response for 15% of the queries in the dataset. Many more examples of such inconsistencies were found in other fields, as well as a high number of near-duplicates in the answer set.

5.3.5 Near-duplicate Answers

We define a near-duplicate as a high degree of word overlap between two sentences, with only minor differences in word choice, casing, and punctuation. For example, both of the following verbatim responses were discovered in the dataset, each with a comparable number of occurrences:

"Please go for sonar from 20 weeks to check baby's gender." - `Answer A`

"Please go to sonar at 20 weeks for baby's gender." - `Answer B`

Even the slightest difference – such as a missing period at the end of the sentence – would mean that traditional string matching methods would not identify them as exact matches. Without additional knowledge of the data-generating process, we can only assume that they are either due to human error, or due to a change in the standardized templates over time.

When observing the frequency distribution of the answers in the dataset, we notice that often the less-frequent answers are simply near-duplicates of more frequent answers. Therefore it was important to establish a protocol for automatically detecting and replacing less-frequent near-duplicates with more frequent counterparts. However, there are some cases where automatically detecting similarity might be more tricky. The following verbatim responses found in the dataset are more-or-less equivalent in their intent but have very little word overlap:

"Your baby will be ready for solid foods at 6 months. Try offering her 1 or 2 spoonfuls of thick, mashed vegetables or fruit. As she gets used to them, start to add other foods like porridge and other mashed foods like beans, lentils and meat. Slowly increase the number of times she has solid food in a day. When your baby is 7 months old, she should be eating solids 3 times a day. Continue breastfeeding your baby, in addition to giving her solid food." - `Answer C`

"Your baby needs ONLY breast milk for the first 6 months. Don't give him water, porridge, formula milk or anything else. Your body can make plenty of milk for your baby. It protects him from disease. If you are HIV-positive make sure you are taking your ARVs while you are breastfeeding. This will help to stop him becoming HIV-positive when he drinks your breast milk." - `Answer D`

If the user had enquired about when she can stop breastfeeding and begin to feed the baby solid food, both of these answers would have been correct. The answer set is simply too large to manually locate all the answers with similar intent, and we should accept that this is a hard limitation on our capabilities. This will be taken into account when discussing the experimental results in Section 6.5.

5.3.6 Informal Language, Typos, Misspellings, and Code-Mixing

We sampled 50 questions from the dataset and manually examined the questions in terms of their language usage and word content. In the WhatsApp and SMS exchanges the language was generally informal and many users used shorthand words like "thanx" instead of "thank you". Typos and misspellings such as "kix" or "kikcs" instead of "kicks" were also encountered. We also encountered code-mixing and language mixing. In South Africa many people speak at least two languages. Bilingual speakers have been known to spontaneously use code-mixing or mixed language in their secondary language (Myers-Scotton, 1993). A mixed language is a socially and culturally influenced phenomenon where the user's dominant (mother-tongue) language influences the spoken grammar, word choice, and syntax of the secondary language. The result is that the secondary language is sometimes spoken with misused prepositions, multipurpose verbs, and the ordering of subjects, verbs, and objects resembling that of the dominant language and not the secondary language.

We found several examples of mixed language within sampled data. The following sentence demonstrates the use of mixed language well (for anonymization purposes we changed the dates and switched out words while still preserving the necessary content): *"If on 15 Dec the scan tells the pregnant is on the 4th month so this month how old is the pregnant."* Here the word "pregnant", which in some languages roughly translates to "carrying a child", has been used in the place of "baby" or "fetus". This is an interesting phenomenon that is usually not encountered in monolingual datasets. Our research partners at Praekelt Foundation performed an independent empirical study to investigate code-mixing in the MomConnect dataset Obrocka *et al.* (2019). They estimate that 10% of the questions posed by the users contained instances of code-mixing. One of the examples of code-mixing they found was the following sentence: *"Kuyenzeka yini kuthi umakuqhume condom kuvele kuthi khulelwe after day".* This sentence is an example of code-mixing between Zulu and

English.

5.3.7 Sample Question-Answer Pairs

Table 5.6 shows what typical question-answer pairs may look like in the dataset. Questions can be posed in any of South Africa's 11 official languages, while the template answers are currently all English.

Question	Template Answer
What causes heartburn?	*Hormone progesterone relaxes smooth muscles, thus the valve that separates the gastric acid relaxes, allowing it to go up causing heartburn. Avoid consuming too much spicy, acidic, fizzy, citrus fruits, chocolates, lots of sugar, rich or fried or fatty food. Eat small meals slowly, wear loose fitting clothes, and do not sleep immediately after eating. Drink peppermint tea and Gaviscon.*
Kungani ngihlale ngigula njalo ekuseni?	*Do you feel like vomiting? Morning sickness or nausea is common in the first 3 months. It should ease from 4 - 7 months. Avoid food with too much fat and spices. Eat dry bread or a dry biscuit when you wake up. If you cannot eat, drink lots of water or tea but not alcohol. Adding some ginger, mint or lemon to your tea may help to ease the nausea.*
What are the signs of labour?	*Signs of labour include a jelly-like discharge, your water breaking, and regular and painful labour contractions. Make sure you can get to a hospital.*

Table 5.6: Typical question-answers pairs found in the MomConnect dataset

5.4 Addressing the Data Challenges

The main objective of this thesis is to address the growing backlog facing the MomConnect helpdesk staff, by using language modelling and question answering techniques. This includes addressing some of the challenges we encountered within the dataset. Firstly, we have a large dataset with many questions and answers occurring only once. We also have a very noisy dataset, with many typos, misspellings, code-mixing, as well as inconsistencies in the manual labelling and answering process. Our dataset contains text in 11 different languages, many of which are low-resource languages. We also have a large imbalance of languages, based on the number of unique users per sign-up language. We also cannot rely on the sign-up language label to distinguish between different languages. Aside from the noise, multilinguality, and low-resource languages, the dataset in its raw state is not ready to be used for model experimentation. We decided to only make use of the question and answer data fields for model experimentation and discarded the other fields. The questions were chosen as the input data, $\boldsymbol{X}$, and the answers the associated class label, $\boldsymbol{y}$. We took the helpdesk response as the ground truth label, even though the answers were sometimes inconsistently mapped to the same questions. We systematically identified and replaced near-duplicate answers with their more popular twins. We decided that the most effective approach to automation would be to only train and test on the questions-answer pairs where there were sufficiently many answers to learn from. Thus, we limited our experimental dataset to question-answer pairs where the answer occurred at least 256 times. This reduced to the dataset to 160,600 question-answer pairs samples, or 75% of the original dataset. The number of unique answers in this reduced dataset were 89. The complete list of answers and their frequencies in the reduced dataset are reported in Appendix C.2. When observing the frequencies, it is clear that we have an extremely imbalanced dataset. For example, the majority class makes up nearly 22% of the reduced dataset. This will make sufficient learning from the smaller classes quite challenging. We separated the reduced dataset into a training, validation, and test set according to a 60:20:20 split. Because we had unreliable language identifiers and many low-resource languages, we decided to rather construct a multilingual vocabulary and make use of cross-lingual sentence embedding techniques. As for the many typos and misspellings in the text, we decided not to limit the vocabulary to words with a frequency above a certain threshold. This also ensured that low-resource languages with words that would have very low occurrence would still be preserved in the vocabulary.

Chapter 6

Experiments

6.1 Introduction

We formalize our problem as an answer selection task for which we have the method of encoding the question and the method of selecting the most appropriate answer. In the absence of reliable language labels in the MomConnect dataset, we resort to a multilingual approach. We have already noted a high level of noise in the questions and answers. This is actually fortunate, as it simulates the type of incoming messages the platform can expect. A good multilingual question-answering model should ideally be language-invariant and robust to noise and domain-changes, so that it might one day be deployed in the MomConnect environment. We design a series of experiments to answer the following research questions:

1. Can we train a machine learning model to automate the answering process of MomConnect?

2. Can we explicitly or implicitly learn a cross-lingual embedding space where knowledge is transferred from the high-resource languages to the low-resource languages in our dataset?

3. How well do our models perform on low-resource languages?

We experiment with both parametric and non-parametric machine learning models. We take two approaches to learning cross-lingual embedding spaces. First we rely on the code-mixing presence in the dataset to provide a cross-lingual signal, and then we use the shared template as an explicit cross-lingual signal. Finally, we conduct a comparative evaluation of our different machine learning models and their associated cross-lingual embeddings on a leave-out test set and low-resource subset of the test set. All the experiments can be found at `github.com/JeannieDaniel/momconnect`. The low-resource subset is created by scoring and ranking questions based on the frequencies of their word content, and taking the bottom 25%. With these experiments, we aim to assess the viability of automation in the MomConnect environment. In Section 6.3 we describe the tools we used, the steps we took to de-duplicate the answer set and to create our training, validation, test, and low-resource test sets, and our text preprocessing methods. In Section 6.4 we describe the different approaches we took to encode questions and then classify the encodings to the most appropriate answer. We also analyze the cross-lingual embeddings produced by some of the models by evaluating the cosine distance of the sentence embeddings of translation pairs. Lastly, we report our results as well as an in-depth discussion in Sections 6.5 and 6.6.

6.2 Answer Selection

We design our experiments as answer selection tasks, where we train models to select the most appropriate answer from a set of candidate answers for a given question. For a question and set of candidate answers, the answer selection task can typically be divided into transforming the natural language question to some form of vector representation, and then ranking and selecting the most appropriate answer using some machine learning model. A Frequently Asked Questions (FAQ) approach relies on the accumulation of human-generated answers that are in response to questions posed by users. It economically reuses previously answered questions to assist in answering as yet unseen questions (Burke *et al.*, 1997). State-of-the-art approaches to answer selection achieve satisfactory results on list, factoid, and definition questions, but fail in real-world scenarios where the questions and answers are more complex (Wang and Ittycheriah, 2015). One reason for this is the challenge of calculating the similarity between questions where there is little word overlap. We explore using the traditional bag-of-words encoding with multinomial naive Bayes, SGNS and FastText with k-nearest neighbours, LSTM networks, bi-directional LSTM networks with pooling layers, multi-head attention encoders, and Siamese networks to learn the sentence embeddings.

6.3 Data Pipeline

The challenges we face are what one can expect when working with a real-world healthcare dataset. The raw dataset is not ideal for language modelling or testing machine learning models, for the following reasons:

- there are missing and/or illegitimate values in many of the fields,
- the questions are posed in multiple, often low-resource languages, and have unreliable language labels,
- the questions have a high level of noise in the form of code-mixing, mixed language, spelling errors, typos, and shorthand,

- and the template answers contain inconsistencies and near-duplicates.

We design a data pipeline for transforming the original, raw dataset to an experiment-ready dataset. We do this by removing unnecessary fields, dropping low-quality or illegitimate data points, de-duplicating the answer set, and finally splitting the dataset into a training, validation, and test set. We also develop a standard preprocessing pipeline for cleaning the text, based on our observations of the training set.

6.3.1 Tools and Libraries

All our preprocessing and machine learning experiments are performed using Python 3. We make use of Github for version control and collaboration with research partners. We constructed the deep learning architectures using TensorFlow 2.0 (Abadi *et al.*, 2015). We used Pandas (McKinney, 2010) to read in and preprocess the raw MomConnect dataset. Other notable Python packages are Scikit-learn (Pedregosa *et al.*, 2011), which provides us with out-of-the-box frameworks for the shallow machine learning models, Numpy (Oliphant, 2015), which provides a framework for linear algebra data structures and functions, and Gensim (Řehůřek and Sojka, 2010) which provides us with out-of-the-box frameworks for SGNS and FastText.

6.3.2 Dataset Creation

Our objective is to adapt the original MomConnect dataset to one that is suited for language modelling and classification. The original dataset contains many noisy data points and near-duplicates in the answer set. We began by dropping all the data fields except `helpdesk_question` and `helpdesk_reply`, as we ultimately came to the conclusion that the other data fields were not used in the manual answering process, and so played a little role in guiding the answer. Subsequently we performed some rudimentary text sanitation on the questions by removing all the delimiters for new lines, tabs and rows and dropping all question-answer pairs where the length of the reply was less than 30 characters or the length of the question was less than 3 characters.

Next we iteratively detected and replaced near-duplicates with their more popular twin until no more near-duplicates could be found. To identify the near-duplicates, we calculate the Jaccard similarity between every pair of answers in the entire answer set. If the Jaccard similarity exceeds a certain threshold, we substitute the less popular answer (lower frequency of occurrence) with the more popular answer (higher frequency of occurrence). The prevalence of near-duplicates was approximately 50% and so the resulting answer set was halved in size.

The frequency of answers in the dataset approximates a power law distribution, where 80% of answers occur only once in the entire dataset. The remaining 20% of unique answers account for 85% of the question-answer pairs. Of the remaining 20% of answers, half occur only twice in the dataset. If there are too few questions per unique answer, we cannot sufficiently learn from and classify new, as yet unseen questions for that answer. This obviously presents a challenge for the answer selection task. We decided to focus our efforts on the most frequently used answers. We chose the frequency threshold to be 256, and only kept answers that occurred more than 256 times in the entire dataset, following our cleaning and de-duplication efforts. This threshold left us with 89 template answers, which made up almost 75% of the entire dataset. Each answer was assigned a unique integer ID.

Then we once-off, randomly split the reduced dataset into a training, validation, and test set, according to a 60:20:20 proportion split. For consistency, the same sets were used throughout our experimentation. Finally, we wanted a way to evaluate the models' performance on low-resource languages. In the absence of reliable language labels and the prevalence of code-mixing, we chose to use the frequency of the words as a proxy for "low-resource language" presence. We ranked all the words found in the training set according to their frequency and chose the 99th percentile as the threshold. Questions were then scored and ranked by the proportion of words below the frequency threshold. The bottom 25% of the test set was chosen as the low-resource test set.

6.3.3 Text Preprocessing

The WhatsApp and SMS text exchanges found in our dataset are informal language with shorthand, code-mixing, typos, spelling errors and often contain unwanted tokens such as newline delimiters or emojis. Therefore we need to take additional steps to sanitize the text before we can proceed with our experiments. We use the preprocessing module from the Gensim library (Řehůřek and Sojka, 2010) to clean and tokenize the questions. Cleaning the text involved casting all text to lowercase, and stripping away all characters that are not part of the Latin Alphabet or Arabic Numerals. Tokenizing means transforming the text into a list of valid tokens by splitting on white spaces.

6.3.4 Multilingual Vocabulary

Due to the absence of reliable language labels, we resort to creating a multilingual vocabulary. To do this, we compile a vocabulary of all the unique words and tokens found in the training set. Our concern was that the low-resource languages within the dataset would have many infrequently-occurring words, and that limiting the vocabulary to words with a frequency above a certain threshold would discard the few occurrences of these low-resource languages. Even though these words would be sparsely distributed, we wanted the models to have the opportunity to learn from these languages. To preserve the low-resource languages within the dataset, we decided to not in any way limit the vocabulary of unique words and tokens found in the training set. This meant that the vocabulary also included many of the typos and misspellings of more frequently occurring words, which represented a real-world scenario where users' questions contain this sort of noise. This resulted in a vocabulary size of $57,545$. Aside from the benchmark test, every model is trained to accommodate this vocabulary. Each word in the vocabulary a unique integer ID as identifier. During preprocessing, each sentence is converted to a variable-length sequence of word IDs. We use the sequences to train the various machine learning models and evaluate their ability to select the most appropriate answer.

6.4 Experimental Design

As previously mentioned, we reduced the dataset to question-answer pairs where the answer occurs at least 256 times in the entire dataset, resulting in 89 unique answers. We approach the problem as an answer selection task, where each machine learning model is trained to classify a question to the most appropriate answer. For the deep learning models, our training objective is to minimize the categorical cross-entropy loss, and we evaluate on the classification accuracy. For the non-parametric models, i.e. k-nearest neighbours, we aim to maximize the accuracy. We use the validation accuracy to tune the hyperparameters and select the best models for testing. We evaluate the performance using four metrics: the test accuracy, the low-resource test accuracy, the top-5 test accuracy, and the top-5 low-resource test accuracy. We use the top-5 accuracy as well as the (top-1) accuracy, as the latter is a particularly strict evaluation metric for a multiclass classification task with 89 classes. The top-5 accuracy is calculated as the proportion of events where the correct label was predicted amongst the top 5 classes, ranked by probability. To evaluate the quality of the cross-lingual embeddings produced by some of the models, we design a small experiment to assess the quality of some of the models' cross-lingual embeddings for English and Zulu.

6.4.1 Multinomial Naive Bayes with Bag-of-words

We train a baseline model against which we intend to compare the results all our models. We make use of the bag-of-words method to construct feature vectors from the incoming questions and multinomial naive Bayes (MNB) as our baseline classifier, as it is fast to train, requires no hyperparameter tuning, and makes the (naive) feature independence assumption (Rennie *et al.*, 2003). The classifier is based on the Bayes theorem that states the following relationship, given a class label y and the elements of the feature vector $x_1, \ldots, x_n$:

$$P(y|x_1, \ldots, x_n) = \frac{P(y)P(x_1, \ldots, x_n|y)}{P(x_1, \ldots, x_n)}. \tag{6.1}$$

Using the assumption that the elements of the feature vector are conditionally independent, we can reduce $P(x_1, \ldots, x_n|y)$ to $\prod_{i=1}^{n} P(x_i|y)$. Then Equation 6.1 can be simplified to

$$P(y|x_1, \ldots, x_n) = \frac{P(y) \prod_{i=1}^{n} P(x_i|y)}{P(x_1, \ldots, x_n)}. \tag{6.2}$$

Given that $P(x_1, \ldots, x_n)$ is constant for all y, we can then deduce that

$$P(y|x_1, \ldots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i|y). \tag{6.3}$$

After training, inference for a test feature vector $\boldsymbol{x}$ is done as follows:

$$\hat{y} = \arg\max_{y} P(y) \prod_{i=1}^{n} P(x_i|y). \tag{6.4}$$

In the multinomial setting, $P(x_i|y)$ is the probability that x_i appears in a sample of class y. This quantity is estimated from the training set using a smoother version of maximum likelihood. We implement the classifier using the Scikit-learn library (Pedregosa *et al.*, 2011). We train the MNB classifier on the training set, and choose the word frequency threshold that maximizes the validation accuracy, resulting in a vocabulary size of the 9286 most frequent words in the training set. The test and low-resource test results for our baseline are reported in Table 6.1. The top-5 test and low-resource test results for our baseline are reported in Table 6.2.

6.4.2 Skip-gram Negative Sampling and FastText with k-Nearest Neighbours

We employ two out-of-the-box cross-lingual word embedding frameworks to learn vector representations of all the words in our multilingual vocabulary. These two models are skip-gram negative sampling (Mikolov *et al.*, 2013*c*) and FastText (Bojanowski *et al.*, 2017). FastText is conceptually similar to SGNS, except it also takes into account subword information. Both models are publicly available for development with the Gensim library (Řehůřek and Sojka, 2010). We rely solely on the code-mixing within the training set to provide a weak cross-lingual signal for these two embedding models. We did experiment with parameter tuning, but found that in the case of SGNS the best performance on the validation set was achieved with the default parameters. In the case of FastText we only changed the default setting from continuous-bag-of-words to skip-gram. After training the embedding models on the questions found in the training set, we extract the cross-lingual word embeddings. We construct a sentence embedding by taking the average of all the word embeddings in the sentence (Wieting *et al.*, 2015). Then we train

k-nearest neighbour classifiers to predict the most appropriate answer, with $k = 1, 5, 25, 50$. The best validation scores were achieved by using cosine as the distance metric and using weighted majority voting, where the contribution of each nearest neighbour is inversely proportion to its distance from the query vector. We report the top-1 test and low-resource test accuracies for SGNS and FastText with the nearest neighbour classifiers in Table 6.1.

6.4.3 Vanilla LSTM Classifier

We construct a vanilla LSTM classifier that takes in variable length sequences of word IDs. The word IDs all correspond to words in our large multilingual vocabulary. Each word ID is fed to an embedding layer that outputs a 300-dimensional embedding. The embedding layer is followed by a dropout layer for regularization. Then the sequence of embeddings is fed to an LSTM layer with tanh activations. The final hidden representation of the LSTM is then passed to the dense output layer, where we perform classification. The weights of the embedding layer is updated so as to minimize the categorical cross-entropy between the model outputs and the ground truth labels. The ground truth labels serve as a weak cross-lingual supervision, and thus the final hidden representations produced by the LSTM network are essentially cross-lingual sentence embeddings. We train the network with a batch size of 32 and early stopping on the validation accuracy to prevent overfitting. We optimize using the Adadelta optimization algorithm (Zeiler, 2012) with a learning rate of 0.5. We train and test networks for 128, 256, 512, and 1024 hidden units, respectively. We report the top-1 test and low-resource test accuracies in Table 6.1. We report the top-5 test and low-resource test accuracies in Table 6.2.

6.4.4 Bi-directional LSTM with Pooling

We explore the architecture of InferSent (Conneau *et al.*, 2017), where a bi-directional LSTM model is combined with pooling layers. We feed the sequence of word IDs to an embedding layer, which outputs a 300-dimensional embedding. Then the sequence of embeddings are fed to the bi-directional LSTM network with tanh activations. We take the entire sequence of hidden representations and then perform max-pooling and average-pooling, respectively, to reduce the sequence of hidden representations to a single vector representation. We attach a dense output layer immediately after the pooling layer for classification, as well as a dropout layer after the embedding layer to regularize the network. We train the network with a batch size of 32 and early stopping on the validation accuracy to prevent overfitting. We optimize using the the Adadelta optimization algorithm (Zeiler, 2012) with a learning rate of 0.5. We train and test separate networks using max-pooling and average-pooling for bi-directional LSTM networks with hidden units 128, 256, 512, and 1024, respectively. We report the top-1 test and low-resource test accuracies in Table 6.1. We report the top-5 test and low-resource test accuracies in Table 6.2.

6.4.5 Multi-head Attention Classifier

As an alternative to using LSTMs, we employ the multi-head attention encoder of the Transformer (Vaswani *et al.*, 2017), to model the questions. The architecture is as described in Section 3.4.6, but with a minor modification. The encoder is followed by an average-pooling layer, which is then followed by a softmax classification layer. The only modification to the original code obtained from the TensorFlow tutorial [1] is a dropout layer between the embedding layer and the positional encoding layer of the encoder. After exploring different hyperparameters during model selection, we decided on using a 2-layer encoder with 8 heads, and for our final results we report for embedding dimensions of 128, 256, 512 and 1024. We train each network with a batch size of 32 and early stopping on the validation accuracy to prevent overfitting. We optimize the training using the Adadelta optimization algorithm (Zeiler, 2012) with a learning rate of 0.25. We report the top-1 test and low-resource test accuracies in Table 6.1. We report the top-5 test and low-resource test accuracies in Table 6.2.

6.4.6 Siamese Triplet Loss Training

With Siamese triplet loss training, we aim to learn a cross-lingual sentence embedding space and then perform k-nearest neighbour classification for answer selection. As previously discussed in Section 2.5, Siamese triplet loss training creates embedding spaces where similar items are pulled closer to one another, and dissimilar items are pushed away from one another. We choose the multi-head attention encoder (Vaswani *et al.*, 2017) as our model to be trained using Siamese triplet loss. For the architecture, we choose the number of layers as 2, with 8 heads, and 128 dimensions. This is similar to one of the architectures explored in Section 6.4.5, with the only difference being that we replace the softmax classification layer with a dense layer with only linear activations. For the sampling of triplets, we employ a technique called online semi-hard mining (Schroff *et al.*, 2015). For a given minibatch, we first compute the embeddings for all the samples in the minibatch. To make up the triplets for the minibatch, all the possible positive anchor pairs $(\boldsymbol{x}_a, \boldsymbol{x}_p)$ are selected, and accompanied with a semi-hard negative that satisfies $D(\boldsymbol{x}_a, \boldsymbol{x}_p) < D(\boldsymbol{x}_a, \boldsymbol{x}_n) < D(\boldsymbol{x}_a, \boldsymbol{x}_p) + m$, where $D(\cdot)$ is the distance function and m is the margin. We perform the Siamese triplet loss training with minibatch sizes of 256, cosine as our distance function and a margin m of 0.5. We optimize using Adadelta (Zeiler, 2012) with a learning rate of 0.05 and early

[1] https://www.tensorflow.org/tutorials/text/transformer

stopping on the validation loss to prevent overfitting. We compare the effect of using unbalanced (as is) batches, versus balanced batches where we undersample the majority (most-frequently occurring answer label) class (Lemaître *et al.*, 2017). Using the embeddings learned from the Siamese training, we train k-nearest neighbour classifiers to predict the most appropriate answer, for models with $k = 1, 5, 25, 50$. As in Section 6.4.2, we use cosine as the distance metric and weighted majority voting for class selection. We report the top-1 test and low-resource test accuracies in Table 6.1.

6.4.7 Analysis of Cross-lingual Embeddings

We design a small experiment to assess the quality of the cross-lingual embeddings for English and Zulu. English is the most common language and Zulu the second-most common, based on the unique sign-up language labels present in the dataset as seen in Table 5.2. We choose these two languages as English is a high-resource language, and although Zulu is considered a low-resource language according to the criteria of Cieri *et al.* (2016), Google Translate supports the language. We hypothesize that due to the injection of cross-lingual information, the source and target sentence embeddings should have a small distance between them. We synthesize English questions based on some of the most common questions found in the dataset, and then translate them to Zulu using Google Translate. The translations were verified and corrected by a native language speaker. Then we construct sentence embeddings for each question using some of the models we explored and measure the cosine distance between each pair of sentence embeddings. We report the results for the embeddings obtained from the SGNS, FastText, Siamese triplet, and balanced Siamese triplet models in Table 6.3. We use the following five parallel sentence pairs for our analysis:

"Can you drink coca cola when you are pregnant?"
"Ungayiphuza yini i-coca cola uma ukhulelwe?" } - Pair A

"When can I stop breastfeeding?"
"Ngingakuyeka nini ukuncelisa ibele?" } - Pair B

"When can I start feeding my baby solid food?"
"Ngingaqala nini ukondla ingane yami ukudla okuqinile?" } - Pair C

"What are the signs of labour?"
"Yiziphi izimpawu zokubeletha?" } - Pair D

"When can I learn the gender of my baby?"
"Ngingabazi ubulili bengane yami?" } - Pair E

6.5 Results

In Table 6.1 we report the top-1 test and LR test set accuracies for all the models. In Table 6.2 we report the top-5 test and LR test set accuracies for all the models that output softmax probabilities, thus excluding all the models that classify using k-nearest neighbours. In Table 6.3 we report the cosine distances for the English-Zulu translation pairs produced by the models that output sentence embedding vectors, thus the word averaging of SGNS and FastText, and both the unbalanced and balanced Siamese triplets models.

Model	**Top-1 Test**	**Top-1 LR Test**
MNB	52.15%	43.96%
SGNS 1-NN	48.69%	37.30%
SGNS 5-NN	53.51%	42.31%
SGNS 25-NN	57.31%	48.00%
SGNS 50-NN	57.54%	48.47%
FastText 1-NN	50.05%	39.42%
FastText 5-NN	54.44%	43.92%
FastText 25-NN	58.48%	49.41%
FastText 50-NN	58.39%	49.76%
LSTM-128	60.45%	52.30%
LSTM-256	60.82%	52.48%
LSTM-512	60.90%	52.73%
LSTM-1024	60.50%	51.89%
BiLSTM-128, avg-pooling	61.37%	53.41%
BiLSTM-256, avg-pooling	61.50%	53.97%
BiLSTM-512, avg-pooling	61.24%	53.37%
BiLSTM-1024, avg-pooling	60.58%	52.87%
BiLSTM-128, max-pooling	60.52%	52.90%
BiLSTM-256, max-pooling	60.56%	52.55%
BiLSTM-512, max-pooling	60.58%	52.15%
BiLSTM-1024, max-pooling	60.69%	52.76%
Multi-head Attention 128-dim	**61.75%**	**54.42%**
Multi-head Attention 256-dim	61.50%	53.17%
Multi-head Attention 512-dim	61.38%	54.37%
Multi-head Attention 1024-dim	61.06%	52.97%
Siamese Triplets, 1-NN	52.98%	42.81%
Siamese Triplets, 5-NN	56.83%	45.87%
Siamese Triplets, 25-NN	58.69%	47.86%
Siamese Triplets, 50-NN	58.56%	48.09%
Siamese Triplets, balanced, 1-NN	52.70%	42.37%
Siamese Triplets, balanced, 5-NN	56.73%	46.91%
Siamese Triplets, balanced, 25-NN	59.29%	51.20%
Siamese Triplets, balanced, 50-NN	59.44%	51.41%

Table 6.1: Top-1 accuracy for test and low-resource test set

Model	**Top-5 Test**	**Top-5 LR Test**
MNB	82.27%	74.51%
LSTM-128	89.61%	80.93%
LSTM-256	89.03%	79.42%
LSTM-512	90.03%	81.48%
LSTM-1024	88.95%	79.31%
BiLSTM-128, avg-pooling	89.99%	80.36%
BiLSTM-256, avg-pooling	90.28%	80.59%
BiLSTM-512, avg-pooling	90.00%	80.14%
BiLSTM-1024, avg-pooling	90.16%	81.23%
BiLSTM-128, max-pooling	89.70%	81.04%
BiLSTM-256, max-pooling	89.74%	80.32%
BiLSTM-512, max-pooling	89.77%	79.82%
BiLSTM-1024, max-pooling	89.86%	81.16%
Multi-head Attention 128-dim	90.69%	82.33%
Multi-head Attention 256-dim	**91.16%**	**83.47%**
Multi-head Attention 512-dim	90.45%	82.83%
Multi-head Attention 1024-dim	90.23%	81.93%

Table 6.2: Top-5 accuracy for test and low-resource test set

	Cosine Distance				
Model	Pair A	Pair B	Pair C	Pair D	Pair E
SGNS	0.3979	0.5983	0.5965	0.5929	**0.6021**
FastText	0.3769	0.5876	0.5461	0.6474	0.6306
Siamese Triplets	**0.2202**	0.3873	0.3462	0.3381	0.6259
Siamese Triplets (balanced)	0.3685	**0.2721**	**0.1551**	**0.1184**	0.9600

Table 6.3: Cosine distance for translated sentence pairs (English-Zulu)

6.6 Discussion

Based on the results we make several observations regarding the performance of the different architectures and machine learning models. The different metrics paint an interesting picture of the generalizability of the models. The low-resource test set is a subset of the test set that essentially contains all the sentences with the most infrequent words. Some of these words would only be observed in one or two training examples. It is thus a good test of a model's ability to generalize and not overfit to the high-resource languages. However, the nature of a machine learning model is to exploit the patterns found in the data, and can only do as well as the data it learns from. Thus the general discrepancy between the accuracies achieved on the full test set versus the low-resource test set is understandable. In almost all our sets of experiments, the more complex models failed to outperform the less complex models. However, models that were more information-efficient than their counterparts produced increasingly better results. For example, the FastText embeddings combined with k-nearest neighbours showed significantly improved performance over the SGNS embeddings in the top-1 test and low-resource test accuracies. This is in line with our expectations given the associated literature, and demonstrates the benefit of including sub-word information. The LSTM models that made use of the entire sequence and bi-directionally processed the sequence showed significant improvement over unidirectional processing and using only the final hidden state of the LSTM.

An interesting and unexpected result was that unlike in the case of InferSent (Conneau *et al.*, 2017), max-pooling did not outperform average-pooling when used to reduce the variable-length sequence of vectors to a single fixed-length vector. One possible explanation could be that max-pooling is more sensitive to the extreme class imbalances, and thus struggles to generalize on unseen data, while the average-pooling smoothes the outliers and thus generalizes better. It is also interesting that the class of multi-head attention encoder classifiers, that essentially treats the input sequence as a bag-of-words and injects information about the relative positions using the positional encodings, outperforms all the LSTM and bi-directional LSTM models that explicitly perform sequential processing. The multi-head attention encoder model with 128 dimensions achieved the best performance on both the top-1 test and low-resource test accuracy, with 61.75% and 54.42%, respectively. Its larger sister, with 256 dimensions, achieved the best scores on the top-5 test and low-resource test accuracy, with 91.16% and 83.47%, respectively. Another interesting observation is the fact that although some of the multi-head attention encoder models achieved results comparable to the bi-directional LSTM models with average-pooling for the test set, their results for the low-resource test set were much higher. This indicates that the attention models are actually performing better on the low-resource examples and worse on the high-resource examples, relative to the bi-directional LSTM models. The unbalanced Siamese triplet models achieves results comparable to the FastText models on top-1 test accuracy, but with comparatively worse scores on the top-1 low-resource test accuracy. This indicates that the model struggles to generalize on low-frequency words. However, down-sampling the majority class in the training data had a positive effect on the generalizability of the model, and resulted in improved scores over the unbalanced training method.

Although Siamese triplet training does not show significant improvement over the FastText and SGNS models in terms of accuracy, they produce cross-lingual embeddings for English and Zulu that seem to be of a higher quality. The cosine distance between the translation pair embeddings produced by the Siamese models were much smaller compared to those produced by the word embedding models. For example, the embeddings produced by the balanced Siamese model for the synthesized translation pair –*"Yiziphi izimpawu zokubeletha?"* and *"When can I learn the gender of my baby?"*– was found to have a cosine distance of 0.1184. When the same translation pair is evaluated, the SGNS model achieves a cosine distance of 0.5929 and the FastText model a cosine distance of 0.6474. The word embedding models only used the code-mixing present in the questions as cross-lingual signal. This indicates that the shared English template answer served as an effective cross-lingual signal to learn a cross-lingual embedding space that maps concepts across languages to similar areas. Down-sampling the majority class further improves the quality of the cross-lingual embeddings, with improved scores for 3 out of the 5 translation pairs.

Despite our best efforts, we could not breach the 62% boundary for the top-1 accuracy. It seems that within the data there might be some upper bound to the classification accuracy, possibly due to high levels of noise, near-duplicates, and inconsistent mappings of questions to answers. While the results for the top-1 accuracy is impressive given the high level of noise and inconsistencies in the data, it still does not impress at the level one would expect from an automated system. On the other hand, the top-5 accuracies are much more impressive and certainly warrant further development. One could, for example, implement a top-5 recommender system to assist the helpdesk staff in answering large volumes of questions.

Chapter 7

Conclusion

7.1 Introduction

Low-resource languages remain underrepresented in the field of natural language processing. This poses significant challenges for advancing digital services in countries with low-resource languages. We were presented with a unique feasibility study for automating a digital healthcare platform called MomConnect, which has provided healthcare information and emotional support to over 2.6 million users in South Africa since 2014. This platform receives questions in all 11 official languages of South Africa, many of which are low-resource languages. All the responses are English template responses. The service has struggled to keep up with the growing user base and increase in incoming questions, and as a result the median response time currently is 20 hours. With this in mind, our research objective was to explore automatically answering questions with the help of natural language processing and machine learning models. This can help increase the efficiency of the MomConnect helpdesk.

7.2 Summary of Findings

Due to the multilinguality, low-resource languages, code-mixing and lack of reliable language labels, we decided to implement cross-lingual embeddings to learn vector representations of the questions. We explored simple models such as skip-gram negative sampling and FastText to learn embeddings for the words in the sentences and then average over these to construct cross-lingual sentence embeddings. These models learned cross-lingual embedding spaces from the signals provided by the code-mixing found in the questions. To automatically select the most appropriate answer, we made use of k-nearest neighbour classifiers. We also investigated deep learning architectures such as long short-term memory networks and the Transformer architecture to classify questions to the most appropriate answer. The cross-lingual embeddings are implicitly learned using the shared English tempate answers. Lastly we investigated using a metric learning model called Siamese triplet loss, where the model explicitly pulls the embeddings of similar sentences closer to one another and pushes the embeddings of dissimilar sentences further apart. The indication of similarity is given by if two questions, regardless of their language, share the same answer or not. The cross-lingual embeddings are then classified to the most appropriate answer using k-nearest neighbour classifiers. We also evaluated the quality of the cross-lingual embeddings produced by some of the models for English and Zulu.

The Siamese triplet loss models failed to improve on the end-to-end classification models and achieved results comparable to the SGNS and FastText models which only used the code-mixing within the dataset as cross-lingual signals. However, our evaluation using parallel English-Zulu question pairs demonstrated that Siamese triplet loss models might have learned a cross-lingual embedding space. The cross-lingual embeddings produced by the Siamese models were of a higher quality than that of the SGNS and FastText models. The multi-head attention encoder model (from the Transformer architecture) demonstrated improved performance over the LSTM models, and also seemed to generalize better on the low-frequency words. The multi-head attention encoder model with 128-dimensional embeddings achieved a top-1 test accuracy of 61.75% and top-1 low-resource test accuracy of 54.42%. The multi-head attention encoder model with 256-dimensional embeddings achieved a top-5 test accuracy of 91.16% and top-5 low-resource test accuracy of 83.47%.

While the results for the top-1 accuracy are impressive given the high level of noise and inconsistencies in the answering process, they still do not impress at the level one would expect from an automated system. Therefore, we propose to rather use the top 5 answers as predicted by one of the models in a recommendation system setting. Such a system can serve in a semi-automated answer selection process, with a human in the loop to choose the final answer. If 75% of the incoming questions can be dealt with in a semi-automated manner, the burden on the current staffing compliment could be reduced significantly. In the case where the human does not agree with any of the suggested answers, the option should remain for the human operator to manually select the correct standardized response, as is currently done. This feedback can help improve the automated response service, and enable future research studies.

7.3 Summary of Contributions

In this thesis we made several contributions to the field of natural language processing. In Chapter 3 we provide a review of the advances in word embeddings, sentence embeddings, and cross-lingual embedding techniques and in Chapter 6 we apply some of these techniques in a very noisy, multilingual, low-resource setting to perform answer selection. Due to the highly sensitive nature of the MomConnect dataset, we had to go through a lengthy process of requesting permission, obtaining ethical clearance, and anonymizing the data according to internationally recognized data protection guidelines. As such, we provide an overview of necessary data protection, anonymization, and ethical clearance processes for working with sensitive data in Chapter 4. Based on our promising top-5 accuracy results in Chapter 6, we also provide a proof of concept for automating the MomConnect helpdesk question-answering pipeline with a top-5 recommendation system. This system can increase the scalability and efficiency of MomConnect by increasing the rate at which the helpdesk staff can answer incoming questions. An early version of our work was published and presented orally at the 57th Annual Gathering of the Association for Computation Linguistics in Florence, Italy in July 2019 (Daniel *et al.*, 2019).

7.4 Suggestions for Future Research

Using subword information showed improved performance over using only the whole words, based on our experiments using FastText and SGNS word embeddings. One can imagine that training on sequences of characters or even sequences of byte-pair encodings would improve on the current method of only training on sequences of word IDs. This would also enable the models to deal with unseen words during testing.

The success of the multi-head attention encoder models warrants further exploration into using attention models for language modelling, rather than the traditional approaches of LSTMs. Rebalancing the dataset showed improved performance, and balanced batching to smooth the distribution of classes should be investigated in future research. It may also be beneficial to enhance the low-resource languages with external sources for improved performance.

Lastly, it would be an interesting research topic to explore answer generation as opposed to answer selection, especially when it comes to addressing the long tail of the answer set.

List of References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y. and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.
Available at: `http://tensorflow.org/`

Agirre, E., Banea, C., Cardie, C., Cer, D., Diab, M., Gonzalez-Agirre, A., Guo, W., Mihalcea, R., Rigau, G. and Wiebe, J. (2014). SemEval-2014 task 10: Multilingual semantic textual similarity. In: *Proceedings of the 8th International Workshop on Semantic Evaluation*, pp. 81–91. Association for Computational Linguistics, Dublin, Ireland.
Available at: `https://www.aclweb.org/anthology/S14-2010`

Artetxe, M. and Schwenk, H. (2018). Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Computing Research Repository*, vol. abs/1812.10464.
Available at: `http://arxiv.org/abs/1812.10464`

Baldi, P. and Chauvin, Y. (1993). Neural networks for fingerprint recognition. *Neural Computation*, vol. 5, pp. 402–418.
Available at: `https://authors.library.caltech.edu/12477/1/BALnc93.pdf`

Barron, P., Peter, J., LeFevre, A.E., Sebidi, J., Bekker, M., Allen, R., Parsons, A.N., Benjamin, P. and Pillay, Y. (2018). Mobile health messaging service and helpdesk for South African mothers (MomConnect): history, successes and challenges. *BMJ Global Health*, vol. 3, no. Suppl 2.
Available at: `https://gh.bmj.com/content/3/Suppl_2/e000559`

Bellet, A., Habrard, A. and Sebban, M. (2013). A survey on metric learning for feature vectors and structured data. *Computing Research Repository*, vol. abs/1306.6709.
Available at: `http://arxiv.org/abs/1306.6709`

Bengio, Y., Ducharme, R., Vincent, P. and Janvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155.

ISSN 1532-4435.
Available at: `http://dl.acm.org/citation.cfm?id=944919.944966`

Bengio, Y., Simard, P. and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166. ISSN 1045-9227.
Available at: `https://doi.org/10.1109/72.279181`

Bojanowski, P., Grave, E., Joulin, A. and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146.
Available at: `https://www.aclweb.org/anthology/Q17-1010/`

Bowman, S.R., Angeli, G., Potts, C. and Manning, C.D. (2015). A large annotated corpus for learning natural language inference. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 632–642. Association for Computational Linguistics, Lisbon, Portugal.
Available at: `https://www.aclweb.org/anthology/D15-1075`

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E. and Shah, R. (1993). Signature verification using a "Siamese" time delay neural network. In: *Proceedings of the 6th International Conference on Neural Information Processing Systems*, NIPS'93, pp. 737–744. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
Available at: `http://dl.acm.org/citation.cfm?id=2987189.2987282`

Bryson, A.E. and Ho, Y.C. (1969). *Applied Optimal Control.* Blaisdell, New York.

Burke, R., Hammond, K., Kulyukin, V., Lytinen, S.L., Tomuro, N. and Schoenberg, S. (1997). Question answering from frequently asked question files: experiences with the FAQ FINDER system. Tech. Rep., University of Chicago.
Available at: `https://www.aaai.org/ojs/index.php/aimagazine/article/view/1294`

Cauchy, M.A. (1847). Méthode générale pour la résolution des systemes d'équations simultanées. *Comptu Rendu des Séances Académie des sciences*, pp. 536–538.

Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I. and Specia, L. (2017). SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 1–14. Association for Computational Linguistics, Vancouver, Canada.
Available at: `https://www.aclweb.org/anthology/S17-2001`

Cer, D., Yang, Y., Kong, S., Hua, N., Limtiaco, N., John, R.S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., Sung, Y., Strope, B. and Kurzweil, R. (2018). Universal sentence encoder. *Computing Research Repository*, vol. abs/1803.11175.
Available at: `http://arxiv.org/abs/1803.11175`

Cho, K., van Merrienboer, B., Gülçehre, Ç., Bougares, F., Schwenk, H. and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *Computing Research Repository*, vol. abs/1406.1078.
Available at: `http://arxiv.org/abs/1406.1078`

Chopra, S., Hadsell, R. and LeCun, Y. (2005). Learning a similarity metric discriminatively, with application to face verification. In: *Proceedings of the 2005 IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1 of *CVPR '05*, pp. 539–546. IEEE Computer Society, Washington, DC, USA. ISBN 0-7695-2372-2.
Available at: `https://doi.org/10.1109/CVPR.2005.202`

Church, K.W. and Hanks, P. (1990). Word association norms, mutual information, and lexicography. *Computational Linguistics*, vol. 16, no. 1, pp. 22–29. ISSN 0891-2017.
Available at: `http://dl.acm.org/citation.cfm?id=89086.89095`

Cieri, C., Maxwell, M., Strassel, S. and Tracey, J. (2016). Selection criteria for low resource language programs. In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, LREC'16, pp. 4543–4549. European Language Resources Association, Portorož, Slovenia.
Available at: `https://www.aclweb.org/anthology/L16-1720`

Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In: *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pp. 160–167. ACM, New York, NY, USA. ISBN 978-1-60558-205-4.
Available at: `http://doi.acm.org/10.1145/1390156.1390177`

Conneau, A., Kiela, D., Schwenk, H., Barrault, L. and Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. *Computing Research Repository*, vol. abs/1705.02364.
Available at: `http://arxiv.org/abs/1705.02364`

Conneau, A., Rinott, R., Lample, G., Williams, A., Bowman, S., Schwenk, H. and Stoyanov, V. (2018). XNLI: Evaluating cross-lingual sentence representations. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2475–2485. Association for Computational Linguistics, Brussels, Belgium.
Available at: `https://www.aclweb.org/anthology/D18-1269`

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27. ISSN 0018-9448.
Available at: `https://doi.org/10.1109/TIT.1967.1053964`

Dacey, D.M. and Petersen, M.R. (1992). Dendritic field size and morphology of midget and parasol ganglion cells of the human retina. *Proceedings of the National Academy of Sciences*, vol. 89, no. 20, pp. 9666–9670. ISSN 0027-8424.
Available at: `https://www.pnas.org/content/89/20/9666`

Daniel, J.E., Brink, W., Eloff, R. and Copley, C. (2019). Towards automating healthcare question answering in a noisy multilingual low-resource setting. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 948–953. Association for Computational Linguistics, Florence, Italy.
Available at: `https://www.aclweb.org/anthology/P19-1090/`

Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K. and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407.
Available at: `http://lsa.colorado.edu/papers/JASIS.lsi.90.pdf`

Demuth, H.B., Beale, M.H., De Jess, O. and Hagan, M.T. (2014). *Neural Network Design*. 2nd edn. Martin Hagan, USA. ISBN 0971732116, 9780971732117.
Available at: `https://hagan.okstate.edu/NNDesign.pdf`

Devlin, J., Chang, M., Lee, K. and Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *Computing Research Repository*, vol. abs/1810.04805.
Available at: `http://arxiv.org/abs/1810.04805`

El Emam, K. and Dankar, F.K. (2008). Protecting privacy using k-anonymity. *Journal of the American Medical Informatics Association*, vol. 15, no. 5, pp. 627–637. ISSN 1527-974X.
Available at: `https://doi.org/10.1197/jamia.M2716`

Engelhard, M., Copley, C., Watson, J., Pillay, Y., Barron, P. and LeFevre, A.E. (2018). Optimising mHealth helpdesk responsiveness in South Africa: towards automated message triage. *BMJ Global Health*, vol. 3, no. Suppl 2.
Available at: `https://gh.bmj.com/content/3/Suppl_2/e000567`

Fano, R.M. (1961). Transmission of information: A statistical theory of communications. *American Journal of Physics*, vol. 29, no. 11, pp. 793–794.
Available at: `https://doi.org/10.1119/1.1937609`

Gittens, A., Achlioptas, D. and Mahoney, M.W. (2017). Skip-gram - Zipf + uniform = vector additivity. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 69–76. Association for Computational Linguistics, Vancouver, Canada.
Available at: `https://www.aclweb.org/anthology/P17-1007`

Golub, G.H. and Reinsch, C. (1970). Singular value decomposition and least squares solutions. *Numererical Mathematics*, vol. 14, no. 5, pp. 403–420. ISSN 0029-599X.
Available at: `http://dx.doi.org/10.1007/BF02163027`

Goodfellow, I., Bengio, Y. and Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Gouws, S. and Søgaard, A. (2015). Simple task-specific bilingual word embeddings. In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp.

1386–1390. Association for Computational Linguistics, Denver, Colorado.
Available at: https://www.aclweb.org/anthology/N15-1157/

Harris, Z. (1954). Distributional structure. *Word*, vol. 10, no. 23, pp. 146–162.
Available at: https://www.tandfonline.com/doi/pdf/10.1080/00437956.1954.11659520

Hastie, T., Tibshirani, R., Friedman, J. and Franklin, J. (2017). The elements of statistical learning: Data mining, inference, and prediction (corrected 12th printing). *Mathematical Intelligence*, vol. 27, pp. 83–85.
Available at: https://web.stanford.edu/~hastie/Papers/ESLII.pdf

HHS.gov (2012). Guidance regarding methods for de-identification of protected health information in accordance with the Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule. https://www.hhs.gov. Accessed: 2019-11-30.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, vol. 9, no. 8, pp. 1735–1780. ISSN 0899-7667.
Available at: http://dx.doi.org/10.1162/neco.1997.9.8.1735

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks*, vol. 4, no. 2, pp. 251–257. ISSN 0893-6080.
Available at: http://dx.doi.org/10.1016/0893-6080(91)90009-T

Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In: *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pp. 168–177. ACM, New York, NY, USA. ISBN 1-58113-888-1.
Available at: http://doi.acm.org/10.1145/1014052.1014073

Iyyer, M., Manjunatha, V., Boyd-Graber, J. and Daumé III, H. (2015). Deep unordered composition rivals syntactic methods for text classification. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1681–1691. Association for Computational Linguistics, Beijing, China.
Available at: https://www.aclweb.org/anthology/P15-1162/

Kiros, R., Zhu, Y., Salakhutdinov, R., Zemel, R.S., Torralba, A., Urtasun, R. and Fidler, S. (2015). Skip-thought vectors. *Computing Research Repository*, vol. abs/1506.06726.
Available at: http://arxiv.org/abs/1506.06726

Koch, G.R. and and, R.Z. (2015). *Siamese Neural Networks for One-Shot Image Recognition*. Master's thesis, University of Toronto, Toronto, Canada.
Available at: https://www.cs.utoronto.ca/~gkoch/files/msc-thesis.pdf

Lemaître, G., Nogueira, F. and Aridas, C.K. (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, vol. 18, no. 17, pp. 1–5.
Available at: `http://jmlr.org/papers/v18/16-365`

Levy, O. and Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. In: *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, pp. 2177–2185. MIT Press, Cambridge, MA, USA.
Available at: `http://dl.acm.org/citation.cfm?id=2969033.2969070`

Levy, O., Søgaard, A. and Goldberg, Y. (2017). A strong baseline for learning cross-lingual word embeddings from sentence alignments. In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pp. 765–774. Association for Computational Linguistics, Valencia, Spain.
Available at: `https://www.aclweb.org/anthology/E17-1072`

Li, X. and Roth, D. (2002). Learning question classifiers. In: *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, COLING '02, pp. 1–7. Association for Computational Linguistics, Stroudsburg, PA, USA.
Available at: `https://doi.org/10.3115/1072228.1072378`

Lin, T., Maire, M., Belongie, S.J., Bourdev, L.D., Girshick, R.B., Hays, J., Perona, P., Ramanan, D., Dollár, P. and Zitnick, C.L. (2014). Microsoft COCO: common objects in context. *Computing Research Repository*, vol. abs/1405.0312.
Available at: `http://arxiv.org/abs/1405.0312`

Marcus, M.P., Marcinkiewicz, M.A. and Santorini, B. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, vol. 19, no. 2, pp. 313–330. ISSN 0891-2017.
Available at: `http://dl.acm.org/citation.cfm?id=972470.972475`

Marelli, M., Bentivogli, L., Baroni, M., Bernardi, R., Menini, S. and Zamparelli, R. (2014). Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In: *Proceedings of the 8th International Workshop on Semantic Evaluation*, pp. 1–8. Association for Computational Linguistics, Dublin, Ireland.
Available at: `https://www.aclweb.org/anthology/S14-2001/`

McKinney, W. (2010). Data structures for statistical computing in Python. In: *Proceedings of the 9th Python in Science Conference*, vol. 445 of *SciPy'10*, pp. 51–56. Austin, TX.
Available at: `https://conference.scipy.org/proceedings/scipy2010/pdfs/mckinney.pdf`

Mikolov, T., Chen, K., Corrado, G.S. and Dean, J. (2013*a*). Efficient estimation of word representations in vector space. *Computing Research Repository*, vol. abs/1301.3781.
Available at: `https://arxiv.org/abs/1301.3781`

Mikolov, T., Le, Q.V. and Sutskever, I. (2013*b*). Exploiting similarities among languages for machine translation. *Computing Research Repository*, vol. abs/1309.4168.
Available at: `http://arxiv.org/abs/1309.4168`

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. and Dean, J. (2013*c*). Distributed representations of words and phrases and their compositionality. *Computing Research Repository*, vol. abs/1310.4546.
Available at: `http://arxiv.org/abs/1310.4546`

Miller, G.A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, vol. 38, no. 11, pp. 39–41.
Available at: `https://dl.acm.org/citation.cfm?id=219748`

Mueller, J. and Thyagarajan, A. (2016). Siamese recurrent architectures for learning sentence similarity. In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pp. 2786–2792. AAAI Press.
Available at: `http://dl.acm.org/citation.cfm?id=3016100.3016291`

Myers-Scotton, C. (1993). Common and uncommon ground: Social and structural factors in code-switching. *Language in Society*, vol. 22, no. 4, pp. 475–503.
Available at: `https://doi.org/10.1017/S0047404500017449`

Nielsen, M.A. (2015). *Neural Networks and Deep Learning*. Determination Press.
Available at: `http://neuralnetworksanddeeplearning.com/`

Noack, R. and Gamio, L. (2015). The world's languages, in 7 maps and charts. `https://www.washingtonpost.com/news/worldviews/wp/2015/04/23/the-worlds-languages-in-7-maps-and-charts/`. Accessed: 2019-11-30.

Nwankpa, C., Ijomah, W., Gachagan, A. and Marshall, S. (2018). Activation functions: Comparison of trends in practice and research for deep learning. *Computing Research Repository*, vol. abs/1811.03378.
Available at: `http://arxiv.org/abs/1811.03378`

Obrocka, M., Copley, C., Gqaza, T. and Grant, E. (2019). Prevalence of code mixing in semi-formal patient communication in low resource languages of South Africa. *Computing Research Repository*, vol. abs/1911.05636.
Available at: `https://arxiv.org/abs/1911.05636`

Oliphant, T.E. (2015). *Guide to NumPy*. 2nd edn. CreateSpace Independent Publishing Platform, USA. ISBN 151730007X, 9781517300074.
Available at: `http://web.mit.edu/dvp/Public/numpybook.pdf`

Pang, B. and Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In: *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pp. 271–278. Association for Computational Linguistics, Barcelona, Spain.
Available at: `https://www.aclweb.org/anthology/P04-1035`

Pang, B. and Lee, L. (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In: *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pp. 115–124. Association for Computational Linguistics, Ann Arbor, Michigan.
Available at: `https://www.aclweb.org/anthology/P05-1015`

Parker, D.B. (1985). Learning-logic. Tech. Rep. TR-47, Center for Computational Research in Economics and Management Sciences, MIT.

Pascanu, R., Mikolov, T. and Bengio, Y. (2012). On the difficulty of training recurrent neural networks. *Computing Research Repository*, vol. abs/1211.5063.
Available at: `http://arxiv.org/abs/1211.5063`

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830.
Available at: `https://arxiv.org/abs/1201.0490`

Pennington, J., Socher, R. and Manning, C. (2014). Glove: Global vectors for word representation. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543. Association for Computational Linguistics, Doha, Qatar.
Available at: `https://www.aclweb.org/anthology/D14-1162/`

Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K. and Zettlemoyer, L. (2018). Deep contextualized word representations. *Computing Research Repository*, vol. abs/1802.05365.
Available at: `http://arxiv.org/abs/1802.05365`

Radovanović, M., Nanopoulos, A. and Ivanović, M. (2010). Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, vol. 11, pp. 2487–2531. ISSN 1532-4435.
Available at: `http://dl.acm.org/citation.cfm?id=1756006.1953015`

Rajaraman, A. and Ullman, J.D. (2011). *Mining of Massive Datasets*. Cambridge University Press, New York, NY, USA. ISBN 1107015359, 9781107015357.

Rajpurkar, P., Zhang, J., Lopyrev, K. and Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. *Computing Research Repository*, vol. abs/1606.05250.
Available at: `http://arxiv.org/abs/1606.05250`

Řehůřek, R. and Sojka, P. (2010). Software framework for topic modelling with large corpora. In: *Proceedings of the Seventh International Conference on Language Resources and Evaluation Workshop on New Challenges for NLP Frameworks*, LREC'10, pp. 45–50. European Language Resources Association, Valletta, Malta.
Available at: `https://radimrehurek.com/lrec2010_final.pdf`

Rennie, J.D.M., Shih, L., Teevan, J. and Karger, D.R. (2003). Tackling the poor assumptions of naive Bayes text classifiers. In: *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, pp. 616–623. AAAI Press. ISBN 1-57735-189-4.
Available at: `http://dl.acm.org/citation.cfm?id=3041838.3041916`

Ripley, B.D. (1996). *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge. ISBN 0-521-46086-7.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, pp. 65–386.

Ruder, S. (2017). A survey of cross-lingual embedding models. *Computing Research Repository*, vol. abs/1706.04902.
Available at: `http://arxiv.org/abs/1706.04902`

Rumelhart, D.E., McClelland, J.L. and PDP Research Group, C. (eds.) (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*. MIT Press, Cambridge, MA, USA.

Samarati, P. and Sweeney, L. (1998). Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Tech. Rep., Computer Science Laboratory, SRI International.
Available at: `http://www.csl.sri.com/papers/sritr-98-04/`

Schroff, F., Kalenichenko, D. and Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. *Computing Research Repository*, vol. abs/1503.03832.
Available at: `http://arxiv.org/abs/1503.03832`

Schwenk, H. and Li, X. (2018). A corpus for multilingual document classification in eight languages. In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, LREC'18. European Language Resources Association, Miyazaki, Japan.
Available at: `https://www.aclweb.org/anthology/L18-1560`

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C.D., Ng, A. and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642. Association for Computational Linguistics, Seattle, Washington, USA.
Available at: `https://www.aclweb.org/anthology/D13-1170`

Taigman, Y., Yang, M., Ranzato, M. and Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In: *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR'14, pp. 1701–1708.
Available at: `https://ieeexplore.ieee.org/document/6909616`

Tjong Kim Sang, E.F. and De Meulder, F. (2003). Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In: *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pp. 142–147. Association for Computational Linguistics, Stroudsburg, PA, USA.
Available at: `https://doi.org/10.3115/1119176.1119195`

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L. and Polosukhin, I. (2017). Attention is all you need. *Computing Research Repository*, vol. abs/1706.03762.
Available at: `http://arxiv.org/abs/1706.03762`

Vulić, I. and Moens, M.-F. (2013*a*). Cross-lingual semantic similarity of words as the similarity of their semantic word responses. In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 106–116. Association for Computational Linguistics, Atlanta, Georgia.
Available at: `https://www.aclweb.org/anthology/N13-1011`

Vulić, I. and Moens, M.-F. (2013*b*). A study on bootstrapping bilingual vector spaces from non-parallel data (and nothing else). In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1613–1624. Association for Computational Linguistics, Seattle, Washington, USA.
Available at: `https://www.aclweb.org/anthology/D13-1168`

Vulić, I. and Moens, M.-F. (2014). Probabilistic models of cross-lingual semantic similarity in context based on latent cross-lingual concepts induced from comparable data. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 349–362. Association for Computational Linguistics, Doha, Qatar.
Available at: `https://www.aclweb.org/anthology/D14-1040`

Vulic, I. and Moens, M.-F. (2016). Bilingual distributed word representations from document-aligned comparable data. *Journal of Artificial Intelligence Research*, vol. 55, no. 1, pp. 953–994. ISSN 1076-9757.
Available at: `http://dl.acm.org/citation.cfm?id=3013558.3013583`

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O. and Bowman, S.R. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding. *Computing Research Repository*, vol. abs/1804.07461.
Available at: `http://arxiv.org/abs/1804.07461`

Wang, J., Shen, H.T., Song, J. and Ji, J. (2014). Hashing for similarity search: A survey. *Computing Research Repository*, vol. abs/1408.2927.
Available at: `http://arxiv.org/abs/1408.2927`

Wang, Z. and Ittycheriah, A. (2015). FAQ-based question answering via word alignment. *Computing Research Repository*, vol. abs/1507.02628.
Available at: `http://arxiv.org/abs/1507.02628`

Werbos, P. and John, P. (1974). *Beyond regression: new tools for prediction and analysis in the behavioral sciences.* Ph.D. thesis, Harvard University, Cambridge, MA, USA.
Available at: `https://www.researchgate.net/publication/279233597_Beyond_Regression_New_Tools_for_Prediction_and_Analysis_in_the_Behavioral_Science_Thesis_Ph_D_Appl_Math_Harvard_University`

Wieting, J., Bansal, M., Gimpel, K. and Livescu, K. (2015). Towards universal paraphrastic sentence embeddings. *Computing Research Repository*, vol. abs/1511.08198.
Available at: `https://arxiv.org/abs/1511.08198`

Williams, A., Nangia, N. and Bowman, S.R. (2017). A broad-coverage challenge corpus for sentence understanding through inference. *Computing Research Repository*, vol. abs/1704.05426.
Available at: `http://arxiv.org/abs/1704.05426`

Wolpert, D.H. and Macready, W.G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82.
Available at: `https://ti.arc.nasa.gov/m/profile/dhw/papers/78.pdf`

Zeiler, M.D. (2012). ADADELTA: an adaptive learning rate method. *Computing Research Repository*, vol. abs/1212.5701.
Available at: `http://arxiv.org/abs/1212.5701`

Zhu, Y., Kiros, R., Zemel, R.S., Salakhutdinov, R., Urtasun, R., Torralba, A. and Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. *Computing Research Repository*, vol. abs/1506.06724.
Available at: `http://arxiv.org/abs/1506.06724`

Zweigenbaum, P., Sharoff, S. and Rapp, R. (2018). Overview of the third BUCC shared task: Spotting parallel sentences in comparable corpora. In: Rapp, R., Zweigenbaum, P. and Sharoff, S. (eds.), *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, LREC'18. European Language Resources Association, Paris, France. ISBN 979-10-95546-07-8.
Available at: `http://lrec-conf.org/workshops/lrec2018/W8/pdf/12_W8.pdf`

Appendices

Appendix A

Data Protections and Regulations

In this appendix we provide a summary of the POPI Act of 2013 and HIPAA Safe Harbor Unique Identifiers for the interested reader.

A.1 POPI Act of 2013

According to Section 11 of the POPI Act, the personal information of individuals can only be processed by the Responsible Party in the following scenarios:

- the individual has given consent, or
- it is specified within a contract to which the individual is a party, or
- it is required by law, or
- it protects a vital interest of the individual or of another individual, or
- it is in the public interest, or
- it is necessary for legitimate purposes pursued by you or a third party which have been given access to the data.

Individuals have the right to withdraw consent to the processing of their data, or object to the processing of their if they can show compelling and legitimate grounds for their objection. Section 13 of the POPI Act states that a "data subject's" personal information can only be collected for an explicitly-defined legal purpose and the subject must be aware of the motivation for which the information is being collected. The Responsible Party has to collect data directly from the subject unless:

- this information already exists in the public domain or has been deliberately made public by the individual, or
- using another source for data collection does not prejudice the individual, or

- the collection of the data is necessary for some public task; or to protect your own interests, or
- it would impair a lawful purpose, or
- it would not be reasonably possible to collect the data directly from the individual.

The requirements above, and others outlined in the POPI Act do not need to be met if the individual has consented to non-compliance or if, by non-compliance, the subject's rights as outlined by the POPI Act would not be prejudiced, or if compliance would result in prejudice to some public interest, or if the collected data will only be used for scientific, historical or statistical research purposes, or if the subject will not be identifiable.

Further, once the data is no longer needed for the specific, lawful task, the data must be destroyed in a way that prevents reconstruction and/or identification unless the Responsible Party is required by law to keep it, or keeping the data is in accordance with the contract between the Responsible Party and the subject, or the subject has consented to them keeping the records. The Responsible Party is authorized to retain the data for historical, statistical or research purposes if the necessary safeguards to prevent the records being used for any other purposes have been established.

A.2 HIPAA Safe Harbor Unique Identifiers

The following identifiers are considered unique identifiers and must be removed from healthcare records in accordance with the HIPAA Act of 2012 (HHS.gov, 2012):

1. a person's first name and family name;
2. geographical subdistricts smaller than a state, including street addresses, all cities, counties, precincts, ZIP codes, and their corresponding geocodes, except for the first three digits of the ZIP code if,
    a) according to the most recent census data, the combination of all the ZIP codes with the same first three digits contains a population of more than 20,000 people, and
    b) the first three digits of a ZIP code is changed to 000 for all such geographic subdivisions containing a population of 20,000 or less people;
3. all components of dates stored in the data that relate to the healthcare patient (except the year), including the birth date, admission date, discharge date, death date, unless the year indicated a person is over the

age of 89, in which case the data must be aggregated to a single category of age 90 or older;

4. telephone numbers;
5. vehicle identifiers and serial numbers, including license plate numbers;
6. fax numbers;
7. device identifiers and serial numbers;
8. email addresses;
9. web universal resource locators (URLs);
10. social security numbers;
11. internet protocol (IP) addresses;
12. medical record numbers;
13. biometric identifiers, including finger and voice prints;
14. health insurance plan beneficiary numbers;
15. photographs that clearly identifies the person;
16. account numbers;
17. any other unique identifier, except as part of implementation specifications necessary for re-identification under the condition that the identifier was not derived from user data and is securely stored;
18. certificate/license numbers.

Appendix B

MomConnect Data Dictionary

Field Name	Description	Example	Identifying
helpdesk_anonymous_id	Anonymous ID	0001113782e33800cc10be949e32	No
helpdesk_communication_channel	Mode of message (WhatsApp / SMS)	WhatsApp	Yes
helpdesk_question	Text of incoming question	“The meaning of burmp plx”	Possibly
helpdesk_reply	Text of helpdesk response	“please google BURP”	No
helpdesk_days_to_edd_question	Number of days to EDD when question was received	21.5845549	No
helpdesk_days_to_edd_reply	Number of days to EDD when reply was sent	21.5373561	No
helpdesk_response_time_days	Number of days to respond to query	0.04719879	No
seed_year_of_birth	Year of birth rounded to nearest 5	1995	Yes
helpdesk_time_between_questions_days	Number of days since the previous question sent by a participant (NA if it is the first question sent)	0.01010482	No
helpdesk_label_name	Label assigned by the helpdesk	Question	No
seed_identity_language	Language of registration of the participant	eng_ZA	Yes
dhis2_province	Name of province where registered	Eastern Cape Province	Yes
dhis2_district	Name of district where registered	Oliver Tambo District Muni	Yes
dhis2_subdistict_code	De-identified code of subdistrict of registration; can be used for aggregation but is non-identifying	606b47322ace1458a340aa7a5d68	No
dhis2_ruralurban	Rural/Urban classifier of the clinic	Urban	No
dhis2_facility_code	De-identified code of clinic of registration; can be used for aggregation but is non-identifying	a10affeb2a1b70a1e6a03de5507c	No
data_extracted_week	Week when the data were extracted from the database	2018-09-03	No
days_to_edd_registration_create_at_10	Number of days until EDD when the participant registered rounded to 10 days	150	Yes

Appendix C

MomConnect Data Analysis

C.1 Regional and Language Distribution

The following data represents the unique user sign-up statistics per language and province in South Africa. A small number of data points have 'Redacted' as the sign-up language, as a result of the anonymization protocols described in Section 4.3.

	EC	**FS**	**GP**	**KZN**	**LP**	**MP**	**NC**	**NW**	**WC**
Afrikaans	627	113	183	29	23	54	275	71	1561
English	3643	3026	18245	6400	5485	4026	578	5062	4737
Ndebele	3	0	59	1	9	22	0	3	0
N. Sotho	24	587	852	65	1375	175	16	276	28
Redacted	2	1	11	7	2	2	0	0	1
S. Sotho	18	429	192	9	44	18	1	28	8
Siswati	0	0	27	5	4	247	0	2	2
Tsonga	1	15	160	4	32	14	88	631	3
Tswana	1	3	450	1	735	141	2	25	2
Venda	0	1	236	1	842	9	0	12	3
Xhosa	3634	72	644	560	50	81	9	164	1603
Zulu	157	92	2406	13199	89	1499	7	59	84
Total	**7762**	**4343**	**23446**	**20288**	**8691**	**6288**	**974**	**6350**	**8038**

Table C.1: Unique users in dataset per sign-up language and province

In South Africa, certain languages are more frequently spoken in some provinces. However, users seem to use English as their sign-up language regardless of the population tendencies of regions, with the exceptions being the Eastern Cape (EC) with Xhosa and English almost on par, and KwaZulu-Natal with Zulu.

C.2 Unique Answers in Reduced Dataset

Unique Answer	**Count**
Thank you for your message but we are unable to help you with this query. Please go to your clinic if you need medical assistance.	35281
Thank you for your request. You will now start receiving messages about your baby.	20004
Thank you for your feedback on MomConnect. We appreciate it. From the National Department of Health.	9786
You may start to feel your baby moving in months 4 and 5. By month 9 your baby is running out of space and will move less. If you don't feel your baby moving for a few hours see if you can wake him up by drinking something sweet, or by listening to some music. If you are worried visit your clinic.	8435
Generally, a full-term pregnancy is considered to be between 38 weeks and 41 weeks. A premature baby is delivered before 37 weeks of your pregnancy. If you are 41 weeks pregnant, go to the hospital. Staff there can give you medicine to help your labour to start. Don't wait for the 42nd week.	6535
Go to the clinic straight away if you have: pain in your stomach, swelling of your legs or feet that does not go down overnight, a fever, pain when you are urinating, if your baby stops moving after five months, a headache and you can't see properly (blurred vision), vomiting and a sudden swelling of your face, hands or feet, fluid leaking from your vagina.	4777
Thank you for your message but we are unable to help you. Please go to your clinic if you or your baby is sick.	4649
Signs of labour include a jelly-like discharge, your waters breaking, and regular and painful labour contractions. Make sure you can get to a hospital.	4317
Pregnancy is a natural time for your body. Your body is going through a lot of changes. Most pregnancy symptoms are quite normal. Some of the most common complaints during pregnancy include: Morning sickness (nausea), feeling tired, pelvic ache and back ache, constipation, swelling legs and feet, difficulty sleeping in late pregnancy. You may also feel angry, emotional or tearful. Check at the clinic if you are worried.	3600
Do you feel like vomiting? Morning sickness or nausea is common in the first 3 months. It should ease from the 4th to 7th months. Avoid food with too much fat and spices. Eat dry bread or a dry biscuit when you wake up. If you cannot eat, drink lots of water or tea but not alcohol. Adding some ginger, mint or lemon to your tea may help to ease the nausea.	2712
Thank you for your feedback on your facility. We appreciate it and we will send your compliment to the clinic. From the National Department of Health.	2403
Vaginal discharge, including light spotting or light bleeding, can be common in pregnancy. But if you are experiencing heavy bleeding, cramps or pain, then go straight to the clinic to have them checked.	2359
You need to visit the clinic at least 4 times during your pregnancy. Make your first appointment as soon as you have missed a period or as soon you think you may be pregnant. The nurses will check your health, and your baby's. If they pick up any problems they can treat you and your baby fast. It will mean fewer problems later on if you get the right care straight away.	2241
Fever, coughing and breathlessness are signs of illness. Always go to the clinic if you have these signs so you can get the right medicines for you and your baby.	2190
Your baby will be ready for solid foods at 6 months. Try offering her 1 or 2 spoonfuls of thick, mashed vegetables or fruit. As she gets used to them, start to add other foods like porridge and other mashed foods like beans, lentils and meat. Slowly increase the number of times she has solid food in a day. When your baby is 7 months old, she should be eating solids 3 times a day. Continue breastfeeding your baby, in addition to giving her solid food	1872

Your baby needs ONLY breast milk for the first 6 months. Don't give him water, porridge, formula milk, tea or anything else. Your body can make plenty of milk for your baby. It protects him from disease. If you are HIV-positive make sure you are taking your ARVs while you are breastfeeding. This will help to stop him becoming HIV-positive when he drinks your breast milk.	1709
Thank you for your message, but we are unable to help you with this query	1674
You will now receive baby messages.	1626
Please go to the clinic for re-registration in your preferred language.	1567
Hormone progesterone relaxes smooth muscles thus the valve that separate the allowing gastric acid to go up causing heartburn. Avoid using too much spicy, acidic, fizzy, lots of sugar, rich or fried or fatty food. Eat small meals slowly, wear loose fitting clothes, and do not sleep immediately after eating. Please see a doctor to prescribe a suitable heartburn medicine.	1524
The Department of Health recommends exclusive breastfeeding for all mothers, whether they are HIV-positive or not, for the first 6 months of a baby's life. Exclusive breastfeeding means your baby needs only breast milk for the first 6 months. It does not need any other fluids or foods. If you choose to formula feed, you must stick to formula feeding exclusively. Switching between breast milk and formula milk is dangerous for your baby.	1303
Your clinic will give you iron and folic acid tablets to take daily. Folic acid is important in early pregnancy. It helps your baby's spine to develop correctly. Take the pills as soon as you know you are pregnant - ideally before. You can also get folic acid from eating dark green vegetables (like spinach), and also eggs and oranges. You may also be given calcium pills. They help to keep your blood pressure stable. High blood pressure is not good for your baby, or you.	1275
After day 4 of life, baby is supposed to have about 2-5 yellow soft stools a day. Newborn babies often poo after every feeding about 6 times per day, in the first few weeks after delivery the babies' intestine are maturing and becoming more efficient at extracting nutrition from breast milk, the time between bowel movements gets longer. Please take your baby to the clinic if constipated	1264
Constipation is a common problem during pregnancy. Make sure you drink lots of water and eat lots of fruit and vegetables. Moderate exercise, such as walking, can also help to ease constipation	1310
Pre-eclampsia is a condition that can occur usually after 20 weeks of pregnancy. Some signs are a sudden or severe swelling in your face, hands or feet, or you may get a bad headache. Your vision may be blurred or you might see flashing lights. Another sign is a pain just below your ribs on your right side. Severe epigastric pain. You need to go straight to the clinic. If you have pre-eclampsia you might need to go to hospital.	1204
It is common to have swollen feet and ankles, particularly during the last months of pregnancy. Your hands may swell too. Your body is holding lots of extra water. Lie down and raise your legs when possible. The swelling should go down overnight. If it does not, tell your clinic and get it checked. It could mean you have high blood pressure. High blood pressure is not good for your baby either.	1218
Breastfeeding is still the best way to feed your baby, even if you are HIV-positive. The ARVs that you are on and the medication that you get from the clinic for your baby will help to stop your baby from getting the HIV through your breast milk. It is very important to stick to the medication that the clinic gives you and your baby, especially while breastfeeding.	1214
For the last 2 months of your pregnancy your womb will have gentle contractions that you may feel as a tightening of your stomach. These are called Braxton Hicks or practice contractions. They are nothing to be worried about. Your body is getting ready to give birth and your womb is keeping itself toned.	1037
Thank you for sending in your complaint. We have taken note of it and will log the complaint with the Department of Health and your facility.	1022

Your baby has the best chance of being born negative if you, as an HIV-positive pregnant woman, start your ARVs as soon as you find out you are pregnant. This is the key to PMTCT (Prevention of Mother-to-Child Transmission). It means you are on ARVs to stop your baby from becoming HIV-positive and to look after your health as well.	988
If you feel a burning sensation when you urinate (pee), or your urine is cloudy, bloody or smelly, you may have a urinary tract infection. Drinking lots of clean water can help to prevent bladder infections. If you do have a bladder infection, your clinic will give you medicine to clear it up.	954
We are undergoing important maintenance so your request to move to baby messages may be delayed by a few days. Thank-you for your patience and understanding.	1018
Drink lots of fluids; eat oatmeal, Green leafy vegetables spinach, broccoli, oranges, tomatoes, apples, banana, eggs, sesame seeds, nuts, carrots, ginger, beans, fruits and vegetables; you can also add garlic to your food. Avoid using pacifiers and bottles when breastfeeding, switch your breasts and make sure the baby is nursing effectively. The more you breastfeed the more milk you will produce.	942
Your clinic will advise you about the safest delivery option for you. It may be recommended that you have a Caesarean-section, if you have: 1) had previous C-sections, 2) a baby that is in breech, 3) or if you have large vaginal warts. If you are taking your ARVs, the best way to deliver is vaginally.	903
You may not feel hungry. Also, you may not like the food you used to enjoy or some food may smell bad to you. But your baby needs good food to grow. Try to eat healthy foods that you like. Or eat a few small meals a day. If you feel nauseous, try drinking some ginger, mint or lemon tea, and rest if you can. Dry biscuits may help with the nausea when you wake up in the morning.	933
It's normal to feel exhausted or very tired during pregnancy. Growing a baby is hard work, even though your baby is tiny. Get as much rest as you can. If that does not help, ask for iron supplements at the clinic, or buy some at your pharmacy if you can. Ask your family to help with household tasks such as shopping, cooking and cleaning	856
Please note that the messages are generic and are meant to empower all pregnant women about HIV irrespective of their HIV status, only an HIV test done at the Doctor or clinic will determine your status.	847
Please feel free to ask. MomConnect is a Health Department programme gives information to pregnant women according to their gestational age, and after birth. You can send in any question related to pregnancy or your baby after birth. Send in complaints and compliments about service received at any of our health facilities.	798
Your baby needs a good mouthful of your breast to feed well. Check that he has most of the dark area around your nipple in his mouth. If you see his jaw moving up and down as he feeds, you know he has latched on. Getting your baby to latch on well will prevent your nipples from getting sore.	736
Labour is a series of contractions of the largest muscle in the body, your uterus. It can feel like period pains at first, or cramp in your back. The contractions may last 30-40 seconds at first and come every 15-20 minutes. In strong labour they may be coming every 5- minutes and last 50-60 seconds.	706
It is safe to have sex whenever you feel like unless your waters have broken, have vaginal bleeding, you were told that you have placenta previa. You are not carrying twins. You are not having contractions. Never had any miscarriages. You and your partnerdo not have active STI's. Please consult with your clinic if you have high blood pressure before having sex.	726
Please take the baby to the clinic for help	667
We are sorry for your loss. You will no longer receive messages from MomConnect. Thank you	671

You must try to get to the hospital as soon as your labour starts. You have to be ready for it. Make sure you have money saved for transport and that there is airtime in your phone. Your hospital bag must have sanitary pads and the clothes in it that you will wear when you go home. Make sure you have clothes and a blanket for your baby in the bag too.	660
The food you like while you are pregnant may not be what you usually eat. Avoid using too much sugar or fried or fatty food. Do not drink any alcohol. Avoid raw eggs. Try not to take any herbal or traditional medicines before first checking with the clinic to see if they are safe for you and your baby.	621
Some bleeding after birth is normal but if the bleeding becomes heavy or clotted (lumpy), go to the clinic for treatment. The bleeding will stop in a few weeks. It is your body cleaning out the lining of your womb. Breastfeeding your baby will help to shrink your womb and reduce bleeding. Make sure you have plenty of pads for the first few days when bleeding can be heavy.	628
Please go to your clinic/doctor and get suitable medication or doctor's prescription which cannot put the baby at risk. Please avoid over the counter medicines during pregnancy they might be harmful to your baby. Some remedies including traditional remedies may not be safe for you or your baby. Before taking any such medicine, please check first with your health worker or the clinic.	622
Keep your baby's cord dry and open to the air to prevent infection. Clean it with surgical spirits. If it becomes red or smelly, please go to the clinic.	594
When you are 30 weeks pregnant, your baby should be in a cephalic (head down) position. Even so, it is normal for a baby to turn their head at 34 weeks. It should be of concern only if your baby does not turn their head after the 36 week mark.	564
If your baby has a high temperature, or if he vomits, take him to the clinic. He will get the right treatment at the clinic. Only give medicine that your health worker recommends. After he recovers, give him and extra feed or meal every day for a week.	565
Perfume free vaseline, baby soap/ head to toe shampoo, baby oil, baby wipes, face cloth, sanitary pads, nappies, bum cream, cotton wool and surgical spirit. Your change of clothing, transport money, your ID book. Baby grows for the baby ,vests, hats and socks. Blanket and receiving blanket.	548
Eat your regular food you eat at home, include fruits and veggies in your diet. Drink at least 8 glasses of water per day. Avoid coffee, fizzy drinks, energy drinks alcohol and soil.	517
Please go to clinic because a headache can be a symptom of something serious such as hypertension leading to complication of pre-eclampsia. During the first trimester, your body experiences a surge of hormones and an increase in blood volume. These two changes can cause more frequent headaches. These headaches may be further aggravated by stress, poor posture, dehydration or changes in your vision.	521
STIs are sexually transmitted infections that you can get by having unprotected sex with an infected person, during pregnancy. STIs can make you and your baby very ill. Some STIs can pass to your baby in pregnancy or at any time. Others can be transmitted during birth. If you suspect that you may have an STI you need to tell the clinic immediately, and get treated. Condoms prevent STIs and HIV transmission, so you need to use protection, especially in pregnancy.	499
Traditional remedies may not be safe for you or your baby. Before taking any traditional medicine, please check first with your health worker or the clinic.	464
Yes you can get pregnant while breastfeeding. Getting pregnant too soon can be dangerous for you, and your next baby could be born weak and early. It's best to not have sex for at least 6 weeks after birth or until you have healed. When you start having sex, use a family planning method. This will prevent you from getting pregnant too early. There are plenty of options to choose from. Condoms will prevent pregnancy and sexually transmitted infections and diseases.	492

Put baby flat on the back and cycle baby's legs, flex baby's legs outwardly, and always burp baby after each feed. Ensure that you avoid gas forming foods such as beans, onions, cabbage.	476
No matter how many weeks pregnant you are when your waters break, you should go straight to the clinic or hospital. Make sure you have enough time to get to the hospital also when you are ready to give birth. If you are HIV-positive there are medicines to help keep you and your baby well. In a normal labour, go to the hospital as soon as labour begins so you can get the care you need.	494
It could be that baby has a cold. Make sure your baby gets plenty of rest, encourage your baby to take extra breast feed, dab a little petroleum jelly onto the outside of your baby's nostrils to reduce any irritation. Breathing in steam may help to loosen your baby's blocked airways and relieve his cough. Try sitting in a steamy bathroom for a few minutes, with the shower on, while holding your baby, that will help to loosen your baby's blocked airways and relieve his cough. Do not give over the counter medication rather discuss with health care provider about your concerns	487
High blood pressure is dangerous especially during pregnancy. The clinic will check your blood pressure each time you go there. High blood pressure can mean that you are getting pre-eclampsia. This can be harmful for you and your baby. It can be controlled but you will need to be checked often. The clinic may give you medicine to help control it.	467
You can express and store milk in the container for 24 hours in a refrigerator or in cool place for 8 hours. Use a glass or hard plastic container with opening and tight lid to store the breast milk. You must warm the milk with boiled water in a bowl. Boil the container for 10 minutes before use, write the time and date of expressed milk on the container before storing. Exclusive breastfeed infant take an average 750 ml between 1-6 months.	480
Premature labour happens before the 37th week of your pregnancy. Generally, a full-term pregnancy is considered to be between 38 weeks and 41 weeks. If you are showing signs of a labour and are less than 9 months pregnant, go to the clinic straightaway. You may be in labour.	402
Caffeine acts as a diuretic and is linked to low birth weight, higher risk of still birth, fetal death or spontaneous abortions	442
Your baby should move at least 20- 50 times a day depending on the stage of your pregnancy, the bigger the baby the less movements you will feel as space gets smaller. Avoid eating too much sugar as it stimulates baby to be active more.	403
Diet Changes ,sudden shift in your food intake can upset your stomach and potentially cause diarrhea and sometimes vitamins may upset your stomach so consult with the clinic let them check you but please drink fluids if having diarrhoea. Wash your hands with soap after using toilet and before handling food	417
There is no fixed term when the baby moves down towards the pelvis, but it happens towards the end of the pregnancy. Baby may drop 2-4 weeks before labor time. Signs include: Frequent urination (pressure to the bladder), pressure to the pelvic area (difficulty in walking) and your bump will be lower. You can also discuss with your health care provider	402
please go to the clinic for assessment	383
Causes of cramping during pregnancy. Cramping typically occurs when the uterus expands, causing the ligaments and muscles that support it to stretch. It may be more noticeable when you sneeze, cough, or change positions. During the second trimester, a common cause of cramping is round ligament pain.	372
Backpain is normal process due to hormone relaxin that make the ligaments to stretch accommodating and prepare for delivery, the growing baby. You can wear comfortable shoes, don't stand for prolonged period in one place, swim, sleep with support of a pillow on the back, put hot or cold compression on your back, let your partner gently massage your back	362

A white non offensive vaginal discharge, including light spotting or light bleeding, can be common in pregnancy. But if you are experiencing heavy bleeding, cramps or pain, offensive, very thick greenish or yellow discharge and itchy vagina then go straight to the clinic to have them checked.	339
Pregnancy bumps come in all shapes and sizes there's no perfect size .Size is no indication of your baby's weight either. With first pregnancy the muscles are tight thus the bump is neat, with subsequent pregnancy your bump may be more spread out or bigger because the muscles aren't holding in the baby so well. Bump size depends on how many babies are in there, how much fluid you've got inside and the way your baby is lying and your stature and posture. Go to clinic.	329
Haemorrhoids or piles are swollen veins in your bottom (anus). They can stick out, be itchy, or even bleed. You may be able to feel them as small, soft lumps inside or around the edge or ring of your bottom or notice blood after you pass a stool. Piles are common during pregnancy. Eating lots of fruit and vegetables, and drinking lots of water can help to relieve the symptoms of piles. If the pain or bleeding continues, have them checked at the clinic.	317
Make starchy foods the basis of most meals: (maize meal, bread flour and bread made from bread flour), eat plenty of vegetables and fruits every day, dry beans, lentils, split peas and soya regularly, chicken, fish, milk, meat or eggs daily. Use fats, sugar and salt sparingly. Drink lots of clean, safe water	307
A well-fed baby should wet 6-8 nappies in 24 hours and feed during the day and night. Breastfeed as often as your baby wants food, or at least every 3-4 hours in the first month. After a feed, his tummy should look full and your breasts should feel softer.	299
It's nothing to worry about, babies teeth differently some may start as early as 4 months while others may only start after 12 months, please be patient it will happen eventually	307
MomConeect is a Health Department programme. It sends SMS messages for you & your baby. To see, change or delete your personal info, dial 1345507#	324
Sorry we have no way of knowing from our side if it's a boy or girl. We send generic messages using he/she. Please go for sonar from 20 weeks to check baby's gender.	310
A woman's period will typically return about 6 to 8 weeks after giving birth, if she is not breastfeeding, if she does breastfeed, the timing for a period to return can vary. Some women might not have a period the entire time they breastfeed, but for others it might return after a couple of months, whether they are breastfeeding or not.	287
For support register at 13455010# then access PMTCT helpdesk for support by calling, SMS and WhatsApp at 0810027727	277
Occasional abdominal discomfort is a common and often harmless complaint during pregnancy, but it can also be a sign of a serious problem. Never ignore severe or persistent abdominal pain. Call your healthcare provider if your pain doesn't go away after several minutes of rest, or if you also have any of the following:spotting or bleeding, unusual vaginal discharge, chills or fever, faintness, discomfort, while urinating, nausea and vomiting check if you are not bloated or constipated aslo. If concerned go to the clinic	297
Normally the baby should move at least 20- 50 times every day depending on the stage of your pregnancy, the more the baby grows; the lesser movements you will feel as space gets smaller. If you don't feel your baby moving for a few hours see if you can wake him up by drinking something sweet, or by listening to some music. If this does not help and worries you, please go to the clinic.	280
Low blood, occurs when the red blood cells that carry oxygen around are few or HB is low. The symptoms will be pallor, dizziness and tiredness. In pregnancy it's commonly caused by iron deficiency thus you have to take your supplements and increase foodstuff such as green leafy vegetables, beetroot, liver, nuts	285

Newborn babies sleep a lot. You can expect him to sleep for up to 18 hours a day for the first few weeks. But he won't sleep for more than 3-4 hours at a time. Put your baby to sleep on his back on a firm surface. Don't use pillows. They could suffocate him. Try to get some rest whenever your baby sleeps.	267
Please go for sonar from 20 weeks to check baby's gender.	260
Make starchy foods the basis of most meals: (maize meal, bread flour and bread made from bread flour),eat plenty of vegetables and fruits every day,dry beans, lentils, split peas and soya regularly, chicken, fish, milk, meat or eggs daily.Use fats sparingly, sugar and salt sparingly.Drink lots of clean, safe water,Drink at least 8 glasses of water per day and you can drink fruit juices but in moderation and rooibos tea. Avoid coffee, fizzy drinks, energy drinks alcohol and soil.	265
No. Your baby needs only breast milk and nothing else, not even water, until he is 6 months old. Breast milk really is the best and only food your baby needs in the first six months. It protects him from disease.	284
Please avoid over the counter medicines during pregnancy they might be harmful to your baby, do not self-medicate always go to the clinic or Doctor they know better.	301
Please reduce/limit the intake of too much caffeine. Avoid energy drinks. Too much caffeine is not good for you and the baby too. Caffeine is a stimulant so it increases your blood pressure and heart rate. High blood pressure during pregnancy need to be avoided. Caffeine and fizzy drinks acts as a diuretic and is linked to low birth weight, higher risk of still birth, fetal death or spontaneous abortions.	261
Normal vaginal discharge during pregnancy is called leukorrhea and is thin, white, milky, and mild smelling.It is caused by increased blood flow to the vagina due to increased oestrogen, You can put cotton panties, panty liner. If it changes in odour, colour and becomes itchy inform your health care provider	269

Table C.2: Frequency distribution of answers