TECHNISCHE HOCHSCHULE NÜRNBERG
GEORG SIMON OHM

Faculty of Computer Science

IT-Based textgeneration using NLP methods

State of the art and design of a prototype

Bachelor Thesis in
Business Informatics and Management

from

Tim Löhr

Student ID 3060802

First advisor: Prof. Dr. Alfred Holl

Second advisor: Prof. Dr. Florian Gallwitz

Hinweis: Diese Erklärung ist in alle Exemplare der Abschlussarbeit fest einzubinden. (Keine Spiralbindung)

TECHNISCHE HOCHSCHULE NÜRNBERG
GEORG SIMON OHM

Prüfungsrechtliche Erklärung der/des Studierenden

Angaben des bzw. der Studierenden:

Name: Vorname: Matrikel-Nr.:

Fakultät: Studiengang:

Semester:

Titel der Abschlussarbeit:

Ich versichere, dass ich die Arbeit selbständig verfasst, nicht anderweitig für Prüfungszwecke vorgelegt, alle benutzten Quellen und Hilfsmittel angegeben sowie wörtliche und sinngemäße Zitate als solche gekennzeichnet habe.

Ort, Datum, Unterschrift Studierende/Studierender

Erklärung zur Veröffentlichung der vorstehend bezeichneten Abschlussarbeit

Die Entscheidung über die vollständige oder auszugsweise Veröffentlichung der Abschlussarbeit liegt grundsätzlich erst einmal allein in der Zuständigkeit der/des studentischen Verfasserin/Verfassers. Nach dem Urheberrechtsgesetz (UrhG) erwirbt die Verfasserin/der Verfasser einer Abschlussarbeit mit Anfertigung ihrer/seiner Arbeit das alleinige Urheberrecht und grundsätzlich auch die hieraus resultierenden Nutzungsrechte wie z.B. Erstveröffentlichung (§ 12 UrhG), Verbreitung (§ 17 UrhG), Vervielfältigung (§ 16 UrhG), Online-Nutzung usw., also alle Rechte, die die nicht-kommerzielle oder kommerzielle Verwertung betreffen.

Die Hochschule und deren Beschäftigte werden Abschlussarbeiten oder Teile davon nicht ohne Zustimmung der/des studentischen Verfasserin/Verfassers veröffentlichen, insbesondere nicht öffentlich zugänglich in die Bibliothek der Hochschule einstellen.

Hiermit ☐ genehmige ich, wenn und soweit keine entgegenstehenden Vereinbarungen mit Dritten getroffen worden sind,

☐ genehmige ich nicht,

dass die oben genannte Abschlussarbeit durch die Technische Hochschule Nürnberg Georg Simon Ohm, ggf. nach Ablauf einer mittels eines auf der Abschlussarbeit aufgebrachten Sperrvermerks kenntlich gemachten Sperrfrist

von Jahren (0 - 5 Jahren ab Datum der Abgabe der Arbeit),

der Öffentlichkeit zugänglich gemacht wird. Im Falle der Genehmigung erfolgt diese unwiderruflich; hierzu wird der Abschlussarbeit ein Exemplar im digitalisierten PDF-Format auf einem Datenträger beigefügt. Bestimmungen der jeweils geltenden Studien- und Prüfungsordnung über Art und Umfang der im Rahmen der Arbeit abzugebenden Exemplare und Materialien werden hierdurch nicht berührt.

Ort, Datum, Unterschrift Studierende/Studierender

Preface I

The following thesis was created during my 7th and last semester at the Georg Simon Ohm University of Applied Science. Within my last three semesters I realized, that my major interest among all IT related topics is artificial intelligence.

My interest started basically with my IT project, in which my team and I programmed an autonomously driving remote control car with a deep neural network together with a Raspberry Pi 3B+. From this first project on, I selected all my further elective courses to be related with machine learning or data science in any possible way. I wanted to further increase my knowledge, so I searched for a website which provides courses related to AI. I found *www.udacity.com*, which offers courses in cooperation with top IT companys, such as Google, Airbnb or Microsoft. Out of curiosity, I bought the course *Natural Language Processing*. After successfully finishing it, I was encouraged to write my bachelor thesis in a *Natural Language Processing* related topic. Together with my professor *Prof. Dr. Alfred Holl* I worked out a method-matrix for the entire structure of this paper. Without his cooperative support overseas while I was studying abroad at the City University of Hong Kong, this thesis would have not been possible for me. Even though Natural Language Processing is just a subfield of machine learning, the current state-of-the-art research is far beyond what I can research within a bachelor thesis. In this way, I decided to write my thesis about the subfield *textgeneration* within NLP. My state-of-the-art research includes all *hot topics* within NLP and my prototyp focuses only on the textgeneration part, to dive deeper into what NLP and especially textgeneration is able to accomplish in the year 2020.

Preface II

For my research I encountered a lot of old and recently published papers, mostly from *https://arxiv.org/*. To read through the papers requires a lot of prior knowledge, especially in mathematics, which I learned in my abroad semester in Hong Kong. To fully understand the mathematics given in this thesis, an enhanced knowledge of calculus and linear algebra is required. Even if this is not the case, I will describe the process in such a way that it can be comprehended without looking on the maths.
Machine Learning and more specifically NLP is not an intuitive study. I provided for the matrix notations the common terminologys origined from top researchers and tried to make the entry into this field as smooth as possible, if the reader has no prior knowledge about this topic. During the five month development process of the bachelor thesis, I gained a lot of knowledge. I recognized that NLP is a huge topic, constantly under research. To keep up to date with the latest publications requires a lot of effort.
To give a full state-of-the-art review about *all* NLP related disciplines is not possible within this thesis. For this reason, I focus entirely on the development of the *Neural Text Generation* (NTP), which includes more fields than the reader might imagine. Thank you very much for reading.

	Titel / Kapitel		Untertitel / Unterkapitel	——Woher?—— Wissensinput	Frageinput	——Wie?—— Methode	——Was?—— Zielbeschreibung
0	**IT-basierte Textgenerierung mit Hilfe von NLP-Methoden** **State of the Art & Entwurf eines Prototypen**			Allgemeingültig: • Fachbücher, Bücher • HongKong, TH-OHM • Online Kurse	1. Was ist der State of Art von NLP - Systemen. 2. In welcher Qualität kann ich den Textgenerierungs- Prototypen selbst programmieren und welche Güte hat dieser?	1. Darstellung des State of the Art der NLP-Systeme. 2. Studium der relevanten Aspekte des NLP und Programmierung eines IT-basierten Textgenerierungs-Prototypen.	1. State of the Art fachlich herausarbeiten. 2. Einen Prototypischen Algorithmus programmieren, der zu einem gegeben Input z.B. ein Buch immer wieder neue kreative Fortsetzungen generiert.
1	**Einleitung**	**1.1**	**Fallbeispiel eines aktuellen NLP-Systems**	• [0.1] • Wissenschaftliches Schreiben und	1. Was sind aktuelle, nützliche Einsatzgebiete von NLP-Textverarbeitungs-Systemen? 2. Was ist der Nutzen meines NLP-Prototypen im Bereich der Textverarbeitung?	1. Recherche über die aktuellen und geplanten NLP-Systeme, im Bereich der Textverarbeitung. 2. Vorstellung meines Beitrags zu NLP-Systemen mithilfe meines Prototyps.	1. Antwort auf die Frage, warum meine Bachelorarbeit sinnvoll ist und welche Motivation ich habe zur Bearbeitung 2. Erläuterung durch einen interessanten leichten Einstieg.
2	**State of the Art**	**2.1**	**Relevante Aspekte der Mathematik**	[1.1]	Welches mathematische „know-how“ ist notwendig, um NLP-Systeme für Textverarbeitung und meinen Prototypen technisch verstehen zu können?	Recherche nach den relevanten Aspekten der Mathematik für dieses Thema.	Beschreibung der anwendungsbezogenen mathematischen Modelle für diesen Themenkomplex anhand von Formeln und Erklärungen.
		2.2	**Geschichte des NLP**	• [0] • [0.1]	1. Seit wann wird an NLP-Systemen geforscht? 2. Ab welchem Punkt konnte man effektiven Nutzen aus diesen Systemen ziehen?	1. Literaturrecherche über die Geschichte des NLP (40 Jahre). 2. Literaturrecherche über die ersten Einsätze der NLP-Systeme.	1. Darstellung der Geschichte des NLP in Form einer zeitlichen Abfolge. 2. Nutzen der ersten NLP-Prototypen oder Technologien die im Einsatz waren.
		2.3	**Aktuelle Trends der Technologie**	• [0] • [0.1] • [2.2] • Fallbeispiele	1. Was sind aktuelle NLP-Systeme imstande zu leisten? 2. Wo sind die Einsatzgebiete?	1. Literaturrecherche über aktuelle Trends (+ - 5 Jahre). 2. Recherche von aktuelle Papern und Veröffentlichungen.	1. Darstellung der aktuellen Technologien. 2. Blick in die kurzfristige Zukunft anhand von aktuellen Fallbeispielen und Forschungsergebnissen.
3	**Prototyp**	**3.1**	**Zielsetzung / Anforderungen**	• [0] • [1] • [2]	1. Was soll mein Prototyp mit gegebenen Mitteln leisten können? 2. Welcher Output ist im besten Fall zu erwarten?	1. Requirements Engineering. 2. Klassifizierung und Analyse möglicher Ergebnisse, z.B. ob der Output grammatikalisch korrekt ist.	1. Erläuterung des Umfangs meines Prototyps. 2. Sammlung und Klassifizierung der Anforderungen an den Algorithmus und dessen Output.
		3.2	**Fachkonzept**	[3]	1. Wie ist mein Prototyp strukturiert? 2. Welche Algorithmen verwende ich? 3. Welche Prozesse durchlaufen die zu verarbeitenden Daten? 4. Wie werden die Daten verarbeitet?	1. Erstellen eines Fachkonzepts 2. Algorithmus modellieren 3. Prozessmodellierung 4. Datenflussmodellierung und, oder Datenmodellierung	1. Fachkonzept fertig erstellt. 2. Der Prototyp wird ohne IT Bezug anhand von verschiedenen Teilmodellen modelliert. 3. Die einzelnen Prozesse werden ohne konkreten Implememtierungs-Vorschlag modelliert. 4. Datenverarbeitung visualisiert
		3.3	**Implementierung**	[3.3]	1. Welche Technologien verwende ich für meinen Prototypen: - „Welche Python Bibliotheken und IDE?“ - „Welche HW & SW-Anforderungen gibt es?“ 2. Welche Probleme traten bei der Programmierung auf?	1. Software-Abhängigkeits-Portfolio erstellen - Vergleich geeigneter Programmiersprachen - Recherche der erforderlichen Bibliotheken - Recherche der erforderlichen Hardware, Software und Auswahl 2. Software entwickeln - Fehler reporten an Hersteller, Bib, etc.	1. Erstellung eines IT-Konzepts in Form einer Beschreibung der notwendigen technischen Mittel anhand von Teilmodellen 2. Problemstellungen erklären und das Auftreten eines Problems „reverse Engineeren“
		3.4	**Evaluation**	[3.4]	1. Wie ist der Output des Prototyps zu bewerten? 2. Wie bewertet man die Qualität des Outputs? 3. Was kann verbessert werden?	1. Soll-Ist-Vergleich der Anforderungen mit dem Output des Prototypen. 2. Vergleich mit verwandten Arbeiten. 3. Recherche über potentielle Verbesserungen des Algorithmus.	1. Evaluation und Analyse des Ergebnisses anhand von grammatikalischer Richtigkeit und Sinn. 2. Bessere Ergebnisse mit meinen vergleichen. 3. Optimierungsmöglichkeiten für meinen Prototypen evaluieren.
4	**Generierung von übertragba-rem Wissen**			[0] bis [3]	Um welche Elemente könnte mein Projekt modular Erweitert werden um ein Anderes oder Besseres Ergebnis zu erzeugen und welchen Einfluss könnte es auf die Forschung haben?	Verallgemeinerung aus den bisher erarbeiteten Ergebnissen.	Einordnung der Evaluationsergebnisse in einen gesellschaftlichen Kontext.

Abstract

– At the end , finally finished :) –

Contents

Chapter 1.

Intro

The boundary between Natural Language Processing (NLP) and Text Generation (TG), or the latest *Neural Text Generation (NTP)*, is relatively blurred and overlap in many ways. Generally speaking, all of the NTG tasks are NLP based, but not all NLP tasks are NTG based.

1.1. Difference of NLP and NTG

In recent months and years, neural networks have produced many *state-of-the-art* results in almost all possible disciplines of machine learning [Xie 17]. The roots of Neural Networks (NN) lie down almost 80 years ago in 1943, when **McCulloch-Pitts** [McCu 43] compared for the first time neuronal networks with the structure of the human brain. This first attempt to approach artificial neurons with neurons from the brain lead to the nowadays commonly used understanding of a simple neuron of a basic neural network (shown in figure 1.1). Those neurons connected together create an aritificial neuronal network, which can calculate any possible logical or arithmetic function.

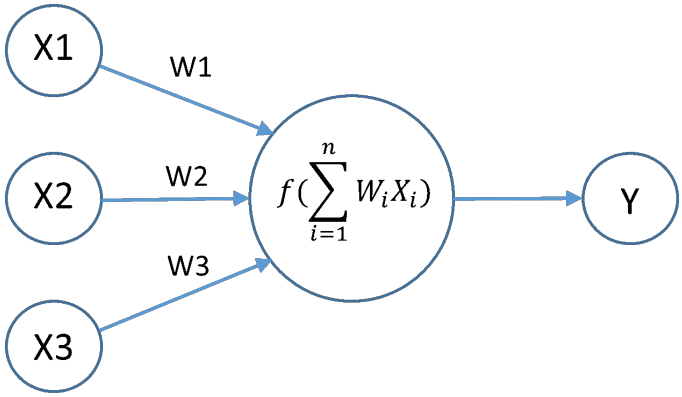

Figure 1.1.: A simple Neuron with 3 inputs and 1 output [Sing 17]

The range in which NN's (2020) can be applied nowadays is wide. Some disciplines have only been created due of the invention of neural networks, because they solve existing- and new problems very effective and efficiently. Many frequently held conferences around the globe contribute continuous evidence of the successes of neural networks. Among those various disciplines counts for example *Pattern recognition* with Convolutional Neural Networks (CNN) [Yann 98] or the famous *CIFAR-10* dataset [Kriz], where many amateurs [Löh 19] and experts attempt annually to further increase the accuracy of predicting the 10 different image classes.
The topic of this thesis *textgeneration* is based from the Natural Language Processing discipline, *NLP* for short. This field covers many other hot research topics, such as

- Sentiment Analysis
- Machine Translation
- Speech Recognition
- Text Generation (Neural Text Generation *NTG*)
- Chatbots

Another term for textgeneration is denoted by *Language Modelling*, because generators use the words and grammar as input for the model. In the past five years were mainly two approaches for modelling NLP, namely the **rule-based** system and the **template-based** system (Figure 1.2) [Xie 17]. Today neural end-to-end systems are is *State-of-the-Art* [Jeka 17]. These new systems offer more flexibility and scale with proportionately better results and less data is required, because the complexity and thus the neccessary computing power has increased. However, this fact leads to a complexity problem, because it becomes very difficult to understand the decisions of the neural network. The neural network is still to a large extent basically a *black box*, although it gives surprisingly good results, especially in NLP. Nevertheless, neural network models for text processing are difficult to understand, so nowadays compromises between rule-based systems still have to be made and hybrid systems are most commonly used.

The neural text generation, also called *NTG*, has many other interesting application fields, which overlap partly with NLP, including

- Speech recording and conversion to text
- Conversation systems e.g. chatbots
- Text summary

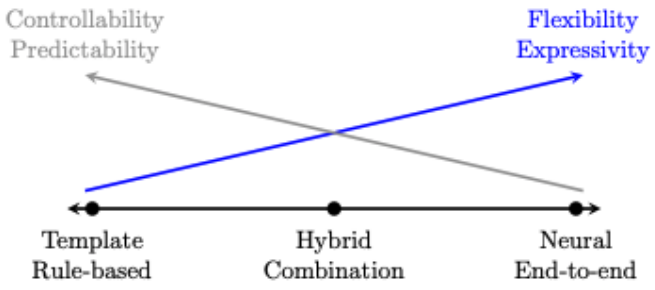

Figure 1.2.: Rule-Based vs. Neural-Text-Generations System [Xie 17]

In order to train language models, they must be taught the probability of occurring words in relation to the preceding words. There are several approaches to achieve this goal. Language models can be trained on the level of words, whole sentences or even whole paragraphs. The granularity in which the training takes place is called *n-grams*, where n represents the number of preceding words.

1.2. Case study of a current NLP system

To give an insight about a current use case, I will describe the basic architecture of the famous **Siri** from Apple. There are many very interesting use cases and I will further discuss the most important hot research topics of 2019-2020 in chapter 2.3, but Siri combines two of the most relevant NLP tasks, namely Speech Recognition and Text Generation. Siri was first released in 2011 for the iPhone 4s, but at this time, users were still required to press the *Home Button* to give Siri a command. For Siri to fulfill the user commands, it needs to understand first the human language itself by recoginizing the words and generate the according text out of it. Secondly it needs to further process those words to a context and figure out what the user wants.

1.2.1. Hands-Free Access to Siri

For the release of the iPhone 6 (iOS 8) in 2015, Apple upgraded Siri to a large extend. Without the need to interacting with the iPhone physically, it can now detect the primary-users voice *"Hey Siri"* to wake up automatically. Even though it might not sound like an innovation, still a lot is going on behind the scences to create an flowless experience. The following figure 1.3 [Team 17] shows the process of detecting the wake-up sentence *"Hey Siri"*.

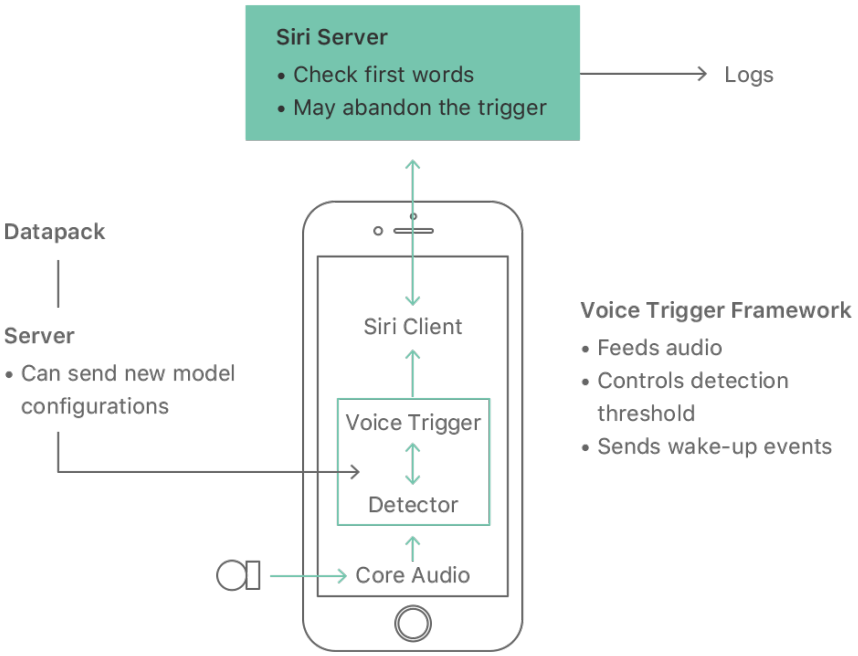

Figure 1.3.: The Hey Siri flow on iPhone (Apple 2017)[Team 17]

Most of the high-tech technologies involved are stored on the cloud. The automatic speech recognition, the natural language interpretation and the various information devices are the major parts [Team 17].
The core In figure 1.3 is basically the acoustic model. The iPhone microphone listens constantly and converts all sounds into a stream of waveform samples. Those samples are transformed into a sequence of frames, which describe approximately 0.01 seconds in time. Those frames are fed into the Deep Neural Network (DNN) acoustic model and produce a log probability to calculate the probabilty of the current sound being the *"Hey Siri"* activation sentence. Adjustments to save the battery, limits of the processing capacities of the Centrul Processing Unit (CPU) and double security checks if the said senetence is really *"Hey Siri"* are not revelant in this aspect.

1.2.2. Personalized Hey Siri

For the introduction of Apple's wake up feature *"Hey Siri"* with the iPhone 6, Apple needs to use multiple machine learning and NLP technologies to function properly on all iPhones. The first task is rather simpler compared to figuring out the context, but still highly researched. When it comes to programming, the language gets split up into basically three parts [Team 18].

- *Syntax*: Composition of the phrases

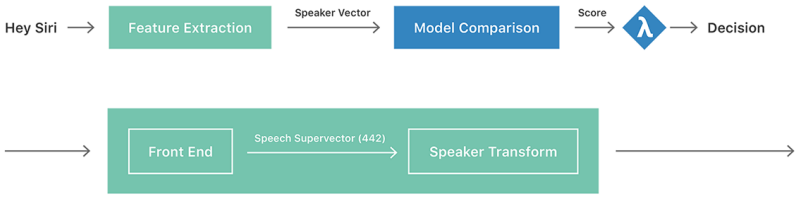

Figure 1.4.: Block diagram of Personalized "Hey Siri" (Apple 2018)[Team 18]

- *Semantics*: Meaning of the phrases
- *Pragmatics*: Composition and context of the phrases

Siri gets activated when it recognizes the words *"Hey Siri"*. This is achieved through the use of Recurrent Neural Networks (RNN) [Team 18], multi-style training and curriculum learning. The RNN will be further explained in chapter 3.2.2. Siri is able to avoid unintended activations which sound similar, but have a different meaning. This is especially challenging, because people all over the world have different accents and dialects, depending on their origins.

When the phone is configurated, in the *enrolment stage* the user is asked to repeat common phrases for a couple of times. This input is fed into a *statistical model* for the user's own voice. In the *recognition stage*, the computer evaluates if the speech input fits to the primary-users-trained model accepts or rejects the request based on that decision.

Figure 1.4 from Apple shows the very basic diagram for this process [Team 18]. The *Feature Extraction* computes a fixed-length speaker vector for the input sentence *"Hey Siri"*. This vector contains information about phonetics, background noise and the identity of the user. In the next step is the vector transformed in such a way, that the environmental background noise is being reduced to a minium with the help of the *Fourier Transform*

Chapter 2.

State of the Art

2.1. Relevant aspects of mathematics

Notationen etc.

2.2. History of NLP and NTG

Zeitabfolge der geschichtlichen Hintergründe

2.3. Current trends in technology

Neuronales Ende-zu-Ende

2.3.1. Application areas of NLP systems

Image-to-Text, Weatherforecast

2.3.2. Application areas of NTG systems

Speech Recognition, Machine Translation

Chapter 3.

Prototyp

In this chapter, we're actually using some code!

```
1 x = 1
2 if x == 1:
3     # indented four spaces
4     print("x is 1.")
```

Listing 3.1: This is an example of inline listing

You can also include listings from a file directly:

```
1 x = 1
2 if x == 1:
3     # indented four spaces
4     print("x is 1.")
```

Listing 3.2: This is an example of included listing

3.1. Zielsetzung

Image Captioning

3.2. Fachkonzept

Fachkonzept

3.2.1. Struktur

The different steps of Text Generation

- Importing Dependencies
- Loading the Data
- Creating Character/Word mappings
- Data Preprocessing
- Modelling
- Generating text

3.2.2. Neuronales Netz

LSTM

Experimenting with different models

- A more trained model
- A deeper model
- A wider model
- A gigantic model

3.2.3. Prozessmodellierung

Funktionen etc.

3.2.4. Datenflussmodellierung

Diagramm

3.3. Implementierung

Code

3.4. Evaluation

Print Ergebnisse

Bild

Image Caption

Chapter 4.

Generierung von Übertragbarem Wissen

Modulare Erweiterbarkeit meines Projekts. Einordnung in Gesellschaftlichen Kontext

Appendix A.

Supplemental Information

List of Figures

List of Tables

List of Listings

Bibliography

[Jeka 17] O. D. Jekaterina Novikova and V. Rieser. “The E2E Dataset: New Challenges For End-to-End Generation”. 2017.

[Kriz] A. Krizhevsky, V. Nair, and G. Hinton. “CIFAR-10 (Canadian Institute for Advanced Research)”.

[Löh 19] T. Löhr and T. Bohnstedt. “Image Classification on the CIFAR10 Dataset”. 2019.

[McCu 43] W. S. McCulloch and W. Pitts. “A logical calculus of the ideas immanent in nervous activity”. *The bulletin of mathematical biophysics*, Vol. 5, No. 4, pp. 115–133, 1943.

[Sing 17] P. Singh. “Neuron explained using simple algebra”. 2017.

[Team 17] A. S. Team. “Hey Siri: An On-device DNN-powered Voice Trigger for Apple’s Personal Assistant”. 10 2017.

[Team 18] A. S. Team. “Personalized Hey Siri”. 04 2018.

[Xie 17] Z. Xie. “Neural Text Generation: A Practical Guide”. 2017.

[Yann 98] L. B. Yann LeCun, Patrick Haffner and Y. Bengio. “Object Recognition with Gradient-Based Learning”. 1998.