MSE | MASTER OF SCIENCE IN ENGINEERING

Hes·so
Haute Ecole Spécialisée
de Suisse occidentale
Fachhochschule Westschweiz
University of Applied Sciences and Arts
Western Switzerland

Master of Science HES-SO in Engineering
Av. de Provence 6
CH-1007 Lausanne

Master of Science HES-SO in Engineering

Orientation: Information and Communication Technologies (ICT)

GraphQA, a Deep Retrieval Chatbot

A Multi-hop Conversational Question-Answering Chatbot using Sub-Knowledge Graphs

Author:

Romain Claret

Under the direction of:
Prof. Dr. Jean Hennebert
HES-SO//Fribourg
Institute of Complex Systems (iCoSys)

External expert:
Prof. Dr. Michael Ignaz Schumacher
HES-SO//Valais
Institute of Business Information Systems

Fribourg, HES-SO//Master, February 7, 2020

Accepted by the HES-SO//Master (Switzerland, Lausanne) on a proposal from:

Prof. Dr. Jean Hennebert, Master's Thesis Supervisor

Place, date: _____

Prof. Dr. Jean Hennebert	M. Philippe Joye
Supervisor	ICT MRU Leader at HES-SO//Fribourg

To my family that believed in me, and still I don't know why.

Contents

Contents

Acknowledgments

I wish I could thank an AGI for doing my thesis, meanwhile here follows my acknowledgments.

People

Jean Hennebert for his advice and the opportunity to work on a meaningful subject to me.

Fiona Baumann For her endless support and proofreading.

Damien Goetschi For his proofreading.

Aymeric Genêt For his support and proofreading.

Jämes Ménétrey For his advice and support.

Lucy Linder For many interesting discussions and her advice.

Luana Martelli For her support and proofreading.

Riccardo Formenti For his advice and support.

The whole iCoSys team For the social interaction and the incredible ambiance.

My open-space To all my colleagues that supported me and helped me stay sane during this project.

My closes friends Julia, Séverine, Jeff, Thomas, Simon, Fabián, Diane, and Marc for their support.

My parents For their support.

Glossary

Adversarial Learning

In Machine Learning (ML), the concept of this technique relies on trying to fool models via malicious inputs. Interpretable as a game, a model plays with itself and modifies the input in such a way that the model will recognize the input as a different one, and then learn from its mistake. 12

Attention Mechanism

In Natural Language Processing (NLP), the Attention Mechanism is an algorithm used to compute the relational weight between elements in a sequence of elements e.g., words. xiii, 22, 51

Bidirectional Encoder Representations from Transformers

In Natural Language Processing (NLP), *Google* BERT is a large Transformer-based model trained at predicting masked tokens within sequences. xix, 19

Bidirectional Language Model

In Natural Language Processing (NLP), a Bidirectional Language Model represents a Language Model (LM) combining the forward pass and the backward pass of the same corpora. 19

BLEU

In Natural Language Processing (NLP), The Bilingual Evaluation Understudy (BLEU) is an evaluation metrics particularly popular in machine translation as the processing is automatic. 28, 34, 35, 39

Close-ended

A closed-ended question is designed to allow a limited amount of responses. 14

Encoder-Decoder

In Machine Learning (ML), Encoder-Decoder are two Neural Networks (NNs) that work in pair. The Encoder generates a fixed-size output vector from any sized input vector, and the Decoder generates from the Encoder's output a vector that could be of any size. xii, 11, 19

F1

In Statistics, the F1 score is used to compute an accuracy metric, using Precision and Recall.

$$2 * ((P * R)/(P + R))$$

. 62

Few-Shot Learning

In Machine Learning (ML), Few-Shot Learning is a technique used to solve tasks with a very small amount of training data. 3

Generative

In the context of the thesis, we are using the generic word “Generative” as the concept of an algorithm able to generate outputs in a meaningful but unpredictable manner from an input, which includes Language Model (LM)s and Generative Models. 4, 40, 68, 69, 72, 90

Generative Model

In Machine Learning (ML), Generative Models generate random outputs from a single input by using the probability of observing the output based on the input. In other words, it models the probability of observation for a given target. x

Generative Pre-Training 2

In Natural Language Processing (NLP), *Open-AI* GPT-2 is a large Generative Model using Transformers to generate outputs based on the probability of the token observation. xix, 19

Ground Learning

In the context of the Artificial Intelligence (AI), Grounded Learning is based on the Grounded theory from the social sciences, which uses inductive reasoning. The mechanism combines structured and unstructured data as small conceptual parts to then apply machine reasoning inductively. 4, 13, 47

Hop

In Question-Answering (QA) Systems, a Hop is a quantitative measure of the number of supporting facts or combinations necessary between indirectly related pieces of information to provide an answer. xi, xii

Knowledge Base

In Information Systems (IS), a Knowledge Base is a Knowledge Representation using a Linked Data database for storing and interlinking structured and unstructured data using a standard. x, xiii, xix

Knowledge Graph

In Information Systems (IS), a Knowledge Graph is a Knowledge Base (KB) organized as a graph using semantics. xix, 45

Language Model

In Natural Language Processing (NLP), a Language Model is a Model trained to provide likelihood probabilities of a following sequence of words in addition to providing the probability for each sequences of words. ix, x, xix, 3, 12, 19

Linked Data

In Information Systems (IS), Linked Data is a structured interlinked database mainly used for semantic queries. x, 42

Machine Reasoning

In Machine Learning (ML), Machine Reasoning represent the ability to apply reasoning for a given input by using knowledge representations and logic patterns such as inductions, analogies, or abductions. 4, 81

Machine Understanding

In Machine Learning (ML), we use Machine Understanding as the ability to represent knowledge as atomic building blocks and fundamental relations. 4, 81

Markov Decision Process

In the context of the Reinforcement Learning (RL), this process models the ability to predict the next state of a finite-state-machine-like process, such as a game, using only the information contained in the present state. xii, 13

Mean Reciprocal Rank

In Information Retrieval (IR), the Mean Reciprocal Rank provides a statistic measure of the quality of a returned ranked list of items for a query. MRR takes into account only the highest-ranked relevant item to the query. 33

Model

In Machine Learning (ML), a Model is the representation of the assumptions made by the algorithm during the training phase. Models are used to output a result based on a provided input and the learned patterns. x, xi

Model Fine-Tuning

In Machine Learning (ML), Fine-Tuning a Model is the technique of using a trained Neural Network (NN) model as a base and tune it for a specific task. xix, 12, 15, 19, 69

Multi-Hop

In Question-Answering (QA) Systems, a Multi-Hop implies that the answer is within multiple Hop of the question. In other words, the answer requires a combination of different pieces of information to be answerable. Generally, extra qualifying Subject-Predicate-Object Tuples (SPOs) are separating the question Subject and the answer Object. xix, 14, 22, 25–27, 33, 40–42, 45, 47, 57, 75, 81

Named-Entity Linking

In Natural Language Processing (NLP), Named-Entity Linking extends the Named-Entity Recognition by providing a unique identifier to each word allowing a mapping in various databases (useful in translations). xix, 26, 27, 50, 56, 58

Named-Entity Recognition

In Information Extraction (IE), Named-Entity Recognition is a technique used to extract, from unstructured texts, words from a predefined vocabulary. xi, xix, 19, 43, 50, 51, 53–56, 77

Open Domain

In Information Retrieval (IR), the support of Open Domain questions provides no restriction to the subject of the questions. 14, 25

Oracle

In Machine Learning (ML), an Oracle is defined as an entity that knows the ground truth to all questions. An Oracle can be a human, or an algorithm querying a database with no errors. 34, 77

Part-of-Speech

In Natural Language Processing (NLP), Part-of-Speech is a technique used to categorize words that behave syntactically similarly. xii, 50, 56

Part-of-Speech Tagging

In Natural Language Processing (NLP), The Part-of-Speech Tagging is extending the Part-of-Speech by adding a label to the word depending on its context (the neighboring words). xix, 19, 50, 56

Reinforcement Learning

In Machine Learning (ML), Reinforcement Learning combines a Markov Decision Process (MDP) environment with an approach similar to Unsupervised Learning (UL) as this type of learning does not require labelled data. The particularity of this technique is that it uses a notion of rewards to predict the best next step by running a large amount of simulations as training. xi, 13

Sequence-to-Sequence

In Machine Learning (ML), a Sequence-to-Sequence or Seq2Seq is an Encoder-Decoder Neural Network that, for a given sequence of elements as input, outputs another sequence of elements. xiii, 11

Shallow Neural Network

In Machine Learning (ML), similar to Deep Learning (DL), Shallow Neural Networks have a Encoder-Decoder approach by having a single hidden layer, which often has a large amount of parameters. 18

Single-Hop

In Question-Answering (QA) Systems, a Single-Hop implies that the answer is within a single Hop of the question. Generally, a unique predicate separates the question subject and the answer object. xix, 41, 42, 57

Supervised Learning

In Machine Learning (ML), this type of learning implies the uses of labelled datasets to perform the training. 11

Transformer

In Natural Language Processing (NLP), Transformers are similar to Sequence-to-Sequence (Seq2Seq) architectures but use a parallelized Attention Mechanism. ix, x, xix, 22, 26, 33, 34, 40, 44, 57, 81

Unsupervised Learning

In Machine Learning (ML), Unsupervised Learning implies the use of unlabelled datasets to perform the training. xii, 11, 17

Wikidata

Wikidata is a community-based Knowledge Base (KB), based on Freebase originally. It stores its data into a linked-data format with Subject-Predicate-Object Tuples (SPOs). 39–44, 47, 48, 50–52, 58, 62, 68, 77

Word Embedding

In Natural Language Processing (NLP), the Word Embedding is a technique for word representation as vectors in an embedding matrix. Additionally, Word Embedding has often the particularity of preserving the semantical analogies of word-vectors. xix, 18, 19, 56, 75, 79

Zero-Shot Learning

In Machine Learning (ML), Zero-Shot Learning is a technique used to solve tasks without training on examples. 3, 4, 47, 48, 61, 81

Acronyms

AGI

Artificial General Intelligence. vii, 1–4

AI

Artificial Intelligence. x, 1–3, 7, 8, 68, 89

AIML

Artificial Intelligence Markup Language. 8

AL

Adversarial Learning. 12, 40, *Glossary:* Adversarial Learning

ANI

Artificial Narrow Intelligence. 1, 3

ANN

Artificial Neural Networks. 8

AT

Adversarial Training. 34

BERT

Bidirectional Encoder Representations from Transformers. xix, 19, 34, 40, 42, 55, 58, *Glossary:* Bidirectional Encoder Representations from Transformers

biLM

Bidirectional Language Model. 19, *Glossary:* Bidirectional Language Model

CE

Character Embedding. 18, 19

CNN

Convolutional Neural Network. 19, 21, 22

CoQa

Conversational Question Answering. 26, 27, 33, 34

CWE

Context-based Word Embedding. 19

Acronyms

DL

Deep Learning. xii, 9, 17, 18, 21, 56

DNN

Deep Neural Networks. 8, 18

FAQ

Frequently Asked Questions. 15

GAN

Generative Adversarial Networks. 12

GLUE

General Language Understanding Evaluation. 35

GPT-2

Generative Pre-Training 2. xix, 19, 23, 27, 42, 55, *Glossary:* Generative Pre-Training 2

GS

Generative System. 67, 68

ICT

Information and Communications Technologies. i

IE

Information Extraction. xii, xix

IR

Information Retrieval. xi, xii, xix, 42, 79

IS

Information Systems. x, xi

KB

Knowledge Base. x, xiii, xix, 25, 26, 33, 39–44, 47, 48, 50–55, 62, 68, *Glossary:* Knowledge Base

KD

Knowledge Distillation. 34

KG

Knowledge Graph. xix, 45, 81, *Glossary:* Knowledge Graph

LM

Language Model. ix, x, xix, 3, 12, 14, 19, 25, 42, *Glossary:* Language Model

LSTM

Long Short-Term Memory. 34

MDP

Markov Decision Process. xii, 13, *Glossary:* Markov Decision Process

ML

Machine Learning. ix–xiii, xix, 3, 7, 8, 13, 14, 17, 22, 81

MN

Memory Network. 22

MR

Machine Reasonsing. 4, 13, 26, 81, *Glossary:* Machine Reasoning

MRR

Mean Reciprocal Rank. 33, 34, 62, *Glossary:* Mean Reciprocal Rank

MRU

Master Research Units. i

MT

Master's Thesis. 4, 70, 71

MU

Machine Understanding. 4, 13, 35, 81, *Glossary:* Machine Understanding

NL

Natural Language. v, 8, 9, 17, 19, 27, 33, 34, 39, 40, 42, 43, 47, 48, 54–56, 61, 62, 64, 76, 81

NLG

Natural Language Generation. 19

NLP

Natural Language Processing. ix–xiii, xix, 1, 3, 4, 7, 8, 11–14, 16–19, 21–23, 25–27, 33–35, 39–42, 47, 48, 50, 56, 57, 67, 68, 70, 71, 75, 76, 78, 81

NLU

Natural Language Understanding. 4, 19, 70, 71

NN

Neural Network. ix, xi

OOV

Out-of-Vocabulary. 18

POC

Proof of Concept. 1, 3, 4, 39, 69, 70, 75, 81

Acronyms

QA

Question-Answering. x–xii, xix, 1–4, 12, 14, 19, 21, 22, 25–27, 33, 39–42, 44, 45, 47, 48, 51, 55, 61, 62, 67–72, 75, 77, 78, 81, 89, 90

QuAC

Question Answering in Context. 27, 40, 44

RL

Reinforcement Learning. xi, 13, *Glossary:* Reinforcement Learning

RNN

Recurrent Neural Network. 19, 21, 22, 34

Seq2Seq

Sequence-to-Sequence. xiii, 11, 19, *Glossary:* Sequence-to-Sequence

SL

Supervised Learning. 11, 18, *Glossary:* Supervised Learning

SNN

Shallow Neural Network. 18, *Glossary:* Shallow Neural Network

SOTA

State of the Art. 3, 4, 16, 17, 19, 25, 33, 39, 41, 42, 48, 61, 67, 68, 70, 71, 75, 78, 81

SPO

Subject-Predicate-Object Tuple. 22, 40, 48, 52, 53, 55

SQuAD

Stanford Question Answering Dataset. 26, 27, 33, 39–41, 75

UL

Unsupervised Learning. xii, 11, 12, 17, 19, *Glossary:* Unsupervised Learning

Weak AI

Weak Artificial Intelligence. *Glossary:* Weak Artificial Intelligences

Abstract

We propose an innovative approach for question-answering chatbots to handle conversational contexts and generate natural language sentences as answers. In addition to the ability to answer open-domain questions, our zero-shot learning approach, which uses a pure algorithmic orchestration in a grounded learning manner, provides a modular architecture to swap statically or dynamically task-oriented models while preserving its independence to training.

In the scope of this research, we realize the proof-of-concept of an open-domain and closed-ended question-answering chatbot able to output comprehensive natural language generated sentences using the Wikidata knowledge base.

To achieve the concept, we explore the extraction, and the use of sub-knowledge graphs from the Wikidata knowledge base to answer questions conversationally and to use the sub-graphs as context holder. Additionally, we are extracting subject-predicate-object tuples from the graph and using language models to join the SPOs and extend the answers as natural language sentences.

The proof-of-concept architecture uses a combination of state-of-the-art and industry-used models with a fine-tuning strategy. As a motivational target, we use a zero-shot learning approach, by combining various models with an algorithmic orchestrator and using pure algorithmic for the graph manipulation and answer extraction.

Finally, we evaluate the answers and compare the results with state-of-the-art single-hop and multi-hop question-answering systems on question-answering datasets. We find out that, aside from the computation time and the computational resources needed, our proof-of-concept performs similarly at question-answering compared to its competitors.

Keywords: Machine Learning (ML), Natural Language Processing (NLP), Single-Hop, Multi-Hop, Question-Answering (QA), Wikidata, Wikipedia, Knowledge Graph (KG), Knowledge Base (KB), Word Embedding, Part-of-Speech Tagging, Named-Entity Recognition, Named-Entity Linking, Language Model (LM), Model Fine-Tuning, Graphs, Sub-Knowledge Graphs, Transformer, Bidirectional Encoder Representations from Transformers (BERT), Generative Pre-Training 2 (GPT-2), Information Extraction (IE), Spacy, GloVe, DeepCorrect, Chatbot, Conversational, Information Retrieval (IR), Queries, Python

How to read this document

This Master's Thesis takes a Storytelling approach while reporting the work done. The document is split into four major parts and is intended to be read in continuous as the previously mentioned notions are often required. To help the reader, we built a glossary, which we recommend to read as a preliminary work to not get overwhelmed.

Project preface

Here, we introduce the project at a high-level and the motivations we have towards it.

State-of-the-art

In this part, we perform state-of-the-art research for Chatbots and Natural Language Processing using a Question-Answering.

Design and realization

In this part, the reader follows us in our state-of-the-art analysis and our adventure at building our research contribution with a proof-of-concept.

Retrospective

Finally, we review our academic journey by taking a step back and commenting on our work.

Part I

Project preface

Chapter 1

Introduction

New technologies are revolutionizing the way humans access knowledge as a service from multiple platforms and providers. Thanks to the emergence of increasingly powerful AI algorithms, particularly in the field of Natural Language Processing (NLP), conversational agents, commonly named chatbots, have come a long way and have become popular among information consumers. As in early 2020, chatbots are all still ANIs[1]. Even though the *chatbots* are continually improving at providing the best outputs for specific tasks as well as providing meaningful human-like sentences, they still cannot generalize the tasks toward human-like conversations. The task of conversation, as humans are applying it, a complex integration of tasks including understanding, reasoning, context linking, context tracking, curiosity, initiatives, Few-Shot Learning or Zero-Shot Learning and on-the-fly learning, has yet to be accomplished. Nonetheless, as research progresses, chatbots are improving with new technics and tools that make them step by step closer to complete human-like discussions, slowly progressing toward AGI chatbots. As for the scope of the thesis, we humbly focus on the combination of few NLP tasks with a Zero-Shot Learning approach to help Machine Learning (ML) and NLP research getting closer to General QA Conversational Chatbots.

1.1 Aim of the Research

The initial goal of the thesis was to explore and combine State of the Art (SOTA) QA Systems and Language Models (LMs) to into an experimental POC of a Conversational QA Chatbots.

During our research journey, we discovered a new purpose to the project, and took a step into the unknown with a Zero-Shot Learning approach with sub-knowledge graphs.

1.1.1 Project's Overall Scope

We focus on the English language as an attempt to increase the number of compatible datasets and make community-accessible solutions. We explore and combine two types of systems as an attempt to build QA chatbots. The first system produces factual answers, and the second system generates human-like sentences from the answers found by the primary system. For the factual answers, we will evaluate the

[1] The State of AI Report 2019 (Benaich et al., 2019)

results of our combined system against SOTA QA systems on QA testing datasets. Humans manually evaluate the answered sentences from our combined system. Finally, as the time allocated for the thesis is 19 weeks, the outcomes are narrowed to providing non-exhaustive research and a POC solution. On a side note, the review of the risks and ethical problems that could be raised by the development of such solutions are not part of this work.

1.1.2 Industrial Interest

iCoSys, the Institut of Complex Systems at the University of Applied Sciences and Arts at Fribourg, Switzerland, is interested in the results of this study for their *AI-News* project[2]. iCoSys' goal is to provide a chatbot-based system as a tool for press readers, to help them narrow their interests and deliver the right information. This project is in collaboration with the *Swiss Innovation Agency* from the Swiss Confederation, *La Liberté*, the daily newspaper from Fribourg and *Djebots*, a startup selling scenario-based narrow chatbots.

1.1.3 Personal Interest

In harmony with the subject of the thesis, as the author is particularly interested in exploring the premises to AGI related technologies such as Zero-Shot Learning, Ground Learning, Machine Understanding, and Machine Reasoning for a Multi-Domain Task Generalization. The human-like QA frame of this project is particularly motivational, as it shows a glance of what the future of NLP when combined with Machine Reasoning (MR) and Machine Understanding (MU).

1.2 Research Questions

We articulate here the initial set of questions as a driver to our research work. From these questions are declined objectives, and from objectives are declined milestones framing the plan.

- What are the components to make QA chatbots?
  - What is the SOTA of chatbots and QA systems?
  - How to tune QA chatbots to make them as human-like as possible?
  - How to tune such systems for the field of journalism?
- What is the SOTA for Generative QA chatbots?
  - What are the components to make Generative QA chatbots?
  - Are Generative chatbots only as good as the data they consume?
  - Could Generative chatbots be a step toward AGI?

[2] `AINews.ch`

Part II

State of the art

Chapter 2

Chatbots

Based on the latest MMC's state of AI report[1], it appears that 26% of the AI Startups studied by Gartner[2] are using or making chatbots (see Figure 2.1). The same study, made a year earlier, in 2018, shows that chatbots are not present as an application, which implies that either chatbots were not referenced as AI or that their popularity exploded within a year.

As currently in the beginning of 2020, based on The State of AI Report 2019 (Benaich et al., 2019) and the two previously mentioned studies, chatbots are commonly present but limited to narrow tasks. In most cases, they are scenario-based with sequences of if-else conditions that we classify as non-learning AI. Moreover, hard-coded scenarios are requiring an infinite amount of human power to create generic Chatbots able to maintain a conversation at a human level. However, the progress in the field of ML and NLP is demonstrating that providing large corpora to an unsupervised algorithm is enough to maintain a passive conversation with users, which results into a shifting of the human power into data engineering. Increasingly complex algorithms and techniques are emerging at a monthly rate in the field, demonstrating a trend toward conversational performance improvements. Note that even though chatbots are getting better at providing meaningful sentences, current Chatbots are still not able to orchestrate the generalization of all the tasks required to a human-like conversation. E.g., understanding and reasoning based on the context, initiatives to search and learn for missing information, initiate dialogue in a meaningful manner, intuition, and much more. As a side note, the generalization of those tasks would reduce the steps significantly toward general Chatbots.

From a user-centric point of view, chatbots are currently trending and rising global interest for various reasons. Big companies such as *Google* or *Apple* believe in the technology and are making a lot of effort at pushing the chatbots into the mainstream. Even if the word "chatbot" is commonly used as a buzzword without a proper definition, people have at least a mental representation of its concept. Indeed, whether called "Digital Assistant", "Siri", "ok Google" or "Alexa", they all expect to a certain extent human-like conversations after using those triggering keywords.

[1] The State of AI 2019: Divergence (Kelnar, 2019)

[2] 2'791 European AI Startups from the 2019 CIO Survey: CIOs Have Awoken to the Importance of AI (Rowsell-Jones et al., 2019)

Interestingly, note that the majority of the following sections could be included in the field of AI in general. The extrapolation of the chatbot subject to AI as a whole is worth further studying, but not in this work. Instead, the focus of this chapter is placed on chatbots; we provide a synthesis and classification of the different methods used to build chatbots. We will define the main categories identified and continue on the main sub-categories and conclude with a cartographical chart of our chatbot vision.

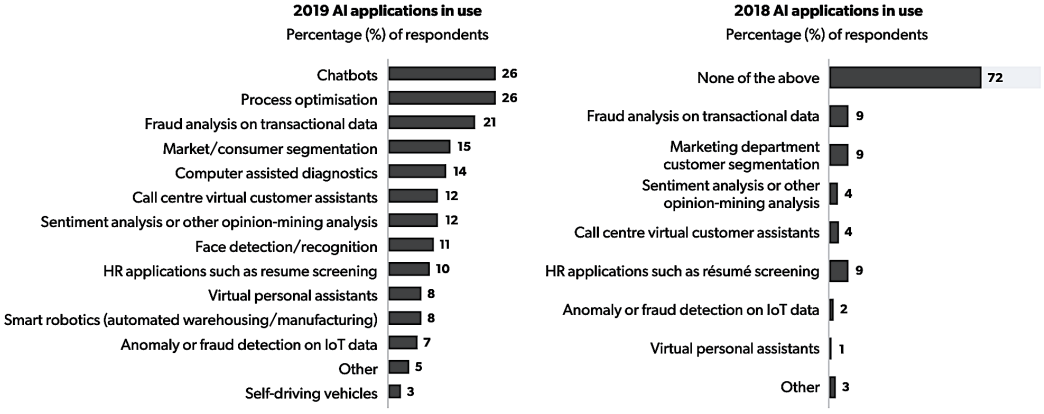


Figure 2.1: Figure 31 from *The State of AI 2019: Divergence (Kelnar, 2019)*. The top AI applications used in European AI Startup in 2019 are Chatbots and Process optimization.

2.1 Chatbot History

Not mentioning *Alan Turing* or *Joseph Weizenbaum*, both considered as the fathers of AI and chatbots, would not be fair to this research. Indeed, in 1950 these two intellectuals forecasted human-like communication with computers and proposed a test to differentiate humans from machines, the Turing Test (Turing, 1950). The test performs as follows: a supervisor asks a human to talk to a masked entity and determine rather the subject is talking to a human or a computer. If the human cannot recognize speaking to a computer, then the machine passes the Turing test.

In 1966, *Joseph Weizenbaum* wrote Eliza (Dunlop, 1999), a computer program simulating a psychotherapist, seen today as one of the first well-documented attempts to make a chatbot designed to pass the Turing test. However, due to techniqueal restrictions, Eliza was not performing particularly well in all contexts. As of today, it is still possible to play with the chatbot on a dedicated website.

Since Eliza, a lot of progress has been made until 2020; from conditional IF-ELSE, Artificial Intelligence Markup Language (AIML), up to ML with Artificial Neural Networks (ANN) and Deep Neural Networks (DNN), the improvements in the field of chatbots increased drastically over the years. Each iterations delivering algorithms being continuously more sophisticated and better at using the NL, resulting in a new field of ML called NLP. As a reminder of the chatbots history

and progress from 1966 to 2016, the infographic (Futurism, 2016) from Futurism is particularly speaking, as it illustrates the timeline of impactful chatbots with their features.

2.2 Main Categories in the Chatbot Realm

While performing the state-of-the-art, we identified three main chatbots categories.

2.2.1 Conversational

We like to call them the Chatty bots. They are great for interaction and structured replies, well designed for their ability to talk. E.g., *User*: “Hello, how are you?”, *Bot*: “Good, what about you?”.

2.2.2 Task-Oriented

The Task-Oriented bots are performing particularly well at specific tasks as smart-assistants. As their design is not toward generalization, their abilities are limited and will fail at off-tasks. A common workflow used by those bots is to detect the Intent and the Entities of the user request, (often in NL), then apply a rule-based matching to perform the command intended by the user. E.g., *User*: “Book the next flight to Geneva from Zürich.”, *Bot*: “Alright! Your ticket number is 00XXYYZZ. Have a great flight!”

2.2.3 Dispatcher

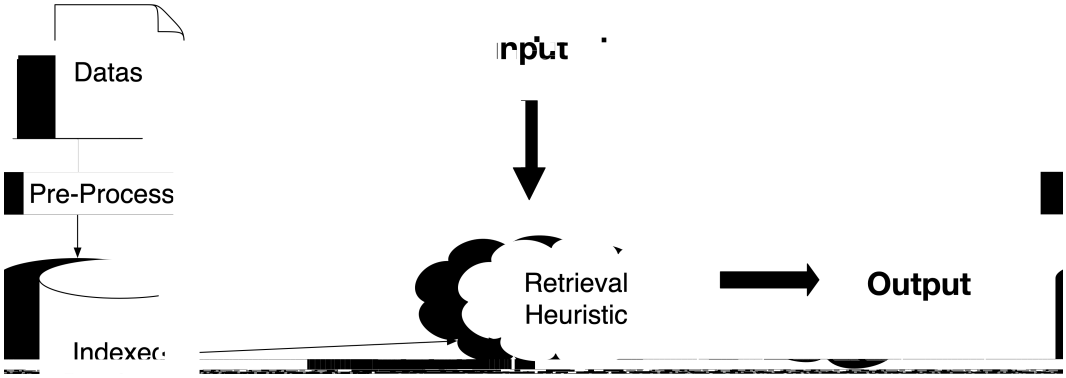


Figure 2.2: Illustrative representation of frequent retrieval chatbots architecture.

2.4 Rule-Based Chatbots

Scenario-based, as we name it, is the oldest and relatively straightforward system for chatbots. The ElizaDunlop, 1999 Chatbot, as mentioned in the Chatbot History (see chapter 2.1), is scanning the input text for keywords, calculates a ranking for each keyword, and finally goes through a series of conditions called rules to reach the best ending leaf. Usually, the bot also includes a default output if the matching process fails, which we can still nowadays see in chatbots (e.g., “Hmm, this is interesting, tell me more.”). Such bots are often used as interactive chatbots, as they can, in a controlled environment, give a sense of deep meaning in the context of the conversation. Note that such systems require a lot of human power to build a frame for the bot to play in, and by this mean makes rule-based chatbots great for the specific scenario but is particularly hard to generalize. See Figure 2.3.

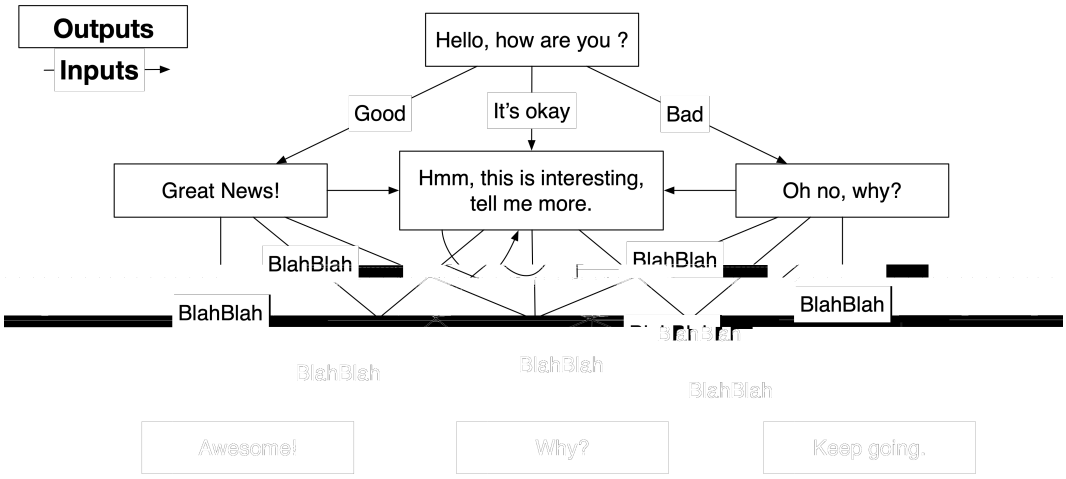


Figure 2.3: Illustrative representation of frequent rule-based chatbots process.

2.5 Generative Chatbots

As the current result of all the incredible innovations made in the past years in NLP, and as a premise to true conversational chatbots, generative methods are overcoming the limitations of the Retrieval (see chapter 2.3) and Rule-Based (see chapter 2.4) chatbots, by its ability to generate new content. In Supervised 2.5.1, Unsupervised Learning (UL) or Adversarial 2.5.2, no pre-defined outputs are used, the models are trained on large corpora to learn the language patterns and output relatively meaningful responses to given inputs.

Another particularity of generative chatbots is that building a domain-oriented chatbot does not require the engineers to have the domain expertise, as the expertise is embedded into the data, which allows relative scalability to new domains. However, even though the trained models can output responses at nearly no timespan, the data-engineering of the datasets and the training phase is most often long and complicated. As a final note, the responses generated by such chatbots are only as good as the data it was fed during the training.

2.5.1 Supervised Learning

Supervised Learning (SL) is probably the most common method used by Generative Chatbots, as it provides relative control over training. Sequence-to-Sequence (Seq2Seq) is commonly used as architecture for those chatbots, a NLP version of the Encoder-Decoder, which encodes the input words sequence and decodes it into a words sequence as an answer into a framed conversation fashion. The training only requires a dataset containing a sentence and its desired response, the model will then map similar inputs with similar outputs. However, a clear limitation for this learning is that the model will for any input always have an answer, regardless of the overall meaning. Additionally, Seq2Seq will prioritize the highest word apparition probabilities, meaning that data duplicate and requiring sentences will create a trend during decoding. E.g., “I don’t know the answer.”. See Figure 2.4

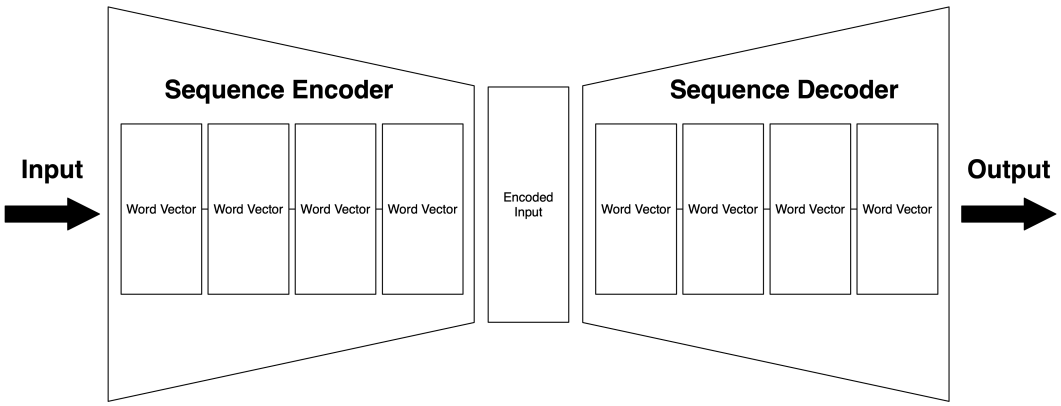


Figure 2.4: Illustrative representation of a Sequence to Sequence architecture.

2.5.2 Adversarial Learning

Adversarial Learning (AL) has driven attention thanks to Computer Vision Generative Adversarial Networks (GAN) (Karras et al., 2019) by proving that it is possible to generate realistic human faces (Wang, 2019). In the chatbots context, it can be extrapolated into a futuristic version of the Turing Test 2.1, in which machines are confronting themselves instead of humans. The concept implies the use of a training dataset containing human conversations, and compare them against the generated answer; the discriminator will then judge which is from a human and which is from an algorithm. Note that adversarial methods such as GAN are working well because of the nature of the data it plays with; indeed, pixels can be deeply noised, but words cannot be due to their discrete nature. See Figure 2.5

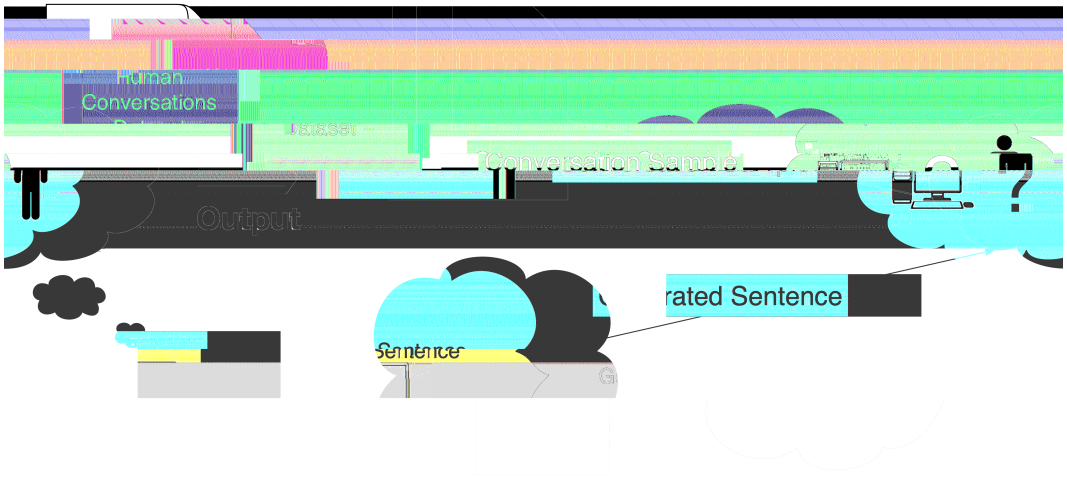


Figure 2.5: Illustrative representation of an adversarial architecture in a chatbot context.

2.5.3 Pre-trained Language Models

Language Models are currently the most recent and the most promising models due to their ability to model language itself instead of conversations and then tune the outputs as a chatbot would. It can been seen as semi-supervised learning, as it uses UL for training and supervised learning 2.5.1 for fine-tuning 2.5.4. We will dive into LM in the NLP chapter 3.3.

2.5.4 Model Fine-Tuning

With Model Fine-Tuning (see Figure 2.6), LM have, by design, the ability to be enhanced to perform particularly a various NLP task such as chatbots. Because pre-trained Language Models (LMs) are based on the grounded blocks of language itself, implying model post-training customization as a light learning task. Indeed, it is relatively easy to fine-tune a QA dataset to a LM, making the model able to answer questions instead of descriptively filling sentences. The main downside to those models is the large memory size required to run them. However, due to their nature, they are trained once and then fine-tuned. Note that training requires an enormous amount of computational power. E.g, The largest form of BERT (Devlin

et al., 2019) was trained on 16 TPUs for 4 days. Fine-tuning, on the other hand, scales down to few hours on a single TPU, which makes it relatively scalable to new domains.

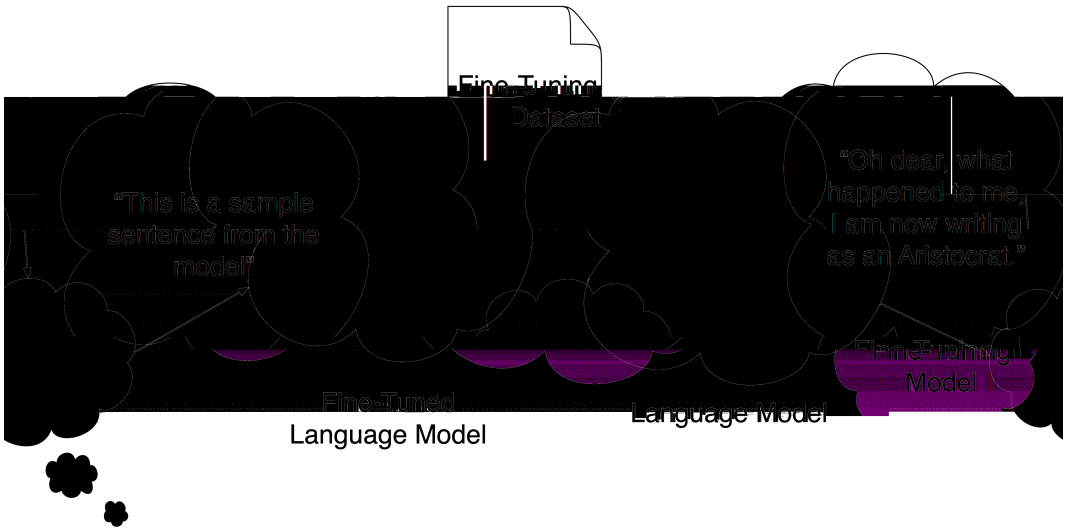


Figure 2.6: Illustrative representation of fine-tuning in a chatbot context.

2.5.5 Reinforcement Learning

Reinforcement Learning (RL) is proven to be very powerful by the latest research made by *Open-AI* with its DOTA2 bot or *Google's Deepmind* with AlphaZero, so we believe that it is worth mentioning it. However, this type of learning requires a finite state similar to a Markov Decision Process (MDP), which matches game cases but not conversations, and impacting by this means the motivation to export the technique to NLP. Indeed, this methodology requires that all information required for the next step are wrapped into a single state to predict it, which makes it hard to use in the dialogue case. For now, NLP research does not provide a conclusion as, even with billions of simulations, RL Chatbots could reach comparative results to Generative Chatbots 2.5.

2.6 Grounded Chatbots

Falling in a particularly rare research field of ML and NLP, Ground Learning can be seen as the future of MU and MR. In a chatbot context, the goal is to simulate, based on the Grounded Theory from the social sciences, how humans are using inductive reasoning to create conversations with unstructured knowledge. The idea is to give the ability to the bot, for any given input, to gather information from any data sources and provide an inductive output. E.g., Combining Knowledge Bases with weather forecaster. As a second example, for the given input: "What is the color in autumn of a leaf in Switzerland?", 1) the bot would have first to identify the context keywords (color, leaf, autumn, switzerland), 2) the bot would select where to gather the information, 3) the bot would investigate the Wikidata Knowledge Base, Wikipedia, and The Weather Channel API, 4) the bot would formulate an answer based on the information it gathered. 2.7

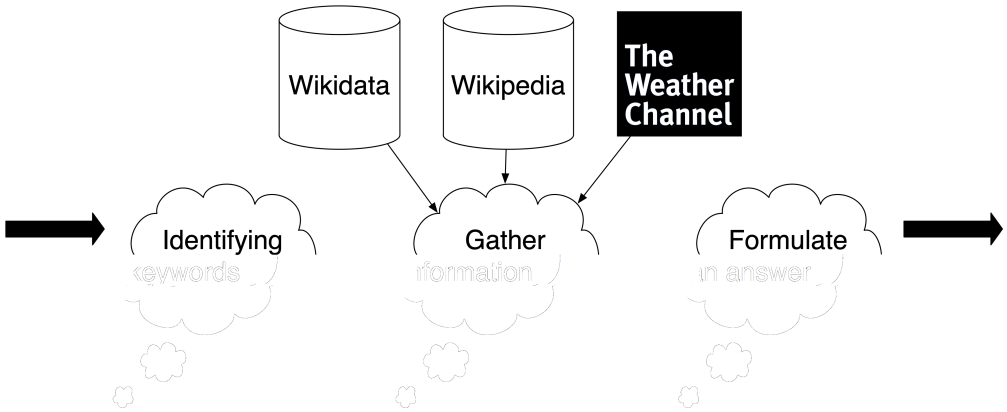


Figure 2.7: Illustrative representation of a grounded chatbot.

2.7 Question-Answering Chatbots

QA is a prevalent task for chatbots; indeed, they are widely used for questioning tasks in either Single or Open Domain, Open or Close-ended, Single or Multi-Hop with applications such as FAQs, Supports, help to find the meaning of life, and so on. Due to the broadness of the field, no defined methodology has been generalized; instead, it uses either one or multiple techniques described in the previous sections. It is interesting to note that the field of QA is raising a lot of interest in NLP research lately, and the benchmarking game of creating the new baselines, with increasingly complex datasets, is still in progress. In this section, we will overview some recent baselines.

Fine-Tunning Language Models Large LM such as BERT (Devlin et al., 2019) or GPT-2 (Radford et al., 2018) are often fine-tuned on QA datasets similar to SQUAD 2.0 (Rajpurkar et al., 2018a) which are particularly tricky, even for humans.

Querying Models Based on QA datasets, a model is trained to fill structured templates. The generated output is a structured query for a particular querying language such as SPARQL for Wikidata.

Retrieval A popular approach in the industry is to use tools such as Elasticsearch for indexing and additional tools using ML heuristics to perform the queries.

2.8 Common Chatbot Features Overview

In this section, we are non-exhaustively naming a few recurring features appearing during our targeted research.

2.8.1 Context

Humans are intuitively and extensively relying on the context for conversational purposes, which implies similar capacities from dialogue-based chatbots. In the scope of the Thesis, we are also using the term Multi-Turns Conversations to mention context holding implicitly. On a side note, one-way style dialogues such as commands or none-nested questions do not need to keep context to perform well.

Short term context Implying the ability for the bot to hold context for at least the current conversation, e.g., few keywords or on-the-fly Model Fine-Tuning.

Long term context Often, chatbots would use user-profiles as part of their architecture to remember information such as the favorite pizza flavor of a client.

2.8.2 Proactivity

Simulating personalized interest as a human would do is not new to chatbots, as it has been proven by becoming a standard in marketing and customer support chatbots. Messages such as “Hey, you have been on our web store for a while, can I help you?”, are carrying a sense of proactivity; however, beyond asking general pre-made questions, limitations are clear, and not much progress has been made yet in the field. Indeed, human-like proactive chatbots imply algorithms capable of initiating conversations by initiating a dialogue or asking information in a meaningful manner based on the long and short term context.

2.8.3 Narrow vs General Chatbots Scope

Beyond the three main categories 2.2 identified during the study, in general, chatbots can additionally be classified within a scope starting at Narrow Chatbots up to General Chatbots. To position them, we defined a two axes classification using Tasks and Knowledge as represented on Table 2.1.

Tasks Axis To name a few examples of task-oriented Chatbots: Talk, Frequently Asked Questions (FAQ), Customer Support, or Ordering.

Knowledge Axis Non-exhaustively, as follows, a few knowledge-centric examples for chatbots: Health, Weather, or Customer Service.

Narrow Chatbots Narrow chatbots are limited by the range of tasks they can accomplish and the knowledge they can use. By design, they are very good at a particular task for a particular knowledge requirement.

General Chatbots They are neither limited by the range of tasks they can accomplish nor by the knowledge they can use. However, they often have an average performance for any task or knowledge. We go in more details at section 2.8.4.

Tasks \ Knowledge		
	Expert in a specific Field Expert at all Tasks	**General Chatbots** Expert in all Fields Expert at all Tasks
	Narrow Chatbots Expert in a specific Field Expert at specific Task	Expert in all Fields Expert at specific Task

Table 2.1: This table represents categories in Narrow and General Chatbots in a Tasks versus Knowledge format.

2.8.4 General Chatbots

As research progresses in the NLP field, chatbots are improving as an effort to perform simultaneously well in various tasks and multi-knowledge bases. As a contemporary goal, in addition to any chatbot related tasks and broad knowledge expertise, General Chatbots must not be limited to their current capabilities, but on the contrary, be able to learn new tasks and subjects continuously. As far as this study went, we could not find SOTA general chatbots as defined. However, companies like *Amazon* are selling to a large public a feel to general chatbots with Alexa. Indeed, apart from ordering goodies from *Amazon* and roughly conversing with Alexa, users can command their smart homes, use it as a personal assistant, or even program *skills* to perform custom actions.

2.9 Chatbots Cartography

As a result of this chapter, we created a chart on Figure 2.8 representing the current state of chatbots from our point of view. Note that a particular use-case could be in multiple leafs.

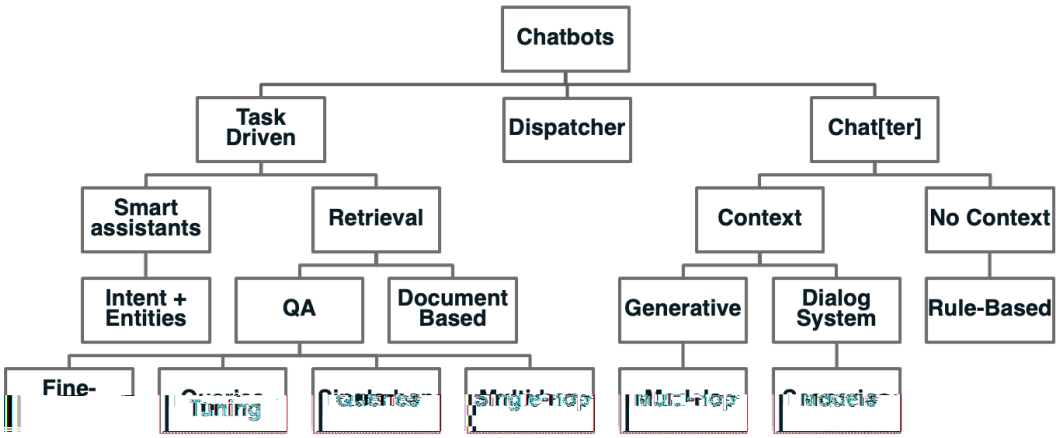


Figure 2.8: Represents the chatbots cartography as conclusion to the chatbot state-of-the-art chapter.

Chapter 3

Natural Language Processing

It is often challenging to realize the complexity behind Natural Language (NL), even to experts. First of all, Language is an academic field of study, implying multi-disciplinary skills. And secondly, staying up to date with evergrowing tools and new SOTA algorithms proves to be challenging. NL is the fundamental communication element for humans, NLP is the field of ML studying NL with the goal of providing the ability to machines to handle and mimic NL to create human-like verbal interactions. Beyond words and grammar rules, NL is a complex orchestration of subtleties, intuitively handled by humans, but not easily handled by machines. Nonetheless, NLPs technologies are massively used in our daily lives, including information extraction, summarization, and conversation simulation. However, even if machines are given the same language rules as humans, they do not yet understand the manipulation they are processing, as humans would do. Indeed, NLP algorithms are applying pre-defined or multiple examples-based learned rules, which may result in ambiguities while applying NL. Using a rule-based approach (as seen in Chapter 2.4) to build a NL model would result into near to infinite amount of conditions, this is the main reason for NLP to be particularly present in ML, particularly in DL.

3.1 Word Embeddings

The technique is commonly used as the first data pre-processing for DL in NLP tasks. Those Unsupervised Learning (UL) algorithms capture syntactical and semantical words representation from large unlabelled corpora datasets as vectors by building a multi-dimensional matrix. On average, dimensions are held in a scope of 100 to 400, and thanks to the vectorized nature of captured words, geometrical operations can be applied, such as the cosine functions to calculate word similarities. Another feature related to word embeddings is the ability to apply analogical operations such as *'king' - 'man' + 'woman' = 'queen'*, which popularize Word2Vec 3.1.1 and gave credits to the method, even if the justification to this effect has been theorized 4 years later [1] by stating that the compositionality is only seen when assumptions are held, in particular when words are uniformly distributed in the embedding space.

[1] Skip-Gram - Zipf + Uniform = Vector Additivity (Gittens et al., 2017)

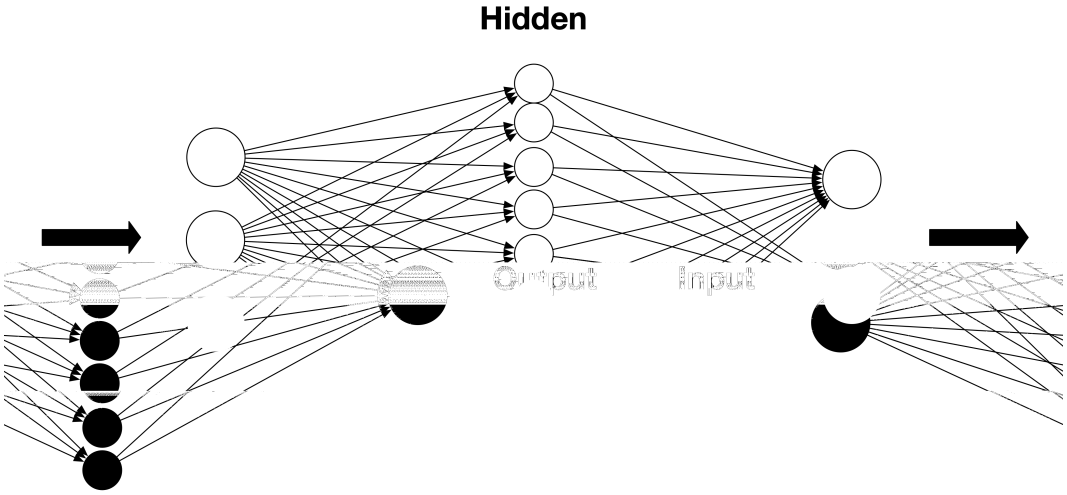


Figure 3.1: Illustrative representation of a Shallow Neural Network

3.1.1 Word2Vec and GloVe

Published by *Google* in 2013, Word2Vec (Mikolov et al., 2013), and its competitor GloVe (Pennington et al., 2014) published by the *University of Standford* in 2014, both use a Shallow Neural Network (SNN), as illustrated on Figure 3.1, similarly to SL by feeding as input a text corpora, and outputting word-vectors with a given vocabulary. Training and testing are straightforward but painful tweaking makes it hard to build good generalized word embedding representations. Even if the SNN could remind a DL approach, it has only one hidden layer; however, the output word-vectors are particularly useful for DNN as input.

3.1.2 Out of Vocabulary Problem

A common issue in Word Embedding is related to the vocabulary itself when words are unknown, called the Out-of-Vocabulary (OOV) issue. The issue occurs when post-training the model is requested to provide a vector representation that it has never seen before. A solution could be to handle the exception by forwarding it to a default or pre-defined error vectors such as a series of zeros. We could approach the problem sophisticatedly, by defining on-the-fly OOV words with at a high learning rate as the sum of word-vectors contextualizing the OOV (Herbelot et al., 2017). Another solution would be to fallback to 3.2 by either training a model to compositional map characters to words (Pinter et al., 2017), or using Character Embedding (CE) as a whole instead of Word Embedding 3.2.

3.2 Character Embeddings

Additionally to Word Embedding similar abilities to capture semantics and syntactic relations, CE handles by design OOV issues 3.1.2, which is common for rich vocabularies languages. Instead of using words as vocabulary, CE uses individual characters and semantics embeds words using the characters compositionally, which avoids word segmentation and makes it useful for languages such as Chinese (Chen et al., 2015). Moreover, CE can also perform complementary NLP

tasks such as Part-of-Speech Tagging (Santos et al., 2014), Named-Entity Recognition (Ma et al., 2016), Sentiment Analysis (Hao et al., 2017) and LM (Kim et al., 2015). As it is at the time of writing, *FastText* based on a morphologically-rich skip-gram approach (Bojanowski et al., 2016) has been popularized due to its ability to be scalabely trained on large corpora fast, and effectively.

3.3 Language Models

Beyond complex semantics and syntaxes provided by Word Embedding 3.1 and CE 3.2, Language Models (LMs) handle Context-based Word Embedding (CWE) by additionally capturing the polysemy across multiple contexts. Indeed, it was discovered that a distributed semantic, such as Word Embedding and CE are not sufficient to infer context within the embeddings (Lucy et al., 2017). A solution is to combine overall word representations from Word Embedding with *ELMo* (Peters et al., 2018), as its authors suggest, a Bidirectional Language Model (biLM) able to build deep contextual word embeddings by handling multiple word representations. As mentioned in the study, handling polymesy is just one of the Language Models (LMs) features as they are theoritized to capture meaningful NL traits used in NLU and Natural Language Generation (NLG). To increase the LM quality, defined by language syntactic and semantical complexities captured, UL on large corpora is popularly used, as no labeled data are required.

3.4 Transformers

The year 2017 has set a milestone in NLP, transformers (Vaswani et al., 2017) are since then defining the SOTA for multiple NLP tasks mainly due to its parallelized attention 3.4.1 architecture. Large multi-directional pre-trained LM such as Generative Pre-Training 2 (GPT-2) or the Bidirectional Encoder Representations from Transformers (BERT) family are, additionally to their ability to capture features at sentence level, out-performing by a large margin previously mentioned NLP technics at tasks such as QA by performing Model Fine-Tuning, an adaptation of the very popular Transfer Learning feature from computer vision. Making those new LM currently trendy among NLP researchers and engineers.

3.4.1 Attention Mechanism

Introduced in 2014, The Attention Mechanism (Bahdanau et al., 2014) solved the problem raised by tasks such as text summarization, machine translation, or sentiment analysis, where the input is often too rich to perform a selective encoding. Originally, the last hidden state of the decoder is used by a multi-layer perceptron to define the attention from an input hidden state. The mechanism even got adapted from NLP to Computer Vision and shown its ability to replace Convolutional Neural Network (CNN) with SOTA results (Ramachandran et al., 2019).

3.4.2 The Architecture

Even if Transformers, Figure 3.2, are using a Seq2Seq approach similar to Encoder-Decoder, which reminds of Recurrent Neural Network (RNN) and CNN, the overall architecture focuses on the attention mechanism to capture the relation between

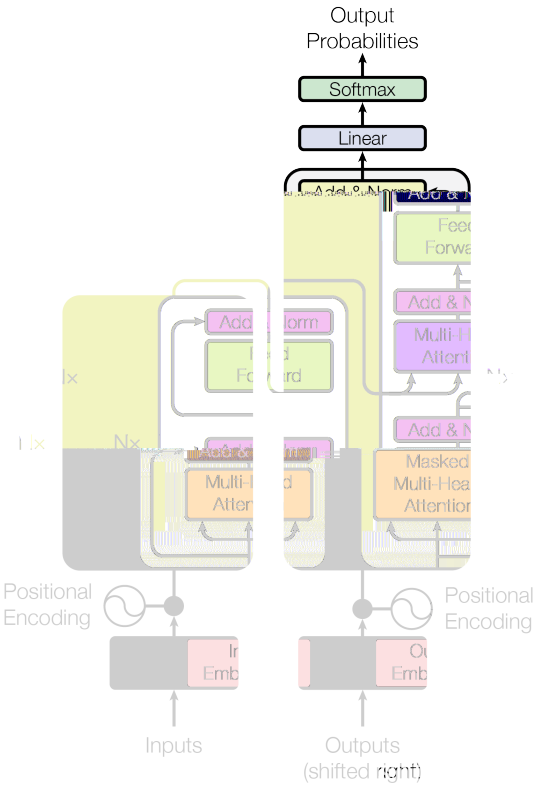


Figure 3.2: Represents the Transformer architecture. Figure 1 from (Vaswani et al., 2017)

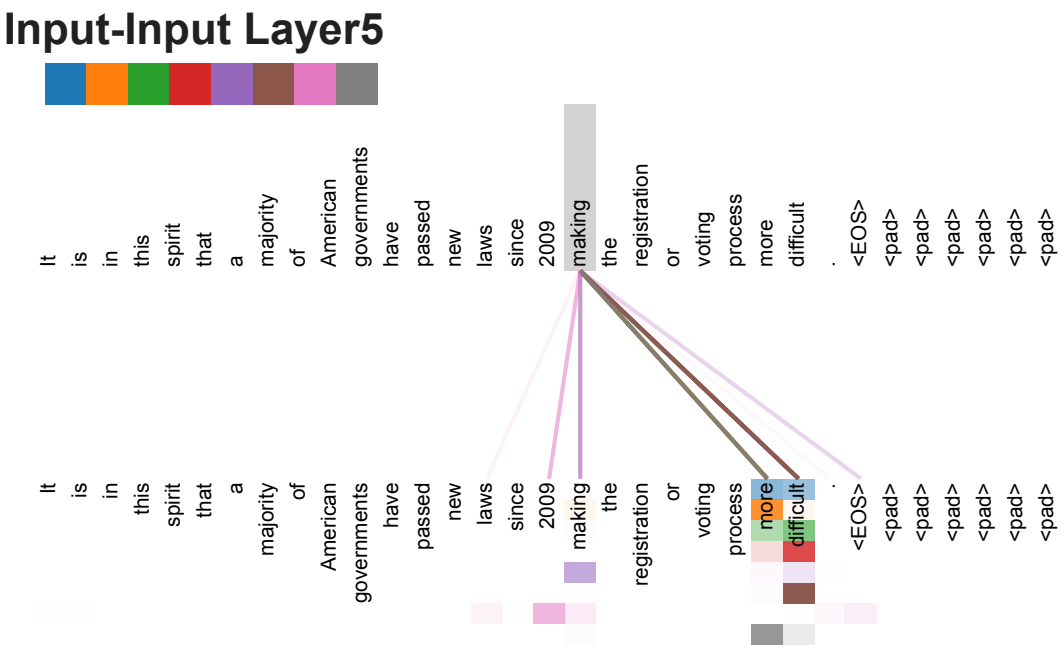
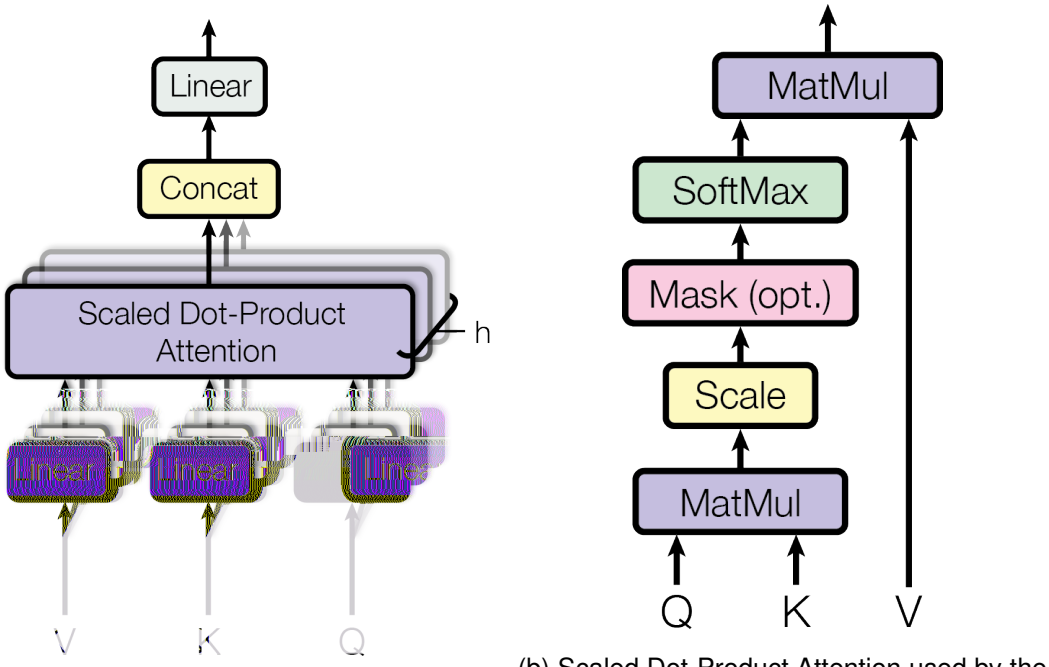


Figure 3.3: Illustrates the attention mechanism for long-distance dependencies handled via multiple attention heads used in transformers. Figure 3 from (Vaswani et al., 2017)

the input and the output, making it well parallelizable and less time consuming during training with its multi-attention heads approach. Multi-heads, Figure 3.4a, uses sets of queries Q, keys K and values V to perform attention with dot-products, Figure 3.4b. In other words, the multi-head attention mechanism builds a multi-dimensional matrix representing each word-vectors with the attention relative to all word-vectors in a predefined window, such as a sentence, then computes the overall attention for each word-vectors. In addition to the attention centric mechanism, transformers are also using proven DL techniques such as layer normalization, dropouts, and positional encodings.



(a) Multi-Head Attention mechanism.

(b) Scaled Dot-Product Attention used by the Multi-Heads Attention 3.4a

Figure 3.4: Multi-head attention anatomy extracted from Figure 2 of *Attention is All you Need* (Vaswani et al., 2017)

3.5 Honorable Mentions

Even if Transformers have deprecated CNN and RNN in NLP by solving their main bottleneck implying the sequential processing during encoding with the Attention Mechanism 3.4.1, we still wanted to mention them, as those techniques have defined baselines at multiple NLP tasks for many years.

3.5.1 Convolutional Neural Networks

Commonly used in sentence modeling thanks to their good ability at mining semantics; however, their models are relatively heavy for the task performed. Additionally, they do not perform well on large windows, resulting in bad context handling for long-distance spread information and order tracking. In the field of QA, interesting approaches has been researched, such as Multi-Column CNN (Dong et al., 2015) able to treat multiple aspects of questions by building compatible representations

with Wikidata's ancestor *Freebase* (Bollacker et al., 2008). In 2016, one of the final promising CNN approach was introduced for QA with a model able to handle relational information by word matching question and answer pairs (Severyn et al., 2016).

3.5.2 Recurrent Neural Networks

By design and compared to CNN, RNNs try to take advantage of their ability to remember previous computations. However, it appears that no clear performance winner at NLP tasks demarks RNN from CNN (Yin et al., 2017); indeed, their parallel performances depends on the global semantics and the task itself. Similarly to CNN, RNN is broadly used for NLP tasks such as Language Modeling, Machine Translation, and Word/Sentence Classification.

3.5.3 Memory Networks

Also named MemNet (Weston et al., 2015), the technique is still actively researched in the field of NLP as it provides an intuitive approach to attention by using Multi-Hop (Tang et al., 2016), and sets the technique as an interesting competitor to Transformers 3.4. As the Attention Mechanism 3.4.1 builds sets of hidden vectors with its encoder, Memory Networks (MNs) uses the hidden vectors as internal memory instead of feeding them to a decoder for token generation. Further in the Transformers competition, MNs can be applied to similar NLP tasks such as QA (Kumar et al., 2015) by extending the *(Representation, Attention, Answer)* tuples to *(Memory, Question, Answer)* tuples. The Figure 3.5 presents a QA Memory Networks based architecture using knowledge base as knowledge source as initial Key-Value Memories provider, Subject-Predicate-Object Tuples (SPOs).

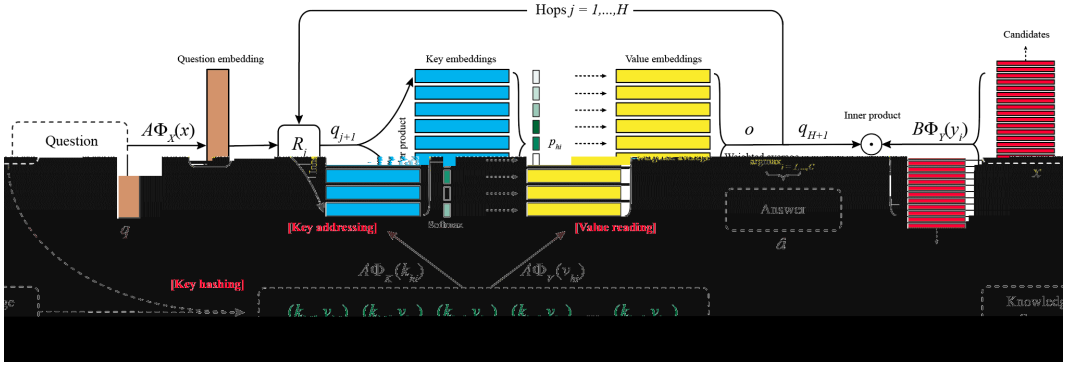


Figure 3.5: Illustrates a Key-Value Memory Network model used in QA. Figure 1 from (Miller et al., 2016)

3.6 Problems

The following will be comments about the current generative algorithms state as we observed, but they may be extrapolated to ML as a whole. Big data is currently starting to make sense in Computer Science as algorithms get incrementally more optimized, complex, and powerful. In addition to continuously improving computational power, it appears that the large amount of data produced for many years can

finally deliver some premise to its potential. However, it raises new questions, such as the privacy implications.

With its impressive 774 Million parameters, GPT-2 uses a large dataset combination of News articles, Reddit comments, IRC conversations, Books, Wikipedia pages, and much more, to train the model, making it humanly impossible to review. This could mean that the model is holding potentially private information: to prove our words, we performed a potential privacy attack on GPT-2 and succeed. Indeed, by merely using meta-information as input, e.g., "[DD/MM/YYYY, HH:MM:SS AM] <USER1>", the model could retrieve a private conversation it once had seen. The method could also recover potential passwords, and so on.

It is essential to repeat that we believe that we are currently at the beginning of a rich potential to NLP, particularly for text generation, as explored in this chapter and confirmed in the next, which implies additional breakthroughs within the incoming year. However, we believe that large models are meant to do good to the world, and must be under a light control to avoid bad actors damaging the image of the upcoming even more impressive technologies.

As a final note, in the scope of our project, we wish not to start a trend toward Sophist Machines, as our study may provide the tools to build algorithms able to manipulate knowledge in the way the author wants.

Chapter 4

Datasets

As the thesis aims at exploring a knowledge-based QA (see Chapter 2.7) and dialogue (see Chapter 2.2) for chatbots combination, this chapter aims at synthesis and compares the current SOTA datasets. With no surprise, we noticed that both NLP research fields are currently in nested competitions, each field playing at defining the new SOTA algorithm and dataset as baselines. In addition to NLP breakthroughs, the competitive attitude makes the community results particularly active and exciting as new techniques, architectures, and incrementally more complex datasets are released at a monthly rate.

4.1 Scope Criteria

The datasets pool base for QA systems and Dialogues may not be as exhaustive as datasets present in other NLP fields such as LM or even in other computer science fields, e.g., Computer Vision, but they still let us dispose of relatively large datasets, lucky to the recent interest in the fields; however, they vary significantly from each other, e.g., based on their features such as Data Sources, Quantity, or Quality. To narrow the research, as many traits are subjective and intrinsically dependent on the required tasks themselves, we defined high priority criterias.

Knowledge-based The use of KB, such as Wikidata, has been defined in the specifications as an ideal knowledge database.

Open Domain The ability to respond to Open Domain question is meaningful to the project as we use a KB containing by design Open Domain relations.

Multiple Supporting Facts As an elegant KB and Open Domain combination, Multi-Hop allows to profit from the KB linked-data architecture to handle more complex questions.

Converstational Support of contexts to handling nested questions is particularly meaningful in our opinion as it provides to QA additional details layers to an answer.

No Reasoning Even if mentioning supporting criteria is essential, we believe that referring to explicitly not supporting criteria is also important. In our case, we ex-

pressly do not support MR in any manner, as reasoning could be a quantifiable task in some datasets.

4.2 Question-Answering

Based on the criterions defined in Chapter 4.1, we made an overview Table 4.1 scoping QA datasets related to our work. The following subsections describe the chosen datasets, ending with the worth mentioning datasets.

4.2.1 ConvQuestions

Late 2019, ConvQuestions, a crowdsourced Multi-Hop datasets of 11'200 augmented questions on 5 domains, released with CONVEX (Christmann et al., 2019). The data augmentation is done by asking the Turker to paraphrase each question once and keeping it semantically equivalent and interchangeable. To the initial 5-turns 350 conversations, a non-reordering permutation is applied to each question and its paraphrasing. Finally, the dataset provides a Wikidata Named-Entity Linking to each answer.

4.2.2 SimpleQuestions casted into Wikidata

Initially built with crowd workers by *Facebook AI Research*, SimpleQuestions(Bordes et al., 2015) was entity-linked to the Freebase KB for its 108'442 (question, answer, language) tuples. For their research, the AskPlatypus (Diefenbach et al., 2017) team used automatically generated mappings of 49'202 SimpleQuestions triples to Wikidata. Note that at the time of building their dataset, only 21'399 were answerable over Wikidata.

4.2.3 Worth Mentioning

Even as part of this work, the opportunity to use the following datasets was not appropriate; we believe that they are worth mentioning due to the attention they attracted lately with fine-tuned pre-trained language models (see Chapter 3.4) QA systems.

Stanford Question Answering Initially presented in 2016, Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al., 2016) was propelled by the raised of Transformers as it was the first massively crowdsourced reading comprehension Wikipedia-based dataset with a 86.6% human performance, making it an exciting challenge for future QA systems. In 2018, SQuAD 2.0 (Rajpurkar et al., 2018b) was released with an additional set of 50'000 unanswerable questions, adversarially similar to answerable questions present in the first version. Note that by design, SQuAD focuses on confusing questions, approachable by combining paraphrasing and text summarization NLP tasks.

Conversational Question Answering Challenge Released in 2019, Conversational Question Answering (CoQa) (Reddy et al., 2018) is a Multi-Hop dataset containing 127'000 questions obtained from 8'000 conversations over 7 difference sources. With a human-performance of 88.8% CoQa implies reading comprehension with coreference and pragmatic reasoning.

Question Answering in Context Since 2018, Question Answering in Context (QuAC) (Choi et al., 2018) is challenging QA system with its Multi-Hop dataset containing 100K crowdsourced evidence-based questions aiming at providing QA in a dialog manner with context holding. Questions are designed to be open-ended, unanswerable without context and focusing on missing information.

Compare SQuAD, CoQa and QuAC In 2019, a comparative study (Yatskar, 2018) have been conducted for the previously mention three datasets, comparing them on the basis of unanswerablility, Multi-Hop, and question abstraction.

4.3 Dialogue Datasets

As the project scope requires a chatbot able to answer NL, we explored the available of conversational datasets. Compared to the QA datasets (see Chapter 4.2), we discovered an underrepresented NLP field. Recapitulated in the Table 4.2, we focused on the datasets featuring on QA setups and multi-domains dialogue openness. We identified a unique dataset matching our requirements and two worth mentioning datasets, reinforcing an overall subjective view implying that dialog-based NLP research currently in standby. Indeed, it would not surprise us, as pre-trained language models have started reaching a popular peak of interest, that the field of Dialogue gains a sudden interest and challenges GPT-2 (Radford et al., 2018).

4.3.1 Natural Questions Corpus

Another jewel of 2019, with over 323'000 dialogues, *Google's* Natural Questions Corpus dataset (Kwiatkowski et al., 2019) is a benchmarking approach for NL generated answers in a QA environment, making it particularly interesting for fine-tuned pre-trained language models like GPT-2 (Radford et al., 2018). Its goal is to provide an appropriate training and testing set for QA systems, by pairing *Google Search Engine*'s real user queries to a large pool of crowdsourced cross-annotations, they call "high quality annotations", to guarantee answer quality over documents. Additionally, their mythology defines new metrics to evaluate answering performances. Interestingly, the dataset provides statistics, a long answer, a short answer, and an answer Named-Entity Linking to a Wikipedia page in most cases.

4.3.2 Honorable Mentions

The following datasets are particularly interesting from a Chatty (see Chapter 2.2) point of view, but no further out-of-the-box features without pre-processing or data augmentation are present. In our case, no Wikidata Named-Entity Linking is available, nor the data are set explicitly in a QA manner, making it particularly random in various contexts. However, the datasets are still impressive by their quantities and their conversational feature.

Twitter Conversation Triple This dataset, form 2015, uses Context-Message-Response triples as storage architecture, making it particularly interesting for parallelized training. Additionally, with its impressive 129 Million tweets tuples, it makes it the most substantial research dataset released until the time of writing. The dataset

is currently combined with BLEU (see Chapter 5.2) to evaluate generated dialogue, often present in the field of machine translation.

Ubuntu Dialogue Corpus *Ubuntu* released its chat logs (Lowe et al., 2015), in 2016, containing over 1 Million multi-turns dialogues. The interest to this dataset comes from its relatively large size containing long technical contexts, and by design, it does not require exhaustive feature engineering to train over out-of-the-box.

Datasets	Release Date	Nested Questions	Hops	Open Domain	Queries	Docs	Query Source	Answer Type
ConvQuestions	2019	**Yes**	**Multi**	**5**	11K	350	**Wikidata**	Spans
Google Natural Questions Corpus	2019	No	Single	**Yes**	323K	??	Wikipedia	Spans
SQuAD 2.0	2018	No	Single	**Yes**	151K	853	Wikipedia	Spans, Unanswerable
CoQa	2018	**Yes**	Single	**Yes**	127K	8K	Children's Stories, Literature, Mid/High School Exams, News, Wikipedia, Reddit, Science	Spans, Unanswerable
QuAC	2018	**Yes**	Single	**Yes**	100K	14K	Wikipedia	Spans, Unanswerable
HotpotQA	2018	No	**Multi**	**Yes**	113K	591	Wikipedia	Spans
DuReader	2018	No	Single	**Yes**	300K	1.5M	Web Search	Spans
TriviaQA	2017	No	**Multi**	**Yes**	650K	95K	Trivia	Spans
RACE	2017	No	Single	No	97K	28K	Mid/High School Exams	Multiple choice
Narrative QA	2017	No	**Multi**	**Yes**	47K	1.6K	Movie Scripts, Literature	Spans
SearchQA	2017	No	**Multi**	**Yes**	140K	6.9M	Jeopardy	Spans

Datasets	Release Date	Nested Questions	Hops	Open Domain	Queries	Docs	Query Source	Answer Type
NewsQA	2017	No	Single	**Yes**	100K	10K	News	Spans
QAngaroo WikiHop	2017	No	**Multi**	**Yes**	51K	528K	**Wikidata**	Spans
QAngaroo MedHop	2017	No	**Multi**	No	2.5K	528K	Medline, Drugbank	Spans
CNN / Daily Mail	2016	No	Single	**Yes**	1.4M	93K / 220K	News	Spans
Children's Book	2016	No	Single	No	688K	108	Children's stories	Multiple Choice
SQuAD	2016	No	Single	**Yes**	108K	536	Wikipedia	Spans
MS MARCO	2016	No	Single	**Yes**	100K	200K	Web Search	Spans, Unanswerable
SelQA	2016	No	Single	**Yes**	8K	486	Wikipedia	Spans
INFOBOXQA	2016	No	Single	**Yes**	15K	150	Wikipedia	Spans
WikiQA	2015	No	Single	**Yes**	3K	29K	Wikipedia, Web Search	Sentence Selection
SimpleQuestions	2015	No	Single	**Yes**	109K	6K	**Freebase**	Spans
bAbI tasks 1 to 6	2015	No	**Multi**	**Yes**	6M	??	??	Spans
MCTest	2013	No	Single	**Yes**	2.6K	660	Children's stories	Multiple Choice

Table 4.1: Overview of Question Answering Datasets. In bold the features identified to be meaningful for the Thesis.

Datasets	Release Date	QA	Open Domain	Dialogues	Utterances	Query Source	Dialogue Type
Google Natural Questions Corpus	2019	**Yes**	**Yes**	323K	??	Wikipedia	Human to Human,
Reddit	2017	No	**Yes**	54M	??	Comments	Human to Human
DSTC4	2016	**Yes**	No	35	??	Chat logs	Human to Human
Twitter Triplets Corpus	2015	No	**Yes**	129M	87M	C-M-R Tweats	Human to Human, blogging
Ubuntu Dialogue	2015	No	**Yes**	1M	7M	Chat logs	Human to Human
Sina Weibo	2015	No	**Yes**	4.4M	8.8M	Posts, Comments	Human to Human, blogging
DSTC2	2014	No	No	3K	24K	Chat logs	Human to Computer
DSTC3	2014	**Yes**	No	2.3K	15K	Chat logs	Human to Computer
DSTC1	2013	**Yes**	No	15K	210K	Chat logs	Human to Computer
Twitter Corpus	2010	No	**Yes**	1.3M	3M	Posts, Tweats	Human to Human, blogging
OpenSubtitles	2009	No	**Yes**	70M	??	Movies	Human to Human

Table 4.2: Dialogues Datasets Overview. In bold the features identified to be meaningful for the Thesis.

Chapter 5

Evaluation

In this chapter, we overview current evaluations in NLP for the two tasks our project is combining, QA systems and computer-generated dialogues. Often, determining if a model is working as expected is a hard task, it depends on the tasks the datasets used. Naively, one could build a complex supervised protocol to evaluate a model's success. Still, often it is not enough due to the chaotic nature of training or to multiple exceptions to handle; indeed, it would require an unrealistic amount of human-power to build a general evaluation protocol, particularly for NL tasks. Instead, it is common to make and combine grounded tasks evaluation and sub-tasks 0(e.15De(not)78(me3(tr)-1iccess)15(s)]TJETQq1001-0.1380.055cm14.)-2q104.5.1kingsystems

dataonsse0(s)0.75(Chapter)0.754.2.1)eed,andnotoftrkeed,notwlr30.753(not)0.1elevalected

corpus (Lopez et al., 2013), and later got evaluated on the ConvQuestions dataset (see Chapter 4.2.1) by the CONVEX authors (see Chapter 5.1.1) with MRR, Top-1 and Top-5 metrics.

5.1.3 Platypus

Similar to qAnswer, Platypus (Pellissier Tanon et al., 2018) is a template-based model (see Chapter 2.7) trained to build SPARQL request for Wikidata as well, but has been released in open-source. It was initially trained, tested, and evaluated with F1 on the Wikidata mapped SimpleQuestions dataset (see Chapter 4.2.2). Later, Playtypus got additionally evaluated on the ConvQuestions dataset (see Chapter 4.2.1) by the CONVEX authors (see Chapter 5.1.1) with MRR, Top-1 and Top-5 metrics.

5.1.4 Honorable Mentions

It is no spoiler to mention that Fine-Tuned Pre-Trained Language Models 3.4 are currently under the spotlights. By curiosity, we wanted to get hold of the phenomenon by investigating and reporting a few comparative results of Transformer, Long Short-Term Memory (LSTM), and RNN Models. We noticed, based on the Table 5.1 that BERT-based (Devlin et al., 2019) models in their base form with pre-training are largely outperforming non-Transformer-based models. As a final note, we didn't judge it necessary to include extended BERT-based models to the table as we believe that it is a field of study by itself and out of our scope. However, we just wanted to mention that, at time of writing, the best baseline in CoQa (see Chapter 4.2.3) leaderboard with an overall 90.7%, is a compositional model combining RoBERTa (Liu et al., 2019) (BERT-based), Adversarial Training (AT), and Knowledge Distillation (KD) (Ju et al., 2019). As a friendly reminder, let's point that Human Performance is set at 88.8% on the same dataset.

5.2 Generative Systems

As mentioned in chapter 4.3, it appeared that so far, progress is in standby for the computer-generated text NLP task, taking machine translation and the Oracle approach apart. As far as our research went, we found two papers supporting interesting facts about computer-generated texts. According to their study (Gehrmann et al., 2019), the authors describe a technique to detect computed generated text by focusing on the induced artifact from generating text. The second paper (Graefe et al., 2018) performed a study on the Readers' perception of computer-generated news. They concluded that people enjoy reading computer-generated texts because it is, in fact, computer-generated, which is currently fascinating to humans; however, a long term study still has to be conducted.

BLEU Originally the Bilingual Evaluation Understudy (BLEU) was created as a completely automated metric for machine translation, but in theory, adaptable to other NLP tasks such as NL generation. By simply comparing the machine output and the ground truth, it appears that BLEU is relatively faster than human translators, computationally friendly, and benchmarkable. It implies that by design, BLEU does not take into account the meaning of the sentence, nor it is taking into account the sentence structure, nor it evaluates how a human would interpret the

sentences. It means that rich languages do not evaluate well and that BLEU performs well in machine translation to measure entire corpora for a good reason, but it is not acceptable in our study as we expect a meaningful human-like evaluation.

ROUGE ROUGE is a BLEU (see chapter 5.2) adaptation focusing on Recall instead of Precision, by evaluating the ground truth to the output.

GLUE Since 2019, the General Language Understanding Evaluation (GLUE) benchmark is a complete framework used to train, evaluate on a collection of datasets and then compare models relative to one another on various NLP tasks. The dataset collection is composed of nine (four are kept private) relatively difficult datasets designed to test MU, which is particularly interesting for fine-tuned models.

Natural Questions Corpus Metric As mentioned in chapter 4.3.1, *Google's* Natural Questions Corpus dataset (Kwiatkowski et al., 2019) is a benchmarking approach that defined a new metric to answer evaluation combining F1 from Long Answers and Short Answers with multiple annotators, making it 25-way Annotated. The technique consists of asking annotators for given questions to say if the question is fact-seeking or not, returns a long and short answer pair. As the next step, the previously annotated questions are sent to 4 annotators, whose goal is to evaluate the annotations. Based on majority expert judgements, the annotated questions are categorized as "Correct", "Debatable" or "Wrong". Finally, the annotated questions evaluation is measured by calculating the Precision and Recall of the long and short answers.

Humans Although it is evident that automatic evaluation metrics are not entirely reliable for text related NLP tasks. The only available solution so far is the use of Humans to perform manual validation, either by Crowdsourcing via Mechanical Trucks, or by asking colleagues. However, even the results obtained from humans are often not qualitatively optimal. Indeed, distraction is in human nature, particularly in Mechanical Trucks setups, resulting in protocols such as *Google's* Natural Questions Corpus dataset (see chapter 4.3.1), which is designed for multiple verifications, with the goal to produced the optimal labelings and evaluations.

Models	Release Date	Handles Nested Questions	ML	**SimpleQuestions F1**	**ConvQuestions MRR**	SQuAD2.0 F1	SQuAD1.1 F1	CoQa F1
ALBERT	2019	**Yes**	Transformer	-	-	92.215%	-	-
BERT-base finetuned	2019	**Yes**	Transformer	-	-	83.061%	93.16%	81.1%
CONVEX	2019	**Yes**	Information Retrieval	-	0.2012	-	-	-
qAnswer	2019	No	Template	-	0.0294	-	-	-
BiDAF++	2018	**Yes**	CNN	-	-	-	-	67.8%
Platypus	2018	No	Template	79.96%	0.0022	-	-	-
QANet	2018	No	Transformer	-	-	-	82.7%	-
DrQA	2017	**Yes**	RNN	-	-	-	79.353%	-
BiDaF	2016	No	LSTM	-	-	-	81.525%	-
MemNet	2015	No	Memory Network	77.97%	-	-	-	-

Table 5.1: Question Answering Benchmarking Overview

Part III

Design and realization

Chapter 6

Analysis

In this chapter, we analyze the SOTA (see Part II) to define the scope of the POC and add knowledgable details to project specifications as our initial understanding of the thesis subject increased. As a kickoff to research the SOTA and for the analysis, in addition to peer consulting with our lab colleagues in the field of NLP, we used curated lists, e.g., awesome-nlp on Github (Keon, 2017). The knowledge accumulation started naturally as we began to chain read articles mentioned in other papers.

6.1 Rescoping and Motivations

Extrapolated from the SOTA part II, we present in this section the process held to get to the final project scope based on the accumulated knowledge through NLP SOTA exploration.

6.1.1 Initial Project

Our research initially started as a satellite to the *AI-News* project (see Chapter 1.1.2), which is currently using a Retrieval approach (see Chapter 2.3) combined with an intents and entities matching techniques, to return highly pondered recent articles from an elastic search in a Ruled-Based chatbot (see Chapter 2.4) format. The *AI-News* project scoped our research indirectly toward finding solutions to bring QA systems to the field of journalism; however, it early rescoped to open domain knowledge as the Wikidata KB provides crowdsourced general knowledge. Additionally to the QA scope, the specification implied NL answers generation. We intended to build a QA system to generate BLEU approved corpora based on the Wikidata KB, then compare them to SQuAD benchmarking (see Chapter 4.2.3), with the ultimate purpose to compare Pre-Trained Language Models such as BERT on your new dataset. As the starting point, we decided to bootstrap our project with SOTA related papers filtered by code availability and reproducibility.

6.1.2 Initial Ideas

To achieve our initial project, we brainstormed the following ideas. Indeed, we expected to explore in detail the field of QA evaluation, in particular Oracle-based solutions. Oracles are particularly meaningful in our context as we would use Wikidata as an oracle. Additionally, we planned to explore in more detail the effects

of fine-tuning pre-trained language models such as BERT, with the intent to create a challenging dataset. Based on a finite state grammar, our evaluation would initially generate a set of 100'000 Wikidata SPO-based questions which uses word-embedding similarity feature to paraphrase those questions in order to increase complexity. We also will discuss an additional feature to generate QA multi-turns conversations (see Chapter 2.8.1) by self-training a model with an AL approach.

6.1.3 Second Brainstorming Iteration

As research progressed, we decided that we did not want to be just another benchmarking system for QA systems, similarly to our initial decision not to become an nth fine-tuned pre-trained language model. As a result, our second brainstorming iteration brought quite a new scope to the project. Indeed, we focused mainly on a Multi-Hop multi-turns conversations approach, by aiming at building an interactive and proactive reversed Akinator-like (Elokence, 2007) QA system. The system would use a "child" learning approach as the model itself starts with no knowledge, and incrementally learns new knowledge by interacting with users. The goal is to teach the model to retrieve information from a KB by itself. The game consists of a randomly selected Wikidata entity as the answer, in which the user is requested to help the bot, via a gamified conversational interaction, to build a path in Wikidata while the bot asks questions to the user. The proactive ability would be a generative approach to scenario-based chatbot like HelloJam (SAS, 2014), which uses intermediary steps as a proactive approach. As a training bootstrap, we initially planned to build a conversational simulator using multi-hop datasets such as ConvQuestions, SQuAD, or QuAC.

6.1.4 Third Brainstorming Iteration

Quickly, the previous brainstorming iteration raised problems such as the true user interest for such games and the bias induced by genuine or intentional human errors, without mentioning the pre-processing needed to build a meaningful training dataset; nevertheless, we kept some ideas. We believed that the Multi-Hop multi-turns conversations are an essential feature to our work, combined with the interactive and proactive approach between the bot and the user, which is a particularly exciting application in the NLP field. We then suggested a new shift in the project scope to build NL QA chatbot allowing the user to interact with the conversation and the knowledge in a meaningful manner. The goal is to give to the user the ability to check returned answers with facts and correct them on the fly if needed. Additionally, we wanted to add a multi-model approach to handle virtual personalities, as a flavor of personalization to the user.

6.1.5 Final Brainstorming Iteration

The project scope shifted one last time. We noticed that the applications in the field of QA and Generative chatbot are extensive, and it appeared that it is possible to find a paper already mentioning, even lightly, what we believed original ideas. As a contradiction, we decided at aim directly at the roots of QA and Generative systems as a whole. To do so, we kept the multi-hops reasoning, multi-turn conversations, and the Wikidata KB as constraints to the project. Indeed, we did not want to be just another Transformer related project trying to define the new baseline.

Our resources at disposal were limited, and we wanted to take a new approach by exploring an original technique for QA, in particular Sub-Knowledge Graphs. To achieve our latest project, we examined existing SOTA systems providing, in addition to the paper, a runnable code to get started. We initially aimed at incrementally improving the original work to impact the NLP field. Our first step was to reproduce the results; then, the second step was to retrain and provide additional value to the original work. And finally, adapt the initial project to the News field with few tweaks.

6.2 Question-Answering Systems Choices

Based on the SOTA (see Part II) and the previous rescoping section 6.1, we aim at building a QA Multi-Hop and multi-turn converstational chatbot using sub-graphs from Wikidata. We found a unique direct competitor to our work, and we are using its baselines and nested baselines to compare our work.

6.2.1 Competitors

We initially explored QA system using the Wikidata KB by default, and providing the must-have features defined in the project scope, Our research revealed a unique candidate and two related candidates, as they define the baseline of our main candidate.

CONVEX As our direct competitor, CONVEX (Christmann et al., 2019) (see Chapter 5.1.1) extracts sub-graph from Wikidata KB. It employs the sub-graphs as context holders and uses them to answer context-related questions via a frontier algorithm. It finally extends the context-graph with the new answers, making the graph more precise at answering detailed questions.

qAnswer and Platypus Both QA systems are using a template-based model (see Chapters 5.1.2 and 5.1.3), and are defined as the baselines to our main candidate, CONVEX, and they are designed to for Wikidata queries. Note that they are Single-Hop and trained on the SimpleQuestions dataset, which we also plan to use as a fair evaluation.

6.2.2 Datasets

Our initial predetermined dataset was SQuAD, due to its popularity and its public leaderboard, as it would have been a pleasant additional comparison to our work. However, SQuAD is designed to answer fact-based questions with a span extracted from the given context paragraph, often implying a mismatch with fact entities like those of the Wikidata KB. It is also a reason for CONVEX to not evaluate on this dataset, similarly to qAnswer and Playtypus. Finally, in the scope of the project, we did not plan to adapt the SQuAD dataset for Wikidata-based QA systems.

ConvQuestions As descried in Evaluation Chapter (see 5.1.1), CONVEX (Christmann et al., 2019) is design to use the ConvQuestions dataset (see Chapter 4.2.1), which is a multi-turn conversations and Multi-Hop QA crowdsourced dataset. CONVEX uses also this dataset with its baselines providing an initial evaluation for all three competitors.

SimpleQuestions As presented in the Dataset Chapter (see 4.2.2), SimpleQuestions has a Wikidata KB adaptation provided by Playtpus authors (Pellissier Tanon et al., 2018). This dataset will be used to benchmark our system in addition to all three competitors, as a fair approach to evaluate Single-Hop systems on Single-Hop design datasets.

6.2.3 Benchmarking

To measure and compare our work, we will set up two benchmarks. The first benchmark will evaluate the Single-Hop capabilities of each competitor on the Wikidata adapted SimpleQuestions dataset. The second benchmark will be focusing on the Multi-Hop and multi-turn conversations capabilities from each competitor. As qAnswer and Platypus are not designed for multi-turn conversations, we will extend them with CONVEX and our work during the multi-turn task, as both projects can work on top of other algorithms (see the next Chapter 7).

6.3 Texts Generation Choices

We plan to used two SOTA pre-trained LM in their vanillia large format, BERT (Devlin et al., 2019) and GPT-2 (Radford et al., 2018), for our text generation NLP task. BERT will be used as a text filler for the SPOs paths extracted complementary to the answer from the Wikidata Sub-Knowledge Graphs. GPT-2 will be used as a complementary facts generator to the answer.

Evaluation We consider *Google's* dataset (see Chapter 4.3.1) as promising; however, we did not plan to evaluate our generated NL answers on this dataset as the project planning is too tight to include this protocol. Instead, we will assess the dialogues generated manually. An evaluation idea would be to run a campaign with 10 to 20 users evaluating 10 to 20 questions, using the evaluation tags: “Good”, “Neutral”, or “Bad”. However, this dataset is worth considering for future extended work.

6.4 Final Project Scope

GraphQA, as we name it, is a QA Sub-Knowledge Graphs Chatbot using the Wikidata KB database. It extracts answers to questions from a complete Linked Data database by extracting small portions of the main database and manipulates it as a graph. The graph manipulation implies context holding, extension, and refining. Initially, based on CONVEX, we plan to improve their Information Retrieval (IR)-based answering to the first answer, described as bottleneck, which is particularly sensitive as it defines a successful anchor to a conversation-driven QA. We wish to also explore sub-graph capabilities with temporally dependant contexts, and their overall performances. Additionally, NL capabilities must be present by combining two complementary pre-trained language models.

Expected Features

Below are defined expected GraphQA features as a Multi-Hop Conversational QA Chatbot.

- Extract question keywords.
- Extract a question-related Sub-Knowledge Graph.
- Compute the answer from the Sub-Knowledge Graph.
- Extract an SPO Tuple meaningful to the question.
- Prune the Sub-Knowledge Graph of context meaningless elements.
- Generate a NL answer from the SPO Tuple.
- Extend the NL generated answer with additional facts.
- Extend the Sub-Knowledge Graph with new questions.

Nice to have

As we do not expect to have the time to add additional features to the primary features described in the above subsection, we are mentioning some features that we would like to see in the project one day. Note that KB databases are by design only as good as the relations they hold, which is limited also by the employed schemas. This implies that if the reasoning information such as quantities were not manually referenced into the database, the answer is impossible to retrieve.

- Give the ability to merge pre-generated sub-graphs.
- Use multiple modules to manage different processes.
- Include reasoning modules such as inductive logic or quantification.
- Provide an auto-correct tool to users.
- Provide an auto-complete tool to users.
- Provide a paraphrasing tool to generated answers.
- Use the multilingual Wikidata's propriety paraphrase into a translation.
- Use Wikidata's multi-properties to paraphrase questions and answers.
- Handle pre-built sub-graphs for particular subjects such as Articles, Countries, Movies, or Famous People.
- Track users custom sub-graphs to track the overall context.
- Use consensus-based modules to complementarily provide the best answers.
- Show visually to users their sub-graph generation.
- Provide on-the-fly tools to modify sub-graphs.

6.5 CONVEX Q0 Solutions

As mentioned in previous sections and as represented on the Figure 6.1, CONVEX (Christmann et al., 2019) has an issue with the first answer. The issue has been briefly explored and confirmed in the CONVEX paper. Indeed, for a0 (the initial answer), CONVEX uses a proprietary Named-Entity Recognition system, TAGME (Ferragina et al., 2010), as its Wikidata entities identify system. It then places into an empty a subgraph the returned entities with their relation extracted from the Wikidata KB. This process is solely relying on a TAGME to answer the initial question, which is often interpreted as lucky guesses. As one of our main contributions, we want to fix this issue, and to do so, we propose five different solutions, including a naive one.

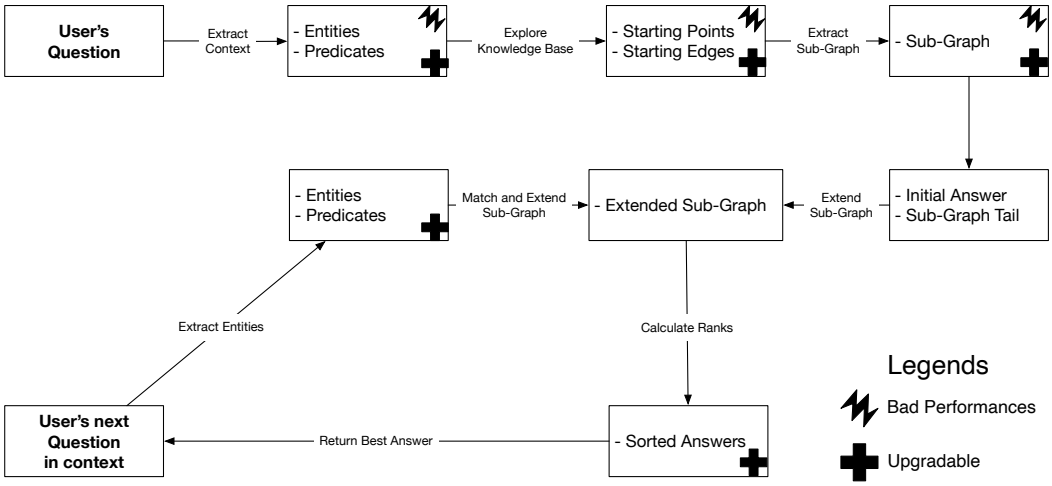


Figure 6.1: Illustrative representation of the high level CONVEX architecture. The diagram includes the identified part having bad performances, and shows the upgradable components.

6.5.1 0th Solution: Naive Approach

Our naive approach is to use text summarization to extract relevant information and build the initial sub-knowledge graph by matching the entities present in the Wikidata KB. It is indeed not an impressive approach, but at least we have control over the extracted data, and we can study the related induced behaviors, then tune the system with additional models.

6.5.2 1st Solution: BiDAF++

The BiDAF++ model, presented with QuAC (Choi et al., 2018), is based on BiDAF (Seo et al., 2016) augmented with self-attention (Clark et al., 2017) and ELMo (Peters et al., 2018), and particularly used as non-Transformer baseline on multiple QA datasets. We could explore training on multiple dataset such as Google's Natural Questions Corpus (Kwiatkowski et al., 2019), ConvQuestions (Christmann et al., 2019), CoQA (Reddy et al., 2018) and NewsQA (Trischler et al., 2016). The model would provide us the answers to the questions, and we would build the initial sub-graph based on the matched entities id found in the question and the entity found as the answer.

6.5.3 2nd Solution: Multi-task learning

Multi-task learning for large scale KB (Shen et al., 2019) is by design handling the conversational QA format, making it particularly appealing to our project. The techniques uses a model trained to parse questions and point them into the KB via pointers, which avoids the propagation of errors and simultaneously exploits the pointer property to share the linked information.

6.5.4 3rd Solution: Knowledge Graph Embedding

Knowledge Graph Embedding (Wang et al., 2019) is a model trained to locate entities (subject) and predicates individually in a Knowledge Graph (KG), and then predict the tail (object). However, this technique, as described by the authors, works only for simple questions, which conflicts with our Multi-Hop constraint, implying to extend the model capabilities to Multi-Hop handling.

6.5.5 4th Solution: Fine-tuned Pre-trained Language Model

Even if we do not expect to use this solution, it is still worth mentioning. Indeed, the final solution would be to fine-tune a transformer-based model on multiple QA datasets as similarily mentioned in the 1st solution above (see Subsection, 6.5.2).

6.5.6 Our representation in the Chatbot Cartography

To conclude this chapter, we updated the chatbots cartography as defined in the chatbot state-of-the-art chapter (see 2.9) to illustrate our position. (See Figure 6.2)

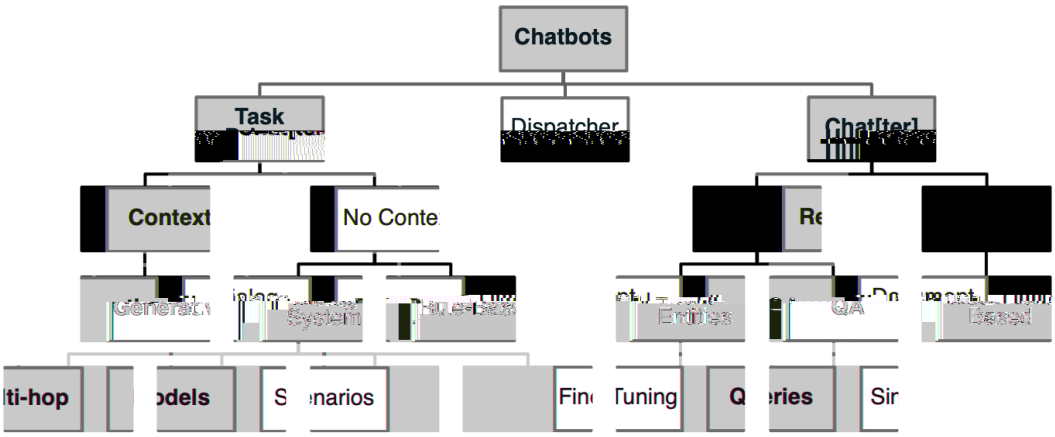


Figure 6.2: In grey, the GraphQA's positions in the chatbots cartography is represented as defined in the chatbot state-of-the-art chapter.

Chapter 7

GraphQA

As stated in the analysis (see chapter 6), GraphQA planned to be a straight forward approach for improving the CONVEX (Christmann et al., 2019) prototype on Q0. As a second step, the project would aim toward NL generated answers based on the Sub-Knowledge Graphs. As this chapter goes, we could not go as planned during the analysis (see Chapter 6) due to CONVEX complications encountered. However, to our relief, we could bounce back and produce original work. Our final approach started with the naive solution proposed during the analysis (see Chapter 6.5.1) and combined it with exciting features we particularly thought meaningful from our analytical brainstorming (see Chapter 6.1). Indeed, in addition to our well defined Multi-Turn Conversations, Multi-Hop, and Wikidata KB Sub-Knowledge Graphs scoped features, we are exploring a Ground Learning approach by orchestrating various specialized NLP models as a global Zero-Shot Learning learning approach for NL QA chatbots. Finally, we keep the same evaluation settings, as stated in the analysis (see Chapter 6.2.3), including the ability for GraphQA to works on top of other QA systems.

7.1 GraphQA Architecture

We aim at highlighting in this section the High-Level Architecture evolution from the initial analysis-based scope to our current architecture.

7.1.1 Initial Architecture

As represented in Figure 7.1, the initial approach for GraphQA was to enhance the CONVEX architecture for the Q0 question (see Chapter 6.5), and then plug a NL generator for answers. Finally, the second step was the upgrade of additional CONVEX modules and suggest new use cases to the project, such as a News extension.

7.1.2 Current Architecture

Our architecture, as represented on Figure 7.2, results from three major incrementations. It is, at first sight, heavier than the initial CONVEX-based architecture (see previous Section 7.1.1); however, its modular and generic design handles contextual graphs with an overall improvement toward the initial architecture as planned in the analysis. Indeed, as we built the project from the ground-up, we focused on the

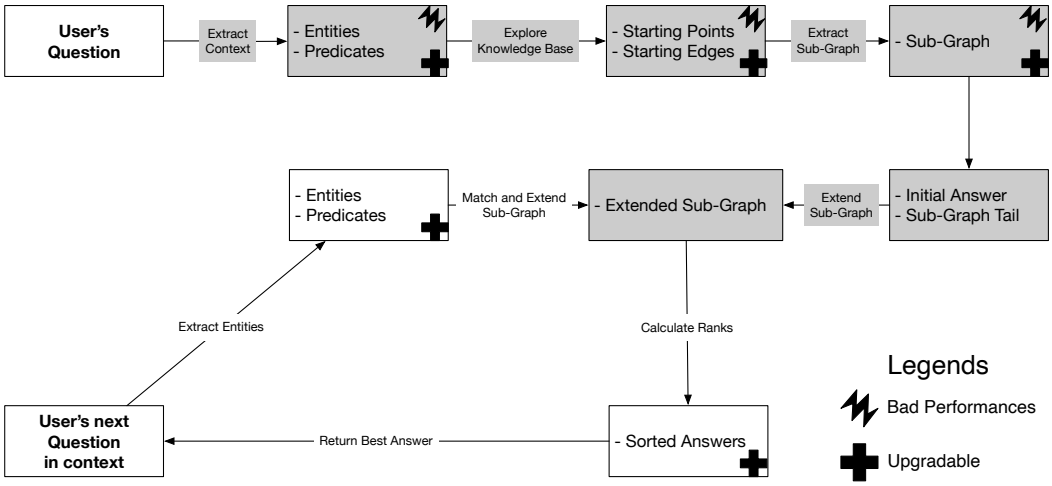


Figure 7.1: Illustrative representation of the initial high level GraphQA architecture improving CONVEX. In grey, the architecture parts for GraphQA to rebuild.

must-have features and pre-designed anchor points for the nice-to-have features as described in the final scope from our analysis (see chapter 6.4). Additionally, as we aimed at a generic approach via modular orchestration, our QA chatbot is by its architecture and design, extensible to further modules by either improve, replace or add new NLP tasks.

7.1.3 Question Answering Pipeline

In more details, we present on Figure 7.3 our grounded tasks approach, implying a Zero-Shot Learning learning QA system, as it doesn't not require any training examples to answer questions.

7.1.4 Dialogue Flow

To illustrate in more details GraphQA Dialogue flow, Figure 7.4 presents our NL answering approach for multi-turns conversations.

7.2 Iteration 0

The initial GraphQA version was CONVEX-based as we originally scoped to extend CONVEX features (see our previous sections 7.1.1). As we dove into the CONVEX implementation, we came across reproducibility issues and unexpected behaviors. After multiple contacts with the authors, we concluded that our interpretation of their paper was not as they intended to. Indeed, what we believed to be features, were from their point of view made-up feature for motivational purposes, which sadly deeply impacted our SOTA analysis. However, during the initial phase, we understood the tools they built, giving us a head start for our architecture. Indeed, tools such as loading the Wikidata KB and fetching SPO statements were reusable out-of-the-box. Optimizations such the HDT (Mario Arias Gallego, 2017) and NetworkX (Hagberg et al., 2008) python libraries were a useful as starting points.

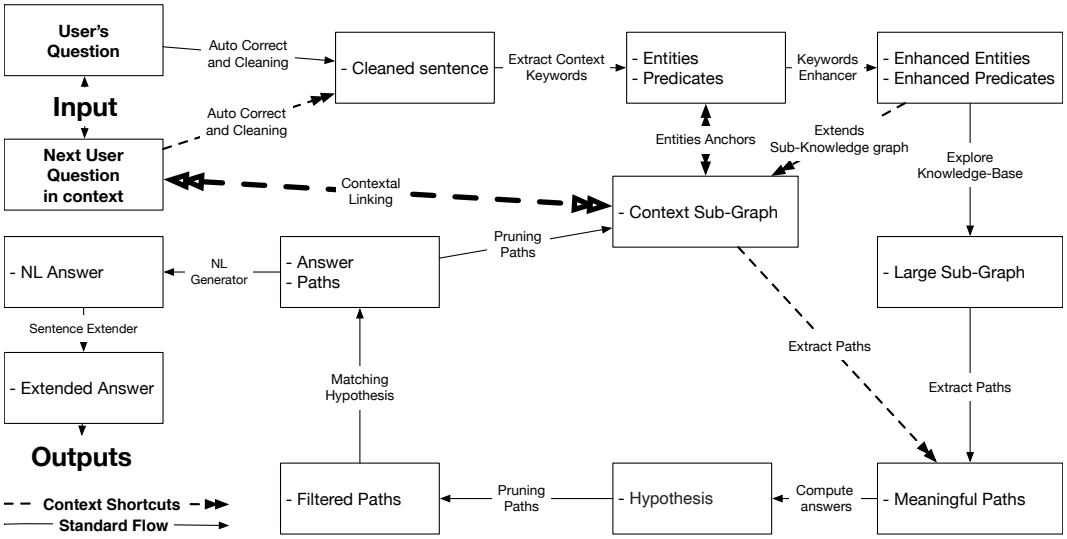


Figure 7.2: Illustrative representation of current high level GraphQA architecture. Double arrows indicates that the flow is related to context.

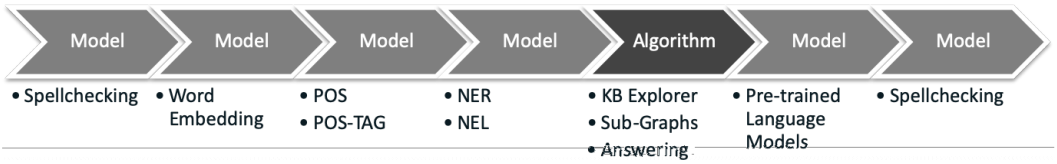


Figure 7.3: Illustrative representation of GraphQA multi-models pipeline.

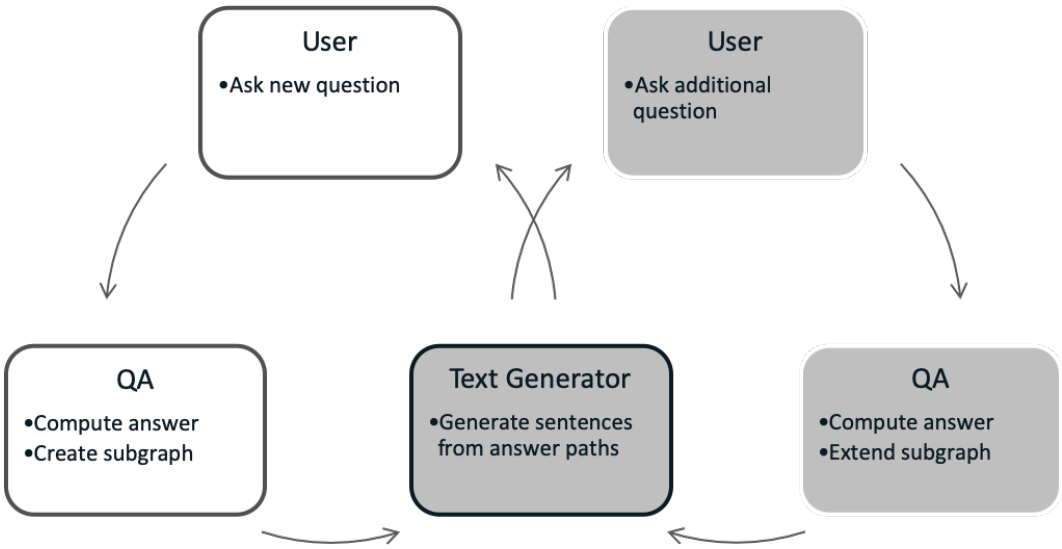


Figure 7.4: Illustrative representation of current GraphQA dialogue pipeline.

7.3 Major Iteration 1

Based on the toolset extracted version 0 from the CONVEX-based extender, GraphQA V1 is composed of the features listed below. During the iteration, we saw the opportunity to contribute to the NLP field by building a recent version of the Wikidata KB optimized HDT (Mario Arias Gallego, 2017) format, as the latest dates from September 2018. Sadly, we were not able to generate an HDF dataset from the latest version of Wikidata KB as we didn't have the required ram at our disposal to complete the task (about 240GB for 7.9 Million tuples).

7.3.1 Library-based Features

We are listing used libraries in the next section (see Section 7.5). Note that we may, in some cases, slightly enhance the basic library features, but we are keeping their main purpose intact.

- Retrieving statements from the Wikidata KB via CONVEX
- Named-Entity Linking labels from the Wikidata KB via CONVEX
- Document/Span/Word Embeddings via Spacy and GloVe
- Tokenization via Spacy
- Part-of-Speech and Part-of-Speech Tagging via Spacy
- Named-Entity Recognition for common word such as towns, countries, or famous people via Spacy
- Extract composed entities via Spacy
- Extract nested word groups within sentences via Spacy
- Build and explore sub-graphs via NetworkX

7.3.2 Custom Features

The following features were designed with modularity in mind, as stated in the earlier sections. We describe them once here, but they are used in next versions. (See their algorithm subsections).

Wikidata fine-tuned Named-Entity Recognition

Spacy's Named-Entity Recognition is particularly impressive out-of-the-box; however, we found the need in the scope of the project to fine-tune Spacy's Named-Entity Recognition model to help it identify Wikidata and Wikipedia pages names.

Wikidata Named-Entity Linking training

Spacy's provides training tools for additional models and includes them natively into its pipeline. We used the opportunity to build a Named-Entity Linking model linking our fine-tuned Named-Entity Recognition model to also be able to linked the recognised pages to Wikidata KB identifiers.

Extract themes

The entities we call Themes are the most meaning words in a question; they define the initial anchor point to our QA system. They have a stronger impact than keywords, as their purpose is to define the question in a meaningful manner.

To extract the question themes, we initially perform an entity check via Spacy's Named-Entity Recognition on the sentence.

To enhanced the Named-Entity Recognition results, we extract nested word groups within sentences via Spacy's Document Embedding property **noun_chunks**, then pass a Named-Entity Recognition on the detected composites.

Additionally, we format the sentences into its Capitalised, Lowercased, Lemmatised, and Determinants-free forms, then apply Spacy's **noun_chunks** property to altered sentences and pass a final Named-Entity Recognition on the detected composites.

Note that the detected composites are the themes in our case.

Enhance themes

During the theme extraction process, not all the words in the sentences are used as they have no meaning for our Named-Entity Recognition model. In this module, we give a second chance to the trashed words by trying to match them to Wikidata entities by exploring the KB.

To do so, we apply a similar technique than for theme extraction by formatting the word in their Capitalised, Lowercased, Lemmatised, and Determinants-free in its forms. We combine the words into multi-sized combination tuples then perform a lookup into the KB to find a matching identity.

Extract predicates

Often, questions are using inductive nouns or adjectives for predicates, making the naive approach of matching verbs to predicate not efficient. Our solution is to query the KB with each word and evaluate the result as predicated. Additionally, we query Wikidata online with the specific type defining predicates, which often provides additional entities to our initial local query. Note that we initially built GraphQA intending to work in standalone without an internet connection, which makes the web querying sensible to our initial intention, but not against the project scope.

Extract sentence focused parts

The module extracts word syntactically focused by the qualifier, such as who, where, or when. We use those words as initial answer anchors as, in most cases, is it possible to replace this word with the answer, and it converts the question into a descriptive sentence.

Extract question keywords

This module aims at capturing the relevant keywords from the question. It categorizes the keywords in three categories, the themes driven keywords, enhanced themes drive keywords, and predicates related to the focused parts. To do so, it uses similar techniques to the Attention Mechanism by exploiting the extract word

properties by Spacy's Document Embedding model, and the entities present in the sub-knowledge graph.

Extract keyword related paths

Based on dynamic thresholds, this module uses previously extracted keywords to extract paths from the raw sub-knowledge graph.

Filter extracted paths

The module parses the extract keyword paths to fix malformed SPO chains by extending further the path and finishes by removing duplicates and sublists.

Compute answer hypothesizes

The process consists of rebuilding the question in its descriptive form with SPO extractions from each filtered path. The final score combines the computed score for each word with an overall rating for each path. The scores are defined either with positive or negative values depending on the evaluated tasks such are word proximities.

Build the sub-knowledge graph

Based on the extracted themes, enhanced themes, and predicates, this module explores the Wikidata KB for statements, uses the extracted themes as subject and extracted predicates as predicates and adds matching statements into an empty NetworkX graph.

7.3.3 Algorithm

Here is enumerated the overall process to compute the answer from a question.

1. Builds a Document Embedding of the question
2. Extracts the themes
3. Extracts the enhanced themes
4. Extracts the predicates
5. Extracts the question focused parts
6. Builds a sub-knowledge graph
7. Rebuilds the graphs with a higher deepness if it contains too many elements
8. Extracts the question keywords
9. Extracts the question-related paths from the sub-knowledge
10. Filters the extracted paths
11. Computes and sort hypothesizes
12. Extracts the paths containing the best hypothesis as the first or last element

7.3.4 Major Iteration 2

In addition to some computational parallelization, tweaks, and optimizations, the second version of GraphQA aims to provide a wiser answer by not trusting the initial best hypothesis score but instead considering all the hypotheses by additionally scoring the matching for each hypothesis within the question itself. We added global banned entities such as linking ids to other KB. Finally, the algorithm can handle various computational decisions via predefined thresholds dynamically.

7.3.5 Library-based Features

We hereby list only the additional features to the first version (see subsection 7.3.1). See Section 7.5 for the libraries listing.

- Auto correcting the questions from the user via DeepCorrect
- Replacing recognised names from Wikipedia Named-Entity Recognition within the question via DeepCorrect

7.3.6 Custom Features

We hereby list additional and modified features to the first version (see subsection 7.3.2)

Compute answer hypothesizes by handling types

The second iteration of this module uses word embedding similarities for its scoring tasks; additionally, it now has the notions of Locations, Persons, Dates, Causes, Quantities, and Selective questions to refine the scoring to the answer types. Finally, we enhanced the scoring formula to handle the overall path score while computing the individual SPO scores.

Evaluate hypothesizes

This module calculates a score for each hypothesis by replacing the hypothesis with the focus part within the question and comparing the new descriptive sentence to the original question. Additionally, it explores the KB to reconstruct the original path by giving a higher weight to qualifiers, and gives a score based on the success of the task.

Answer paths extractor

Use the fact that we can trust the hypothesizes evaluator to return the best-fitting answer to the question, we enhanced the answer path extractor also to include paths where the answer is not in the first of the last place.

7.3.7 Algorithm

Here is enumerated the overall process to compute the answer from a question. In bold are highlighted the new tasks.

1. **Removes unsupported characters from the question**

2. **Autocorrects the question and replaces with Wikipedia names via Named-Entity Recognition**
3. Builds a Document Embedding of the question
4. Extracts the themes
5. Extracts the enhanced themes
6. Extracts the predicates
7. Extracts the question focused parts
8. Builds a sub-knowledge graph
9. Rebuilds the graphs with a higher deepness if it contains too many elements
10. **Rebuilds the graphs with a lower deepness if it does not contain a minimum of elements**
11. Extracts the question keywords
12. Extracts the question-related paths from the sub-knowledge
13. Filters the extracted paths
14. Computes and sorts hypothesizes
15. **Evaluates and sorts the hypothesizes**
16. **Extracts the paths containing the best hypothesis**

7.4 Major Iteration 3

In addition to now handle multi-turns conversations within a context sub-graph, GraphQA is completely parallelized, which improves the overall answering speed by 50%, but does not solve the speed impact of computation (see the results Chapter 8). Numerous tweaks have taken place and made the overall answering better compared to the previous iteration. Finally, GraphQA in the current state returns answers in NL, with the option to extend them with KB and random facts.

7.4.1 Library-based Features

We hereby list only the additional features to the second version (see subsection 7.3.5). See Section 7.5 for the libraries listing.

- Use of BERT-large pre-trained language model via Huggingface
- Use of GPT-2-XL pre-trained generative language model via Huggingface

7.4.2 Custom Features

We hereby list additional and modified features to the first version (see subsection 7.3.2)

7.4.3 Sub-Knowledge Graph optimnizations

We enhanced the initial sub-knowledge graph extractor by additionally filtering clusters and removing clusters unrelated to the questions.

7.4.4 Binary answers

This module handles binary questions by exploring the sub-knowledge graph to detect the presence of the focused part.

Context holding in sub-graphs

We prune the sub-knowledge graph from clusters entities to the computed answer. We call the pruned sub-knowledge graph Context Graphs, as they only hold the context to the question and previous answer.

Multi-turn conversations

In addition to handling context graphs during QA, by using the entities already present in the context graph to drive the KB exploration and replace pronouns in the new question, the module keeps track of the previous direct questions and weights the keywords to it.

7.4.5 NL answers

Until the second version, we returned the fact answer and the best paths from the answer. This module uses the masking feature from BERT (Devlin et al., 2019) to fill the gaps between SPO within the answering paths.

7.4.6 Context facts extender

By default, the NL answers computes the returned SPO based on the question, but it is also possible to manually increase the amount of SPO to return, which increase the original NL answers to additional facts extracted from matched paths.

7.4.7 Random facts extender

For gamification purposes, it is also possible to extend the NL answers with generated random facts using GPT-2.

7.4.8 Handling Pre-built graph

GraphQA can be plugged to any QA system, as it only requires a context graph, which can hold from none to any amount of nodes and edges, and optionally keywords from the previous conversation. This feature makes it a nice candidate to give the ability to non-conversational QA systems to handle Multi-turn conversations.

7.4.9 Algorithm

Here is enumerated the overall process to compute the answer from a question. In bold are highlighted the new tasks.

1. Filters the sentence from unsupported characters
2. Autocorrects the question and replaces with Wikipedia names via Named-Entity Recognition
3. **Handles previous context**

4. Builds a Document Embedding of the question
5. Extracts the themes
6. Extracts the enhanced themes
7. Extracts the predicates
8. Extracts the question focused parts
9. **Extends previous context graph**
10. Builds a sub-knowledge graph
11. Rebuilds the graphs with a higher deepness if it contains too many elements
12. Rebuilds the graphs with a lower deepness if it does not contain a minimum of elements
13. **Answers binary questions based on the graph**
14. Extracts the question keywords
15. Extracts the question-related paths from the sub-knowledge
16. Filters the extracted paths
17. Computes and sorts hypothesizes
18. Evaluates and sorts the hypothesizes
19. Extracts the paths containing the best hypothesis as the first element
20. **Extracts context graph from sub-knowledge graph**
21. **Builds a NL answer**
22. **Extends the NL answer with context graph**
23. **Extends the NL answer with random facts**

7.5 Technologies

The following are the technologies used by GraphQA.

HDT (Mario Arias Gallego, 2017) It is a query compression format for linked data; it compresses the RDF database and loads its index into the RAM for querying.

Spacy (Honnibal et al., 2017) It is an impressive framework used in the industry for its multiples NLP tools, pre-builts models and easy to use fine-tuning pipelines. It allows handles Document Embedding, Word Embedding via GloVe (Pennington et al., 2014), Named-Entity Recognition Named-Entity Linking, Part-of-Speech, Part-of-Speech Tagging, Tokenization, and much more. On a side note, we used the version 2 released in 2019.

DeepCorrect (bedapudi6788, 2018) It is a project exploring the DL for text and punctuation correction with a pre-trained model on Wikipedia.

NetworkX (Hagberg et al., 2008) It is a Python package used for the creation and manipulation of graphs.

Huggingface (Wolf et al., 2019) It is a startup focusing on NLP tools, in particular Transformers. They provide frameworks to use pre-trained language model out-of-the-box, making them popular in the NLP field and the industry.

7.6 Dev Benchmarking Questions

For each GraphQA version, we used the same six development questions to evaluate our work superficially. Additionally, we incrementally increased a pool of questions triggering error while performing benchmarks on the SimpleQuestions and ConvQuestions datasets, which we kept as debugging purposes and tracker for the consistency across our incremental versions.

7.6.1 Single-hop Questions

Here is our two development Single-Hop questions.

- “Who is the wife of Barack Obama?”
- “of what nationality is ken mcgoogan”

7.6.2 Multi-hop Questions

Following is our two development Multi-Hop questions.

- “What is the name of the writer of The Secret Garden?”
- “Which actor voiced the Unicorn in The Last Unicorn?”

7.6.3 Multi-hop and Multi-Turns Questions

Following our two development Multi-Hop and multi-turn questions.

Conversation 1

1. “Which actor voiced the Unicorn in The Last Unicorn?”
2. “And Alan Arkin was behind..”
3. “Who did the score?”
4. “So who performed the songs?”
5. “Genre of this band”
6. “By the way, who was the director?”

Conversation 2

1. “Who is the author of the Harry Potter series?”
2. “What was the year of publication for the first book?”
3. “The first book was called what?”
4. “It was set in what country?”
5. “Which book has the highest page count?”

7.7 Further Exploration Ideas

This section provides a non-exhausting list of exploratory ideas for future related studies.

- Use a pre-trained language model such as BERT as word embedding instead of GloVe.
- Fine-tune a pretrain language model such as BERT for Named-Entity Linking.
- Fine-tune a pretrain language model such as BERT to add a multi-brains approach as slightly mentioned in the analysis 6 brainstorming. Where multiple models are answering simultaneously the same question, and reach a consensus for the final answer.
- Query Wikidata online for cross-checking local entities and enhanced themes.
- Prebuild context graph for contents such as Articles to increase speed and target the information.
- Context graph bridging multiple prebuilt graphs to gather information faster and long-distance context handling
- Extend context graphs for user experience personalization by keeping long-term contexts.

7.8 Interesting Facts

On a final note, we also wanted to mention a non-exhausting list of problems that occurred during the making of GraphQA that we believe are interesting facts.

- The uncompressed version of the latest Wikidata dump is 500 GB.
- Knowledge Bases do not have the same conventions for Subject-Predicate-Object Tuples (SPOs).
- Nested Tensorflow sessions are not working as expected when called.

Part IV

Retrospective

Chapter 8

Results

In this chapter, we review our work results split into the methodologies investigation and our comments on the benchmarks.

8.1 Methodologies

As stated in our analysis (see Chapter 6), we planned to evaluate our QA system by comparing our results with our defined baseline, CONVEX, which is the most similar QA system we found during our SOTA study (see Chapter 5). We additionally planned to evaluate our generated NL answers by subjective humans as a way to frame the time available for the project.

8.1.1 Question Answering

It is interesting to note that we originally started benchmarking the second version of GraphQA at the beginning of the 5th sprint (see Chapter 9). Sadly, we could not collect significative data from this initial evaluation, as the benchmark raised multiple bugs and resulted in the development of a new GraphQA iteration designed to handle the errors and, we believe, improved the overall results. We restarted the benchmark at the beginning of the 6th sprint on the third GraphQA iteration, in parallel to minor feature implementation, and bug fixes.

Constraints

During the benchmarking process we noticed two issues. The first is the computation time for GraphQA, and the second is a memory leak, that we did not have time to investigate, occurring every 24 hours of non-stop benchmarking on the latest version of GraphQA. Indeed, GraphQA is very computation-intensive and requires some time to run, which resulted in a relatively small amount of results (see the next benchmarking section). However, we estimate that for a Zero-Shot Learning approach, we have a reasonable answers pool for a first evaluation review, even if we can only agree that it is better to evaluate the most data available.

Evaluation Fairness

ConvQuestions We did a complete evaluation of all algorithms for two reasons. First, because we wanted to evaluate the reproducibility of the data published with CONVEX (Christmann et al., 2019) and secondly, for consistency as the F1 metric

is missing and since GraphQA won't evaluate on all questions from the dataset. We planned to compare fairly the four competitors on the same sampled dataset.

SimpleQuestions We are are also performing a complete evaluating of the dataset as we do not have the benchmark results for CONVEX on this dataset, and to be fair to GraphQA, which is slower, we are evaluating the four competitors on the same dataset sample.

Measures

We mentioned in the evaluation chapter 5 our intention to use the MRR and F1, which we use in the final stage of our work. Indeed, as described in our previous section about Fairness, we evaluate the four competitors equally on the same sampled datasets, allowing us to measure MRR and F1 fairly. F1 and MRR will be used on ConvQuestions, and F1 on SimpleQuestions.

8.1.2 Generated Natural Language Answers

Concerning the generated sentences evaluation, we asked humans to do it, as we didn't have enough time to implement the *Google*'s natural questions protocol (Kwiatkowski et al., 2019). Additionally, as the primary focus for GraphQA is the ability to build and use sub-knowledge graphs, the evaluations of how well performs the pre-trained language models we used are not particularly meaningful overall. However, we still believe that our approach at using pre-trained language models is an original solution to QA tasks, as demonstrated with the generation of NL answer based on paths extracted from Wikidata KB.

8.2 Benchmarks

8.2.1 Hardware

iCoSys provided two Dedicated Servers, a Lambda Lab, and a CPUs-based machine .

Lambda Lab Specification

- CPU: 1x Intel(R) Core(TM) i9-9820X CPU @ 3.30GHz
- RAM: 126GB
- GPUs: 2x Nvidia Titan RTX

CPUs-based machine Specification

- CPU: 8x 1.2Ghz AMD Opteron 6176
- RAM: 192GB DIMM

8.2.2 Tables

SimpleQuestions

The sampled dataset contains 4486 questions. Based on the following results, we can preliminary conclude that GraphQA performs poorer than its competitors, but

CONVEX is leading the chart. However, we cannot, with the sampled dataset, discuss of the statistical significance. (See Table 8.1)

Competitors	F1
qAnswer	0.23375
Platypus:	0.00878
Convex	0.36367
GraphQA	0.13724

Table 8.1: Benchmark F1 results for the SimpleQuestions dataset with 4486 sampled Questions

ConvQuestions + GraphQA as multi-turn extension

The sampled datasets contains 1444 questions. Based on the following results, we can preliminary conclude that GraphQA performs similarly, but CONVEX is leading the chart. However, we cannot, with the sampled dataset, discuss of the statistical significance. (See Table 8.2)

Competitors	F1	MRR
qAnswer + GraphQA	0.05756	0.03044
Platypus + GraphQA	0.0	0.0
Convex + GraphQA	0.09452	0.05619
GraphQA + GraphQA	0.05981	0.04724

Table 8.2: Benchmark F1 and MRR results for competitors extended by GraphQA on the ConvQuestions dataset with 1444 sampled Questions

ConvQuestions + CONVEX as multi-turn extension

The sampled datasets contains 1444 questions. Based on the following results, we can preliminary conclude that GraphQA performs similarly, but CONVEX is leading the chart. However, we cannot, with the sampled dataset, discuss of the statistical significance. (See Table 8.3)

Competitors	F1	MRR
qAnswer + CONVEX	0.03257	0.01964
Platypus + CONVEX	0.0	0.0
Convex + CONVEX	0.12459	0.08445
GraphQA + CONVEX	0.08221	0.05312

Table 8.3: Benchmark F1 and MRR results for competitors extended by CONVEX on the ConvQuestion dataset with 1444 sampled Questions

Natural Language Answer generation

The sampled datasets contains 10 correct questions-answer tuples. The user were required to give a satisfaction categorization grade from 1 “Bad” to 5 (“Good”), 0 being “No Opinon”. Based on the following results, we can preliminary conclude that the NL answer satisfactory to the user. However, we cannot, with the sampled dataset, discuss of the statistical significance. (See Table 8.4)

8.2.3 Question-Answering Results conclusion

GraphQA is not in the current do not perform better than CONVEX or qAnswer. We also observe that the results for Platypus are way below its competitors; a reason could be due to the small sample dataset used. We also could retrieve the average computing time for GraphQA, which is 182 seconds for an average of 5 seconds for other competitors.

8.2.4 Natural Language Answers Results conclusion

We observed that for the answer that GraphQA can answer, the NL answers are generated with a relative quality. All 5 testers testers enjoyed the overall experience. We are satisfied with the results, even if the answers are more descriptive than natural; however, we do not believe that the showroom made with pre-trained language models is a sign of quality and overall success. We end up with an average score of 4.14 / 5, an exciting score.

Questions	Natural Language Answers	Avg. Score
Which actor voiced the Unicorn in The Last Unicorn?	Mia Farrow, the voice actor of The Last Unicorn in a character role as The Unicorn.	4.6
Who is the wife of Barack Obama?	Michelle Obama, first spouse of Barack Obama.	4.7
When was the publication date of the movie Grease?	02 August 1978, - start date of Grease.	3.4
Which is the nation of Martha Mattox	United States of America: first country of citizenship of Martha Mattox	4.4
Where did roger marquis die	Holyoke is the place of birth of Roger Marquis and his given name is Roger.	3.2
Who is the author that wrote the book Moby Dick	Herman Melville, the author of Moby'dick.	4.2
of what nationality is ken mcgoogan	Canada is his country of citizenship of Ken McGoogan.	4.9
whats the name of the organization that was founded by,frei otto	I don't know.	5
which stadium do the wests tigers play in	Stadium Australia is the home venue of Wests Tigers.	5
By whom was Misery written?	Commune of France, an instance of national Misery.	2

Table 8.4: Benchmark the score for the dataset with 10 sampled working Question-Answer (tuples). The score is averaged from a grade from 0 to 5.

Chapter 9

Project Management

In this chapter, we evaluate the overall project management during the thesis. Indeed, as defined in the initial specifications, the milestones were meant to be adjustable based on the project iterations. Our primary constraint for the Master's Thesis was the time: indeed, the project is formally framed to start on the 17th of September 2019 and end on 7th of February 2020, for a total of amount of 900 hours. To evaluate the project management, we plan three steps; first, we take a high-level overview and comment on the two project phases, the State of the Art (SOTA) research and GraphQA as our research contribution. Secondly, we will review and reflect on the initial specification, and finally conclude on the overall management.

9.1 High-Level Overview

Initially defined as *Back to Level* and *Diving into the Subject*, the two phases had the same meaning overall to, what we believe to be, our academic vision defined as Research and Contribution. In this chapter, we take a step back to visualize the entire work done as a whole to summarise the exciting adventure of our first academic research.

9.1.1 State-of-the-Art Research

Our first step to avoid being overwhelmed with knowledge from the most advanced NLP papers in the field of QA systems and Generative Systems (GSs) was to plan the research. As we did not have the tools to understand the papers properly, we decided to define a workflow to gather valuable information, such as the initial tools to get started with more complicated techniques. The listing below shows our procedure.

- Get up to date with the NLP technologies used at our lab, *iCoSys*.
- Explore community-made curated lists[1].
- Subscribe to various specilized social medias to stay informed of the latest NLP breakthroughs [2].
- Read reviews and article summarises of recent papers[3].

[1] *Awesome* NLP lists from `github.com`

[2] Examples from `reddit.com` /r/MachineLearning, /r/LanguageTechnology, /r/deeplearning

[3] Particularly from community based `medium.com` articles

- Deeply analyse the latest breakthrough papers and read all the mentioned paper.
- Filter and read the latest preprints [4].

Using this workflow, we could in 6 weeks read about 40 papers and gently examine 40 others, which we believe gave us an approximately fair overview of the NLP field, and particularly of the QA systems and GSs.

9.1.2 Research Contribution

Based on the accumulated knowledge from the SOTA research, we could analyze the current techniques used and their applications in the field of NLP, in particular, for QA systems and GSs. Which helped us define a scope for the project that would make sense in the scope of a Master's Thesis to contribute to NLP. Gladly, our constraint to use Wikidata KB could sharpen the possible contributions. As stated in our analysis 6, we went through multiple brainstorming and project iterations to get to GraphQA. From a management point of view, it appears that we respected the initial planning and honored the objectives defined in the initial project specification. Finally, we believe that our work could contribute to NLP, making the second phase as a success.

9.2 Specification Review

In this section, we review the original specifications by adding comments as a retrospective approach. We keep the structure and often paraphrase the original content. To improve reading, when content is reused or paraphrased, we set it in *italic* and in **bold** for comments. Additionally, for use the ✓and × bullets to mark items as realized or not realized.

9.2.1 Intrinsic Objectives

Primaries

In this section we presented the tasks that we believed to be essential to get started with the master's thesis.

- ✓ *Propose a project specification and planning.*
- ✓ *Analyze the SOTA of existing technologies and technics of QA systems and Generative AI.*
- ✓ *Overview digital transformation in journalism* **Even if we did the study, we did not include the search as the project shifted toward chatbots and NLP.**
- ✓ *Review the current status of the AI-News project.*
- ✓ *Document the study and write the thesis.*

9.2.2 Fact-based Question-Answering Chatbot Objectives

The first objective is to make, based on the State of the Art (SOTA), an algorithm that takes a question as input and outputs a response, as illustrated on Figure 9.1

[4] Most of the articles are coming from `arxiv.com` and `aclweb.org`

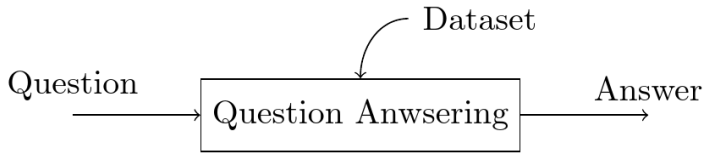


Figure 9.1: Suggested QA diagram

Primaries

- ✓ *Select existing papers and projects treating the subject as a starting point.*
- ✓ *Identify relevant datasets.*
- ✓ *Develop one or more POC.*
- ✓ *Test and evaluate solutions.*
- ✓ *Suggest improvements, possible continuation, and future outcomes.*

Secondaries

- × *Extend the QA chatbot using "tailored" knowledge, e.g., Model Fine-Tuning with press content.* **As mentioned in the final notes from the GraphQA chapter 7, this item can be extrapolated to GraphQA by adding a fine-tuned pretrain language model for a multi-brains approach to reach a consensus-based answer.**

9.2.3 Natural Language Question Answering Chatbot Objectives

The second objective was to extend the output from the QA system, from the first objective, by enhancing the answers and generate human-like sentences from the enhanced answers. The initial vision for this objective is as illustrated in Figure 9.2, a two parts system. The Enricher enriches the answer from the QA system, e.g. using a knowledge base[5]*. The Generator aims at creating readable text from the enriched answer. Besides, we could also use user profiles*[6] *as input to those two parts.*

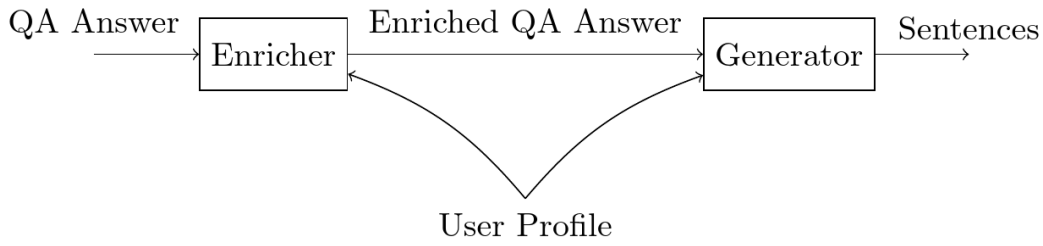


Figure 9.2: Suggested Generative QA diagram

Primaries

- ✓ *Investigate a rule-based system for keyword enrichment.*
- ✓ *Generate sentences with keywords.*

[5]Wikidata.org, a Freebase-based (Bollacker et al., 2008) knowledge base or Google's Knowledge Graphs (Singhal, 2012)

[6]Fictive profiles in the context of the thesis

- ✓ *Identify relevant datasets.*
- ✓ *Develop one or more POC.*
- ✓ *Test and evaluate solutions.*
- ✓ *Suggest improvements, possible continuation, and future outcomes.*

Secondaries

- ✓ *Use advanced strategies to enrich keywords.*
- ✓ *Use advanced text generation technics such as GTP-2*[7] *.*
- × *Use user profiles to customize the outputs.* **This item is mentioned in the final notes from the GraphQA chapter 7. GraphQA could build long-term context graphs for each user to hold their preferences. It could hold particular interests (entities), injected for the user each time a new Sub-Knowledge Graph is generated. The result would be that GraphQA will try to find a path to the answer using the users injected interests.**

9.2.4 Objectives Retrospective

For the objectives, we indeed honored the primary functions and even could add secondary functions to the GraphQA. Even if initial planned, we did not build two distinct POCs. Indeed, we started with an hybrid model combining the required features into a single POC (see Figure 9.3).

9.2.5 Methodologies

For consistency, the project was separated into two methodological parts. In the first third, as the project targets information gathering and self-study, we used a standard sequential project management methodology. For the next two-thirds of the project, we used an agile methodology to perform incremental progress while exploring.

Back to level Milestones

First third of the study, from 16.09.19 to 25.10.19 (6 weeks).

✓M1. *Initial MT plan and project specification*

✓M2. *Review the SOTA for the NLP and NLU technologies and refine the plan if needed.*

Diving into the subject Milestones

From 28.10.19 to 07.02.20 (13 weeks), the following two-third of the work is composed of 6 sprints of two weeks each and one week to finalize the thesis.

✓M3. *Basic QA Chatbot*

✓M4. *Evaluation of basic QA Chatbot*

✓M5. *Basic generative QA Chatbot*

✓M6. *Evaluation of basic generative QA Chatbot*

[7] OpenAI's GTP-2 Algorithm (Radford et al., 2018)

9.2.6 Initial Gantt

The Figure 9.4 represents the chart for the initial plan.

9.2.7 Methologies Retrospective

The two phases split were respected from a methodological and temporal point of view; however, the objectives hybridization (see Figure 9.3) made the milestones slightly altered as the evaluation milestones M4 and M6 are combined.

9.3 Management Conclusion

We believe that it is important to note that even if the objectives and the results are positives, it is difficult from a management point of view to validate the statement that the end results justify the means, which we think happened in the scope of our master's thesis. Indeed, even if we enjoyed every minute, we did massive overtime for the project to reach our objectives, which means that either the initial project scope or the post-analysis redefined scope was too large for our time constraint. We blame the rescoping as the project shifted toward an understudied field of NLP and QA systems, which made us notice that we could define a potential new field of the NLP research. On a final management note, even if from an industrial point of view, the current overtime would not be easily accepted. In our case; however, from an academic point of view, we justify our overflow as passionate dedication and as a fair attitude to contribute to research.

9.3.1 Final Milestones

M1. Initial MT plan and project specification

M2. Review the SOTA for the NLP and NLU technologies and refine the plan if needed.

M3. GraphQA 1 (see Chapter 7.3)

M4. GraphQA 2 (see Chapter 7.3.4)

M5. GraphQA 3 (see Chapter 7.4)

M6. Evaluation of basic generative QA Chatbot

M7. Turn in Master's Thesis

9.3.2 Effective Gantt

The Figure 9.5 represents the chart for for the effective plan.

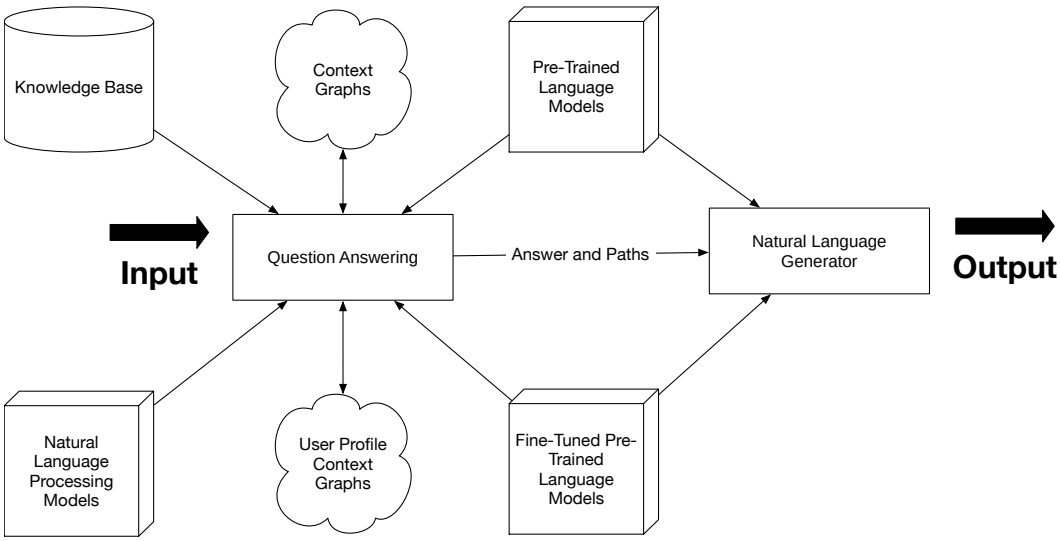


Figure 9.3: Suggested Generative QA diagram

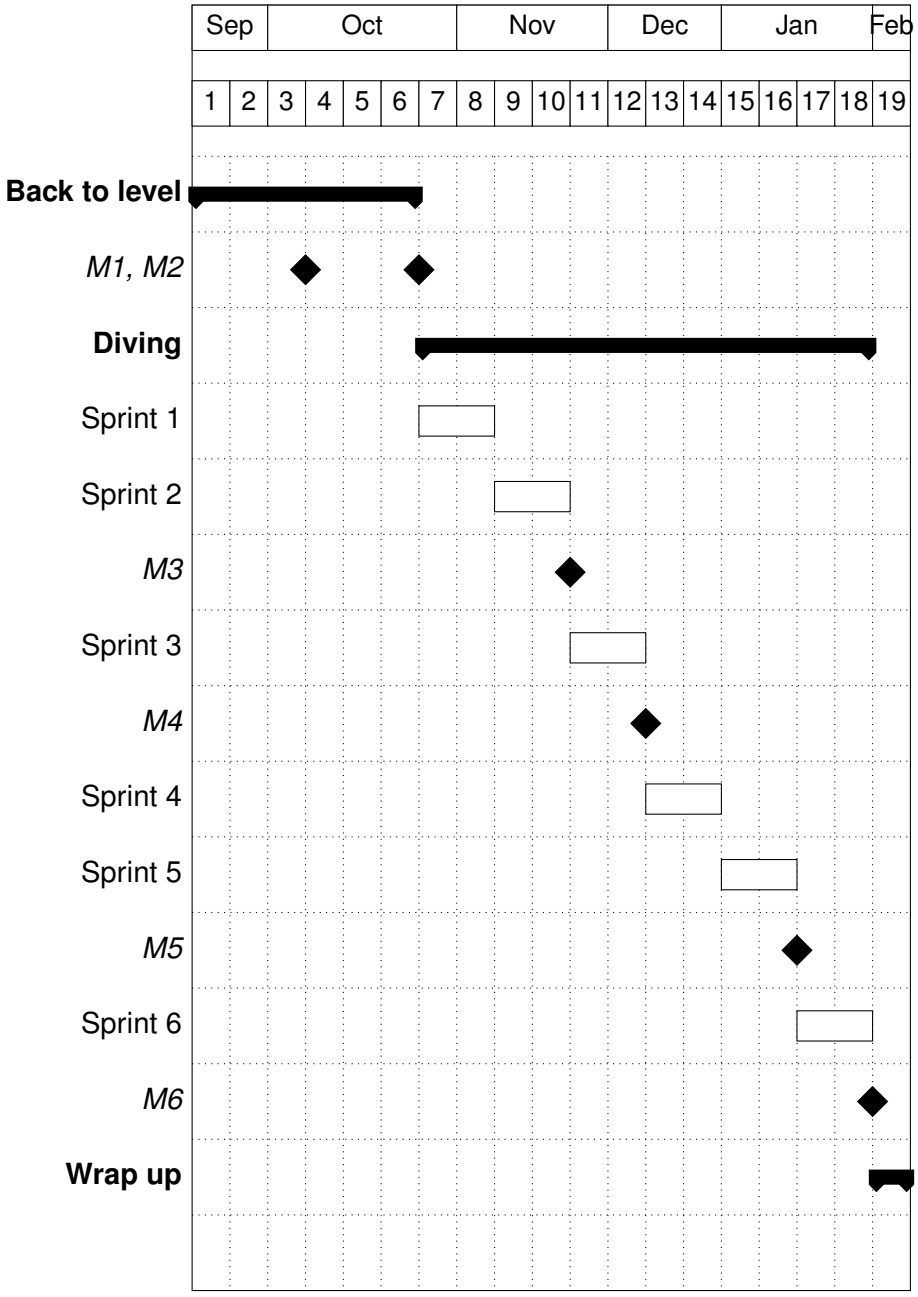


Figure 9.4: Initial Gantt Chart

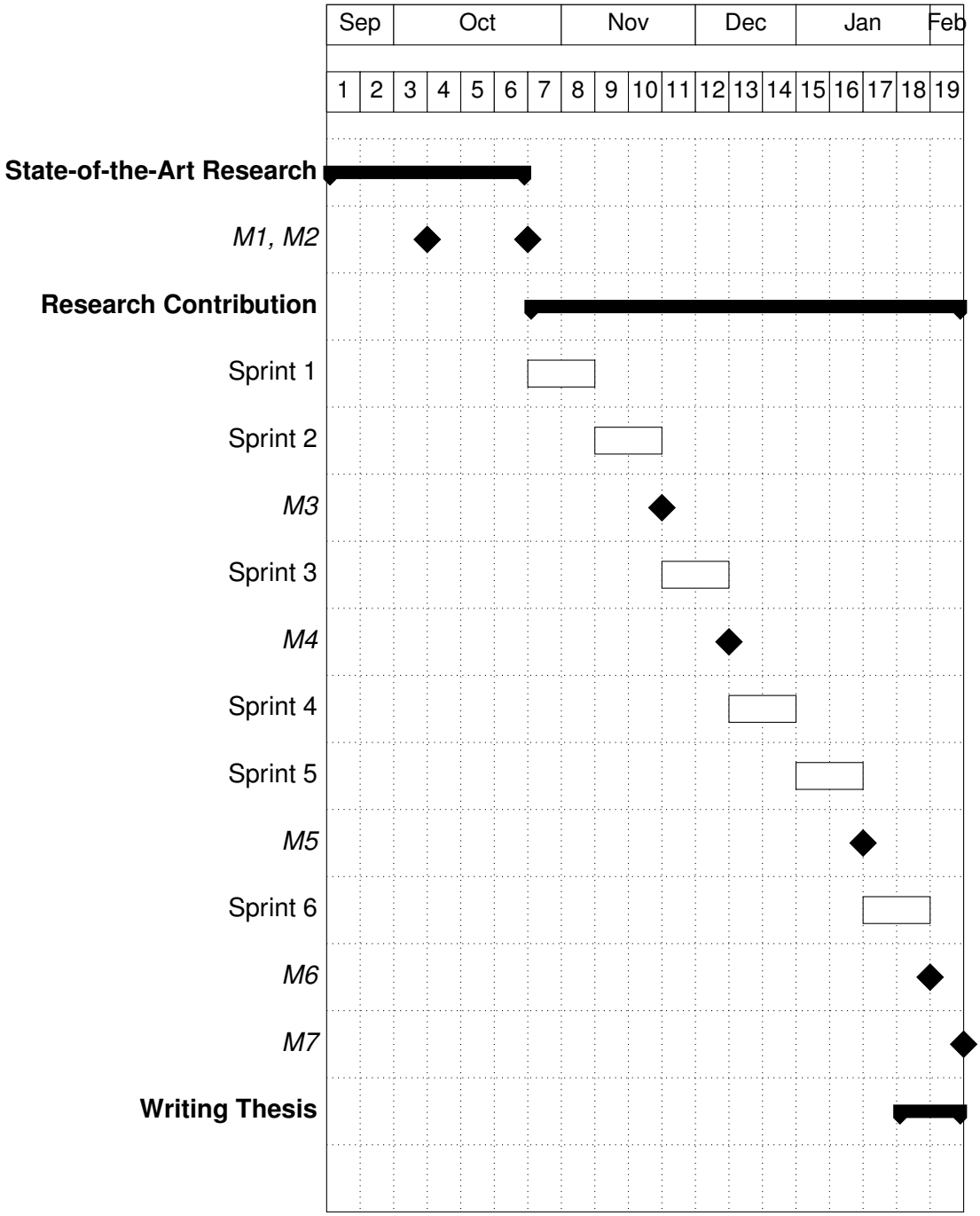


Figure 9.5: Effective Gantt Chart

Chapter 10

Discussion

The following chapter summarizes our state of mind about the project. As we choose a storytelling approach for our Master's Thesis, discussions were diluted within all chapters making the traditional academic discussion chapter, the only place to discuss our humble opinions. First of all, we want to remind that GraphQA is a POC, and its performances are not particularly important, as it is the project itself, with its original approaches that must be evaluated. As an observation, we are using SOTA technologies mostly released in late 2019 (and even for some, such as CONVEX was released during our last SOTA sprint), reinforcing our statement that NLP is driving a lot of interest lately. To structure this chapter, we will mostly talk about our main competitor and its dataset. Indeed, we indirectly spent the Master's Thesis analyzing their work, as GraphQA is using a similar sub-knowledge graph approach. We believed that it is important to discuss how our competitors perform. We will finally overview what we learned from our overall research and threw a few questions that are left to answer.

10.1 ConvQuestions Dataset

As we praised this dataset the whole thesis, we believe for good reason, as it a SOTA in the field of Multi-Hop and multi-turns. However, in this section we will overview the dataset their contribution to NLP, the limitation, and the required improvement for future Multi-Hop and multi-turns conversational datasets.

10.1.1 Data Augmentation

As a contribution, ConvQuestion, they suggested a new data augmentation approach for NLP. Data augmentation is very popular in the field of Computer Vision; however, it is often a bad solution in NLP as it often uses computer-generated paraphrasing by applying Word Embedding similarities. With ConvQuestion, the authors took a clever shift by asking humans to generate the paraphrasing, with the constraint that the paraphrased sentence must be semantically equivalent and interchangeable with its original sentence. Besides, ConvQuestions guarantees that the conversations are not permuted to not alter nested questions. It won't surprise us that QA evaluation-based datasets such as SQuAD will use this method to enhance their datasets. On a final note, ConvQuestions is a relatively small dataset with its 11'200 questions compared to the 150'000 questions from SQuAD 2.0;

however, the plot twist is that ConvQuestion used a total of 1'750 unique questions to generate their dataset with only a single permutation per questions.

Paraphrasing examples

Who is the author of the Harry Potter series? Who wrote Harry Potter?
What was the year of publication for the first book? When was it first published?
Title of the first book? The first book was called what?
What country was the book set in? It was set in what country?
Which book has the highest page count? What's the longest book?

10.1.2 Human Errors

The main limitation we noticed is the crowdsourcing itself. Even if this solution is at the time of writing, probably the best to generate NLP datasets, Mechanical Truckers often make mistakes. Indeed, it is difficult for large datasets to have no mistakes; it is human nature to get distracted, particularly for repetitive tasks such as being a Mechanical Trucker. A common inattentiveness we noticed, in addition to laziness while paraphrasing (see the previous subsection), is that the truckers do answering mistakes and are not always respecting the format guidelines, implying 32 (2^5) wrong question-answers tuples for a single mistake. For a small dataset such as ConvQuestion, the ratio of unanswerable tuples can rise dramatically and induce important bias while training models. We regret that the author didn't take the time to review correctly their relatively small dataset, or implement a crowd-sourced cross-checking protocol like *Google* did for their Natural Dialogue dataset (Kwiatkowski et al., 2019).

Mechanical Trucker error examples

Answering Mistake In this situation, the Trucker provided the wrong answer to the question.
Question: When did the first The Fast and the Furious film come out?
Expected answer: 1955
Wrong answer: 22 June 2001

Inattentiveness For this example we extrapolate the expected answer from the full NL answer span "The first film came out 22 June 2001.".
Question: When did the first The Fast and the Furious film come out?
Expected answer: 22 June 2001
Trucker's answer: https://www.wikidata.org/wiki/Q155476

Not Respecting the Answer Guideline In this example, the Trucker did not providing a NL answer span.
Question: When was he born?
Answer: 1 August 1819
Answer Span: 1 August 1819

10.1.3 GraphQA

The previous sections would probably not exist if GraphQA didn't find them. Indeed, as we were monitoring GraphQA answers, we noticed that sometimes the answers were correct, but the benchmark said otherwise. To our regrets, due to time constraints, we could not go through the whole dataset; however, this observation implies that an application for GraphQA and its competitors could be to Oracle crowdsourced datasets.

10.2 CONVEX

In our opinion, CONVEX is a pioneer in the field of sub-knowledge graphs and we amire their approach as they inspired GraphQA. They were chosen during our analysis as a starting project for our QA system and planned to enhance their work. However, as stated in the GraphQA chapter 7, we took another path and, in the following sections, we want to discuss this decision and discuss their work as our main competitor.

Initial Issues

Our initial grip with CONVEX code was not the smoothest, as the authors provided their code in the state with few refactoring bugs. We contacted the authors and debugged the code together; then, we decided to fork the project and continue on our own by fixing and adapting the code to our needs as GraphQA progressed. After the project investigation, we noticed that we could not reproduce the results in the paper. We contacted the authors to notify the issues: they explained that they used an undisclosed split of the dataset to obtain their results, and confirmed that the code works as expected.

Q0 Problems in more details

Based on the previous authors' statement, we started to investigate further how CONVEX answers questions. As the paper acknowledges that Q0 is their bottleneck, we decided to explore further why it is indeed a bottleneck; we observed that often the answers are lucky guesses as it returns the first Object from a multi solutions Subject-Predicate tuples. E.g., "Who is the actor of this role in this movie?": by design, the query would return the list of all actors to the requested movie, which we call hypothesis in GraphQA. CONVEX, in this particular case, will return the first element. Their Named-Entity Recognition depends on TAGME, which is proprietary, meaning that it's up to a black box to return the Wikidata entity identified based on the context of the sentence. It is indeed hard to trust a proprietary system to return the right entities when multiple entities have the same name.

GraphQA from Scratch

Based on the previous analysis, it was clear that building an open-source module for Q0 was the priority task. We investigated in details how to integrate our module to the existing system, by going deeper into the code and understanding all functions. Aside from odd computations, we could not find the context-graph generation as described in the paper, so we asked the author about it. They explained that the part we are looking for, which we imagined to be the most exciting, were written

in the paper with made-up values for motivational purposes. From this point, we lost all motivation to enhance CONVEX as its key feature was not implemented and required an architecture remodeling to get it to work. Annoyingly, the clock was ticking, and we could not step back to evaluate a new solution. Based on the CONVEX's motivational purpose feature, we built from scratch an open-source architecture handling scoring by design. (See chapter 7)

10.3 Lesson Learned

This project gave us lessons about the academic experience and the field of NLP.

10.3.1 Only trust yourself

Preprints Some papers are good, but they are mostly bad as they are either repetitive or pointless with name-dropping.

Published in journal Mostly good, they often do nested research by publishing a paper at each iteration of their project.

Published in conferences Often useful, but be careful with the origin country.

Repetitivity We realized that after a reasonable amount of paper read, we are not surprised anymore about the breakthroughs.

Never trust claims in articles

- Would it be possible to build a tool to parse articles and generate context graphs automatically?
- Investigate the correlation between human thinking and reasoning with knowledge graphs representation.
- Compare knowledge graphs representation to IR and access structure.
- Compare Wordnet and Word Embedding in the scope of GraphQA.

Chapter 11

Conclusions

This chapter concludes our Master's Thesis. During our journey, we had the chance to get a glance of the academic life by starting with an in-depth review of the current SOTA for Chatbots and NLP techniques, and finishing by a potential contribution to the NLP field. Driven by passion, we cumulated well over 900 hours, but hope that our work as the purpose to enlarge, what we believe, a relatively unexplored field for QA and potentially a new NLP approach for grounded learning. We predict to see pre-trained language models and pre-trained generative language models being fine-tuned on sub-knowledge graphs shortly.

Even if GraphQA, in its final form, is different from the expected version from the analysis, we believe that we could shift the project in a better direction as an overall contribution to NLP. Indeed, we achieved at building a POC of a Multi-Hop and multi-turns conversations Sub-KG QA Chatbots outputting NL answer, by orchestrating multiple models with Zero-Shot Learning approach, which is not currently a trend in the field of NLP. We hope that our contribution will serve newcomers in grounded tasks. Note that our work can be used for training models to perform similarly on subgraphs.

11.1 Final words

We observed a valuable statement with this project: simpler a concept or a model is, the best it is. Compared with nature, the simplest survives the best. In NLP at the time of writing, Transformers are currently leading with their relatively simple architecture; we believe that their success results from the simplicity.

We wish that additional work would be done towards a multiple-brain strategy with grounded tasks, and we believe that GraphQA could be taken as an example to break complexes tasks such as NLP into smaller and simpler tasks to handle by models.

A lot of adaptation is being made from a ML field to others. However, we believe that instead of adapting technologies from other fields, the next breakthrough is the combination of various ML fields such as Machine Vision, or Sensory Robotics. Building a Multi-Domain Grounded Task Generation model using Grounded Learning is set to become a new standard in Machine Reasoning (MR) and Machine Understanding (MU).

We conclude by assuming that our GraphQA approach is slightly analogically to human's knowledge representation and reasoning, as we use multiple modules to accomplish a composite result. However, we hope that more research will be

done on subgraphs and perform a comparative study with the way humans are processing knowledge.

Bibliography

BAHDANAU, Dzmitry; CHO, Kyunghyun; BENGIO, Yoshua, 2014. Neural machine

DEVLIN, Jacob; CHANG, Ming-Wei; LEE, Kenton; TOUTANOVA, Kristina, 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186. Available from DOI: 10.18653/v1/N19-1423.

DIEFENBACH, Dennis; TANON, Thomas Pellissier; SINGH, Kamal Deep; MARET, Pierre, 2017. Question Answering Benchmarks for Wikidata. In: *Question Answering Benchmarks for Wikidata*. *Proceedings of the ISWC 2017 Posters & Demonstrations and Industry Tracks co-located with 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 23rd - to - 25th, 2017.* Available also from: http://ceur-ws.org/Vol-1963/paper555.pdf.

DONG, Li; WEI, Furu; ZHOU, Ming; XU, Ke, 2015. Question answering over freebase with multi-column convolutional neural networks. *ACL-IJCNLP 2015 - 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Proceedings of the Conference*. Vol. 1, pp. 260–269. ISBN 9781941643723.

DUNLOP, Michal Wallace & George, 1999. *Eliza, the Rogerian Therapist* [http://psych.fullerton.edu/mbirnbaum/psych101/Eliza.htm]. (Accessed on 10/09/2019).

ELOKENCE, Scimob, 2007. *Akinator* [https://en.wikipedia.org/wiki/Akinator]. (Accessed on 10/09/2019).

FERRAGINA, Paolo; SCAIELLA, Ugo, 2010. TAGME: on-the-fly annotation of short text fragments (by wikipedia entities). In: *TAGME: on-the-fly annotation of short text fragments (by wikipedia entities)*. *Proceedings of the 19th ACM International Conference on Conference on Information and Knowledge Management*. ACM, pp. 1625–1628. Available from DOI: 10.1145/1871437.1871689.

FUTURISM, LLC, 2016. *The History of Chatbots Infographic* [https://futurism.com/images/the-history-of-chatbots-infographic]. (Accessed on 10/09/2019).

GEHRMANN, Sebastian; STROBELT, Hendrik; RUSH, Alexander M., 2019. GLTR: Statistical Detection and Visualization of Generated Text. *CoRR*. Vol. abs/1906.04043. Available from arXiv: 1906.04043.

GITTENS, Alex; ACHLIOPTAS, Dimitris; MAHONEY, Michael W., 2017. Skip-Gram â^' Zipf + Uniform = Vector Additivity. In: *Skip-Gram â^' Zipf + Uniform = Vector Additivity*. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 69–76. Available from DOI: 10.18653/v1/P17-1007.

GRAEFE, Andreas; HAIM, Mario; HAARMANN, Bastian; BROSIUS, Hans-Bernd, 2018. Readers' perception of computer-generated news: Credibility, expertise, and readability. *Journalism*. Vol. 19, no. 5, pp. 595–610. Available from DOI: 10.1177/1464884916641269.

HAGBERG, Aric A.; SCHULT, Daniel A.; SWART, Pieter J., 2008. Exploring Network Structure, Dynamics, and Function using NetworkX. In: VAROQUAUX, Gaël; VAUGHT, Travis; MILLMAN, Jarrod (eds.). *Proceedings of the 7th Python in Science Conference*. Pasadena, CA USA, pp. 11 –15.

HAO, Yazhou; ZHENG, Qinghua; LAN, Yangyang; LI, Yufei; WANG, Meng; WANG, Sen; LI, Chen, 2017. Improving Chinese Sentiment Analysis via Segmentation-Based Representation Using Parallel CNN. In: CONG, Gao; PENG, Wen-Chih; ZHANG, Wei Emma; LI, Chengliang; SUN, Aixin (eds.). *Advanced Data Mining and Applications*. Cham: Springer International Publishing, pp. 668–680. ISBN 978-3-319-69179-4.

HERBELOT, Aurélie; BARONI, Marco, 2017. High-risk learning: acquiring new word vectors from tiny data. *CoRR*. Vol. abs/1707.06556. Available from arXiv: 1707.06556.

HONNIBAL, Matthew; MONTANI, Ines, 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. *To appear*.

JU, Ying; ZHAO, Fubang; CHEN, Shijie; ZHENG, Bowen; YANG, Xuefeng; LIU, Yunfeng, 2019. Technical report on Conversational Question Answering. *arXiv e-prints*, pp. arXiv:1909.10772. Available from arXiv: 1909.10772 [cs.CL].

KARRAS, Tero; LAINE, Samuli; AITTALA, Miika; HELLSTEN, Janne; LEHTINEN, Jaakko; AILA, Timo, 2019. Analyzing and Improving the Image Quality of StyleGAN. *CoRR*. Vol. abs/1912.04958.

KELNAR, David, 2019. The State of AI, pp. 151. Available also from: https://www.stateofai2019.com/summary/.

KEON, 2017. *keon / awesome-nlp* [https://github.com/keon/awesome-nlp]. GitHub.

KIM, Yoon; JERNITE, Yacine; SONTAG, David A.; RUSH, Alexander M., 2015. Character-Aware Neural Language Models. *CoRR*. Vol. abs/1508.06615. Available from arXiv: 1508.06615.

KUMAR, Ankit; IRSOY, Ozan; SU, Jonathan; BRADBURY, James; ENGLISH, Robert; PIERCE, Brian; ONDRUSKA, Peter; GULRAJANI, Ishaan; SOCHER, Richard, 2015. Ask Me Anything: Dynamic Memory Networks for Natural Language Processing. *CoRR*. Vol. abs/1506.07285. Available from arXiv: 1506.07285.

KWIATKOWSKI, Tom et al., 2019. Natural Questions: a Benchmark for Question Answering Research. *Transactions of the Association of Computational Linguistics*.

LIU, Yinhan et al., 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*. Vol. abs/1907.11692. Available from arXiv: 1907.11692.

LOPEZ, Vanessa; UNGER, Christina; CIMIANO, Philipp; MOTTA, Enrico, 2013. Evaluating question answering over linked data. *Web Semantics Science Services And Agents On The World Wide Web*. Vol. 21, pp. 3–13. ISSN 1570-8268. Available from DOI: 10.1016/j.websem.2013.05.006.

LOWE, Ryan; POW, Nissan; SERBAN, Iulian; PINEAU, Joelle, 2015. The Ubuntu Dialogue Corpus: A Large Dataset for Research in Unstructured Multi-Turn Dialogue Systems. *CoRR*. Vol. abs/1506.08909. Available from arXiv: 1506.08909.

LUCY, Li; GAUTHIER, Jon, 2017. Are distributional representations ready for the real world? Evaluating word vectors for grounded perceptual meaning. *CoRR*. Vol. abs/1705.11168. Available from arXiv: 1705.11168.

MA, Yukun; CAMBRIA, Erik; GAO, Sa, 2016. Label Embedding for Zero-shot Fine-grained Named Entity Typing. In: *Label Embedding for Zero-shot Fine-grained Named Entity Typing. Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. Osaka, Japan: The COLING 2016 Organizing Committee, pp. 171–180. Available also from: https://www.aclweb.org/anthology/C16-1017.

MARIO ARIAS GALLEGO Javier D. Fernández, Claudio Gutierrez Miguel A. Martínez-Prieto Axel Polleres, 2017. *RDF/HDT* [http://www.rdfhdt.org/]. (Accessed on 10/09/2019).

MIKOLOV, Tomas; CHEN, Kai; CORRADO, Greg; DEAN, Jeffrey, 2013. Efficient Estimation of Word Representations in Vector Space. *arXiv e-prints*, pp. arXiv:1301.3781. Available from arXiv: 1301.3781 [cs.CL].

MILLER, Alexander H.; FISCH, Adam; DODGE, Jesse; KARIMI, Amir-Hossein; BORDES, Antoine; WESTON, Jason, 2016. Key-Value Memory Networks for Directly Reading Documents. *CoRR*. Vol. abs/1606.03126. Available from arXiv: 1606.03126.

PELLISSIER TANON, Thomas; ASSUNÇÃO, Marcos Dias de; CARON, Eddy; M. SUCHANEK, Fabian, 2018. Demoing Platypus – was A Multilingual Question Answering Platform for Wikidata. In: *Demoing Platypus – was A Multilingual Question Answering Platform for Wikidata. ESWC 2018 - Extended Semantic Web Conference*. Heraklion, Greece. hal-01824972.

PENNINGTON, Jeffrey; SOCHER, Richard; MANNING, Christopher D., 2014. GloVe: Global Vectors for Word Representation, pp. 1532–1543. Available also from: http://www.aclweb.org/anthology/D14-1162.

PETERS, Matthew E.; NEUMANN, Mark; IYYER, Mohit; GARDNER, Matt; CLARK, Christopher; LEE, Kenton; ZETTLEMOYER, Luke, 2018. Deep contextualized word representations. *CoRR*. Vol. abs/1802.05365. Available from arXiv: 1802.05365.

PINTER, Yuval; GUTHRIE, Robert; EISENSTEIN, Jacob, 2017. Mimicking Word Embeddings using Subword RNNs. *CoRR*. Vol. abs/1707.06961. Available from arXiv: 1707.06961.

RADFORD, Alec; WU, Jeffrey; CHILD, Rewon; LUAN, David; AMODEI, Dario; SUTSKEVER, Ilya, 2018. Language Models are Unsupervised Multitask Learners.

RAJPURKAR, Pranav; ZHANG, Jian; LOPYREV, Konstantin; LIANG, Percy, 2016. SQuAD: 100, 000+ Questions for Machine Comprehension of Text. *CoRR*. Vol. abs/1606.05250. Available from arXiv: 1606.05250.

RAJPURKAR, Pranav; JIA, Robin; LIANG, Percy, 2018a. Know What You Don't Know: Unanswerable Questions for SQuAD. In: *Know What You Don't Know: Unanswerable Questions for SQuAD. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 784–789. Available from DOI: 10.18653/v1/P18-2124.

RAJPURKAR, Pranav; JIA, Robin; LIANG, Percy, 2018b. Know What You Don't Know: Unanswerable Questions for SQuAD. *CoRR*. Vol. abs/1806.03822. Available from arXiv: 1806.03822.

RAMACHANDRAN, Prajit; PARMAR, Niki; VASWANI, Ashish; BELLO, Irwan; LEVSKAYA, Anselm; SHLENS, Jonathon, 2019. Stand-Alone Self-Attention in Vision Models. *CoRR*. Vol. abs/1906.05909. Available from arXiv: 1906.05909.

REDDY, Siva; CHEN, Danqi; MANNING, Christopher D., 2018. CoQA: A Conversational Question Answering Challenge. *CoRR*. Vol. abs/1808.07042. Available from arXiv: 1808.07042.

ROWSELL-JONES, Andy; HOWARD, Chris, 2019. *2019 CIO Survey: CIOs Have Awoken to the Importance of AI* [https://www.gartner.com/en/documents/3897266/2019-cio-survey-cios-have-awoken-to-the-importance-of-ai]. (Accessed on 10/09/2019).

SANTOS, Cícero Nogueira dos; ZADROZNY, Bianca, 2014. Learning Character-level Representations for Part-of-Speech Tagging. In: *Learning Character-level Representations for Part-of-Speech Tagging. ICML*. JMLR.org, vol. 32, pp. 1818–1826. JMLR Workshop and Conference Proceedings. Available also from: http://dblp.uni-trier.de/db/conf/icml/icml2014.html#SantosZ14.

SAS, Blackbird, 2014. *HelloJam.fr* [https://www.hellojam.fr]. (Accessed on 10/09/2019).

SEO, Min Joon; KEMBHAVI, Aniruddha; FARHADI, Ali; HAJISHIRZI, Hannaneh, 2016. Bidirectional Attention Flow for Machine Comprehension. *CoRR*. Vol. abs/1611.01603. Available from arXiv: 1611.01603.

SEVERYN, Aliaksei; MOSCHITTI, Alessandro, 2016. Modeling Relational Information in Question-Answer Pairs with Convolutional Neural Networks. Available from arXiv: 1604.01178.

SHEN, Tao; GENG, Xiubo; QIN, Tao; GUO, Daya; TANG, Duyu; DUAN, Nan; LONG, Guodong; JIANG, Daxin, 2019. Multi-Task Learning for Conversational Question Answering over a Large-Scale Knowledge Base. *arXiv e-prints*, pp. arXiv:1910.05069. Available from arXiv: 1910.05069 [cs.CL].

SINGHAL, Amit, 2012. *Official Google Blog: Introducing the Knowledge Graph: things, not strings* [https://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html]. (Accessed on 10/09/2019).

TANG, Duyu; QIN, Bing; LIU, Ting, 2016. Aspect Level Sentiment Classification with Deep Memory Network. *CoRR*. Vol. abs/1605.08900. Available from arXiv: 1605.08900.

TRISCHLER, Adam; WANG, Tong; YUAN, Xingdi; HARRIS, Justin; SORDONI, Alessandro; BACHMAN, Philip; SULEMAN, Kaheer, 2016. NewsQA: A Machine Comprehension Dataset. *CoRR*. Vol. abs/1611.09830. Available from arXiv: 1611.09830.

TURING, A. M., 1950. Computing Machinery and Intelligence. *Mind*. Vol. 59, no. 236, pp. 433–460. ISSN 00264423. Available also from: http://www.jstor.org/stable/2251299.

VASWANI, Ashish; SHAZEER, Noam; PARMAR, Niki; USZKOREIT, Jakob; JONES, Llion; GOMEZ, Aidan N.; KAISER, Lukasz; POLOSUKHIN, Illia, 2017. Attention Is All You Need. *CoRR*. Vol. abs/1706.03762. Available from arXiv: 1706.03762.

WANG, Phil, 2019. *This Person Does Not Exist* [https://www.thispersondoesnotexist.com]. (Accessed on 10/09/2019).

WANG, Quan; HUANG, Pingping; WANG, Haifeng; DAI, Songtai; JIANG, Wenbin; LIU, Jing; LYU, Yajuan; ZHU, Yong; WU, Hua, 2019. CoKE: Contextualized Knowledge Graph Embedding. *arXiv e-prints*, pp. arXiv:1911.02168. Available from arXiv: 1911.02168 [cs.AI].

WESTON, Jason; CHOPRA, Sumit; BORDES, Antoine, 2015. Memory Networks. *CoRR*. Vol. abs/1410.3916.

WOLF, Thomas et al., 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv e-prints*, pp. arXiv:1910.03771. Available from arXiv: 1910.03771 [cs.CL].

YATSKAR, Mark, 2018. A Qualitative Comparison of CoQA, SQuAD 2.0 and QuAC. *CoRR*. Vol. abs/1809.10735. Available from arXiv: 1809.10735.

YIN, Wenpeng; KANN, Katharina; YU, Mo; SCHÜTZE, Hinrich, 2017. Comparative Study of CNN and RNN for Natural Language Processing. Available from arXiv: 1702.01923.

List of Figures

List of Figures

List of Tables