Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Implementing ChatBots using Neural Machine Translation techniques

Degree's Thesis
Telecommunications Sciences and Technologies

Author: Alvaro Nuez Ezquerra
Advisors: Marta R. Costa-Jussà and Carlos Segura Perales

Universitat Politècnica de Catalunya (UPC)
2017 - 2018

Abstract

Conversational agents or chatbots (short for chat robot) are a branch of Natural Language Processing (NLP) that has arisen a lot of interest nowadays due to the extent number of applications in company services such as customer support or automatized FAQS and personal asistent services, for instance Siri or Cortana.

There are three types: rule-based models, retrieval-based models and generative-based models. The difference between them is the freedom they have at the time of generating an answer given a question. The chatbot models usually used in public services are rule-based or retrieval-based given the need to guarantee quality and adecuate answers to users. But these models can handle only conversations aligned with their previous written answers and, therefore, conversations can sometimes sound artificial if it goes out of the topic.

Generative-based models can handle better an open conversation which makes them a more generalizable approach. Promising results have been achieved in generative-based models by applying neural machine translation techniques with the recurrent encoder/decoder architecture.

In this project is implemented, compared and analyzed two generative models that constitute the state of the art in neural machine translation applied to chatbots. One model is based on recurrence with attention and the other is based exclusively in attention. Additionally, the model based exclusively on recurrence has been used as a reference.

Experiments show that, as in translation, an architecture based only in attention mechanisms obtains better results than the recurrence based models.

Resum

Els agents conversacionals o chatbots (abreviació de chat robot) és una branca del Processat de Llenguatge Natural (PLN o en anglès NLP) que ha despertat gran interès avui en dia degut a la gran quantitat d'aplicacions en serveis com atenció al client, FAQs i sistemes d'assistència personal com Siri o Cortana.

Hi ha de tres tipus: els sistemes basats en regles, els models basats en recuperació i els models generatius. La diferència entre ells resideix en la llibertat que tenen a l'hora de generar una resposta donada una pregunta. Els models de chatbots utilizats comunament en serveis públics son de tipus recuperació o basats en regles a causa de la necessitat de garantir respostes correctes i de qualitat als usuaris. El problema d'aquests models és que tan sols poden mantenir converses relacionades amb les seves respostes escrites prèviament i, aleshores, les converses poden ser molt artificials si un usuari es desvia del tema.

Els models generatius, per altra banda, poden desenvolupar-se molt millor en converses obertes, el que els converteix en un enfocament més generalitzable. S'han aconseguit prometedors resultats en l'ámbit dels models generatius gràcies a l'aplicació de tècniques de traducció neuronal amb arquitectures encoder/decoder basades en recurrència.

En aquest projecte s'implementa, es compara i s'analitza dos tipus de models generatius que constitueixen l'estat de la qüestió en traducció neuronal aplicats a chatbots. Un dels models es basa en recurrència i mecanismes d'atenció i l'altre es basa exclusivament en atenció. Adicionalment, el model basat exclusivament en recurrència s'ha utilizat com a referència per als experiments.

Els experiments demostren que, com succeïa en traducció, una arquitectura basada exclusivament en mecanismes d'atenció obté millors resultats que aquells basats en recurrència.

Resumen

Los agentes conversacionales o chatbots (abreviación de chat robot) es una rama del Procesado de Lenguaje Natural (PLN o en inglés NLP) que ha despertado gran interés hoy en día debido a la gran cantidad de aplicaciones en servicios como atención al cliente, FAQs y sistemas de asistencia personal como Siri o Cortana.

Existen tres tipos: los sistemas basados en reglas, los modelos basados en recuperación y los modelos generativos. La diferencia entre ellos reside en la libertad que tienen a la hora de generar una respuesta dada una pregunta. Los modelos de chatbot utilizados comúnmente en servicios públicos son de tipo recuperación o basado en reglas dada la necesidad de garantizar respuestas correctas y de calidad a los usuarios. El problema de estos modelos es que tan solo pueden mantener conversaciones relacionadas con sus respuestas pre-escritas y, por tanto, las conversaciones pueden llegar a ser muy artificiales si un usuario se desvía del tema. Los modelos generativos, por otro lado, pueden desenvolverse mucho mejor en conversaciones abiertas, lo que los convierte en un enfoque más generalizable. Se han logrado prometedores resultados en el ámbito de los modelos generativos gracias a la aplicación de técnicas de traducción neuronal con arquitecturas encoder/decoder basadas en recurrencia.

En este proyecto se implementan, comparan y analizan dos tipos de modelos generativos que constituyen el estado del arte en traducción neuronal aplicados a chatbots. Uno de los modelos está basado en recurrencia y mecanismos de atención y el otro está basado exclusivamente en atención. Adicionalmente, el modelo basado exclusivamente en recurrencia se ha utilizado como referencia para los experimentos.

Los experimentos demuestran que, como sucedía en traducción, una arquitectura basada exclusivamente en mecanismos de atención obtiene mejores resultados que aquellos basados en recurrencia.

Acknowledgements

First of all I want to thank my tutors Marta R. Costa-Jussà and Carlos Segura Perales for letting me be part of this project and motivate me to continue investigating about the fascinating world of AI. It has been such a pleasure to work with you. Also I want to express my gratitude to Carlos Escolano and Noe Casas for helping me with some problems I found during the realization of the project.

I want to thank my parents for all the dedication and effort they have had with me to bring me here. All this is thanks to you.

And thank you Judith for the immense patience you have had with me, listening and supporting all my ideas day after day and being at my side whenever I needed it.

Revision history and approval record

Revision	Date	Purpose
0	11/01/2018	Document creation
1	20/01/2018	Document revision
2	23/01/2018	Document revision
3	25/01/2018	Document approbation

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Alvaro Nuez Ezquerra	alvaronuez.eis@gmail.com
Marta R. Costa-Jussà	martaruizcostajussa@gmail.com
Carlos Segura Perales	carlos.seguraperales@telefonica.com

Written by:		Reviewed and approved by:		Reviewed and approved by:	
Date	11/01/2018	**Date**	25/01/2018	**Date**	25/01/2018
Name	Alvaro Nuez Ezquerra	**Name**	Marta R. Costa-Jussà	**Name**	Carlos Segura Perales
Position	Project Author	**Position**	Project Supervisor	**Position**	Project Supervisor

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A conversational agent or chatbot is a language recognition system able to maintain a conversation with an user using a question/answer protocol. The communication can be done by audio or text media. As far as concerned in this project, we are going to focus on the textual models.

The first chatbot models were rule-based, for instance ELIZA[13], PARRY[1] and A.L.I.C.E[2]. These models require a programmer to write some rules and patterns beforehand for the analysis and decomposition of a sentence and then, create an answer by the combination of a template and keywords.

Thanks to the latest advances in machine learning and more specifically in artificial neural networks, it is possible to create chatbot models that do not longer require previous written rules. Instead, given a set of examples, the chatbot learns the pattern inherent to the samples. There are two different approaches depending on the freedom they have at the time of generating an answer: retrieval-based and generative-based.

The retrieval-based systems determine which answer, from a set of answers previously written, is the most appropriate given a sentence/question as an input. These models are quite useful when the target domain is limited (e.g. a model trained only for sport or medicine conversations) and the chatbot is not allowed to commit grammatical or semantic errors during its service, for instance in FAQs[3] and costumer support services. The problem is that they barely handle unseen questions and became impractical in open domains (e.g. general knowledge).

Generative models, on the other hand, are trained to generate data as a response word by word. Nevertheless, not having rules implies that they have to learn to build sentences during their training. For that reason, they are more complex and harder to train than the retrieval-based systems. Usually generative models are prone to commit grammatical and semantic errors but on the other hand, they better handle new data and can answer with more natural sentences. What makes them an interesting approach is that they are an advance towards what is known as Strong Artificial Intelligence[4] (Strong AI), is the system itself who analyze, compute and build an answer thanks to an autonomous learning without any human intervention.

A great step forward to the area of generative-based chatbots was the implementation of a model using an encoder/decoder architecture with recurrent neural networks known as sequence to sequence (Seq2Seq) [12] used in translation [2] [9]. This project is motivated by the good results shown in the experiment, which achieve a new state of the art in the generative-based chatbot area.

[1]PARRY was a chatbot that simulated a person with schizophrenia. It was created by psychiatrist Kenneth Colby as a counterpart to the ELIZA model.

[2]Artificial Linguistic Internet Computer Entity (A.L.I.C.E) is a chatbot created by Dr. Richard S. Wallace. The chatbot uses the Artificial Intelligence Markup Language (AIML) for the definition of patterns and rules.

[3]Frequent Asked Questions services

[4]Strong Artificial Intelligence is the word used to refer to systems that can perform any intellectual task that a human brain can.

1.1 Statement of purpose and contributions

The main goal of this project is to apply to generative-based conversational agents (chatbots) two encoder/decoder architectures from the state of the art in translation techniques and determine, with an experiment, which has a better performance in general topic conversations.

The main contribution of this project is the first implementation of a generative-based chatbot using the Transformer architecture. Results show that this architecture outperforms the state of the art in generative-based chatbot models. Additionally, some improvements have been applied to the basic model implemented by Google Brain resident Etienne Pot[6] by adding a bidirectional encoder, the attention mechanism and a beam search algorithm to improve the quality of the answers. This improved version has also been shown to outperform the basic model.

1.2 Requirements and specifications

As one of the main languages used in machine learning nowadays, this project has been developed completely in Python 3.5.3. using the open-source software library for machine learning TensorFlow [5] for both implementation and training of models.

All the software has been launched in a cluster of 8 servers from the TSC department of the UPC, each with 2 Intel® Xeon® E5-2670 v3 2,3GHz 12N processors, and a total of 16 NVIDIA GTX Titan X GPUs. Each GPU has 12GB of memory and 3072 CUDA Cores.

1.3 Methods and procedures

The project's main idea was originally proposed by my supervisors and starts from previous work [12] [11]. The baseline of this project is the model used in [12], which is a generative-based chatbot implemented with the neural machine translation architecture Seq2Seq. For its implementation it has been used a basic model published by Google Brain resident Etienne Pot at his GitHub page[6]. After testing the model with different data sets, it was improved by adding an attention mechanism that allows the model to focus in the most relevant characteristics from the input sentence in the decoding phase. Additionally, as in [1], it was incorporated to the model a bidirectional encoder and a beam search algorithm at the decoder to improve the quality of the answers.

Finally, the main contribution uses an architecture recently proposed by the Google research team [11] which consists in an encoder/decoder architecture based exclusively in attention mechanisms without recurrent neural networks. The motivation to use the architecture is because it has been shown to outperform the state of the art in translation systems. It has been built using a library from TensorFlow called Tensor2Tensor [7].

[5] https://www.tensorflow.org/
[6] Seq2Seq Chatbot model code available at https://github.com/Conchylicultor/DeepQA
[7] https://github.com/tensorflow/tensor2tensor

1.4 Work Plan

This project has followed the work plan described below:

- WP 1: Project propose and work plan.
- WP 2: Information research
- WP 3: Project development
- WP 4: Critical review
- WP 5: Test and results
- WP 6: Final Report
- WP 7: TFG presentation

Additionally in Figure 1.1 can be seen the Gantt diagram of the project.

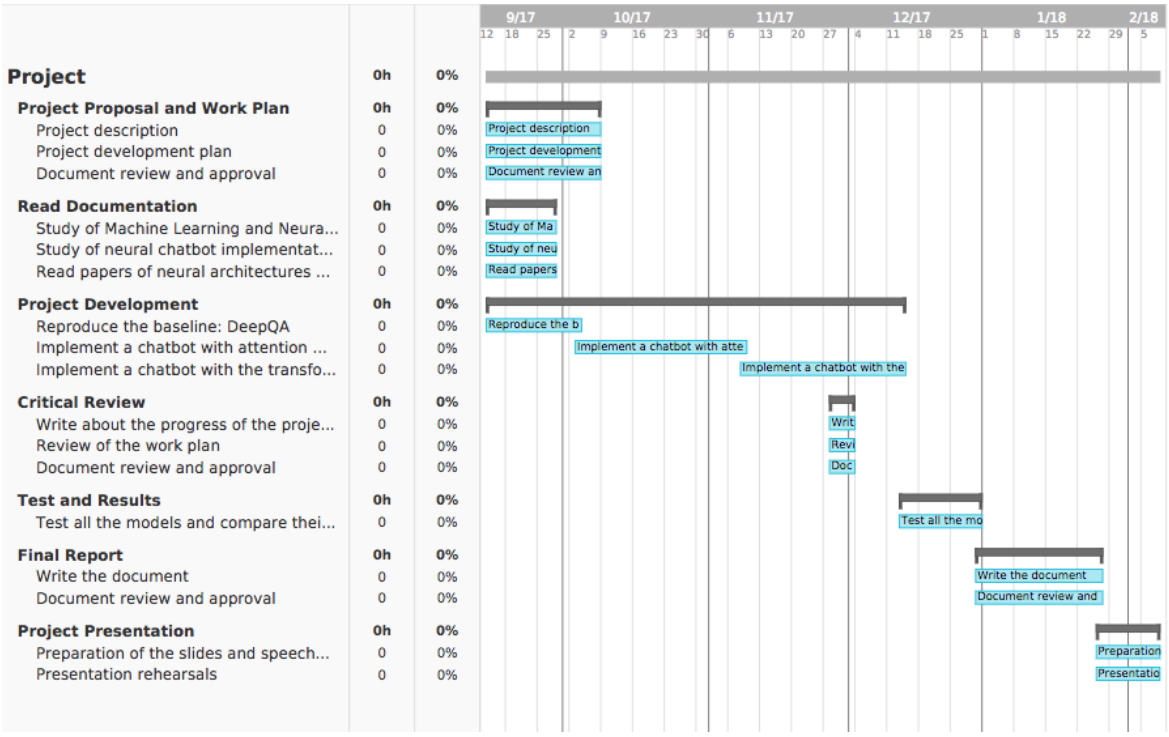


Figure 1.1: Gantt Diagram of the Degree Thesis

Chapter 2

State of the art

This chapter explains the theoretical background that holds this project. First, this section defines the area which studies and develops chatbot models, Natural Language Processing. Then, this section provides a global vision of what are the learning algorithms and how they work. After that, this section goes deeper in the explanation of the specific ones used along the project. Finally, this section ends reporting the type of architectures chosen.

2.1 Natural Language Processing - NLP

Natural Language Processing (NLP) is a research area of Artificial Intelligence (AI) which focus in the study and development of systems that allows the communication between a person and a machine through natural language.

Chatbots belong to the area of NLP given the importance of their ability to understand natural language and know how to extract relevant information from it. Both models, retrieval-based and generative-based must be able to identify some information from the input sentence in order to pick or create an answer.

2.2 Machine Learning

Machine Learning is a field of study of AI that studies and develop techniques capable to learn tasks as classification or regression from a data set. There are different algorithms without being any of them, in general, better among the others (No Free Lunch Theorem) [1]. The suitability of one algorithm in particular, depends exclusively on the nature and type of the problem addressed.

The aim of a learning algorithm is to estimate the behaviour of a training set by the identification of their inherent pattern. Once accomplished, it must be capable to perform tasks as classification or regression given unseen samples.

All the learning algorithms require a learning phase at which, an objective function is defined as a metric to optimize in order to get a reference of how well our model fits to the problem (e.g. Minimization of the error function). Then, the algorithm iterates through the training set looking for the optimization of the metric. It is important to have three disjoint sets of samples in machine learning algorithms: training, validation and test set. The training set is used as examples for the objective function optimization. A validation set is required when it is necessary to compute the optimal parameters of an algorithm. Finally, the test set is used to test how well the algorithm has learned and generalized the problem.

[1] No Free Lunch Theorem - Wolpert (1996): Without any prior information about the problem, there are not any superior pattern classification algorithm.

There are two types of learning algorithms: supervised and unsupervised. Difference lies in, if during the training process, training samples are labeled with information of the class they belong or conversely there is no additional information and is the system who must determine which class they belong to.

These algorithm have become popular because they reduce the human intervention at the time of defining rules or patterns to the systems, letting to them to extract that information. They have changed lots of areas as image processing, audio and speech processing, translation and conversational systems among others.

Most of them where created long time ago but their true potential has been possible thanks to recent computing capacity improvements and the availability of big data bases.

2.3 Biological Neuron

The learning algorithm at which focuses this project, is inspired by the biological neurons of the brain.

Neurons are a type of cell from the nervous system composed by a cell body called soma, some input signal branches called dendrites and a single output signal branch called axon (Figure 2.1).

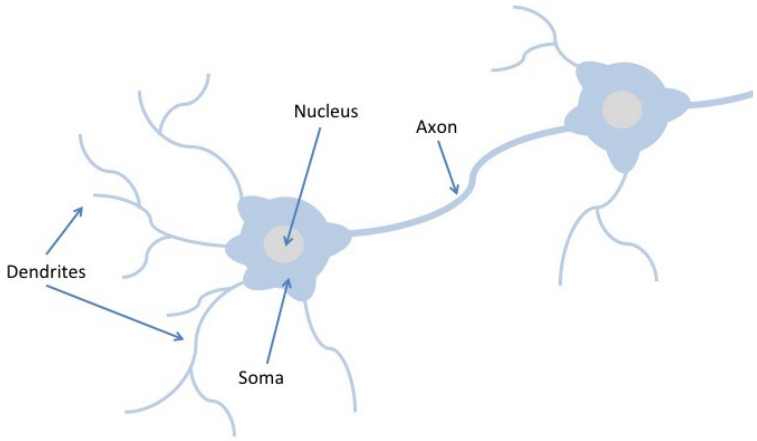


Figure 2.1: Structure of a neuron cell

Axons split in their extremities into different sub-branches called tellodendrites. Connection between tellodendrites of a neuron and dendrites (or directly the soma) of another is performed by the synaptic terminal, which is a small structure that contains neurotransmitter molecules responsible for the exchange of the nervous signal (also known as synapses).

Neurons, emit electrical impulses by the axon if the amount of electrical excitation received by the dendrites exceeds a threshold.

2.4 Artificial Neural Networks

A neural network is a type of machine learning algorithm that is inspired by the behaviour of biological neurons in the brain. It consists of a group of basic units called artificial neurons (AN) or perceptron (see Figure 2.2) which are connected among them composing a complex network. They can compute an output given input data by decomposing it in different representations in order to identify different characteristics.

The first model of AN was proposed by neurophysiologist Warren McCulloch and mathematician Walter Pitts in 1943 [4]. The proposed model is a simple mathematical approximation of the operation of a biological neuron, capable to compute basic operations as identity function, AND, OR and NOR.

Many other models have been proposed since then, but the most simple AN architecture is the perceptron which was proposed by Frank Rosenblatt in 1957 [4]. Whilst the AN model proposed by McCulloch and Pitts used binary values, the perceptron can operate with any numbers. The algorithm computes an activation function over the weighted sum of the input values. Additionally, in order to give one extra degree of freedom, a bias is added as shown in the following equation:

$$output = f(\sum_i x_i * w_i + w_0) \tag{2.1}$$

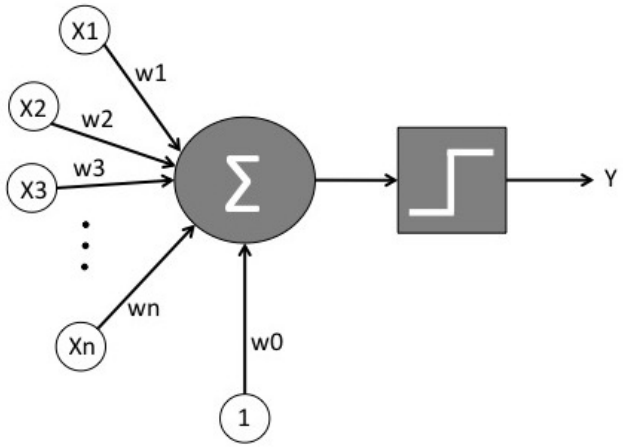


Figure 2.2: Structure of a perceptron

where w_i and w_O are the weights and bias respectively. The optimal values of these parameters are computed using gradient descent techniques starting from a labeled training data set[2]. Gradient Descent is an iterative optimization algorithm used to find the global minimum of a function.

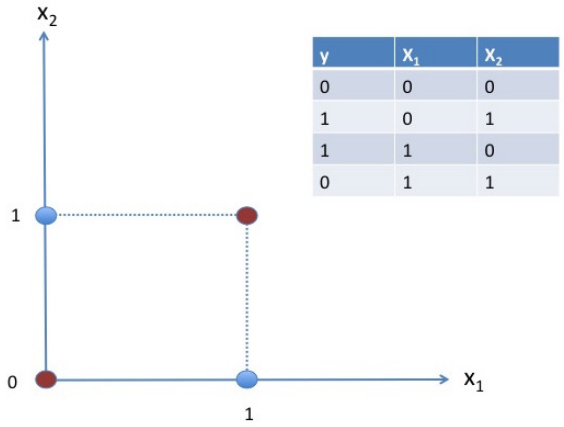
Activation functions are continuous and differentiable non-linear functions. They are required to be smooth in order to be able to learn from gradient descent techniques. The non-linearity

[2]In unsupervised learning, it is required to apply clustering techniques first, in order to label the data.

is an important condition that ensures a non linear discriminant expression at the output of the neural network, on the contrary a multilayer and single layer network perform alike.

Output values are binary, if the weighted sum exceeds a threshold imposed by the activation function (originally the Heaviside step function), then the output is activated on the other hand the output value is deactivated.

The perceptron operates as a linear discriminant, which means that every unit can linearly separate samples into two classes. It is possible to compute basic operations as AND or OR, but functions as XOR, are non-linear separable problems and therefore not implementable (Figure 2.3).



y	x_1	x_2
0	0	0
1	0	1
1	1	0
0	1	1

Figure 2.3: XOR operation is a non-linear separable problem

The solution is a multilayer perceptron (MLP) which is a network composed by multiple layers of perceptrons (Figure 2.4). The basic structure of a MLP is composed by an input layer where all data is fed to the network, one or more hidden layers for multiple representations of data and characteristic identification and finally an output layer. The output layer can use a different activation function depending on the nature of the task. For classification, the output layer uses a softmax function that represents, for each target class, a probability of success.

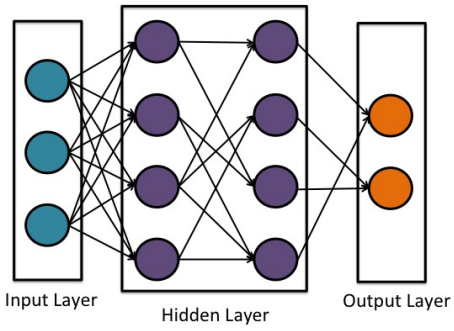


Figure 2.4: Structure of a multilayer perceptron

MLP is a basic architecture of neural network, but there are many more with different structures and types of unit cells. Depending on the nature of the problem, some architectures are preferred over the others, for instance, convolutional neural networks for image processing or recurrent neural networks for input signals in time.

2.5 Recurrent Neural Networks - RNN

Two of the models covered by this project use a special type of artificial neural network called recurrent neural network (RNN). RNN have the ability to retain information from previous data as a temporal memory. They can be viewed as a sequence of concatenations of the same unit (see Figure 2.5) like a chain, where each one computes an output given an input and the information provided by the last network.

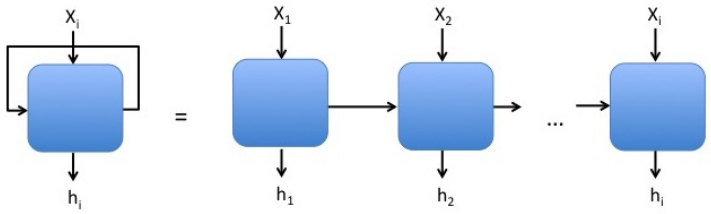


Figure 2.5: Diagram of a recurrent neural network

Due to its temporal memory, they are quite useful for sequential data were each element of the sequence is related to the the others.

Nevertheless, RNN can only retain recent information from a sequence, which means that they can only perform correctly when the element to be processed is near at the sequence to the relevant information.

Long Short Term Memory Networks (LSTM networks) [5] are a special type of RNN wich can recall long term dependencies due to its internal structure (see Figure 2.6). Whilst a basic RNN is composed by a single operation layer, LSTM networks use four. Internally they can perform three operations: forget information, update information and output information.

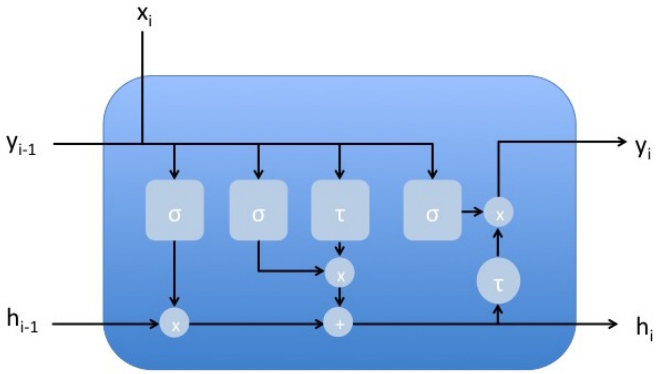


Figure 2.6: Diagram of a LSTM network

A variant of LSTMs are the Gated Recurrent Unit (GRU), presented at [2], which have a simpler structure, making them computationally more efficient and faster to train.

2.6 Encoder/Decoder architectures

The architectures used in this project are inspired in a specific neural network model known as encoder/decoder. The encoder projects information from a variable-length input into a fixed-length vector (also known as thought or context vector) from a representation vector space. Then the decoder projects the vector into the original space of symbols. To train these models, given a pair input/target, the minimization error between the network's output and the target is sought.

Chapter 3

Architectures

In the following chapter will be explained the different encoder/decoder architectures used in this project. First, it will be described the RNN architecture used in [12]. Second an overview of the attention mechanism used by the other two models will be provided. Finally, it will be explained the most recent architecture based exclusively in attention.

3.1 RNN Encoder/Decoder architecture (Seq2Seq)

A very successful implementation of Encoder/Decoder architecture for NLP tasks (specially in neural machine translation) is the RNN Encoder/Decoder [2] [9] also known as Sequence to Sequence (Seq2Seq). The encoder and decoder are recurrent neural networks, which allows the model to be fed with variable-length input sentences.

Given an input sentence, the encoder iteratively computes for each word a hidden state vector using the word and previous hidden state of the RNN. Once the whole sentence has been analyzed, the relevant information of the input sentence is contained in the last hidden state of the RNN, known as context or thought vector. The decoder computes, word by word, an output in the original representation space using the information contained in the context vector and previous decoded words.

The architecture implementation can vary depending on the type of RNN cell used (genuine RNN cell, a LSTM cell or a GRU cell), number of cells per layer or the number of hidden layers among other parameters. Figure 3.1 shows a diagram of the sequence to sequence architecture.

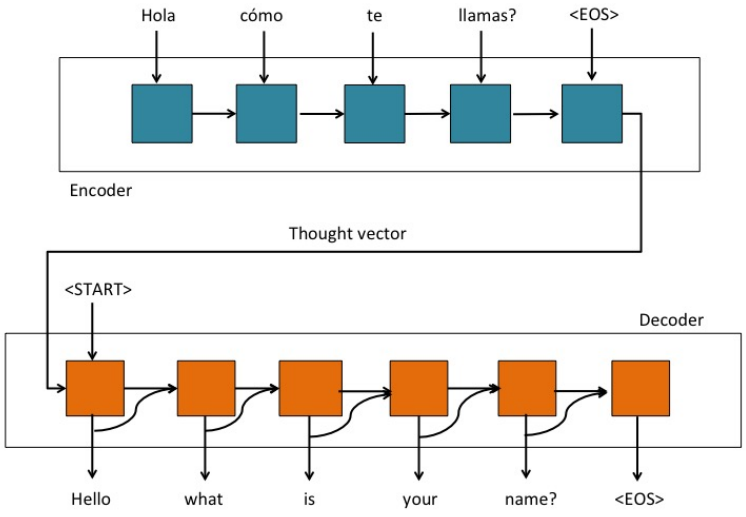


Figure 3.1: Diagram of the sequence to sequence architecture

As the size of the sentence increases, it is needed to encode a large quantity of information into a fixed-length vector, so some of it is lost at the encoding process resulting in a poor performance of the chatbot.

3.2 Attention Mechanism

A solution to the problem due to the fixed-length nature of the context vector is to allow the decoder to "see" the most relevant words of the input sentence during the decoding process (Figure 3.2). This method is called Attention Mechanism [1].

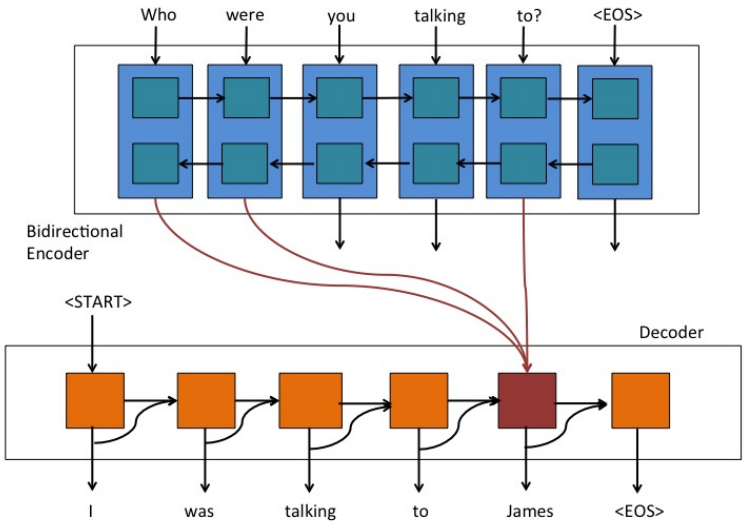


Figure 3.2: Diagram of the sequence to sequence architecture with a bidirectional encoder and applying attention in the decoding process

Instead of only using the last hidden state vector as the context vector, for each word to decode, the decoder computes a context vector with a weighted sum of all hidden state vectors of the encoder. Unlike the Seq2Seq previously presented, for the attention mechanism a bidirectional encoder is used to encode the input sentence word by word into two hidden cell state vectors. The first one, going from the first to the last word of the sequence (forward) and the other one, reversing the sequence going from the last to the first word (backwards). The final hidden cell state vector is a concatenation of the hidden state going forwards and the hidden state going backwards (bidirectional hidden state vector). The bidirectional encoding allows to encode more information of a word from the sentence.

Weights are computed by an alignment model and normalized over all values to get a percentage of how relevant the word from the input sentence is, in relation to the word to be decoded. [1]

[1]For further technical explanation of how weights are computed see [1].

3.3 Transformer architecture

The transformer architecture is an encoder/decoder model based entirely on attention mechanism [2] recently proposed by the Google team [11] as a new state of the art neural machine translation (NMT) architecture. Additionally, they proved that the model can be generalized for other NLP tasks as English constituency parsing [6].

RNN are intrinsically sequential, which is a problem at parallelizing RNN models as Seq2Seq. This problem is solved in the Transformer architecture due to be based only in attention mechanism and lack of RNN. Moreover, it has been proven that they require less training time than the RNN encoder/decoders.

There are three main stages in the encoder (see Figure 3.3). The first one is where input words are projected into a vector representation space by an embedding matrix and then, given that there is no information of the order and position of words in the input sentence [3] a positional encoding is added to the embedded input vectors. The second stage is a multi-head attention block (of Self-Attention in this first case) that linearly projects the input information into different space representations and performs attention over all of them. This method allows the model to identify different semantic, morphological and lexical characteristics of the input sequence and attend them separately at the decoding process. Finally a position-wise feed-forward network is used, which applies two linear transformations to each position separately.

The decoder has five stages, the first two only used at the training phase: an output embedding and positional encoding (similar to the one used in the encoder but for target sentences in the training phase), a masked multi-head attention (also Self-Attention), a multi-head attention, a feed forward network and finally a softmax layer to compute the output probabilities. Given that at the decoding process we can not know the future words, the attention can only be applied to previous ones. This is what the masked multi-head attention does, which is a multi-head attention block with a mask that restricts the attention only to past words. For a deeper technical explanation of the architecture see [11].

[2] Unlike previous encoder/decoder models as Seq2Seq which uses RNN.
[3] In RNN encoder/decoder models, due to their sequential nature, no positional information is required.

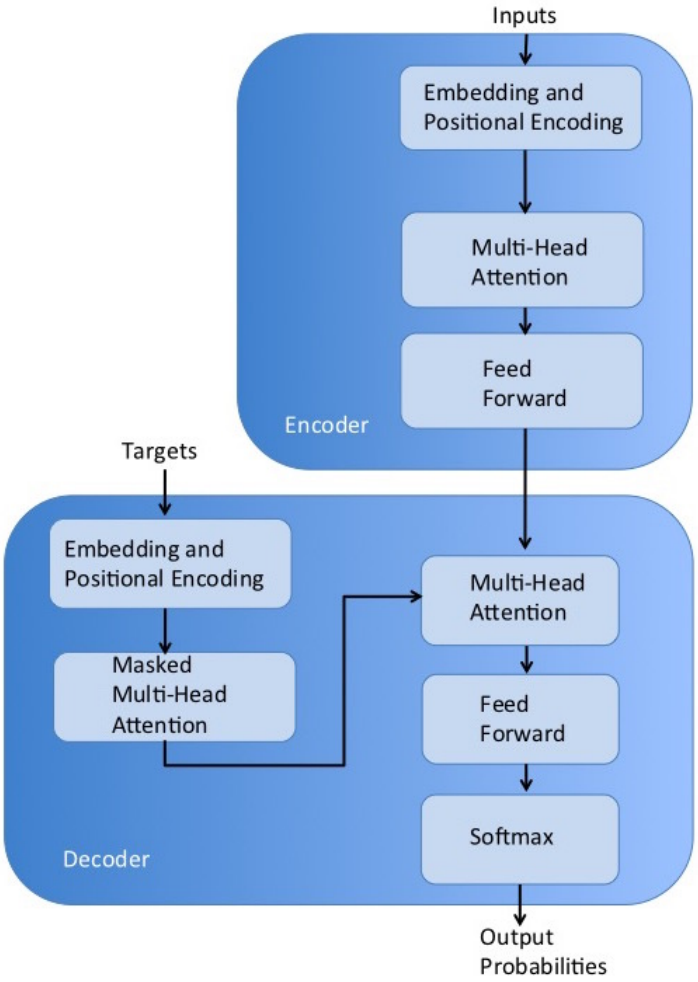


Figure 3.3: Simplified diagram of the Transformer architecture

Chapter 4

Implementation

As previously stated, the project has been divided in four parts along the semester, being the first three parts the implementation of previously described models and the last one their performance test. In the following chapters it will be explained how the models where implemented for chatbot tasks, which parameters and data sets where used and why, and how the models where trained. Finally, it will be explained how the models were tested, which was the criterion used for the evaluation and which were the results.

All code has been written in Python using Tensor Flow libraries. For the implementation of the baseline model and the Seq2Seq + Attention model, it has been used code written by Google Brain resident Etienne Pot[1] in TensorFlow available at his GitHub page[2]. The software consists of a simple implementation of the Seq2Seq architecture plus an interface with a wide range of adjustable parameters and tools.

On the other hand, the transformer model has been created using TensorFlow's library Tensor2Tensor, which was published by researchers from [11] as an open-source code for the recreation of their model.

4.1 Data bases

Along the project, three data sets have been used: Ubuntu Dialogue Corpus[8], a small version of the English Open Subtitles Corpus [10] and the full English Open Subtitles Corpus[3].

The motivation to use these data sets comes to test the baseline in two different environments, a closed one related to technical issues and another with a wide topic range conversations and decide which one to use in the future as a standard for all models.

First experiments were performed with Ubuntu Dialogue Corpus and Open Subtitles Corpus (small version). Given the need of a bigger corpus for the final experiments, it was used the full Open Subtitles Corpus.

4.1.1 Ubuntu Dialogue Corpus

Ubuntu Dialogue Corpus[8] is a set of 7 million sentences with 100 million words extracted from IRC[4] networks of Ubuntu. The data set is a task-specific domain corpus of Ubuntu technical support.

[1] More about him at his personal page: http://e-pot.xyz/
[2] https://github.com/Conchylicultor/DeepQA
[3] Corpus available at `http://opus.lingfil.uu.se/download.php?f=OpenSubtitles2016/en.raw.tar.gz`
[4] Internet Relay Chat

4.1.2 Open Subtitles Corpus

The Open Subtitles Corpus is composed by a wide range of movie and TV series scripts translated to multiple languages. It is generally used by video platforms to show subtitles of their movies/TV series.

As for the project, a small version of the English corpus with 1,651,332 sentences was used for first experiments and lately, given the need of a bigger data set, the full English corpus, composed by 337,847,902.

4.2 Pre-Processing of the full Open Subtitles Corpus

Although the Ubuntu Dialogue Corpus and the small version of the Open Subtitles Corpus were already processed, the full Open Subtitles Corpus required a pre-processing to clean a little bit the data set and adjust it to the experiments.

The initial data base format of the full Open Subtitles Corpus was a set of XML files distributed in different directories corresponding to movies and TV series. For simplicity, all scripts were extracted and written into a single file, were each row corresponded to the dialog of a single speaker. The final file contained 337.847.902 sentences which was incredibly big. The corpus had a lot of noise, it required a pre-processing to reduce it. First, with a python script the data set was cleaned of symbols as: **"**, *****, -, **#** and musical note symbols. After the pre-processing the data set was reduced to 335,190,993 sentences.

Nevertheless, it continued to be large, so some statistics were computed (Table 4.1) in order to get the maximum length that ensured to be covering the 99 % of the data set.

Percentage of covered corpus	**50%**	**75%**	**90%**	**95%**	**99%**
Maximum sentence length	4	7	11	15	24

Table 4.1: Table of percentage of covered data set for different lengths

Additionally the maximum and minimum sentence lengths computed were 8251 and 0 respectively. After some post-analysis it was found that some sentences were long sequences of random numbers and others just null.

By limiting the data set to 24 words as the maximum sentence length it is ensured to be covering the 99 % of the corpus and avoiding random sequences (noise is reduced). After limiting the length and discarding null sentences the corpus was reduced to 331,595,588 sentences.

Given that it is necessary to establish a vocabulary size as a parameter for the models, from the processed corpus it was computed statistics of covered corpus given maximum vocabulary sizes (Figure 4.2). Among the 331,595,588 sentences there were 2,188,717,613 words from which 2,420,428 of them were different.

Percentage of covered corpus	50%	75%	90%	95%	99%
Vocabulary size	40	281	2365	8049	72827

Table 4.2: Table of percentage of covered data set for different vocabulary sizes

Given the results, the vocabulary size choose is 72,827 words which ensures to be covering the 99 % of the corpus.

4.3 Parameters

4.3.1 Baseline model

At the experiments in [12], the architectures had 1024 unit cells for the Ubuntu model and 4096 for the OpenSubtitles. Due to computational limitations, our model had to be simpler, for that reason we used a two layered LSTM model with 512 unit cells per layer. Although the model could not achieve the results of the mentioned experiments, it would give quite good results. The initial code used a 32 dense dimension for the embedding matrix, but given that the vocabulary size of the data set was quite large, it was double to 64.

As for the training, it was used ADAM [7] which is an optimizer algorithm that has been shown to achieve good results in deep learning applications and has been established as a reference. Also, it has been use the recommended parameter values for generic deep learning tasks in [7]: a learning rate of 0.2, $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 10^{-9}$. Additionally, in order to avoid over fitting, a dropout mechanism[5] was applied with a keep probability of 90%. Also a 256 sample batches were used as input data.

For better results, a beam search algorithm has been added at the decoder with a beam size of 4 and a penalty length factor of 0.6.

4.3.2 Seq2Seq + Attention Mechanism model

The Seq2Seq + Attention Mechanism model (henceforth +Attention model) uses the same parameters as the baseline model for the Seq2Seq architecture and training.

Due to the Attention Mechanism and large vocabulary sizes, the +Attention models require lot of memory. A solution is to use a sampled softmax loss function which instead of training with all possible words uses only a smaller random set, which in this project is set to 512.

Due to problems with TensorFlow's new version, it was required to modify some of the native function of RNN encoder with attention in order to gift a bidirectional encoding to the model.

[5]The dropout mechanism applies a probability, to each unit in the neural network, to be dropped. The elimination of units forces the model to lose some specificity over the training set and therefore avoid over fitting.

4.3.3 Transformer model

The configuration parameters are the same as the ones used in [11] with few simplifications. The model has 8 attention head at the Multi-Head Attention block and 6 hidden layers with 512 unit cells each one. As the baseline model and the +Attention model, a beam search algorithm is used at the decoding process with a beam size of 4 and a length penalty factor of 0.6.

For the training, it has been used ADAM optimizer algorithm but with the recommended parameters for the model in [11]: a learning rate of 0.2, $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 10^{-9}$. The model also had a dropout mechanism with the same keep probabilities as the Seq2Seq and +Attention. The batch size of the input data is of 4096 samples.

Chapter 5

Evaluation

During the project three main evaluations were performed. First, in order to choose a suitable corpus for the experiments, two models were trained using Ubuntu Dialog Corpus and Open Subtitles Corpus (small version).

5.1 Evaluation between Ubuntu and Open Subtitles models

At the beginning of the project, it was necessary to choose between the Ubuntu and Open Subtitles corpus as a reference for the whole project. To compare them, two baseline architectures, each one with a different corpus, were trained. Limitations were imposed to the data sets as a maximum vocabulary size of 40,000 words and a maximum length of 30 words.

Training a conversational model is quite difficult. The training error, computed as the difference between the model's output and the target sentence, is not a good metric to have an idea of the learning process. Whilst most of the AI areas have some specific metrics to compare, in conversational models there is no mathematical metric to compute in order to have an idea of how well the model answers, it is all subjective. The problem constrains the evaluation to human supervision. For training the initial models, it was established an initial number of epochs[1] to 30 and then evaluate the answers. Depending on the quality, the models were trained more until achieve decent results.

After 30 epochs, the models were tested. Some of the answers were:

Ubuntu Dialogue Corpus Model

Human	Hi
Chatbot	I need help
Human	Are you conscious?
Chatbot	Yes
Human	How are you?
Chatbot	I'm not sure.
Human	What is your name ?
Chatbot	I'm not sure
Human	Are you alive ?
Chatbot	Yes
Human	Nice!
Chatbot	:)

Open Subtitles Corpus Model

[1] Number of iterations through the whole data set.

Human Hi
Chatbot I' m the king
Human Are you conscious?
Chatbot No.
Human How are you?
Chatbot I' m fine.
Human What is your name ?
Chatbot I' m not.
Human Are you alive ?
Chatbot Yes.
Human Nice!
Chatbot I' m sorry.

The results were quite good as a first approach although Open Subtitles model seemed to perform better with less training epochs. At that point, it was decided to use the Open Subtitles data set as a standard for the three models given that allowed the model to maintain a more generic conversation.

5.2 Comparison between Baseline model and +Attention model

In order to establish a standard test to compare the models, it has been used the same test set used by [12] in their experiments,composed by 200 random sentences[2].

Both models were trained using all the sentences from the data set as a question for their next sentence and as an answer for their previous sentence.

A first evaluation of the two models consisted of passing the test set of 200 sentences to both and compare the answers, for each question, by which sentence were the most convincing. If there was a tie or none of them had a good answer then it was marked tie. The +Attention model scored 86 points while the Baseline only 26, there was 89 ties. These results gave an idea of how the attention mechanism improved the performance of the Seq2Seq chatbot with large sentences.

Here are some examples of the answers:

Baseline

Human what is two plus two ?
Chatbot I don' t know.
Human what is your name ?
Chatbot I' m not.
Human how old are you ?
Chatbot Eighteen.
Human my name is david. what is my name?
Chatbot I' m sorry.

[2]The test set can be found at the experimental results of [12] in `http://ai.stanford.edu/~quocle/QAresults.pdf`

+Attention

Human	what is two plus two ?
Chatbot	Two hundred dollars.
Human	what is your name ?
Chatbot	Rusesabagina
Human	how old are you ?
Chatbot	Pardon me.
Human	my name is david. what is my name?
Chatbot	Porter.

Although the answers are not quite correct, it is possible to notice that the +Attention model is able to identify when an answer requires to be a number, an age or even a name.

5.3 Transformer model using the small version of the Open Subtitles Corpus

After having implemented and tested two models for the SeqSeq and +Attention, a Transformer model was implemented. After a first training, the model shown a really bad performance by almost always answering "I don't know" or "I'm not sure". The model was then trained more times but results improved very little. At that time, it was discussed to change the data set which, as previously mentioned, is a small version of the Open Subtitles data set to the full corpus. With the new corpus it was expected to achieve better results.

5.4 Final Experiments with full Open Subtitles Corpus

For the final experiments the Baseline, +Attention, and Transformer models were trained using the full English Open Subtitles Corpus. As it has been explained in 4.2, the pre-processed corpus has 331,595,588 sentences which is too large for the training process.

For that reason, the training process was divided by pre-training the models with a small set of the corpus and go increasing the number of sentences in posterior training. The pre-training was performed using the first 5 million sentences of the data set. All models were configured with a maximum vocabulary size of 72,827 words and a maximum sentence length of 24 words as explained in 4.2. Although the maximum vocabulary size has changed for the sub data set, in order to make possible train again the models with a different sub set, the models were configured with the values computed for the full pre-processed corpus.

All the models were trained using all sentences as questions for their next sentence and as an answer for the previous one.

For the evaluation it was establish a new criterion, ranking the answers of each model by if they are good, neutral or bad. The neutral label is for those answers which are not completely correct neither incorrect. For instance, when a model does not know what to answer says "I don't know" or "I'm not sure" which depending on the context can be valid answers.

Given that the evaluation depends on the subjectivity of the evaluator, it was asked to two persons to rank the answers. From their rankings, an agreement is computed to know how many sentences they have rank alike. Results are shown in table 5.1. Given that the purpose of chatbots is to model language and not apply some logic at their answers, the evaluators were required to rank positively an answer if the type of answer matches with the question but the information is not correct (for instance, ask the result of an operation to the model and answer with a number but not the correct one).

System	Ranking	Evaluator 1	Evaluator 2	Agreement
RNN	Good	46	49	43
+Attention		56	56	46
Transformer		83	94	77
RNN	Bad	140	145	136
+Attention		62	98	78
Transformer		87	30	24
RNN	Neutral	14	5	3
+Attention		62	45	40
Transformer		87	75	68

Table 5.1: Manual Evaluation of the models trained with 5 million sentences

Some examples of the results on different styles of conversations: basic, philosophical, numerical and general knowledge.

Conversation 1: Basic

Context	My name is Mary Johnson. What is my name?
RNN	I'm sorry
+Attention	Jinha.
Transformer	Mary.

Context	What time is it ?
RNN	I'm sorry
+Attention	8:15
Transformer	It's after 10:00 .

Conversation 2: Philosophical

Context	Why are you here ?
RNN	I'm sorry
+Attention	What do you want?
Transformer	I'm meeting someone called Natalie.

Context	What is the purpose of this conversation ?
RNN	I'm not a little
+Attention	I'm not going to be a man.
Transformer	I don't know.

Conversation 3: Numerical

Context	How old were you when you were in school ?
RNN	I'm not sure.
+Attention	Novalee
Transformer	Six years.

Context	i have two apples, Paul gives me two oranges, how many apples do i have?
RNN	No.
+Attention	I haven't been here since i came to you.
Transformer	One.

Conversation 4: General knowledge

Context	ronald reagan is ...
RNN	I'm not a child.
+Attention	Yes
Transformer	He's a friend of mine.

Context	The Pacific ocean is ...
RNN	I'm not gonna be a little more.
+Attention	The ocean.
Transformer	If you want to give me a hand, I'll die.

The agreement between both evaluators shows that the Transformer model performs better than the recurrence-based ones.

After the evaluation, the models were trained again using another 15 million different sentences from the full corpus, being trained in total with 20 million sentences. In this case it was asked to three evaluators to rank the chatbot performances, results are shown in table 5.2.

System	Ranking	Evaluator 1	Evaluator 2	Evaluator 3	Agreement
RNN	Good	41	47	54	32
+Attention		61	51	71	43
Transformer		74	57	70	51
RNN	Bad	75	123	53	46
+Attention		90	116	57	46
Transformer		25	110	25	20
RNN	Neutral	84	30	93	15
+Attention		49	33	72	6
Transformer		101	33	105	18

Table 5.2: Manual Evaluation of the models trained with 20 million sentences

At the second training, the Baseline and +Attention models improved significantly in comparison to the pre trained models. From the agreement between evaluators the transformer architecture is good or neutral in 34.5% of the questions whilst the baseline plus attention is good or neutral in 24.5% of the questions. Finally, with the worst rank, the baseline architecture

is good or neutral in 23.5% of the questions. Results show clearly that the models with attention have a better performance than the model used in [12] based exclusively in RNN. Furthermore, the transformer model (a model based exclusively in attention) performs better than the +Attention model (RNN model with attention).

Again, some examples of the results on different styles of conversations: basic, philosophical, numerical and general knowledge.

Conversation 1: Basic

Context	what is your name ?
RNN	Anouk.
+Attention	Adriano faustino sotolongo.
Transformer	i don't know .

Context	what time is it ?
RNN	I do n't know.
+Attention	6:30.
Transformer	i don't know.

Conversation 2: Philosophical

Context	I think you are a machine
RNN	I'm sorry
+Attention	Vlanne:
Transformer	I'm not a machine .

Context	Life is sad .
RNN	I'm not a monster.
+Attention	I do n't want to go to jail.
Transformer	It's not sad .

Context	Is there a god ?
RNN	No.
+Attention	Surprisingly, yes.
Transformer	no .

Conversation 3: Personality

Context	How old were you when you were able to speak ?
RNN	Eighteen.
+Attention	Seventeen.
Transformer	six months.

Conversation 4: General knowledge

Context	ronald reagan is ...
RNN	I'm not a good person.
+Attention	And i'm not a spy..
Transformer	he's in the hospital.

Context	Is five plus five equal to ten ?
RNN	No, no.
+Attention	That's exactly what i'm gon na do.
Transformer	yes .

Chapter 6

Conclusions and Further Research

Chatbots are systems with the ability to model natural language and establish a conversation with a user through a question/answer protocol. There are three types of approaches depending on the freedom they have at the time of answering: rule-based, retrieval-based and generative-based. The two first approaches are the most used nowadays due to its effectiveness at the time of maintaining a close-domain conversation.

The generative-based models, on the other hand, arise as a powerful alternative in the sense that they can handle better an open topic conversation. They are very related to the idea of strong AI, no human intervention at the time of answering, everything is learned by the machine.

Promising results have been achieved in generative-based chatbot models by applying neural translation techniques with RNN encoder/decoder architectures. In this thesis, it has been shown that chatbot models based on encoder/decoder architectures using exclusively attention outperforms RNN models.

It is important to mention that all models shown in this project shape and mimic natural human language but do not apply any logic to their answers. That is why most of the answers are not coherent between them and the final model lacks of a "personality". Encoder/decoder architectures are a very interesting approach to solve the last problem. Between the encoder and decoder a logical block could be added. Then, once an input sentence has been encoded, apply some reasoning to compute what it is intended to answer. Finally, all the information is passed to the decoder which models an answer in natural language to describe what the logical block has reasoned.

As direct further research of this thesis, the author and his supervisors are organizing a hackathon competition in the *4 Years from Now* Conference 2018 (Barcelona). The objective of this hackathon is to build a multilingual chatbot model based on a modified version of the transformer architecture with an additional intermediate block that will allow to separate the translation modelling part from the conversational one.

Chapter 7

Appendix

This appendix contains the paper [3] currently under review at the recognized international conference of CICLING 2018.

Experimental research on encoder-decoder architectures with attention for chatbots

Marta R. Costa-jussà, Álvaro Nuez, and Carlos Segura*

TALP Research Center - Universitat Politècnica de Catalunya, Barcelona
* Telefònica I+D, Barcelona
marta.ruiz@upc.edu, alvaronuez.eis@gmail.com, carlos.seguraperales@telefonica.com

Abstract. Chatbots aim at automatically offering a conversation between a human and a computer. While there is a long track of research in rule-based and information retrieval-based approaches, the generation-based approach is quite recent and can be dramatically improved by adapting recent advances in close areas as machine translation. In this paper, we offer an experimental view of how alternative encoder-decoder deep learning architectures perform in the context of chatbots. Our research concludes that a fully attention-based architecture is able to dramatically outperform the recurrent neural network baseline system.

Keywords: Chatbot, Encoder-Decoder, Attention Mechanisms

1 Introduction

A chatbot stands for the short version of *chat* plus *robot* and it is a computer program that conducts a human-machine conversation in any topic.

One of the very first chatbots was rule-based. It was proposed in 1966 by Joseph Weizenbaum's program ELIZA [11]. Input sentences were analyzed using several predefined decomposition rules, and after that key words were used to generate responses to them. The Artificial Intelligence Markup Language (AIML) is an evolution of these first rule-based chatbots. This AIML follows the idea of defining written *patterns* and the corresponding *templates* which are responses to the patterns. Then, in inference, if the robot identifies a pattern in a sentence from a user, the robot is able to reply taking the corresponding template [9]. To reduce the amount of work that developing these patterns and templates requires, alternative chatbots, no longer rule-based, but retrieval-based were proposed. These systems use different dialogue databases to train an information retrieval system [2]. The big advantage of these retrieval-based systems is that their training requires little human dedication. However, these systems still rely on giving the most appropriate response from a set of sentences. Thanks to the emergent deep learning techniques, the novel generative-based approaches have arisen offering chatbots that are capable, for the first time, to respond to non-predefined sentences. First successful approach is based on the popular

encoder-decoder architecture implemented with recurrent neural networks [8] and inspired by previous work in machine translation [5, 3].

The main contribution of this paper is the application of the experimentation of attention-based mechanisms [1, 7] to chatbots. Taking [8] as starting point, we compare the encoder-decoder architecture with attention [1] and the transformer [7]. A manually performed evaluation shows that the latter is able to outperfom the encoder-decoder with attention which is already better than the encoder-decoder baseline architecture.

The rest of the paper is organized as follows. Section 2 briefly introduces the deep learning architectures used in this work which basically are encoder-decoder based on recurrent neural networks (with or without attention mechanism) and the transformer which uses a fully attention-based encoder-decoder without recurrent neural networks. Section 3 details the experimental framework, particularly, data statistics and parameters from systems; and also reports a description of the manual evaluation. Section 4 discusses insights of results and contributions of this study.

2 Encoder-decoder architectures

An autoencoder is a type of neural network that aims at learning a representation of the input while allowing for a decoding of this representation by minimizing the recovering error. A generalization of this architecture is the encoder-decoder which allows for input and outputs to be different. This architecture has emerged as an effective paradigm for dealing with variable-length inputs and outputs and much more than NLP applications, it has been extended to image and speech processing applications [8, 10]. In this section, we briefly provide a high-level description of two successful implementations applied to MT that we are adapting and testing for chatbots.

2.1 RNNs with attention

One successful implementation of the encoder-decoder in natural language processing has been the recent concatenation of recurrent neural networks (RNNs) [5, 3]. In fact, this architecture builds on top of RNN language models [4] by adding an encoder step and a decoder step. In the encoder step, a RNN converts an input sequence into a fixed representation (called thought vector). This representation is fed in the RNN from the decoder step which allows the decoder model to output a more intelligent predictions given the context from the encoding. While this implementation has shown some results in chatbots [8], the main drawback is that long sequences are not well codified into a single vector. To deal with this problem, [1] propose the attention mechanism, which allows the decoder to put different amounts of attention on the encoder states. The main idea is to iteratively train a context vector by using a weighted average of the encoder states and learning the weights by means of a multilayer perceptron. Equations and details can be found in the original paper [1].

2.2 Transformer

While previous architecture has been successfully applied to MT, there are still some issues to solve. The architecture in practice can be really slow to train and given the way RNNs deal with sequences, it is not easy to parallelize the algorithm and take advantage of recent computational resources such as Tensor Processing Units (TPUs). Motivated by this issue, Google team proposed the Transformer model in [7] which has been proven to be competitive in the task of machine translation. The Transformer model is able to improve state-of-the-art results in a couple of academic benchmarks while speeding up training by an order of magnitude in comparison to RNN-based sequence-to-sequence approaches.

The Transformer architecture is basically an encoder-decoder which concatenates attention-based mechanisms allowing to model relationships between words without requiring recurrence. More specifically, the encoder (enc) is composed of three stages: (enc1) input embedding and positional encoding; (enc2) multi-head attention; and (enc3) feed-forward layer. The decoder (dec) is composed of 5 stages: (dec1) input embedding and positional encoding; (dec2) masked multi-head attention; (dec3) multi-head attention; (dec4) feed-forward layer; and (dec5) softmax layer. Both (enc1) and (enc2) are standard word embeddings combined with positional encoding. The latter allows to record information about word position. The multi-head attention module is composed by different submodules of scaled dot product attentions with different linear projections. The single scaled dot product attention is a variation of the attention proposed by Bahdanau et al. [1] with less parameters to train (the multilayer perceptron is changed to the dot product). While a single scaled dot product attention computes how words in the sequences are relevant to each others, linear projections of several dot product attentions allows to jointly attend information from different representations. The masked multi-head attention forces to attend only to past words making training similar to inference. The attentions are either self-attentions or standard. Self-attention means that the attention is performed over the same sequence, while standard is performed from target sequence to source. The feed-forward layers are a linear transformation and they allow the model for further adaptability and learning capacity. Finally, the softmax layer allows to map target word scores into target word probabilities. More details about the architecture can be found in the original paper [7].

3 Experimental framework and evaluation

3.1 Data and preprocessing

Models were tested on the OpenSubtitles dataset [6]. This dataset consists of subtitles from movies conversations, which is open-domain since movies come from broad scopes.

The subtitles do not contain identity nor turn information. Therefore, similarly to [8], we assumed that consecutive sentences were uttered by different

characters. We constructed a dataset consisting in pairs of consecutive utterances, using every sentence twice as context and as target. Due to computing and memory constrains, we extracted a subset of the first 10 million sentences for training using each sentence as context and as target. Therefore, we end up training with 20 million sentences for context and targets. Preprocessing of the database consisted on removing XML tags, limiting the sentence size and removing strange symbols (e.g. #). Details on training and evaluation split are reported on Table 1.

Table 1. Size of the parallel corpora

Set	Role	Words	Vocab	
Training	Context/Target	20,000,000	131,708,317	349,603
Evaluation	Context	200	1,446	399

3.2 Parameters

In order to ensure that we cover the 99 % of the dataset, we have limited the vocabulary size to 72,827 words and the length of sentences to 24 words. All words that are used only once are discarded. All three models have a 64 dense size for the embedding matrix.

The recurrent architecture for both models (with and without attention), have 2 LSTM layers with 512 units. Additionally, the model with attention uses 512 softmax samples. For the training, we used ADAM with a learning rate of 0.002, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$; a batch size of 256 and a dropout rate of 0.1.

The transformer architecture has 8 attention heads and 6 hidden layers with 512 units. For the training we have used ADAM with a learning rate of 0.2, $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 10^{-9}$; a batch size of 4096 and a dropout and attention dropout rates of 0.1.

For better responses we have added to the three models a beam search with a beam size of 4 and 0.6 of alpha.

3.3 Evaluation

In order to compare our three conversational models, we asked three evaluators to rank chatbot responses among: good, bad or neutral. Results are shown in table 2 showing ranking from Evaluator 1, Evaluator 2, Evaluator 3 and agreement between them. Results show clear improvement of the attention mechanisms over the baseline system. In fact, all evaluators agree that the transformer architecture is good or neutral in 34.5% of the questions whereas the baseline architecture is only good or neutral in 23.5% of the questions and the baseline plus attention is good or neutral in 24.5% of the questions.

Table 2. Manual Evaluation

System	Ranking	Evaluator 1	Evaluator 2	Evaluator 3	Agreement
RNN	Good	41	47	54	32
+Attention		61	51	71	43
Transformer		74	57	70	51
RNN	Bad	75	123	53	46
+Attention		90	116	57	46
Transformer		25	110	25	20
RNN	Neutral	84	30	93	15
+Attention		49	33	72	6
Transformer		101	33	105	18

In addition to the results, we report several examples on different styles of conversations present in the test set: basic, philosophical, numerical and general knowledge.

Conversation 1: Basic

Context	what is your name ?
RNN	Anouk.
+Attention	Adriano faustino sotolongo.
Transformer	i don't know .
Context	what time is it ?
RNN	I do n't know.
+Attention	6:30.
Transformer	i don't know.

Conversation 2: Philosophical

Context	I think you are a machine
RNN	I'm sorry
+Attention	Vlanne:
Transformer	I'm not a machine .
Context	Life is sad .
RNN	I'm not a monster.
+Attention	I do n't want to go to jail.
Transformer	It's not sad .
Context	Is there a god ?
RNN	No.
+Attention	Surprisingly, yes.
Transformer	no .

Conversation 3: Personality

Context	How old were you when you were able to speak ?
RNN	Eighteen.
+Attention	Seventeen.
Transformer	six months.

Conversation 4: General knowledge

Context	ronald reagan is ...
RNN	I'm not a good person.
+Attention	And i'm not a spy..
Transformer	he's in the hospital.
Context	Is five plus five equal to ten ?
RNN	No, no.
+Attention	That's exactly what i'm gon na do.
Transformer	yes .

4 Conclusions

Attention-based mechanisms are revolutionizing natural language, speech and image processing applications. In this paper, we are implementing a couple of recently proposed attention mechanisms into the chatbot application. Experiments trained on a open-domain database show that a fully attention-based architecture performs significantly better in a variety of contexts including basic, philosophical, personality and general knowledge. Three evaluators agreed on rating the fully attention-based mechanism 34.5% of the time either good or neutral, while the responses in the baseline system where only 23.5% of the time either good or neutral.

Taking advantage of this generic encoder-decoder architecture, among further research, we plan to introduce further contexts while training the system so as to allow the system to keep coherence in longer dialogues and to train our system on multiple languages.

Acknowledgments. This study has been funded by the Spanish Ministerio de Economía y Competitividad and the European Regional Development Fund, through the postdoctoral senior grant Ramón y Cajal and the contract TEC2015-69266-P (MINECO/FEDER,EU).

References

1. Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.
2. Rafael E. Banchs and Haizhou Li. IRIS: a chat-oriented dialogue system based on the vector space model. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the System Demonstrations, July 10, 2012, Jeju Island, Korea*, pages 37–42, 2012.

3. Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734, 2014.
4. Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, pages 1045–1048, 2010.
5. Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112, 2014.
6. Jörg Tiedemann. News from opus : A collection of multilingual parallel corpora with tools and interfaces. In *Recent Advances in Natural Language Processing V*, volume V, pages 237–248. John Benjamins, 2009.
7. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 6000–6010, 2017.
8. Oriol Vinyals and Quoc V. Le. A neural conversational model. *CoRR*, abs/1506.05869, 2015.
9. Richard Wallace. The elements of aiml style, 2003.
10. Ron J. Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Zhifeng Chen. Sequence-to-sequence models can directly transcribe foreign speech. *CoRR*, abs/1703.08581, 2017.
11. Joseph Weizenbaum. Eliza: a computer program for the study of natural language communication between man and machine. *Commun. ACM*, 9(1):36–45, January 1966.

Bibliography

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

[2] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014.

[3] Marta R. Costa-Jussà, Carlos Segura Perales, and Álvaro Nuez. Experimental research on encoder-decoder architectures with attention for chatbots.

[4] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, 2017.

[5] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[6] Lukasz Kaiser, Aidan N. Gomez, Noam Shazeer, Ashish Vaswani, Niki Parmar, Llion Jones, and Jakob Uszkoreit. One model to learn them all. *CoRR*, abs/1706.05137, 2017.

[7] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[8] Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. *CoRR*, abs/1506.08909, 2015.

[9] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215, 2014.

[10] Jörg Tiedemann. News from OPUS - A collection of multilingual parallel corpora with tools and interfaces. In N. Nicolov, K. Bontcheva, G. Angelova, and R. Mitkov, editors, *Recent Advances in Natural Language Processing*, volume V, pages 237–248. John Benjamins, Amsterdam/Philadelphia, Borovets, Bulgaria, 2009.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[12] Oriol Vinyals and Quoc V. Le. A neural conversational model. *CoRR*, abs/1506.05869, 2015.

[13] Joseph Weizenbaum. Eliza: a computer program for the study of natural language communication between man and machine. *Commun. ACM*, 9(1):36–45, January 1966.