NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

DEPARTMENT OF
COMPUTER SCIENCE

FREDERICO MIGUEL GUERRA PAULO PEREIRA VICENTE
Bachelor in Computer Science

TALK COMMONSENSE TO ME!

ENRICHING LANGUAGE MODELS WITH COMMONSENSE KNOWLEDGE

COMPUTER SCIENCE

NOVA University Lisbon
September, 2022

DEPARTMENT OF
COMPUTER SCIENCE

TALK COMMONSENSE TO ME!

ENRICHING LANGUAGE MODELS WITH COMMONSENSE KNOWLEDGE

FREDERICO MIGUEL GUERRA PAULO PEREIRA VICENTE
Bachelor in Computer Science

Adviser: João Magalhães
Associate Professor with Habilitation, NOVA University Lisbon

Co-adviser: David Semedo
Invited Auxiliar Researcher, NOVA University Lisbon

COMPUTER SCIENCE

NOVA University Lisbon
September, 2022

Talk Commonsense To Me!

I dedicate this dissertation to my family, who always made
their impossible to make sure I could succeed and be happy :)

Acknowledgements

This dissertation would not have been possible without the support of some people, to whom I would like to show my appreciation to.

I would like to thank professor João Magalhães and professor David Semedo for supervising this dissertation.

I would, secondly, like to thank the members of the TWIZ team (our Amazon team competing in the Amazon's Alexa Challenge), for their feedback and further technical support on some key ideas.

I would like to further thank Professor Ludwig Krippahl, who for years gave me essential intuitions on tackling machine learning problems.

I would also like to thank my friends Rui Margarido and Mariana Bonito who helped me to keep sane with healing conversations and insightful meditations about both the state of the world and my work.

Furthermore, I thank my family and Marta who were the pillars of my well-being throughout this dissertation.

I would further like to thank Professor João Lourenço for providing the template for this dissertation [66].

Lastly, I would like to thank the support provided by NOVA.id.FCT and Amazon Media EU S.à r.l. through their Alexa Prize Research Sponsorship: project nº3097.

"- Falhámos a vida, menino!
- Creio que sim... Mas todo o mundo mais ou menos a falha. Isto é, falha-se sempre na realidade aquela vida que se planeou com a imaginação. Diz-se: «vou ser assim, porque a beleza está em ser assim». E nunca se é assim, é-se invariavelmente assado, como dizia o pobre marquês. Às vezes melhor, mas sempre diferente." (Eça de Queirós)

Abstract

Human cognition is exciting, it is a mesh up of several neural phenomena which really strive our ability to constantly reason and infer about the involving world. In cognitive computer science, *Commonsense Reasoning* is the terminology given to our ability to infer uncertain events and reason about Cognitive Knowledge. The introduction of Commonsense to intelligent systems has been for years desired, but the mechanism for this introduction remains a scientific jigsaw. Some, implicitly believe language understanding is enough to achieve some level of Commonsense [90]. In a less common ground, there are others who think enriching language with Knowledge Graphs might be enough for human-like reasoning [63], while there are others who believe human-like reasoning can only be truly captured with symbolic rules and logical deduction powered by Knowledge Bases, such as taxonomies and ontologies [50]. We focus on Commonsense Knowledge integration to Language Models, because we believe that this integration is a step towards a beneficial embedding of Commonsense Reasoning to interactive Intelligent Systems, such as conversational assistants.

Conversational assistants, such as **Alexa** from Amazon, are user driven systems. Thus, giving birth to a more human-like interaction is strongly desired to really capture the user's attention and empathy. We believe that such humanistic characteristics can be leveraged through the introduction of stronger Commonsense Knowledge and Reasoning to fruitfully engage with users.

To this end, we intend to introduce a new family of models, the Relation-Aware BART (RA-BART), leveraging language generation abilities of BART [51] with explicit Commonsense Knowledge extracted from Commonsense Knowledge Graphs to further extend human capabilities on these models.

We evaluate our model on three different tasks: Abstractive Question Answering, Text Generation conditioned on certain concepts and a Multi-Choice Question Answering task. We find out that, on generation tasks, RA-BART outperforms non-knowledge enriched models, however, it underperforms on the multi-choice question answering task.

Our Project can be consulted in our open source, public GitHub repository (Explicit Commonsense).

Keywords: Natural Language Generation • Commonsense Knowledge • Knowledge Graphs • BART • Transformers

Resumo

A cognição humana é entusiasmante, é uma malha de vários fenómenos neuronais que nos estimulam vivamente a capacidade de raciocinar e inferir constantemente sobre o mundo envolvente. Na ciência cognitiva computacional, o raciocínio de senso comum é a terminologia dada à nossa capacidade de inquirir sobre acontecimentos incertos e de raciocinar sobre o conhecimento cognitivo. A introdução do senso comum nos sistemas inteligentes é desejada há anos, mas o mecanismo para esta introdução continua a ser um quebra-cabeças científico. Alguns acreditam que apenas compreensão da linguagem é suficiente para alcançar o senso comum [90], num campo menos similar há outros que pensam que enriquecendo a linguagem com gráfos de conhecimento pode ser um caminho para obter um raciocínio mais semelhante ao ser humano [63], enquanto que há outros ciêntistas que acreditam que o raciocínio humano só pode ser verdadeiramente capturado com regras simbólicas e deduções lógicas alimentadas por bases de conhecimento, como taxonomias e ontologias [50]. Concentramo-nos na integração de conhecimento de censo comum em Modelos Linguísticos, acreditando que esta integração é um passo no sentido de uma incorporação benéfica no racíocinio de senso comum em Sistemas Inteligentes Interactivos, como é o caso dos assistentes de conversação.

Assistentes de conversação, como o Alexa da Amazon, são sistemas orientados aos utilizadores. Assim, dar origem a uma comunicação mais humana é fortemente desejada para captar realmente a atenção e a empatia do utilizador. Acreditamos que tais características humanísticas podem ser alavancadas por meio de uma introdução mais rica de conhecimento e raciocínio de senso comum de forma a proporcionar uma interação mais natural com o utilizador.

Para tal, pretendemos introduzir uma nova família de modelos, o Relation-Aware BART (RA-BART), alavancando as capacidades de geração de linguagem do BART [51] com conhecimento de censo comum extraído a partir de grafos de conhecimento explícito de senso comum para alargar ainda mais as capacidades humanas nestes modelos.

Avaliamos o nosso modelo em três tarefas distintas: Respostas a Perguntas Abstratas, Geração de Texto com base em conceitos e numa tarefa de Resposta a Perguntas de

Escolha Múltipla . Descobrimos que, nas tarefas de geração, o RA-BART tem um desempenho superior aos modelos sem enriquecimento de conhecimento, contudo, tem um desempenho inferior na tarefa de resposta a perguntas de múltipla escolha.

O nosso Projecto pode ser consultado no nosso repositório GitHub público, de código aberto (Explicit Commonsense).

Palavras-chave: Geração de Linguagem Natural • Conhecimento de censo comum • Grafos de Conhecimento • BART • Transformers

Contents

Appendices

Annexes

List of Figures

List of Tables

List of Listings

Glossary

Entity Entities represent a term or an object, which are normally linked with knowledge repositories. 2

Intent Refers to the end-goal a user has in their mind when typing in a question, a comment, or a request. For example, an agent can learn that words such as buy or acquire are often associated with the intent to Purchase. 2

multi-modal Referring to a mixture of many types of media (eg. Natural Language text, images, relational concepts). 1

n-gram n-grams correspond to a set of n consecutive words (e.g. The phrase: Darkness is shy and polite, has four 2-grams and three 3-grams. 28

Sequence to Sequence Refers to encoder-decoder models that learn sequential data (eg. natural language sentences) representations and attempt to generate them. xi, 6

stopwords Stopwords are a collection of words which are filtered out before or after a processing procedure regarding Natural Language text, supposedly because they add no relevant information. 37

Acronyms

AI	Artificial Intelligence 26
ANN	Artificial Neural Networks 1
API	Application Programming Interface 91, 93
CTG	Controllable Text Generation xi, xiv, 13, 14
DIY	Do It Yourself 3, 31, 89, 91, 92, 93
GPU	Graphics Processing Unit 26
GRU	Gated Recurrent Unit 6
HLP	Human Level Performance 28, 61, 63
KG	Knowledge Graph 31, 33, 34, 41, 42, 49
LSTM	Long Short Term Memory 6, 7
NLG	Natural Language Generation 13
NLP	Natural Language Processing 4, 10, 13, 27, 65
NLU	Natural Language Understanding 7, 20, 36
OWA	Open World Assumption 46
QA	Question Answering 21
RA-BART	Relation-Aware BART vii, 49
RL	Reinforcement Learning 13
RNN	Recurrent Neural Network 6

seq2seq	Sequence to Sequence 6, 7, 33, 71
SOTA	State of the art 2, 7, 15, 26, 29, 42, 70
Symbolic AI	Symbolic Artificial Intelligence 1
VL	Vision-Language 9, 24

1

Introduction

This chapter shall be regarded as an introduction to our proposed work, passing through the contributions made. Lastly, a brief overview of the document structure is covered.

1.1 Cafe Contextualisation

Human cognition is remarkably complex as it is powered by a biological *computer*: **the brain**. Yet we are far way from totally deciphering it. The brain, stunningly, is a versatile framework carrying the ability to store knowledge, to formulate thoughts and morph them, to strengthen them with communication tools like language, where thoughts can be dismantled in multi-modal finite mediums and be converted back to neural stimuli. Even though the brain is a biological black box in most aspects, it is fruitful in intelligent complex computation, which researchers have not looked away, but instead joined forces into deeply introspecting it and taking inspiration onto artificial computations. Artificial Neural Networks (ANN) were initially inspired by the brain, where the first network, the Perceptron had it's genesis in 1958 [89].

Introducing cognitive abilities to machines has then been attempted mainly through the means of Symbolic Artificial Intelligence (Symbolic AI) and ANN. Symbolic AI [75] was an attempt to create human-readable logical patterns which could reason about mundane situations, based on knowledge bases, symbols manipulation, logic rules and deduction engines.

Technical limits, however, such as knowledge building and rules system specification, led to a discouragement in the field, but with the rise of ANN, a different take on the field was born. Subsymbolic artificial intelligence builds as the field which will leverage the reasoning of logical systems with the co-occurence inference capabilities and brain-efficient inspired architectures used by ANN, promising rich computational cognitive intelligence [28].

On a different avenue, deep learning purists attempt to model approximations of knowledge and human-like capabilities using neural networks of all sorts of forms. In

2015, Yann LeCun [49], alluded to the ability of deep neural networks in solving problems where implicit patterns and relations were fundamental (eg. Natural Language and Vision). Furthermore, he addressed and explained the core of all these deep neural models: the backpropagation learning algorithm, which was for several decades put aside, but now played a key role in guiding the learning process of these deep models. He was optimistic with the way science was being pursued in the field, however, human cognition could not be narrowly solved by supervised or non-supervised neural nets, as he later pointed in his report: "A Path Towards Autonomous Machine Intelligence" [1]. In this report, he points out that there is a need for focused computational components, which are modelled together to battle several different problems in order to create a more aware, knowledgeable system.

In contrast, usually, neural systems are built to tackle tasks independently (eg. Machine Translation, Question Answering, Image classification, etc), even though there are systems which require a crossing of different tasks and knowledge sharing between them, because they are complex and offer multiple ways of interaction.

One playground, where such "intelligent" systems are much desired is in the field of conversational assistants, such as **Alexa**, designed by Amazon. Even though these assistants may seem to represent a single computational unit, they are an abstraction of multiple layers of both rule-based and deep models, empirical and automate strategies, among other components, as an attempt to better accommodate satisfaction in system's users.

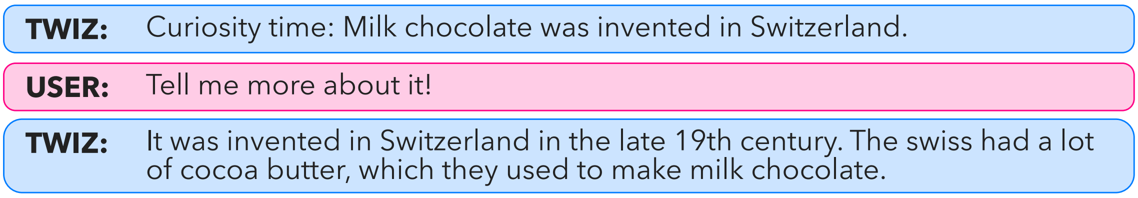
Under the hood, to be more concrete, we find several components such as: Intent detection, Entity recognition, Dialog Policy mapping, Ranking strategies, Specific task models (eg. Question-Answering, Sentiment Analysis, Text Generation, etc), among others.

All components have their relevant impact and importance on the flow of an assistant-user conversation, from which we have decided to grasp a more deeper research into the Text Generation component, having observed problems with Natural Language generation lacking obvious Commonsense.

Neural Language Models, more precisely, in the last 4 years have really become competent in generating language and in discriminant language tasks (Take a glance at the Fig 1.1 for a better view of current State of the art (SOTA) language model's (BART [51]) Natural Language generation quality).

However, these SOTA Language Models are certainly not bullet proof, neither are they indefinitely reliable. One question which ignites dispute on Language Models is whether the seemingly language inference of such models is a result of a deep meaning understanding of language concepts or whether the inference is coerced by mere word co-occurrence statistics. There has been research on this matter, for instance on the BART architecture [53], which points out that indeed the attention mechanism rooted in these

[1] A Path Towards Autonomous Machine Intelligence: https://openreview.net/pdf?id=BZ5a1r-kVsf



TWIZ: Curiosity time: Milk chocolate was invented in Switzerland.

USER: Tell me more about it!

TWIZ: It was invented in Switzerland in the late 19th century. The swiss had a lot of cocoa butter, which they used to make milk chocolate.

Figure 1.1: Adapted from our work (TWIZ team) for Alexa Prize TaskBot Challenge [59]

models tend to create a semantic understanding of concepts and textual scenarios diminishing the hypothesis that these models are nothing but good mappers at relating words in similar contexts. Language Models, can therefore adhere to the creation of an intrinsic conceptualisation of semantics, limited by the complexity of the model. One thing to consider, however, relates to the fact that language understanding transcends the aspect of building a morphed textual idea of a concept (vector representation) and even combining a set of them in a sentence. When we, humans, conceive a mental idea of a concept or a set of related concepts, we cognitively are bounded by a broader Commonsense representation of them and a reasoning system which relates them. **Machine Commonsense**, is a subset of human knowledge, which usually is absent in corpus of texts, due to *human reporting bias* (Humans tend to not state the obvious). Commonsense is multi-modal by definition and can be dismounted in several layers, such as: concepts' textual view, concepts' abstract visual representations, concepts' abstract sounds, the symbolic entailment underlying knowledge, social conventions, any representation of knowledge explicitly consulted or deduced over memory knowledge, among others.

Provided a vast grasp over the conceptualisation of Commonsense, it is not realistic to expect that a single model trained upon linguistic corpus to be able to extract the human thoughtful experience of understanding and reasoning about matters, nor consequently be able to generate coherent language without reasoning deficits.

Having in mind the vast human cognitive playground and previous work on leveraging neural Language Models with external data modality, such as KG-BART [63], it is our goal therefore to work with Commonsense Knowledge and expand Language Reasoning and Generation to a space where language is learnt along with structured external Commonsense Knowledge.

We expect, ultimately, with our goal to further improve conversational assistants, such as Alexa, by providing new generative models, which are more aware of the world using our generative Commonsense-Aware Encoder-Decoder Language Model (Relation-Aware BART Model).

Consideration This thesis was written in parallel with an international project (the Amazon Alexa Prize TaskBot Challenge[118]) sponsored by Amazon. My role in the project was to ensure Alexa was able to answer questions of users in regards to Natural Language contexts, such as recipe articles and Do It Yourself (DIY) articles. To this extent, this thesis targets the enhancement of current methodologies concerning the generation of Natural Language, enriching them with an approximation to human-like Commonsense.

1.2 From the Cafe to the World: Contributions

With this work, we hope to empower 2 major contributions to the research field of Commonsense in Language Models:

- A rich and different approach to the integration of Commonsense Knowledge on Encoder-Decoder Language models.
- Study strategies for the generation of Natural Language, when boosted with Commonsense Knowledge.

On a different take, we also hope to have made a useful contribution to the Computational Curiosity field (Annex I):

- Creation of a Curiosity Dataset and exploration of Natural Language Processing (NLP) use-case studies (eg. Curiosities clarification using Commonsense).

1.3 Document Structure

We have segregated this document in what we believe better guides the reader through the contributions being proposed. Next, follows a brief description of each section:

- Chapter 2 outlines the major ideas and knowledge recommended for proceeding with the reading of the full dissertation, along with the research that has been done in the corresponding field. Attention mechanisms are covered since most of the work make use of the Transformer architecture. The problem of question answering is explained. Computers require a more concise and thoughtful way to process data, so a coverage on processing different modalities of data is essential to better understand the difficulties behind Natural Language Understanding and Generation. Since our work mostly focus on upbringing textual output rich in semantic meaning, the field of Commonsense is extensively discussed. Finally, in order to evaluate generative models, we need to use metrics and benchmarks so as to compare with related work on this field. Therefore, we overview the current state of the art metrics for such evaluation.
- Chapter 3 introduces in depth *how* and *what* we propose to challenge Language Models Generation in regards to the Commonsense issue. To achieve this, we start by covering the tasks we want to tackle, followed by the Commonsense Knowledge Base we intend to use to aid in the tasks. We go through how such Commonsense external knowledge can be merged with the tasks data to further feed a custom altered BART model (RA-BART), engineered to make use of Commonsense Knowledge.

- Chapter 4 presents the performance of our proposed work, in several different tasks, considering automatic metrics. Moreover, human evaluation is discussed as well as some ablation studies done to the Commonsense BART model. Finally, we present a demo for the reader to test the models.

- Chapter 5 focus on the impact of this dissertation, considering both the carbon and memory footprint of the RA-BART model.

- Chapter 6 concludes the dissertation and mentions next possible challenges: what was left undone and what other approaches to the problem could be pursued.

- Appendix A showcases further model comparisons and text generations for the reader to have a hands-on intuition over their quality.

- Annex I covers the concept of Curiosity. It explores the creation of a Curiosity Dataset, and how we have used it to relate curiosities to both food recipes' articles and Wikihow articles. Lastly, we present a technique to generate textual explanations/clarifications over our Curiosity Dataset using a BART model. We also present examples which show that using Commonsense (RA-BART) can further help creating more sounding curiosities' explanations.

2

Background & Related Work

2.1 Neural Language Models

2.1.1 Sequence to Sequence Learning

Sequence to Sequence models are a type of Neural Language Models. They were introduced while sharing the belief of the possibility of extracting an intermediate, compact representation of sequences, which could in turn help to better understand and generate alike sequences. Sequences have unique characteristics such as: 1. unbounded sizing; 2. strong dependency between sequence units (which may not be adjacent). A Sequence to Sequence (seq2seq) abstract model is illustrated in Fig 2.1, where sequences here represent Natural Language phrases in two different languages.

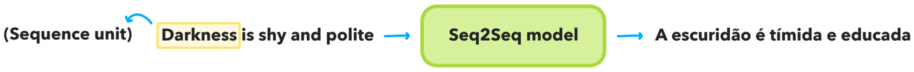


Figure 2.1: Seq2Seq model example: [Translation: English → Portuguese]

Tackling this sequence engineering theme, Recurrent Neural Network (RNN) [40] had its genesis at the turn of century, but only had its success around a decade ago. This architecture proved interesting since it learns patterns from sequences by auto feeding itself along with new sequence units, preserving a sense of sequential/timed memory of what is passed-through the network. The problem with such network, however, points to the learning process itself, because the mechanism of being fed one sequence unit at a time the network introduces issues like: computing speed, due to the impossibility of utilising parallelism techniques. Also, the vanilla **RNN** does not possess any special mechanism to handle a memory state, therefore when considering long sequences, the past context tends to meet the oblivion. This issue relates to the notion of vanishing gradients [78], where the weight of past neural states start having diminished importance after each learning update step, stagnating the updates on the learning weights.

Long Short Term Memory (LSTM) [33] and Gated Recurrent Unit (GRU) [17] were proposed as an attempt to address the deficiencies present in RNN, such as the short term memory issue. Internal mechanisms called gates, assuming the promise of a controlling memory unit, are optimised in the direction of preserving at best the memory thought to be relevant in the learning process and discard irrelevant information towards next feeding cycles.

Even though these network variants provide better memorisation capabilities, they are still upper bound limited to what they can filter and recapture from the past. Other feature which is interesting and which these networks fail to capture is the look-ahead sequence data. Reasonably one can acknowledge that some patterns are not solely dependent on past sequence units, but rather also carved by a notion of a future compass.

2.1.2 Transformers: Deep Learning Spring

As mentioned previously, these seq2seq models even though some attempt to combat architectural deficiencies such as vanishing gradients, limited memory absorption and reasoning, do not solve them completely, thus limiting their potential in useful applications. In 2017, “Attention is all you need” [108], was published revolutionising the way researchers looked at explicit attention, due to the introduction of Self-Attention along with the transformer architecture.

Meanwhile Self-Attention was gathering “attention”, a new attempt to perform transfer learning called ULMFiT [35] paved new ground by using a pre-training methodology on LSTM networks over a long corpus of text and further fine-tuning the same model on an Natural Language Understanding (NLU) task. ULMFiT method reached State of the art (SOTA) results in the text classification task with little labelled data, revealing the potential of pre-training models and fine-tuning the same architecture over a different (target) task.

Putting all the breakthroughs mentioned together and other past methodologies, deep learning gained a new direction, where the following points are crucial: 1. the Encoder-Decoder models; 2. transformers’ utilisation of attention: Self-Attention and Multi-Head Attention; 3. Usage of transfer learning: Pre-training models and in a later stage use them to learn a specific domain task (fine-tuning);

Next, the attention mechanism, the transformers architecture and transfer learning technique will be covered.

2.1.2.1 Cognitive Attention

Like human cognition is bombarded with sensory information, deep learning models implicitly face a weighted amount of digital stimulus [58].

Attention, as the name implies relates to the idea of focusing more on some things rather than on others. Regarding data processing, this is associated to the idea of coming up with probabilistic scores to data units that amplify or neglect the attention that should be paid to them, when considering a certain goal.

Three popular attention mechanisms are 1. Self-Attention 2. Cross-Attention 3. Graph-Attention

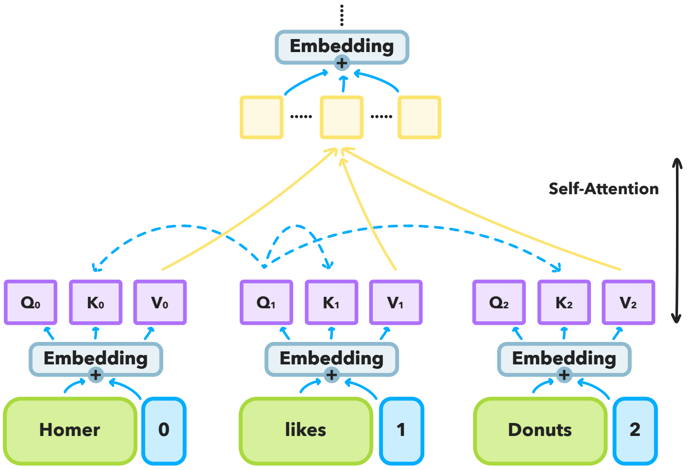


Figure 2.2: Self-Attention Mechanism

Self-Attention Mechanism: Self-Attention is an introspective contained set of operations which try to predict which input units have the most influence considering the current input unit being processed (See Fig 2.2 for a better understanding of the Self-Attention mechanism, used extensively within the most popular language transformer architectures such as BERT [22], BART [51], among others).

Self-Attention, can analogically be seen as the process of searching something in the web. Consider the Fig 2.2 and the word ⟨likes⟩, embedded for machine understanding purposes, with which we want to query the web. In this analogy, let's say that the web represents the others words from which our **word query** was taken from "homer likes donuts". We want to search something with our word, so the process is the following: our **word query** will attempt to match with the keys present in the web, which resemble all possible matches ⟨ Homer , likes , donuts ⟩. The **key** that better matches our search query, will consequently make the mechanism retrieve the related **value** information. Most of the times, in attention mechanisms the queries, keys and values all correspond to the same set of sequence units. However, this not always the case, see next **Cross-Attention Mechanism** paragraph for more information. As an example, in computer science, commonly, images are fetched through a textual address, but what you get when fetching them are the actual images' files. In this situation, thus you would have an exact matching between queries and keys, but obtaining a value which is in a different medium. See equation 2.1 to visually grasp the mathematics behind the queries, keys and values interaction, just explained.

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2.1}$$

Cross-Attention Mechanism: Cross-Attention is an attention mechanism which attempts to score the relevance of one sequence units of data with another different sequence units of data. Ernie architecture [102], for instance, uses this technique for question answering where a question attends to an external information context to better retrieve contextualised answers. It is also used a lot when combining multiple modalities, in Vision-Language (VL) model LXMERT [105], images attend to questions, to answer questions over the medium of images. BART [51], also uses this technique to merge the knowledge/vector representation extracted from the Encoder with queries from the decoder to generate sequences, which are conditioned on the input text.

Graph-Attention Mechanism: Graph-Attention, used for example in KG-BART [63], is based on the Self-Attention Mechanism but applied to sub-graphs, instead of textual sequences. In Graph-Attention, nodes get confronted with neighbour nodes to score their relative importance.

Even though Attention was first used to alleviate the problem of handling long term dependencies in regards to machine translation, it has been broadly adopted and researched on further deep learning fields, such as Computer Vision [23] and Commonsense Reasoning/Generation [63], which normally involves the usage of external knowledge graphs.

Relation-Aware Self-Attention Mechanism: Relation-Aware Self-Attention [95], was designed with the goal of introducing pre-known relative knowledge information between tokens alongside the Self-Attention mechanism in order to have a more knowledgeable learning process, which combines explicit and implicit knowledge. RAT-SQL [113] is an example project which uses this attention mechanism on the Encoder part of the model.

2.1.2.2 Transformers Architecture

Transformers follow an architecture where there are no recurrent layers, rather Self-Attention is used to map the relevance of sequence units. Even though the initial transformer possessed both an encoder and a decoder module, these days we have architectures which consist of Encoder-only layers such as BERT [22], we have Decoder-only architectures such as the GPTs [14] and we have hybrid models reassembling more closely the original proposal of the transformer, such as BART [51].

Behind these models there is a manifest which is characterised by 2 main ideas: 1. Having a big enough corpora of language, we can use the data itself to comprehend the underlying base rules of a language and syntax (**pre-training**). From a **Self-Supervised learning** process, without implicit labelled data we mask parts of a text corpora and try to predict the gaps of text since the integral corpora is known. 2. Assuming the pre-training has been performed and there is control over the understanding of the syntax of a language, the task becomes how to solve specific problems such as summarization, language translation, questions answering, semantics enhancement. This process is coined as training (**fine-tuning**) for "downstream-tasks" on labelled data (**supervised learning**) [107, 126].

The standard Transformer's Encoder architecture is designed to leverage an understanding of the provided data into a compact representation. In a probabilistic context, the Encoder attempts to model the probability distribution which best fits the provided input. Learning a lower representation of Natural Language, the model tries to understand inter-token patterns and what tokens/words/expressions matter the most to another. Apart from pattern reasoning, there is also a relevant feature which is *flexibility*. From an encoding we can further use this low-dimension representation into other models, or even to solve specific tasks.

The Decoder, on the other hand, can be seen as a mathematical compass which tries to learn the sequences' probability distribution. From the learnt distribution, the decoder gains the ability to sample sequence units from it, given a conditional impulse (eg. other sequence units as input). This sampling characteristic, makes such module architecture ideal for generative purposes, such as generating Natural Language phrases.

2.1.2.3 Pre-Training

Pre-Training is a training technique, where a learning process is conceived on a model to obtain a general understanding of a certain type of data, which based on the concept of **Transfer Learning** can be used to further train a model more accurately on a specific "downstream-task" (**fine-tuning**). This has been a rather popular technique in the Computer Vision field for a decade [54], since AlexNet showed promising results [45] in the ImageNet problem [21] and further since Residual Nets made its appearance [32]. Pre-Training, more objectively in the Natural Language Processing (NLP) field, is often associated with gathering one, or a set of gigantic corpus of data, whom is learnt via a learning procedure called Self-Supervised Learning [22, 51]. Not relying on labelled data is what enables this procedure to learn from billions of words (or any other modality unit), hence requiring annotation on such a big corpus of data would be very costly and time demanding. Apart from the bureaucratic rational, Self-Supervised Learning, which usually works by masking pieces of the data it is learning from and later predict those masked units of data (eg. The <mask> meows → The cat meows), leads to an implicit understanding of patterns which reside within the data [22, 53]. This feature of implicitly learning patterns and "storing" the knowledge encountered, makes this learning process as an accelerator for later learning specific tasks, such as **summarization** or **question answering** in the Natural Language Processing field and **object recognition** in the Computer Vision field.

2.1.2.4 Pruning and the ugly 'head'ling

Dating back to the birth of Transformers [108], as mentioned previously, both the encoder and decoder module are populated with multiple independent attention heads, which are stacked in parallel since they have the potential to learn different patterns, resulting in the promise of a more profound language knowledge acquisition. However, as observed in Voita's work [111], there are heads which unfortunately and apparently fail to learn useful representations of the domain data. Having this in consideration, one can scrutinise the quality of each attention head in regards to each module layer, and even remove or adapt an attention head [96, 111].

Techniques to analyse Transformer Heads The stacking of layers rich in further consecutive layers of mathematical operations harden a human grasp on the learning process of neural networks, therefore to better dissect the undergoing of transformers, visual frameworks, such as bertViz [110], visToolkit [106] and ecco [1], were brought onto existence. BertViz, for instance allows for a visual understanding of how an head is behaving conditioned on a certain input data. Ecco using interpretative methodologies, such as salience mappings, allow for a more mature analysis over how data individually is captured. As pointed by Voita [111], one other way to systematically introspect these models is to use the technique: Layerwise Relevance Propagation (LRP) [5], initially created for understanding the weight of pixels in the output of visual models.

2.1.3 Generative Language models

Regarding Neural Language Models which compose Natural Language, there are two main strategies for generating text:**1.** the **Encoder-Decoder** architectures and **2.** the **Decoder-only** architectures. Encoder-Decoder architectures tend to be more versatile, since they invest in creating contextualised latent representations, before passing these representations to aid the decoding (language generation) process. Differently, Decoder-only architectures infer the next piece of data having only in consideration the causal probability of what has been seen before (generated). The decoder in Encoder-Decoder, on the other hand, would also be coerced by the encoder output (embedding) over the several Decoder layers.

Three of the most popular architectures for Natural Language Generation are the following: GPT architecture family, being the more recent one GPT-3 [14], BART [51], T5 [83].

GPT-3 (Generative Pre-trained Transformer-3): OpenAI launched GPT-3 in 2020 without open-sourcing it, which was presented as an improvement to the previous open-sourced GPT-2 model. The GPT family of models follows the decoder-only architecture.

BART (Bidirectional Auto-Regressive Transformer): Presented by the Facebook research team, BART was inspired in the Encoder-Decoder framework, from which they envisioned the BERT [22] as the encoder and the GPT-2 [82] as an inspiration for the decoder component.

T5: Likewise BART, T5 was Google's attempt at following a similar architecture rational, adopting different solutions.

The key differences between T5 and BART, apart from the training procedure, lie in the way input is dealt with and consequentially the intrinsic model behaviours regarding this choice. The T5 model is conditioned only by the means of textual input, meaning that for different tasks, some task prefix must reside within the textual input for the model to understand what task is being asked to perform. BART, on the other hand, distances itself from the textual input when selecting tasks. Tasks are coerced by using special tokens representing the tasks, which are concatenated to the encoded textual input when fed to the model. These models' architectures have a powerful ability: Custom Special Tokens can be used on the input part of the model to teach the model categorisations of the input (See Fig 2.3 as an example).

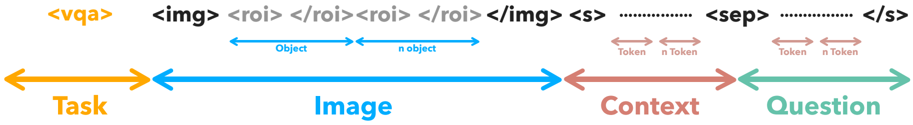

<vqa>
<img> <roi> </roi><roi> </roi> </img> <s> .... <sep> .... </s>
Object
n object
Token
n Token
Token
n Token
Task
Image
Context
Question

Figure 2.3: Abstract input structure for a VQA task

In this example, there is an intent to teach the model that on the task of <vqa> (visual question answering) the input will be divided in several sub elements: An image, a contextual text and the question to be asked. The special tokens in this case are presented with the distinctive characters "<" and ">". This technique of using special tokens has proven to be rather powerful for "downstream-tasks", such as visual question answering, among many others. MBART [64], for instance, used this technique to create a multilingual BART, where a language special token was placed before a textual input to teach the model a specific language.

2.1.4 Natural Language Generation Strategies (Decoding Strategies)

Theoretically, on an Encoder-Decoder architecture, the decoder takes the Encoder output and in an iterative way generates 1 language unit token each time while feeding itself the language already generated so the next tokens are coerced by the language generated so far and the output from the Encoder. In practice, this process described, usually, only happens at inference time, whereas at training a technique called **Teacher Forcing** [30] is used. On inference time, the iterative approach of generating sequence units, is called *decoding* and the algorithm for deciding which tokens to choose *decoding strategy* (see

Fig 2.4 for a decoding example). The most popular and general decoding strategies are the following: 1. Greedy Decoding 2. Beam Search Decoding 3. Top-k Decoding [24] 4. Top-p Decoding [34]

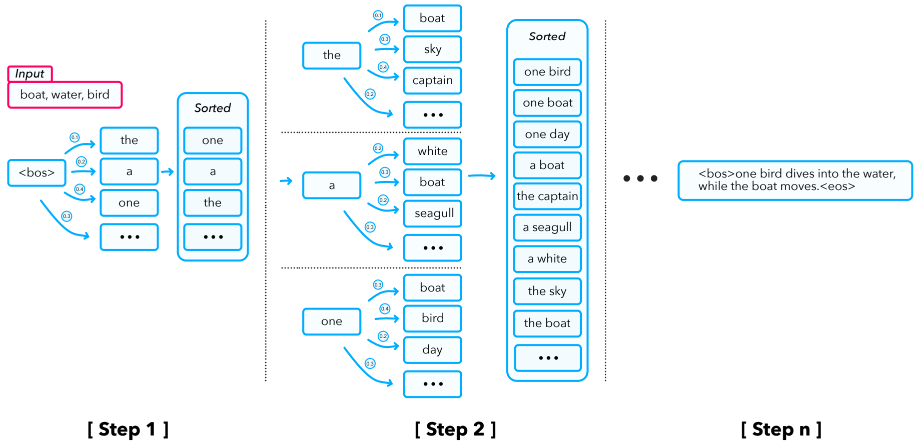


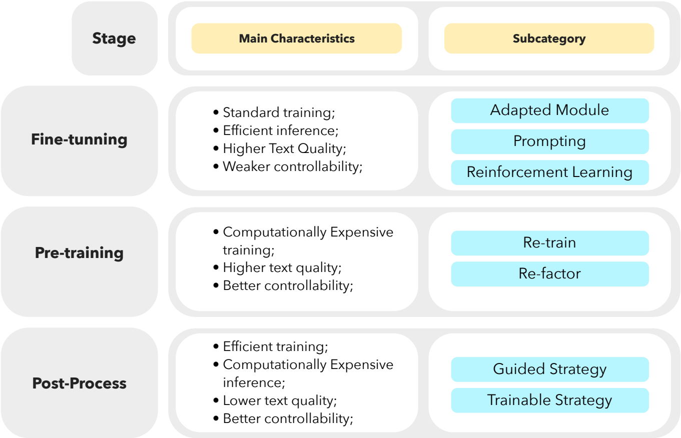
Figure 2.4: Beam Search Decoding Example

Greedy Decoding, is simplest decoding strategy, where each next token is selected based on the highest likelihood of being the best match over the pre-generated tokens. Beam Search Decoding attempts to convey better sequences by maintaining a top-n sequence likelihood, that at each step of generation sorts the likelihood and selects the top-n sequences. This strategy ends up having nice properties, since even if a certain token has a lower probability, after this token we can have a very interesting text which ends up being considered in beam search, whereas on greedy decoding such sentence would have been discarded. There are also use cases, where we want to be quite diverse and not use the same language units to describe a scenario/idea. Sampling is also a strategy, which ignites a more poetic literacy, by considering a sample of our vocabulary and applying the same techniques as before, or others. Top-k, for instance, from a sample selects the first k tokens with higher likelihood, whereas top-p is more relaxed and considers the selection of the top-n tokens until reaching the accumulative probability of p value.

2.1.4.1 Controllable Text Generation (CTG)

There are times, where we want to exhibit a more controlled output, following a certain constrained domain of sequence units or rules. For instance we might want to to force out swear words from a generated text. In NLP, the field of controlling the generated output text, is called CTG.

As pointed by this recent survey [127], CTG is rampantly emerging in the field of Natural Language Generation (NLG) and it can be pursued throughout three main stages. **1.** In the fine-tuning phase it can be performed by an adaptor procedure [124], prompting [61] or an Reinforcement Learning (RL) optimisation procedure [130]. **2.** One other way is to retrain or refactor the pre-training models. **3.** Finally, having a model already trained, we can guide it's decoding generation using decoding strategies, which is the main focus of this section (See Fig 2.5 for a summary over these techniques).



Stage	Main Characteristics	Subcategory
Fine-tunning	• Standard training; • Efficient inference; • Higher Text Quality; • Weaker controllability;	Adapted Module Prompting Reinforcement Learning
Pre-training	• Computationally Expensive training; • Higher text quality; • Better controllability;	Re-train Re-factor
Post-Process	• Efficient training; • Computationally Expensive inference; • Lower text quality; • Better controllability;	Guided Strategy Trainable Strategy

Figure 2.5: Overview of the current approaches to CTG (adapted from [127])

Constrained Decoding Strategies Regarding controllable Decoding Strategies, there are two main approaches: **1.** using pure Conjunctive Constraints and **2.** using Disjunctive Constraints.

Introducing constraints on decoding procedure is not trivial, imagine we want to force tokens t_1, t_2, which form a word, in a generated sequence S in the form:

$$S = (s_1, ..., s_k, t_1, t_2, s_{k+1}, ...s_n)$$

Generally, tokens are selected one at a time, which makes it difficult to know at what $step_i$ will be best to place the tokens, especially considering that there are multiple tokens and, for instance, when we have multiple constraints simultaneously. In order to deal with constraints, at every step the tokens from the constraints are forcefully introduced in parallel with the most likely ones. To control the constraints fulfilment and the quality of output, **Banks** are introduced. A bank [55] relates to a constraint, which has n-tokens to be fulfilled on a generated sentence. Each constraint forces the existence of n-token banks, where each bank defines the list of phrases (in current hypothesis) that have made n steps progress in fulfilling the constraints.

2.2 Learning Data Representations: Building Blocks

Knowledge, reasoning and communication are intrinsically present in our conceptualisation of intelligence, which come in several shapes and in a multi-dimension fashion. Transforming data into a compact understandable dimension has been researched for many years, for example H.Poincaré in 1900, first transfigured a graph unto a mathematical representation, the incidence matrix. Creating compact representations of data is not much to battle our humanistic limited reasoning on diverse kinds of modalities, but rather to ease communication, facilitate operations over them and ultimately to encompass the indeed limits machines have interpreting different modalities. Learning lower representations of data also relates to capturing the essence underlying it which allows for a better generalisation when facing unseen data. The caveat in this process is trying to preserve at best the information present in the initial modality type. Fernando Pessoa in the 19th century would already meditate in a lyric paradigm about such a concept. In "Autopsicografia" he says:

And those who read what he may write, | E os que leem o que escreve,
Upon reading his pain feel all too well, | Na dor lida sentem bem,
Not the two pains that he has, | Não as duas que ele teve,
But rather only those pains that they do not have. | Mas só a que eles não têm.

Fernando Pessoa (1931)

In a natural sense, this can be interpreted as the corruption that happens when converting a medium/representation of data to a different one. This "Lost in Translation" consequence of transforming data is rather impossible to avoid, however diminishing the impact of it is what matters to scientists. Concerning this, hereafter 3 types of medium (Natural Language, Knowledge and Symbolism) are mentioned along with some techniques used in SOTA research to preserve at best their natural medium throughout "encoding" transformations.

2.2.1 Natural Language

Concerning textual data, machines fail to grasp it in its raw form, therefore there is a need in converting it in a medium a machine unit can understand it. It's not by chance computers usually encode text in ASCII or Unicode encoding (numerical encoding), among others. Regarding Natural Language reasoning and processing however these text representations are further transformed to vector numerical contexts [8].

It would be common one decade ago to see textual units (words) being represented in one-hot encoding and as a balanced weight of word occurrences encoding (TF-IDF). The problem with the former, apart from the potential unfeasible size of such vocabulary encoding, is that the orthogonality of such representation would result in non-human representations of text, since different and similar words would be treated in the same way. The latter, even though, trying to balance the importance of uncommon words with noisy common ones, such as stop words, it is still itself a naive approach with no syntax reasoning.

Posteriorly, word embeddings (eg. Word2Vec [69], GLOvE [80]) were presented as a means to approximate similar concepts and keep away intangible concepts. Opening doors for spacial operations, such as translation and interpolation between concepts, on a dense vector level space. Adopted in mass by the deep learning field, however, since the words embedded result in a static representation, meaning the same word on different contexts result in the same vector output, such models fail to reach human-like word semantic representation.

Trying to fill the gap on textual encodings, a new norm has taken place along with the recent wave of natural language transformers. Coined as context-informed word embeddings, many architectures, such as BERT [22], process language by first passing it through a sub-word tokenizer and lastly by feeding these tokenized language units through multiple attention layers in order to learn the encoding of each word/sub-word within a context.

Behind these new advances, as mentioned, there is this concept of **tokenization**. The reason for this is that language tokenization has become the norm when pre-processing natural language phrases. However, what is it exactly?

2.2.1.1 Sub-word Tokenizers

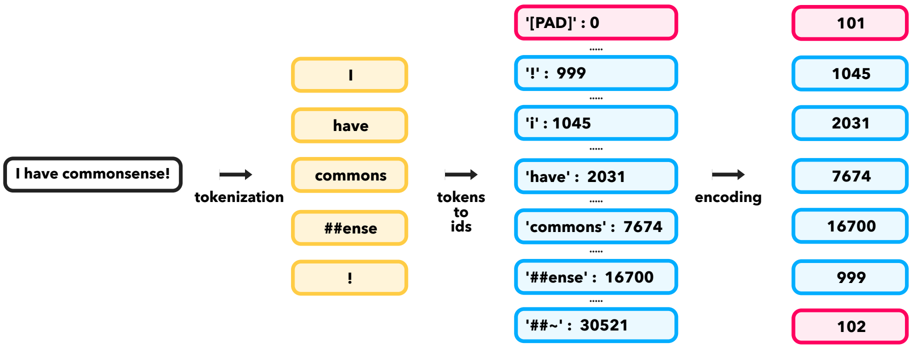


Figure 2.6: BERT Tokenizer pipeline

Tokenizing a text consists in splitting it into words or sub-words and later convert them to numerical ids through a look-up table. Tokenizers represent the fundamental basis of transformers success, being the building blocks of uniformly transforming free-form language into computational units machines can understand. This representation standard proves to be even more powerful, apart from language representation, as it allows for further combination and concatenation of multiple modality mediums in the same granularity level (eg. images, knowledge vector embeddings).

Considering tokenizers, there are two main types: 1. Rule-based tokenizers 2. Neural tokenizers . Rule-based tokenizers are simpler to imagine, since they assemble a word tokenization (eg.space separation process), which resonates to our human natural sense of splitting a sentence at the word level, however, they face some issues. One is the potential poor entanglement to rules not well defined, which may not capture every natural language sentence split case. The other relates to size of the vocabulary it creates, which is rather large due to repetitive sub-words not being reused.

One concern with having a gigantic vocabulary (hundreds of thousands of words) is related to enforcing neural models, which make use of such tokenized representations to have input layers prepared for enormous embedding matrices, resulting in their learning process being slower.

Neural tokenizers (transformers tokenizers) on the other hand are dynamic tokenizers, which generally don't overcome more than some dozens of thousands of elements. They are able to be much shorter on the vocabulary, due to the attempt to not capture every single possible word, punctuation and symbols, but rather fuse sub-elements and reuse them to encode a full word. The way this mechanism is implemented differs on some cases. Four of the most popular tokenizer models techniques are: 1. Byte-Pair Encoding (BPE) [27] 2. WordPiece [94] 3. Unigram [46] 4. SentencePiece [47]

We will focus mainly on **Byte-Pair Encoding (BPE)** and **WordPiece** tokenizers, since they are respectively used as input pre-processing for the BART and BERT architectures. The main goal of both of these tokenizer algorithms is to find a way language representation encoding with utilises the least amount of tokens, since an infinite vocabulary is unfeasible to compute and several variations of a word exists. Both algorithms start out with gathering all unit characters the language uses.

Byte-Pair Encoding (BPE) Tokenizer: This is a multi turn algorithm, which utilises a frequency approach to build its vocabulary, over one byte characters. At each step, pairs of current vocabulary units are merged if combined they represent the best counting of usages in a textual dataset. The algorithm stops when a limit amount of vocabulary units have been reached.

WordPiece Tokenizer: This is a multi turn algorithm, which utilises a probabilistic maximum likelihood approach to build its vocabulary, over many byte characters. The algorithm is similar to the BPE one, deferring on the merging function. WordPiece, instead of depending only on the frequency of the combining tokens to match, cares about the impact that merge will have on the vocabulary. The rule of merging is therefore: the difference between the probability of the new merged pair occurring minus the probability of both individual tokens occurring individually should be the greatest (See Fig 2.6 to see the WordPiece Tokenizer (BERT tokenizer) in action).

Both algorithms ensure that the most common words will be represented in the new vocabulary as a single token, while rare words will be split into two or more sub-word tokens.

2.2.2 Knowledge Graphs

Natural Language is a powerful input resource for learning syntax rules and semantics, but it lacks the expressiveness of world knowledge. In reality, language builds as a communication tool through knowledge information, pre-defined rules, and relationships that follow concrete meaningful patterns. Therefore, it is reasonable to mentally map that human knowledge should be presented in a different modality from text or even images or audio, since what we want to extract is the reasoning behind mental concepts. Knowledge graphs do then fill this gap, by gathering nodes of concepts connected to others through meaningful relationships forming a knowledge mesh. **Facts**, or **Relational facts** which compose knowledge graphs are often represented in the form of triples: [subject] - [relation] - [object] . One problem that using Knowledge Graphs incites falls over the fact that deep models can't reason a Knowledge graph as a whole without further processing.

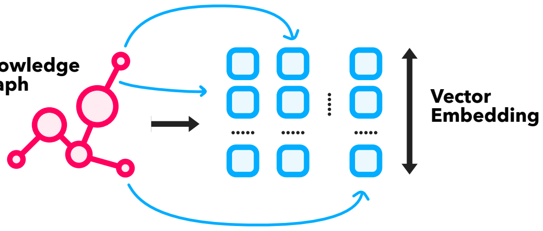


Figure 2.7: Knowledge Graph embedding

In an attempt to solve this issue, the paradigm of **Graph Representation Learning** ignited as a way to convert a Knowledge Graph into a compact embedding, which a Natural Language Model, for example, could intrinsically embed and ultimately understand. One way to achieve this is converting a graph into a well established shape of data, a vector representation. One thing to consider, however, is that such embeddings should at their finest preserve knowledge patterns or else the relevance of representing knowledge within a graph diminishes. Such patterns are, for instance: 1. Symmetry 2. Asymmetry 3. Inversion 4. Composition 5. Hierarchies 6. Type Constraints 7. Transitivity 8. Homophily (similarity clusters) 9. Long-range dependency

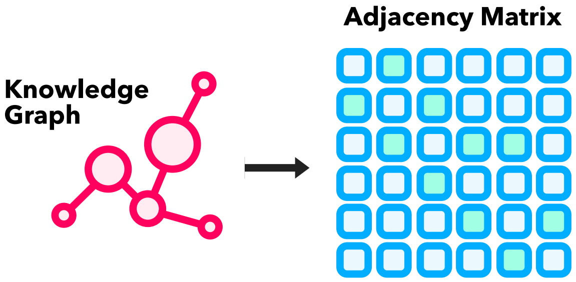


Figure 2.8: Knowledge Graph → Adjacency Matrix

There are, however, other simpler ways to represent a Knowledge Graph, such as using an Adjacency Matrix (See Fig 2.8), an Incidence Matrix or an Adjacency List. Using such kinds of graphs representations come as an explicit knowledge encoding, in contrast to Graph Representation Learning.

This does not mean, however, that one can not use simple graph representations, such as Adjacency Matrices and further learn representations of the relations (nodes that intertwine them), using learnable Embedding Layers. RAT-SQL [113], for instance, uses

Adjacency Matrices to learn relations between keywords phrases and SQL Schemas.

2.2.3 Symbolism and Neural models

Symbolic Logic is the study of reasoning over knowledge captured in human-readable symbols to solve both formal problems and daily life scenarios [39]. **Subsymbolic logic** emerged as an inspiration from symbolic logic relaxing its premises by considering **uncertainty**[1], which is leveraged by neural networks modelling and learning procedures.

Methods merging both types of symbolism have since gained popularity, as a means to bridge the gap of the shortcomings and benefits between the symbolic and sub-symbolic paradigms. These methods follow, then, a design which attempt to combine the advantages of both: the subsymbolic ability to learn from the environment and the symbolic ability to reason about a certain domain [38].

Symbolic and Subsymbolic mechanisms are indeed very promising regarding reasoning capabilities inherent in them. However, what mechanisms are behind them and what challenges do they face?

Symbolic logic programming formalisation: Based on formal logic, a symbolic logic program is a set of facts and rules, also known as predicates, which attempt to cover a certain domain. **Prolog** is one famous example of a logic programming language and a **backward-chaining** reasoner [100]. Predicates, in Prolog, are defined by a set of logical rules or Horn clauses following the structure (Head :- Body), and facts (Head), where Head is a predicate; Body is a conjunction of predicates; :- is a logical implication. A **Fact** is a **rule** that contains a head element but has no body, whereas a **Rule** contains both a head and a set of body elements. A **Fact** is considered to represent a knowledge building function, and a **rule** is considered to be a function which makes use of variable units to extrapolate new knowledge from existing **Facts**. The simplest unit in these programs are called: **Atoms**, which are textual constants. An example of facts and rules can be observed next:

Listing 2.1: Fact-Rule Comparison

```
H.                          (fact)
H :- B1, ..., Bn.           (rule)
```

Having defined a set of facts and rules, what is then intend with logical programs is to query it to check if a certain goal can be met. For this, a proof system needs to be in place (See a real problem in prolog example 2.2).

[1] Uncertainty, in the context of subsymbolism, relates to a symbolic mentality paradigm shift. Whereas symbolic AI produces logical conclusions, subsymbolic produce an approximation to them, hence uncertainty. Subsymbolism utilises inference and differential techniques to computationally deal with large data, and learn knowledge/rules from it. Additionally, algorithms for proving queries/goals often make use of approximations (eg. instead of syntax-exact string matching, semantic similar matching can be used)

Listing 2.2: Prolog example

```
zombie(john).                                   (fact)
human(allie).                                   (fact)
hungry(john).                                   (fact)
brain_eater(X) :- zombie(X), hungry(X).         (rule)
eat_brain(X,Y) :- brain_eater(X), human(Y).     (rule)
?- eat_brain(john, allie)                       (goal)
```

Neuro-Symbolic challenges: Symbolism is increasingly finding its way towards neural models [53, 68, 97, 115], because it can provide an easier explainable framework, expand knowledge with logical rules and even serve as a knowledge-deductive control tool. However, this adoption is not straightforward to NLU models. Two of the major Neuro-Symbolic challenges are:

- Free-form text and open-domain contexts can not be easily constrained with rules (How does one establish a rule system which covers everything?);
- A vanilla symbolic engine cannot out of the shelf backpropagate gradients, which neural networks are dependent on for learning; Vanilla symbolic engine can be intractable in real-time dynamic reasoning queries and in situations dependent on huge empirical data streams [38].

Neural Theorem Provers [86] to battle the lack of differential ability, have relaxed the backward-chaining algorithm, by using a differential unification algorithm. The evaluation of a proof, thus departs from a boolean output, to a continuous truth score. The Neural Theorem Prover, however does not deal well with the intractable amount of candidate proofs it potentiates, making it not suitable for Natural Language tasks.

2.3 QA: Question Answering

Question Answering is a Natural Language downstream task related to having a system which attempts to answer questions from users. This task can vary both in terms of the questions specifications, and in terms of the knowledge environment from which the answers should be obtained: direct text context; external knowledge source; Commonsense Knowledge; pre-training knowledge.

QA problems can be categorised in several different families:

Retrieval QA: The task of Retrieval QA is related to the gathering of textual passages from documents retrieved which may contain an answer to a user question. Example datasets for this task are: TrecQA[2] and MSMARCO-QA [6].

[2] https://trec.nist.gov/data/qa.html

Extractive QA: This QA problem is associated with obtaining an answer to a provided question, solely based on a provided textual context. Therefore, either the answer exists and is extracted from the context, or there is no answer within the provided context and no answer is retrieved. SQuAD2.0 [84] is a dataset covering this task.

Classification QA: This QA task relates to answering questions over a discrete set of choices, also known as multi-choice QA task. Some examples of such task are: CommonsenseQA [104], TextbookQA [42].

Open Generative QA: Unlike **Extractive QA**, this type of QA problem is related to the generation of an answer that best matches a provided question, resorting to either a context or an external knowledge source. There is a version of ELI5 [25] dataset, containing supporting documents (external context) to provide more reasonable answers, which is a good example of this task.

Closed Generative (Abstractive) QA: Similarly to **Open Generative QA** the task is to fully generate an answer, however it is bounded by the limits of the knowledge embedded in the model (no external knowledge is provided). An example dataset for this task is ELI5 [25], without using corresponding answer supporting documents.

RAG QA: This task, like Retrieval QA makes use of relevant retrieved documents, but the mission here is to fully generate an answer to a question, conditioned by the retrieved data [52].

Regarding the questions context domain, Question Answering (QA) models can further be distinguished as either **open-domain** or **close-domain**. Closed-domain models are domain specific, targeting a certain field of knowledge (e.g. finance, biology context), whereas open-domain models are unbounded to any knowledge context.

2.4 Building up the Commonsense Cognition

It is a social utopia thought to envision a human-brain transfusion into a synthetic corpus, aspiring our biological and neurological capabilities onto them. One of the key characteristics we would like to neurologically tame is the Commonsense processing. The ability to leverage, in a controlled scenario, the fusion of our intrinsic knowledge, a priori human experiences, social accepted conventions and logical deduction. In a way, **Commonsense** capabilities is the promise to achieve human-like reasoning, within non-biological environments.

Commonsense Reasoning can sometimes be confused with open-domain knowledge understanding & retrieval [16, 26, 102], which intrinsically relate to different ideas. Open-domain knowledge is associated with information about the world, history, science, geography, important people, among other domains, whereas Commonsense is more closely regarded as the awareness of the concepts underlying what is not directly expressed: the deep understanding of what is left or right [19] or even what it means to be under or upper of something in a visual context [77].

2.4.1 Commonsense: $\alpha \rightarrow \omega$

Deriving from the **No Free Lunch theorem** [117], machine algorithms are intrinsically bounded to a learning caveat. There is no general algorithm which can beat all other algorithms in all tasks, which strengthens the importance of really capturing the essence of any task, hence improving the efficiency in learning a specific task. Commonsense should therefore not be misplaced as a second effect to learning, but rather as a primary and specific cognitive domain we want to learn from. There is the belief that Commonsense is the holy grail factor [29] holding us from successfully teaching machines to more closely mimic our cognition [19].

When the sky is heavily clouded and we intend to leave the house, we undoubtedly grab an umbrella or coat. Also, the mentality behind *running for a charger when seeing the battery on the phone being on the 1%* is a trace element of commonsense processing. However, what exactly is conveyed as commonsense, in computational terms?

Computational Commonsense is commonly and formally rooted to 2 major aspects [19]:

Knowledge bases: Networks of human-knowledge concepts, normally represented by nodes in a graph, connected to other concepts through meaningful associations.

Plausible inference: This terminology refers to a deduction process based on uncertain information and events, which can result at either factual true events/information or false information in a further future moment, but reasonably inferred at some period of time.

2.4.1.1 Knowledge bases: Taxonomies, Ontologies & Knowledge Graphs

As an attempt to comprehend and structurally store human knowledge, knowledge bases have seen the day of light, such as DBPedia [4] and Wikidata [112]. With the rise of Semantic Web [10], the W3C standard Web Ontology Language came to existence as a means to express web ontologies' networks. Many public ontologies following the standards have then been made public, such as "The National Center for Biomedical Ontology" [74]. More oriented to the Commonsense problem and structured in "triples", semantic networks such as ConceptNet [99], Swow [60], and Atomic [91] have been created. ConceptNet consists of a semantic knowledge graph covering several different languages and containing general information about objects and concepts, while Atomic focus on having a knowledge graph which covers events: how concepts interact and situations unfold given a specific event. Like ConceptNet and Atomic, there has been other relevant attempts to map sub-knowledge repositories, such as WordNet [70] (an English lexical database) and ImageNet [21] (a repository of categorised images and captions). Having in consideration the amount of different knowledge repositories in existence, CSKG [37] tried to gather and conform several knowledge networks in a single graph with the goal of achieving a more general Commonsense Knowledge-grounding. Apart from these knowledge repositories, which are either manually curated or rule-based automatically extracted from third-party sources, there are already neural models with generative knowledge capabilities. Some examples of these are COMET [13] and Visual COMET [77], which leveraging from knowledge-graphs have learnt the ability to generate new knowledge phrases based on a given context and prior knowledge. Some of this generated knowledge is, however, not correct and that is a concern.

2.4.1.2 Plausible Inference

CYC [50] has battled with the Commonsense problem for over 3 decades now. From early on, they conceived that Commonsense Reasoning was dependent on a robust taxonomy (opencyc), but urged the importance of an higher level logic language from which statements could be inferred, using knowledge networks and baseline logic rules.

Regarding the concept of plausible inference, there has been extended research in a similar task: **Knowledge base completion** which is related to automatically inferring facts from knowledge graphs which are not present in them; See for example the work of [67]. In simple terms, **Knowledge base completion** is often associated with the prediction of relations between concepts from reasoning over the knowledge accessed from the same or other knowledge graphs.

Differing from the technique of designing models to infer over missing knowledge, there is also the take of **neuro-symbolic reasoning**, which try to encompass knowledge representations with both symbolic rules and inference engines, similarly to the CYC project.

As mentioned in 2.4.1.2, the take on plausible inference is not really to have strong deductive systems, but rather relaxed ones. The analogy here relates to our human brain Cognition Reasoning: merging Logic Reasoning with Commonsense Reasoning. There is interest in leveraging Commonsense Reasoning over the logic one, since our humanistic behaviour is mostly derived from our instincts and not from a formal logical deduction process.

2.4.2 Neuro-Symbolic Commonsense Reasoning

Considering the challenges that coexist in transferring the human trivial ability to deduce a thought and judge events, mainly due to the difficulty of finding ways of gathering human knowledge and logically reason over it, there is the neuro-symbolic commonsense reasoning research field trying to mix symbolic reasoning with neural models to battle these challenges [71].

One of the key problems with the other approaches mentioned previously relates to the knowledge graphs used having an upper-bound limitation over the facts that are expressed on them and therefore, these techniques are prone to ignorance over the facts which are absent. Most texts and knowledge graphs focus on what happened or can happen, whereas the information sauce regarding what cannot happen (known as **negative knowledge**) is most of the times implicit or absent.

Neuro-symbolism attempts then to combine the benefits of symbolic logic to tackle the problem of negative knowledge, deductive reasoning and the lack of explainability found in neural models. On the deductive reasoning field, trying to merge the problem of answering questions in a more deductive manner, NLProlog [115] was proposed in 2019. It presented a unification relaxation to the Prolog logical language system to tackle Question Answering tasks over Natural Language phrases. Their idea was to logically reason about what answer made sense, regarding a given question, deviating from the common approach of solely using transformers co-occurrence driven Language Models, such as BERT [22].

Neuro-symbolism, as mentioned in the NLProlog work, can help extract Commonsense Reasoning for answering questions, however, at the cost of language expressiveness. Further mixing it with Generative Language Models is a promise for better language expressiveness, by combining Natural Language with knowledge over relaxed logical deduction.

To tackle the merge of conversational systems and neuro-symbolic reasoners, F. Arabshahi et al [3] present **CORGI** a system that performs **soft** logical inference. CORGI uses a neuro-symbolic theorem prover, also presented in their work, which they further use to extract multi-hop reasoning chains of Commonsense presumptions over a knowledge base. Their work is, however, bounded to solving syntax phrases in the form: if-(state), then-(action), because-(goal). F. Moghimifar et al.[72] attempted to extend their work, by providing the logical paths taken to reach the envisioned goal.

Apart from the generative mesh-up of merging the generative abilities of Neural Language models with neuro-symbolism techniques, there is also a promising take on letting the Language Models take the generative wheel, but controlling them using relaxed symbolic reasoners [103]. The rational behind this is under the assumption that human's primary' Symbolic Reasoning goal is to exclude dangerous and wrong inferences.

2.4.3 Language Models & Commonsense Reasoning

Human-like sensibility in communication and reasoning is sought in VL models. Commonsense Reasoning in the context of Language Models, is the task of gathering strategies to understand better Natural Language phrases, which are aware of the relationships that coexist between concepts. The concern is not on the syntax level of the language, but rather on the semantics. In other words, the syntax granularity is rooted to grammatical rules, whether semantics is rooted to words' underlying meaning and logic. Considering the sentence: "The bird is flying on the water", the sentence is grammatically correct, however the natural meaning of the words were not captured correctly, since rationally "flying" is not coined to the maritime subspace of concepts [29].

More and more researchers are beginning to adjust the models for capturing better Commonsense, such as the case of KEAR [121], K-BERT [62], manipulating the transformer architecture's Self-Attention with Knowledgeable External Attention. KG-BART [63], further attempts to absorb Commonsense Knowledge from external Knowledge Graphs and using Graph-Attention puts language concepts on the same granularity level to Commonsense information.

As seen in work such as "Recognition to Cognition" [125], this can be further explored to the field of Visual Commonsense Reasoning which takes the language medium understanding onto the visual one.

2.4.4 Language Models & Commonsense Generation

As mentioned, Commonsense Reasoning is a cognitive field where the focus lies in studying the understanding of the semantics on a given medium (eg. textual, visual), and not exactly on the use of this Knowledge to generate phrases rich in Commonsense. "Recognition to Cognition" [125], for instance explores multiple choices questions, which from a Commonsense playground try to better predict the right answer. However, apart from making sure Commonsense Knowledge is captured within the models learning process, the generation of alike meaningful phrases is also rather scientifically sought. Some previous work, such as Visual COMET [77] and KM-BART [120] have coined this problem of generating language rich in commonsense as Commonsense Generation.

KM-BART [120] attempts to reason about Commonsense Knowledge over multi-modal data, such as images and text. Built over an Encoder-Decoder BART architecture, external knowledge graphs representation, image embeddings and Natural Language encodings are merged together as a means to generate textual sentences better aware of concepts and their nature relationships. In this work, the knowledge is textual based contrasting to a graph enriched based approach. Knowledge is captured by the usage of a pre-trained Transformer model (Visual COMET [77], a visual approach to the classic COMET [13]), whose task is to reproduce knowledge triples in a Natural Language phrase rich in Commonsense.

In contrast with using the COMET approach to feed Commonsense Knowledge, KG-BART [63] proposed a different approach to knowledge embedding targeting Natural Language Generation. They focus their work on textual input, while leveraging Natural Language Understanding with a custom knowledge embedding technique which merges Natural Language tokens representation with the paradigm of graph representation learning. For this they have created a custom Knowledge Graph Transformer which attempts to capture structural information and relations between concepts, taken from ConceptNet [99] Commonsense Knowledge Graph, by bringing textual data and knowledge concepts to the same granularity level.

Even though these models mentioned have reached SOTA results in the corresponding field benchmarks, a question that resides is whether this is the "right" path for really capturing Commonsense and generating Commonsense rich phrases.

2.5 Data is all you need? - Datasets Importance

Deep learning has become more popular and easier to do research on due to the continuous improvement and availability of Graphics Processing Unit (GPU) components, which are fully competent on vector operations. Additionally, platforms such as Google Colab, Kaggle and Paperspace have emerged to proportionate free GPU computation, reducing the money-gap bridge around Artificial Intelligence (AI) education. Libraries like the Hugging Face [116], have further extended the reach of SOTA NLP reserach providing a platform for the sharing of neural models, metrics, benchmarks and datasets.

As mentioned, GPU machines are essential for the learning procedures of Neural Models and sometimes even for inference latency reduction, however neural models without diverse and robust datasets to be trained on are rather useless. To this reason, we will devise an overview on the importance of datasets and the current SOTA datasets used in tasks relevant to our research.

2.5.1 Data balance juggling

Even though, Neural Models are often called **black-boxes**, due to our difficulty in understanding their results, there is no doubt that model behaviour is deeply impacted by its underlying training data. The world contains natural invariant bias, such as gender misrepresentations, often seen on some professions, for instance. Carefully sampling data is then a concerning ignition to capturing a realistic perception of a distribution of data. Not deposing care on such matter, can result in unfair and poisoned biased models, acting as a statistical segregator. Summing up, bias, in its raw form is not a negative concept, since it relates to any correlation found in data, which is generally how Neural Models learn. The problem arises when the correlations result from human mistakes.

Another characteristic to be aware in datasets is misrepresentation of features to be learned. In our work, for instance, there is the objective to create a Neural Model more aware of Commonsense. The assumption that such cognitive characteristic can solely be learnt through language correlation is rather faulty. The reason for this is that language and other mediums, such as images, do not express Commonsense Knowledge in their raw form. Concepts and an understanding of the world are intrinsically and indirectly encoded in our way of communication. Language datasets easily may contain a sentence like: Person X is eating an ⟨Apple⟩. But maybe there will not exist a sentence: ⟨Apple⟩

is a fruit, or: An ⟨Apple⟩ is generally green and red, or finally: An ⟨Apple⟩ can be eaten, cooked or even used as a flying object to hit something. Meaning, that perhaps we need to think more broadly about datasets instead of solely focusing on textual datasets. In the end, even though they are normally huge (dozens of GB), as mentioned they have natural limitations of coverage.

Therefore, a strong meditation about datasets and their possible faults, misconceptions, data limitations and bias is needed for a more understandable neural learning process.

2.5.2 Where's wally? | Finding the right data for the task

There are plenty of open datasets which drive the growth of research and applications in deep learning models. Regarding the work in debate in this dissertation, we mention datasets covering linguistics, Commonsense and Knowledge Reasoning, since they present the most importance, both for training the proposed model and also for evaluating it.

Next follows some datasets, normally considered for NLP tasks related to our work:

Abstract Question Answering: ELI5 [25] is a long form Question Answering dataset, which being diverse and containing forum-like knowledgeable answers can be used to answer general questions in an informative and generative way.

Commonsense Understanding Task: To evaluate how robust a Language Model performs over Commonsense Reasoning, datasets such as CommonsenseQA (CSQA) [104] and Social IQA [92] were created. They are question-answering datasets, which gather the assumption that answering correctly the questions imply a better sense of Commonsense Reasoning. In an attempt to better handle reading comprehension, the ReCoRD dataset [129] was created and combining both Commonsense Reasoning and Reading Comprehension, there has been COSMOS QA [36].

Commonsense Generation Task: CommonGen [56] can be used to train models to more coherently generate text, bounded by a set of concepts. Since some concepts have been fixed, the model needs to quite well understand them to figure out how to build an every-day scenario phrase using them.

2.6 Evaluation Metrics & Benchmarks: Baseline

The Generation of Natural Language sentences is exciting but is associated with a critical caveat. Ideally, the quality of generated text should be measured, not only by whether phrases are well behaved concerning the syntax of a selected language, but also even more importantly if they semantically make sense and are able to establish richful connections between sentence concepts and even provided images (in visual tasks). Natural

Language is, however, subjective and generally hard to measure concerning abstract criteria (eg. Commonsense, Politeness, etc). To this extent, evaluation metrics for Natural Language generation are broadly researched. Some popular metrics to analyse n-grams match between generated sentences and human-reference ones are BLEU-n [76], and Rouge-n [57], which complement each other, hence normally are used along with F1-score [79]. Consensus-based Image Description Metric (CIDEr) [109], which is also captive on n-grams overlapping, focus its evaluation on a better human judgement correlation, meaning that a consensus between the several proposed references, in a dataset, must be acquired for a good score. It has become a reference quality measure due to better mimicking human judgement.

More relaxed measures exist, which do not account solely the exact matching between the generated text and the target one: Meteor, for example, evaluates the similarity of texts having in consideration synonyms and the root of words, apart from the standard exact matching. On a similar ground of conveying a metric concerned with the semantics of a sentence, Semantic Propositional Image Caption Evaluation (SPICE) [2] emerged. SPICE, is a semantic evaluation metric that measures how effectively language text recover real world objects, their attributes and the relations between them, using semantic graphs. For open-ended text generation domain, a new metric: Mauve [81] was proposed so as to battle some inefficiencies of previous metrics by capturing more closely the distribution of the texts being compared.

As some of the metrics previously mentioned the focus of them lay in the potential healthiness in terms of syntax and in some subtle form on the semantics captured from a Language Model. However, ideally, we are much concerned about several other characteristics of cognition mirrored in language. Concerning the more concrete problem of Commonsense Reasoning within sentences, google's BIG-bench benchmark present key tasks to better understand the quality of commonsense reasoning. Some of these tasks are: 1. anachronisms 2. causal judgement, 3. cause and effect, 4. com2sense .

Human Level Performance (HLP) [18] plays also a major role in evaluation, especially when working with unstructured data, for establishing a point of comparison which helps deciding on what problem/task to tackle with further strength. Humans, perform really well on unstructured problems, such as textual, audio or images tasks making them ideal golden baselines. HLP is important to establish how relative an 100% score really is. For instance, provided an HLP score of 70%, achieving a similar score of 70% would mean a proposed model is extremely good on such task.

3

Commonsense-Aware Language Model

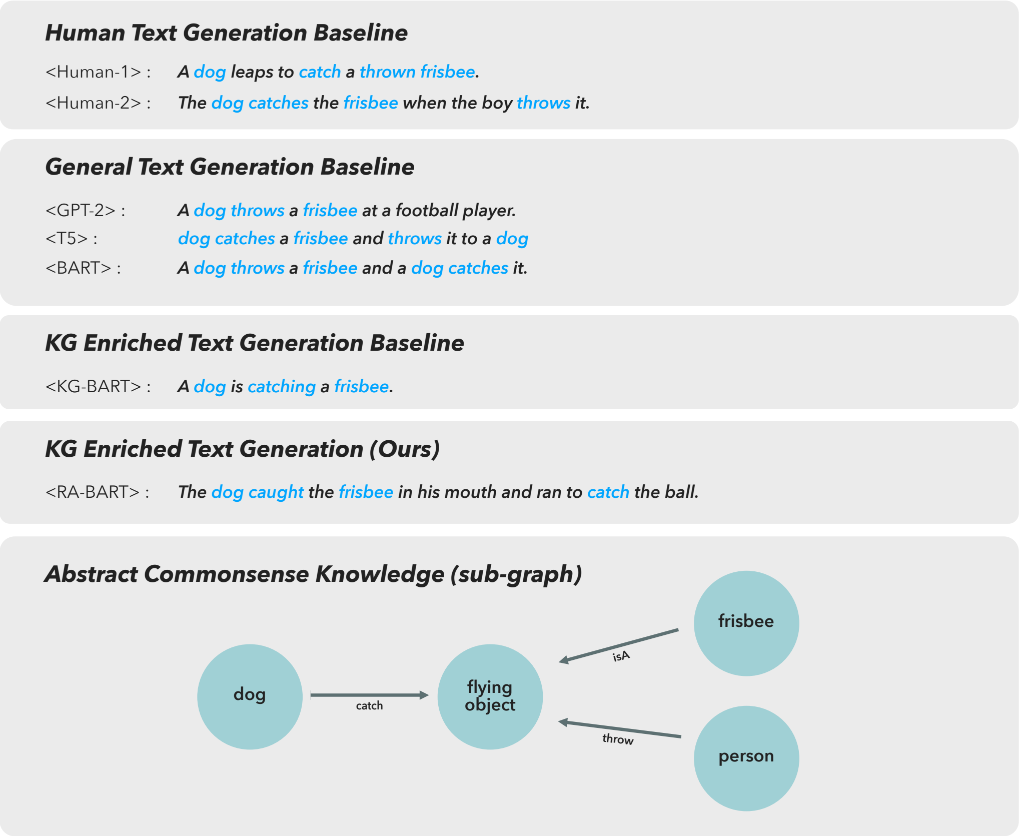
Enriching Language Models with explicit and structured Commonsense Knowledge.

In this Chapter, we take a deeper look into the motivation behind this dissertation. We go through the tasks/problems we want to address, along with the data related to them. We explore the importance of Commonsense Knowledge and from where we can capture it to improve Language Models. Lastly, we propose a Commonsense Encoder-Decoder Model architecture, enriched with Commonsense Knowledge Bases whose objective lies in generating Natural Language which is richer in semantic meaning. Some fine utilities for this can be to answer questions, as seen in section 2.3, or, for instance, to generate realistic sentences based on given concepts, in a more human-like manner.

To intuitively depict the prominent Commonsense issue in Language Models, let's imagine we have the concepts: ⟨ dog , catch , throw , frisbee ⟩ and we want to generate a realistic, rich in content, sentence. In Fig 3.1, we cover this hypothetical scenario and present a general baseline comparison between Language Models not enriched with external knowledge and others enriched, for a more intuitive perception of the interest in enriching these models with external knowledge. From the figure, we gain an intuition that SOTA Language Models generate sounding syntax sentences, however most of them lack on delivering believable snapshots of reality (eg. A dog throws a frisbee at a football player. - Do dogs throw frisbees?)

Motivation: We share the belief that most SOTA Language Models even though capturing the linguistics in some form, they are only barely capturing human-like understanding capabilities rightly.

Large corpora of texts, well-behaved, when trained with Attention mechanisms allow for a good understanding of language. However, the generation of fine looking syntax phrases give the impression of capturing language semantics, deduction reasoning and world domain knowledge, when the models producing them only capture them substantially [53], without further help. One reason for this is that Commonsense Knowledge



Human Text Generation Baseline

<Human-1> : A dog leaps to catch a thrown frisbee.

<Human-2> : The dog catches the frisbee when the boy throws it.

General Text Generation Baseline

<GPT-2> : A dog throws a frisbee at a football player.

<T5> : dog catches a frisbee and throws it to a dog

<BART> : A dog throws a frisbee and a dog catches it.

KG Enriched Text Generation Baseline

<KG-BART> : A dog is catching a frisbee.

KG Enriched Text Generation (Ours)

<RA-BART> : The dog caught the frisbee in his mouth and ran to catch the ball.

Abstract Commonsense Knowledge (sub-graph)

Figure 3.1: Baseline comparison of generative models (adapted from CommonGen [56])

and "Reasoning rules" are most of the times absent on the large corpus used to train these models (eg. on a big text corpus, one can maybe find that an umbrella protects a person from rain, but does it encoder that fact that: it avoids one getting wet; it can be shared in a romantic walk through the rain; it can also protect one from the sun; it can be lost, etc).

We advocate that training language or language-vision models in a fashion, which only leverages the linguistics is insufficient for guarantying generalisation to unseen world (use-case) scenarios. Therefore, we want to focus on enriching Language Models with Commonsense Knowledge so as it may be more semantically human-aware.

Adding to this, there is interest in leveraging human knowledge and language generation abilities with logical reasoning and even structured memory. After all, Transformers' Language Models are mainly co-occurrence driven when understanding and generating text, due to their probabilistic nature. This results in syntactically correct phrases, which is beneficial, however, we would like to provide explicit Commonsense Knowledge and not only implicit one, in an attempt to bound generated text to a more controlled knowledge grounding (eg. an abstract view of what we think is lacking in models learning process is the next exemplary scenario: "A cat is chasing a mice" + "By the way, did you know that a mice is prey food in regards to cats?"). To achieve this, we intend to merge a language model, the BART architecture with Commonsense

Knowledge Graph (KG) information.

3.1 Long live Commonsense: Tasks & Commonsense

How can we leverage Commonsense Knowledge on Language Models? We share the hypothesis that one issue with current Language Models research, is the ultimate trust on Neural Networks to capture intrinsic patterns (implicit knowledge) within unstructured data (eg. text). We believe that, explicit knowledge, if rightly introduced, can be healthy for models learning procedures. If we know that generally monkeys like bananas, why shouldn't we explicitly provide this information to the model? Considering this case, it is reasonable to believe that the model will encounter examples in data which would implicitly provide such information. However, what if it misses this relevant information, due to machine learning scientists believing such information would be within billions lines of text (training dataset)? This is the issue we propose to overcome. Possibly, our approach can help models generalise better, especially to unseen scenarios.

To test our hypothesis, there were three major key elements to consider: **1.** the Commonsense (explicit) Knowledge source; **2.** the datasets used to train and test our models enriched with Commonsense Knowledge; and lastly, **3.** the foundations of our proposed model architecture to leverage Commonsense Knowledge on given data tasks.

We will now cover the tasks we have worked with, followed by the Commonsense Knowledge considered and finally walktrough the reader through the proposed model, data processing and model alterations to test extensively our model.

3.1.1 Tasks & Datasets

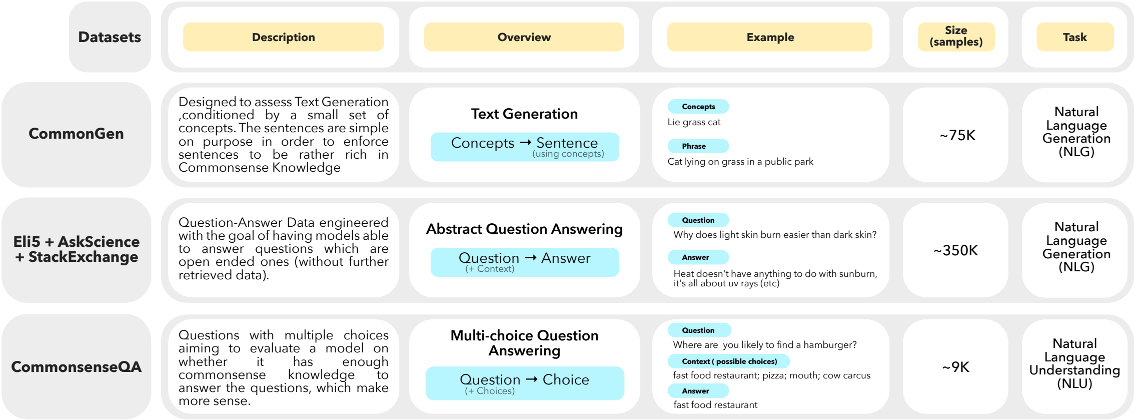
On the Alexa Challenge there was a need for a system which could answer abstract (general) questions (eg: "Why is it healthier to boil in the microwave?" or "How do i boil an egg?", etc), which made us consider 2 main ideas. First, there was a need for a set of data, which could transmit to a learning model, general world knowledge and domain related knowledge to the cuisine and Do It Yourself (DIY) tasks; Second, we were dealing with a Conversational System, which intrinsically enforced a bottleneck need of human-like interaction in order to provide a natural engagement.

Having pursued some research on related available data, we resorted to the ELI5, AskScience subredit data [25] and the StackExchange[1] forum data. These datasets, theoretically, would allow us to train and deploy a model, rich in general world knowledge, intrinsically encoded in the sentences found in ELI5 and AskScience subredit. To absorb focused in-domain knowledge (cuisine and DIY), the StackExchange specific forums ⟨ cooking , crafts , diy , gardening , lifehacks , pets ⟩ were fundamental. One interesting note about these datasets is their intrinsic internet forum sentence structure, which

[1] https://archive.org/download/stackexchange

motivates models trained on them to naturally formulate sentences with human-like characteristics. The datasets mentioned, however, make it difficult to evaluate the models on Commonsense, due to their indirect richness in knowledge and complexity.

To this end, it was ideal to also have specialised datasets to analyse whether our Commonsense boosting techniques could indirectly help achieve more realistic-human like generation. So, to objectively put our models to the test regarding our Commonsense enrichment methodology, we used the CommonsenseQA (CSQA) [104] and the CommonGen [56] dataset. These datasets have been designed to deeply capture Commonsense Understanding about the world, either regarding discrete or generative tasks. To have a more fine-grained understanding over the datasets and to observe some samples of them refer to Fig 3.2.



Datasets	Description	Overview	Example	Size (samples)	Task
CommonGen	Designed to assess Text Generation ,conditioned by a small set of concepts. The sentences are simple on purpose in order to enforce sentences to be rather rich in Commonsense Knowledge	Text Generation Concepts → Sentence (using concepts)	Concepts Lie grass cat Phrase Cat lying on grass in a public park	~75K	Natural Language Generation (NLG)
Eli5 + AskScience + StackExchange	Question-Answer Data engineered with the goal of having models able to answer questions which are open ended ones (without further retrieved data).	Abstract Question Answering Question → Answer (+ Context)	Question Why does light skin burn easier than dark skin? Answer Heat doesn't have anything to do with sunburn, it's all about uv rays (etc)	~350K	Natural Language Generation (NLG)
CommonsenseQA	Questions with multiple choices aiming to evaluate a model on whether it has enough commonsense knowledge to answer the questions, which make more sense.	Multi-choice Question Answering Question → Choice (+ Choices)	Question Where are you likely to find a hamburger? Context (possible choices) fast food restaurant; pizza; mouth; cow carcus Answer fast food restaurant	~9K	Natural Language Understanding (NLU)

Figure 3.2: Information overview about each data task, used in this dissertation.

3.1.1.1 ELI5, AskScience & StackExchange: The elephant data in the room

These three data sources are reddit like forums, which consist of collections of ranked social-media-like posts in the form of Question-Answer pair of posts. When working with such data, one must be careful and aware that the content underlying them can contain text, images, markdown text, links among other data fragments, which deeply hardens the extrapolation of useful information from them. To ensure noisy textual fragments and posts were put away, we developed a cleaning data pipeline, selecting only the top-1 post (*the correct answer*), in regards to each question and applied **Regex** rule based filters to simplify the answers. Such, Regex rules, were heuristically defined after having deeply reviewed the datasets' data. "Geek" web forum references were removed or converted to regular human vocabulary (eg. subreddit→place); Hyper links were removed; Images and Textual fragments mentioning either images or hyper-links were removed to the best of our abilities, since they would not be encoded textually and could be captured as noise.

3.1.1.2 Commonsense Evaluation Datasets

The CommonGen and CommonsenseQA datasets are standard Commonsense related benchmark datasets, respectively, for Commonsense Generation and Commonsense Understanding, which made them require none to minimal data manipulation.

CommonsenseQA: The Commonsense Question Answering dataset is by default taken as a discrete multi-choice Question Answering dataset, so normally when used to evaluate models on Commonsense this dataset is considered as a Classification Dataset. This, consequently, means that the architecture of a model working on a classification problem gets conditioned by the dataset itself, since the model would need a classification head, which in the end would have n-choices of neurons providing a probability for each question answer choice. In this work, however, we are using the BART architecture, which is a seq2seq generative model and we did not want to alter it's functioning. Therefore we turned the CommonsenseQA dataset into a generative task, by guarantying that the choices would be transmitted to the model along with the questions.

3.1.2 Commonsense Knowledge

In the Computational Commonsense Literature, we have seen that Commonsense is mostly depicted in dual lens. **1.** We have the pillar of static Commonsense Knowledge built on empirical knowledge, conventions, beliefs, world understating. **2.** The stimulation of a static to a kinetic knowledge, through logical deduction mechanisms result in human judgements, events evaluations: Commonsense Reasoning. Modelling such reasoning mechanism is a building block to successful model commonsense intuition. We intend to simplify the integration of Commonsense in Language Models and achieve this by approximating humans' static knowledge and their neurologically connected mesh up of concepts using a structured Commonsense KG. Let us share the following terminology: A concept isolated, or even a list of concepts can be considered as static knowledge. A snapshot of connected concepts can be thought of as a metamorphosis approximation of a dynamic concept. Traversing such knowledge snapshots, can also be introspected as a mere take on Commonsense Reasoning, bounded by the completeness and richness of such knowledge. We now pursue with a solid mathematical formulation of a Commonsense KG.

Mathematically, we can consider a Commonsense KG as a Graph, $G = (I, R)$, which connects concepts/ideas (I) through semantic relationships (R). Concretely, a Commonsense KG can be described as a collection of triplets in the form of (I_i, R_{ij}, I_j) which represent knowledge units, with I_i being a concept instance, I_j being another concept instance and R_{ij} the respective relation between I_i and I_j. Provided a Commonsense KG, $G = (I, R)$, then one can capture commonsense concepts and their relationships from any given sequence $S = (s1, ..., s_i, .., s_n)$, consisting of multiple s_i sub-words, by applying a parsing function such as $f : S, G \rightarrow S_c, S_r$, where S_c corresponds to the concepts present at sequence S and S_r the relations found at sequence S. S_c can then be defined as $S_c = \{c_1, ..., c_i, ..., c_n\} : c_i \in S$, being a collection of concepts within sequence S. S_r can then be defined as $S_r = \{(c_1, r_{12}, c_2), ..., (c_i, r_{ik}, c_k), ..., (c_n, r_{nm}, c_m)\} : c_i \in S, c_k \in S, r_{ik} \in G_R$, being a collection of relationships within sequence S.

3.1.2.1 Commonsense Knowledge Bases: n-relations Vs 1-relations KGs

Diving to a more concrete space, in Chapter 2 we got introduced to Knowledge Bases either in the form of models, graphs or ontologies. In this dissertation, there was a need for finding concepts, which relate to one another in the form of a semantic meaning (Commonsense enrichment), therefore, the selected Knowledge Bases had to mimic the Commonsense universe in meaningful inter-connected ideas. We will now dive deeper to the knowledge bases used in this work.

ConceptNet [99] consists of triplets of concepts connected through a small finite number of relations. Even though, ConceptNet contains around 3 million nodes (concepts), it only compounds around 40 different type of relations (eg. is_a, at_location, causes, etc). As pointed by the work [60], ConceptNet can be noisy and indirect, when it comes to having relations that are somewhat far-fetched and having direct missing relations which force hops in the graphs to get to some obvious concept relation. However, one key aspect of the nodes in ConceptNet is that the bounding to symbolic concepts are often 1-to-2 words, making them ideal for mappings with attention operations, which work on a sub-word level. **Swow** [20], contrasting with ConceptNet, has more direct associations with concepts, consisting merely of around 100 thousand nodes with 1.5 million relations. One less positive aspect about Swow is the lack of richness in the relationships between concepts, since they are binary (either they exist or not). See Fig 3.3 for a better intuition on the aspect of these Commonsense knowledge graphs. If we look closely to this figure, we can observe that there plenty of similarities, especially in more obvious concepts. However, if we take a glance at the concept “water”, we can see that ConceptNet starts to deviate subtly to not so direct concepts. Water can be frosted and that is an obvious fact, but we could argue that before thinking about such concept, we would most certainly think first about more direct concepts such as “bottle”.

Note: A simple/concise representation of a Commonsense KG can be seen as a collection of triplets in the form ⟨ concept_1/idea_1 , relationship , concept_2/idea_2 ⟩ (eg. ⟨ banana , related_to , monkey ⟩. In this work we use both this type of representation or it’s matrix representation format, where a relationship between two concepts get symbolised in a matrix as an entry relationship value between two tokenized concepts (**1.** row; **2.** column).

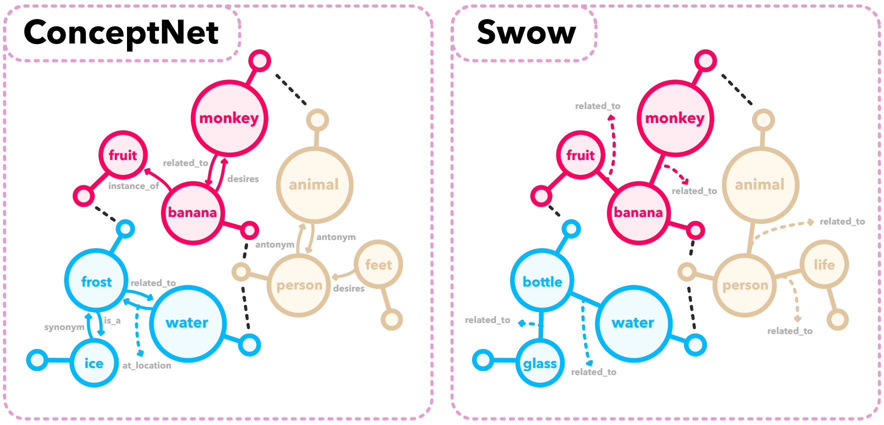


Figure 3.3: ConceptNet vs Swow Comparison

3.1.3 Raw Text data and Commonsense Knowledge Fusion

To understand the importance of fusing explicit Commonsense Knowledge with Natural Language one must be aware of an issue, which often prevails in Natural Language Generation, which is the hallucination problem. This issue consists in producing text which does not follow the semantic standards of our world or, simply put, it is the nomenclature given to the generation of non-sense text. Notably, Language Generation Models, being smaller or larger, are merely probabilistic models which provided some data input attempt to best map their output according to a learnt probabilistic distribution *happiness*. Learning the perfect distribution parameters of the world is a hard problem, easier for Larger Language Models, but nevertheless hard. To tackle this challenge a merge between implicit and explicit knowledge should be optimal. Implicit learning is the foundation of neural networks, however the importance of guiding these models in a more knowledgeable fashion should not be ignored. If we know before hand that some concepts are connected somehow it should be our obligation as machine teachers to inform the model about it and motivate it to retain the human knowledge. Now, we will cover, how we can extract explicit Commonsense Knowledge from both textual data and Commonsense Knowledge Graphs.

3.1.3.1 Detecting and Extracting Commonsense Knowledge from Text Data

In this dissertation, we have a recurrent issue which relates to having two different mediums of data interacting. We want to use external knowledge found in Commonsense Knowledge Graphs and extract from it relevant information which could prove to be useful to better comprehend a span of raw text. Therefore, there was an urgent need to find a way to bound a Knowledge Graph (structured) data to a text (raw & unstructured) data.

Imagine the example: "The race car is moving rather fast " in Fig 3.4. Conceptually we are in presence of a car, but more concretely, a race car. We have three choices now.

1. We can enlighten our model that a "race car" is "fast" **2.** We can parse the text such that "race" is related to "car" and that "car" is related to "fast" and **3.** We can merge option 1 and 2. The problem with the third option, however, is that for each language unit (token), only one relation is applicable, due to our model limitation (this issue is covered in detail in the model section 3.2). So, in practice, we either choose option 1 or 2, or we further populate more language units (tokens) which can provide us more Commonsense Relations. In terms of a realistic semantic scope, option 1 is preferred; in terms of the amount of Commonsense Knowledge captured, option 2 is more abundant (the number of simpler concepts extracted may be higher, but at a possible cost of adding noisy information). Moreover, populating new tokens or repeated ones can patch the issues found in the first and second option, however, at the cost of augmenting the encoding length (which has a pre-defined maximum value) or also at the cost of adding knowledge noise. One technique to approximate the third option might be to replace padding tokens (eg. **<pad>**), which are "useless" but"inevitable", with neighbour concepts relevant to the context (with relations with other concepts present in the context). With this latter technique the hypothesis it that we ensure a maximisation of the memory reserved for a sequence and the knowledge we have about a given topic (We will see this technique in section 3.2.1.2).

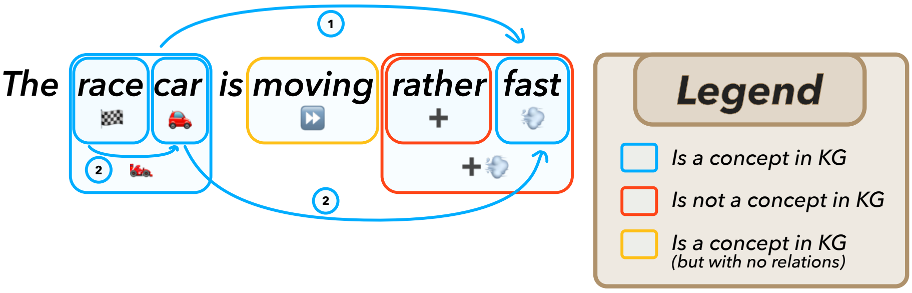


Figure 3.4: Overview of the text-to-concept mapping

In NLU literature, some libraries to parse Natural Language have gained popularity, such as Spacy [73] and NLTK [12]. Especially, for **keyword phrases extraction**, there are good performing techniques such as: KEYBert [31], Rake [88], YAKE! [15]. In our work, however, we chose a simpler keyword concept extraction approach using n-gram parsing, since it offered a very direct way to extract multi/uni-word concepts. Our Commonsense concept extraction procedure can be described as the following: We define a maximum word number length (n-gram) for a concept/idea, which we established it to be $n = 3$. We create all possible 3-grams and attempt to match them with a Commonsense Knowledge Graph (eg. ConceptNet). For any positive match we store the characters index of the concept/idea, in regards to the position it appears on the given textual context. If two concepts have a relationship on the Commonsense Knowledge Graph, we define a relationship between the characters range of both concepts, storing the type of relation they have (See Fig 3.6 in the Relation Context part, for an example). For the words still not matched we lower the *n*-gram matching. We try 2-ngram matching, 1-gram matching, and we stop, even if there are words with no match or words that have been discarded for being common words. In the end, we have concepts extracted from which we can later index our Commonsense Knowledge Graph to extract relevant knowledge information. During the iterations mentioned, we also make sure that within the matching searching phase, the concepts extracted have existing relations with other concepts within the same context. If concepts are found, but they lack relations with the concepts in the same context, they get discarded, simply because we can not add any value to that concept, or we risk adding knowledge noise.

In terms of accuracy, this Commonsense concept linking approach is optimal, because it's oriented to the Commonsense domain and we consult the Commonsense Knowledge Graph directly to know if the concepts exist and have relations between other context concepts. SOTA techniques, such as the ones mentioned (eg.YAKE!), are general domain techniques which would output both relevant but also *noisy* extracted content. There would also be a burden of proof-checking if the concepts extracted had relations between them and it would not be possible to exactly find the index of the words (needed for our use-case). We believe the accuracy would be lower using these more formal techniques and the extra burden of applying them would not be worth it (See Fig 3.5 for an example to gain intuition of their results differences).

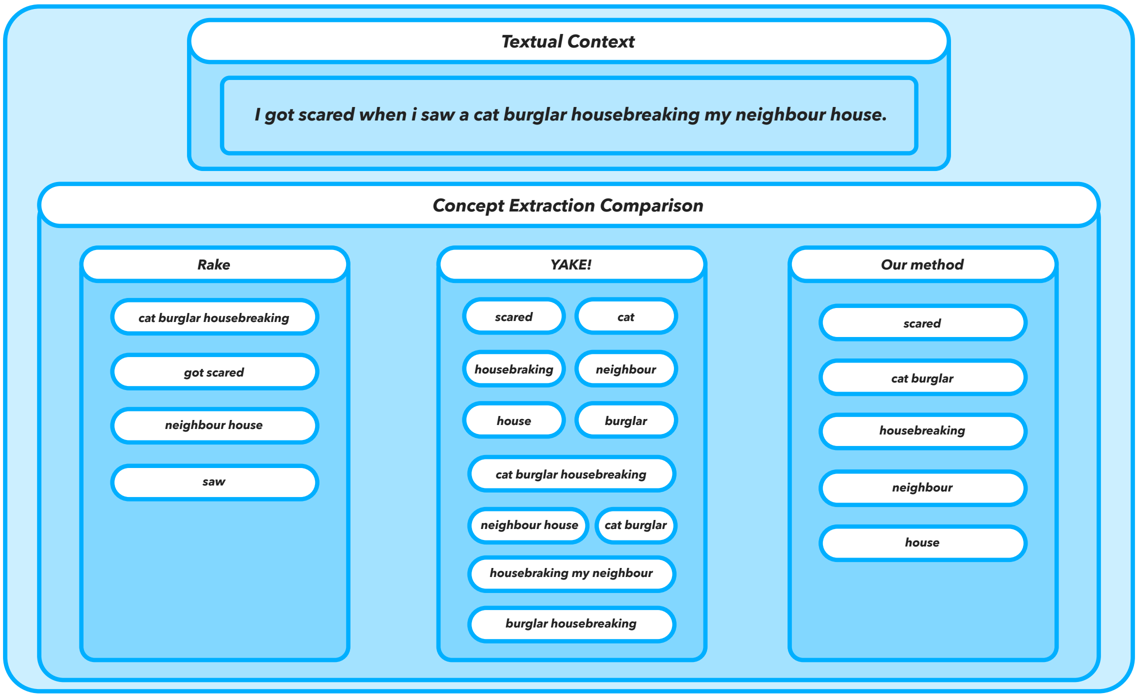


Figure 3.5: Comparison overview of keyword phrases extraction tools.

One other relevant facet of the pre-processing phase of mapping language units (tokens) to concepts in the Commonsense Knowledge Graphs relates to the removal of stopwords, consisting of around ∼1000 words. Our hypothesis is that, adding such words would confuse the models with **information noise**, especially when Language Models are already powerful at understanding the importance of stopwords syntactically, adding to the fact that they end up not providing further relevant information to the meaning of a phrase. Using the lemmatisation technique, we also convert the nouns and verbs to their root form so we can index the Knowledge Graph. It is common that plurals and verbs conjugations reside within the Commonsense Knowledge Graphs, however, their relations are less relevant and lower in magnitude than their root forms, being a solid reason for not considering them.

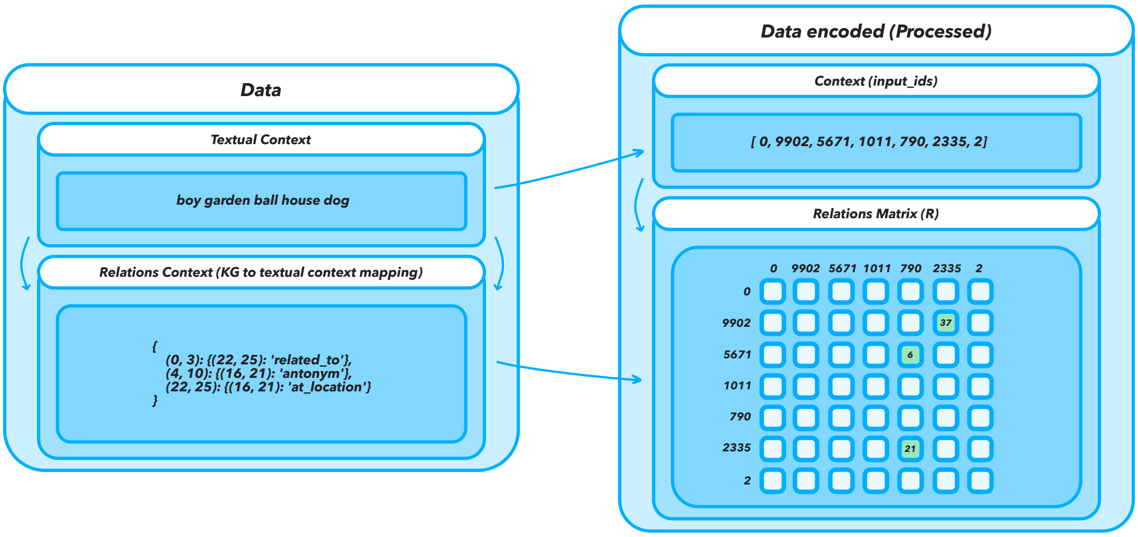
One important last note, we do not directly attempt to solve the **polysemy** [2] problem. As mentioned, we recur to a Commonsense Knowledge Graph for concept detection, if the concept as multiple meanings and those different meanings get reflected on concrete relationships with other knowledge concepts then our approach works in extracting the right concept meaning (the right relationship between concepts), if not: it suffices in leveraging the right meaning of a concept when multiple exist.

3.1.3.2 Textual Commonsense Knowledge - Knowledge Encoding

Textually extracting concepts from a given text is a start, however models don't understand text, so we have to find an intelligent way to: **1.** encode the text in order for a learning model to comprehend it. **2.** Encode the Commonsense Knowledge and their relations within such text. As seen in Chapter 2, in subsection 2.2.1.1, standard neural tokenizers are great because they are able to encode any collection of words, being them rare ones or not into an index form. This actually lightens up a complication related to sub-word tokenization, which we will go through briefly. Tokenizers solve the encoding of the text, but what about the concepts and their relationships? It is a crucial moment to remember that our learning models work at the token level and operate with them over a strict positioning scheme (since they are indexed in matrices). With this in mind, we have therefore to meditate about ways to identify which tokens relate to what concepts and map their relations. One nuance, however, related to the issue mentioned previously is that we have to decide what to do with words/compound words which consist of multiple multiple tokens/words. In our work, we have decided that the tokens related to a concept (word/compound word) get associated with the relation n token times related to other concept (See Fig 3.6 for a simple approximation of Commonsense Knowledge Encoding pipeline).

Having both, the *Relations Context* and the correspondent input text tokenized, we can align the tokens produced for each concept with their correspondent connected concepts (in the token format). Due to the model infrastructure chosen, explained in the next section, we decided to map the tokenized concepts' relationships in an Adjacency Matrix

[2] polysemy problem: consists of one concept having many possible meanings depending on a certain context.



```
{
   (0, 3): {(22, 25): 'related_to'},
   (4, 10): {(16, 21): 'antonym'},
   (22, 25): {(16, 21): 'at_location'}
}
```

	0	9902	5671	1011	790	2335	2
0							
9902						37	
5671					6		
1011							
790							
2335					21		
2							

Figure 3.6: Overview of the Commonsense Knowledge extraction from text and the text preparation to input the model.

format. This, however, limits the amount of possible relations between 2 tokens to be only one, since there can only be one matrix entry per each two tokens.

Systematically, we can describe the text pre-processing phase with Commonsense Knowledge Enrichment as a pipeline following the steps (refer back to Fig 3.6 to visually capture each step (2-4,6):

1. Clean text (remove stop words, punctuation, certain word contractions (lematization);

2. Observe which words/compound words are concepts in a Knowledge graph and for these apply the next step;

3. Find the index range (concept first character index, concept last character index) of the concepts and map their relations;

4. Tokenize the original text to obtain the words' tokens;

5. Align the tokens with the words/compound words relations;

6. Create a sparse matrix with the corresponding relations between each token;

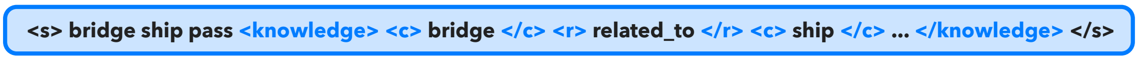
3.2 Commonsense-Aware Encoder-Decoder Transformer

On chapter 2, we saw different types of transformer models, such as the Encoder-only models, Decoder-only models and a merge of these two: Encoder-Decoder Models. Since we want to tackle Natural Language Generation problems we leave out Encoder-only models aside, which are mainly focused for encoding a certain input and use it for a certain discrete task. Using Decoder-only models were also not very interesting to us, since we needed a way to fuse input text data with knowledge extracted from a Commonsense Knowledge Graph and later use this encoding as an intermediate representation to better generate new fragments of text. For this reason, we use an Encoder-Decoder (Seq2Seq) Transformer model. As we will see further, using a Seq2Seq model gives us the ability to create a meaningful abstraction of the input enriched with external knowledge and additionally use this succinct representation as an helper (through cross-attention mechanisms) to generate more sounding text.

Regarding Seq2Seq models, we chose the BART architecture, which offered a balanced set of qualities: **1.** The biggest BART model is ~1.5GB (400M parameters), which comfortably fits within a standard GPU card and **2.** due to their proven quality on generation tasks [51].

BART, however, was not designed to leverage Commonsense Knowledge in a structured way. Therefore, using default BART, the simplest things one could do to explicitly teach Commonsense to the BART model would be to convert knowledge triplets in sentences (eg. ⟨ monkey , desires , banana ⟩ → monkey desires banana) and perform 2 possible tasks. **1.** Mask concepts or their relationships, predict these masked tokens and later fine-tune the model on a generative task. **2.** Mix sentences from a fine-tune dataset with the converted Commonsense sentences, in an attempt to teach Commonsense, while attempting to solve a specific task.

One other approach one could take, without manipulating the BART architecture would be to use special prompting techniques [61] to inject knowledge in structured ways. Among other utility factors, Special Tokens are designed to teach Language Models input patterns to help distinguish, for instance, one sample from another, or even within the same sample, some input segment from another. In our Commonsense use case, one could then use a Special token to suggest the start of a Commonsense Concept and the end of it, the same for another concept and the same for their relationship, for instance (See Fig 3.7 for an abstraction view of this prompting technique).



<s> bridge ship pass <knowledge> <c> bridge </c> <r> related_to </r> <c> ship </c> ... </knowledge> </s>

Figure 3.7: Commonsense Knowledge injection prompting technique on model input.

We could blindly apply these mentioned techniques, but if we meditate about the intrinsic nature of both a Commonsense Knowledge Graph and the Self-Attention Mechanism, we can acknowledge a certain behaviour overlap. In the Self-Attention Mechanism we encounter multiple word-to-word relationships, which are dynamically scored. Similarly, in a Commonsense Knowledge Graph we have multiple word-to-word connected relationships, with or without a fixed score. One big difference between both, is the granularity alignment of what is meant with a “word” in the Attention mechanism and in a graph. A word in the attention mechanism is usually a token, representing a sub-word in a Knowledge Graph. In section 3.1.3.2 we saw the importance for aligning multiple tokens with a KG concept (word), mainly because of this reason. One further and important difference between a graph and the attention mechanism is that the Attention mechanism can not depict the whole graph, but rather several independent sub-graphs of it if there is an existent alignment between the two.

It was clear for us that we needed a special way to explicitly inform the model about pre-defined Commonsense Knowledge, while it implicitly learns to perform on some task. Therefore we envisioned a model, abstractedly illustrated in Fig 3.8, where we take the normal BART Architecture and set out to replace the standard Encoder with a Commonsense-Aware one.

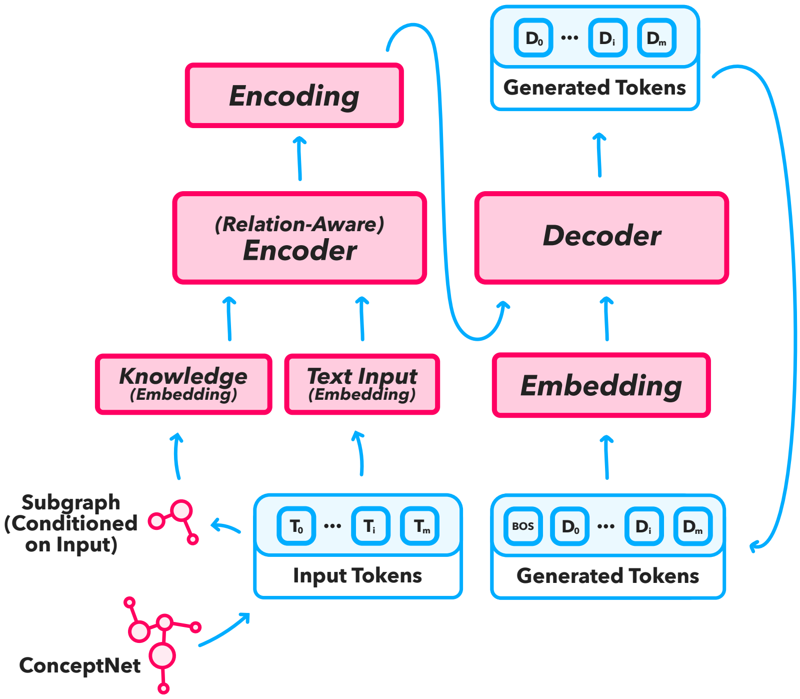


Figure 3.8: Default BART Model with Commonsense Integration alteration (RA-BART).

The question that resides is then: how could we alter the BART's Encoder to encompass both a Natural Language Input and the Commonsense Knowledge extracted from the input and Commonsense Knowledge Graph?

To answer this question, we relate to the RAT-SQL [113] work, a method to convert Natural Language questions onto the SQL language. In RAT-SQL they were aware that terms occurring on natural language questions could strongly relate to explicit tables and metadata which most certainly would be present either in a database schema or within SQL tables data. Therefore, they made sure to model that prior explicit knowledge within the Transformer's attention mechanism, making use of a special Relation-Aware Self-Attention [95].

This take on modelling explicit knowledge to leverage Language Models, came as a strong inspiration for our work, which we adapted for our Commonsense use case.

Whereas in RAT-SQL, the explicit relations exists between tables/attribute names exposed in Natural Language text and tabular data regarding those tables and attributes, in our work we could foresee the explicit knowledge as any word-like format concept in a provided text and their realisation on a Commonsense KG. See Fig 3.9 for an abstract view of the Commonsense Knowledge which can lie within a Natural Language phrase. Considering, the sentence: *"On the air , heavy clouds shadow a plane , which is flying to transport people over an ocean"*, according to our Commonsense Knowledge we would know that the ⟨air⟩ is a place for ⟨flying⟩, that ⟨clouds⟩ exist in the ⟨air⟩, and even that ⟨people⟩ can be ⟨transported⟩ by a ⟨plane⟩ object, which also ⟨flies⟩.

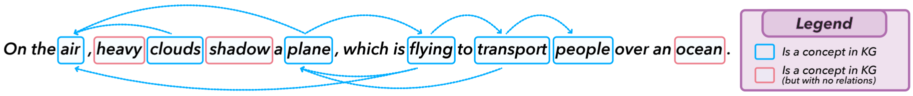

On the air , heavy clouds shadow a plane , which is flying to transport people over an ocean .
Legend
Is a concept in KG
Is a concept in KG (but with no relations)

Figure 3.9: Natural Language phrase with correspondent concepts and Commonsense relations.

Mathematically speaking, let's consider the standard Self-Attention Mechanism (Equation3.1), covered in chapter 2, and consider how one could enrich it with Commonsense Knowledge and map the relations between concepts (multi-token level units) within the Attention Mechanism.

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{3.1}$$

Normally, in standard Self-Attention, we make the queries attend to the keys and respectively extract the values, based on the best matching between the queries and keys. This works immensely well, being the foundational mechanism for the current SOTA deep learning architectures, but what if we wanted to inject explicit Commonsense Knowledge? Could we also weight the concepts (keys) which are more meaningful to the concept queries, whether they have pre-established relationships, found, for example, on a Commonsense Knowledge Graph?

In order to answer this question, one can try to fuse Commonsense relations between input sequence units by adding a relation term both to the keys mapping (R_k) and the values mapping (R_v). This way, we guaranty that Commonsense Knowledge will also be attended to, since we encode the Commonsense relations between concepts on these new terms. The following question becomes then: what exactly is R_k/R_v?.

The relation term (R_k/R_v) represent a matrix, which encodes a **single** embedding of the relations connecting each two tokens within a context. R_k/R_v are then obtained by applying an embedding to matrix elements, initially encoded through a Matrix (R), which for each two tokens store an index corresponding to a certain relation (eg. index 0: no-relation, etc). See equation 3.2 to see matrix R containing the relation code between two tokens ($token_i$,$token_k$).

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} & \dots & r_{1n} \\ r_{21} & r_{22} & r_{23} & \dots & r_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{m1} & r_{m2} & r_{m3} & \dots & r_{mn} \end{bmatrix} \quad , r_{ik} \in G_R \tag{3.2}$$

Matrix R, is an adjacency matrix ($maxSequenceLength \times maxSequenceLength$), where instead of only storing if there is a connection between tokens, the type of the relation (relation identifier) is stored. To further obtain (R_k/R_v), we build an embedding layer, where each relation id gets mapped to a learnt embedding of size d, where this d size corresponds to the BART's Encoder head dimension. R_k/R_v is then obtained by fetching the correspondent embedding vector for each relation id value stored within Matrix R (See equation 3.3 to see element-wise embedding of matrix R, which corresponds to R_k/R_v).

$$R_k(R) = R_v(R) = \begin{bmatrix} E(r_{11}) & E(r_{12}) & E(r_{13}) & \dots & E(r_{1n}) \\ E(r_{21}) & E(r_{22}) & E(r_{23}) & \dots & E(r_{2n}) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ E(r_{m1}) & E(r_{m2}) & E(r_{m3}) & \dots & E(r_{mn}) \end{bmatrix} \tag{3.3}$$

The RA-BART Self-Attention can then be written as equation3.4, which takes the advantage of the mentioned relative Commonsense relation information.

$$Attention(Q,K,V,R) = softmax(\frac{Q(K^T + R_k^T)}{\sqrt{d_k}})(VR_v^T) \tag{3.4}$$

For a visual intuition, we present Fig 3.10 for an abstract view of the Relation-Aware Attention in regards to Commonsense Knowledge). In this figure, we can see that not only a query attends to the neighbouring concepts, but also to the previously mentioned structured Commonsense knowledge Matrix (R) informing whether such query relates to other concepts. In Fig 3.8, one can also have a look of general overview of the RA-BART architecture and reckon that the Relation-Aware Self-Attention is a layer which gets enclosed over the each Encoder sub-module.

We only use this special type of Attention in the Encoder module, due to the bi-directional nature of the Encoder, which means that all input units can see and operate on the other input sequence units. In contrast, in the Decoder module, we have an auto-regressive masking technique, which hides future sequence units, since it's objective is to generate tokens only conditioned by past information (tokens). Additionally, we believe that within the Encoder, it's nature behaviour allows for a relevant Commonsense Knowledge injection, whereas in the Decoder this Knowledge injection would not be trivial and would perhaps cause harm in the learning process by forcing masked knowledge.

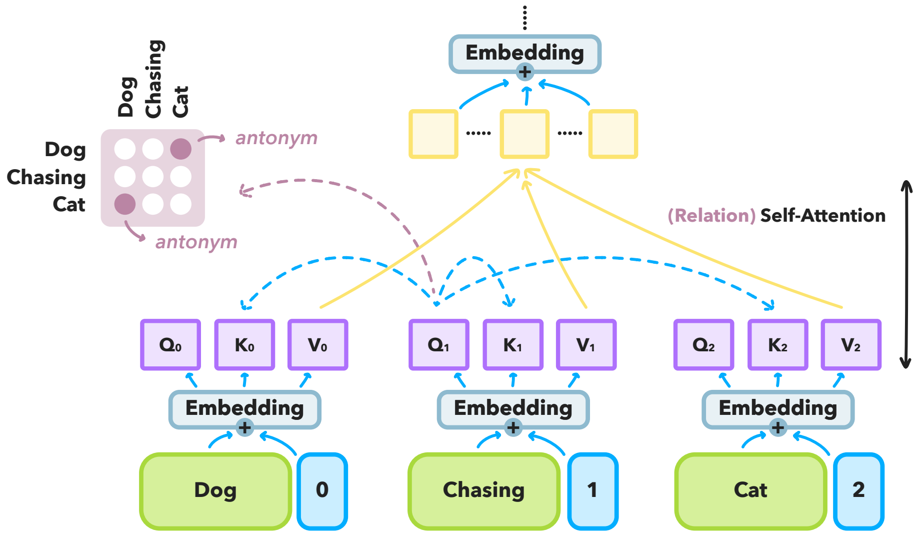


Figure 3.10: Relation Self-Attention mechanism supported by external Commonsense Knowledge (eg. ConceptNet).

3.2.1 Learning: Manipulating RA-BART (Loss, Input & Decoding Strategy)

As an attempt to further study *how* we can maximise the Commonsense Knowledge acquisition considering the proposed RA-BART model, we prepared 3 further experiments on the **CommonGen Task**, consisting on: **1.** changing the standard Seq2Seq loss function to a Commonsense One (**Commonsense Loss**) **2.** providing neighbour concepts to the input, which we call **Concept Expansion**; and lastly **3.** changing the decoding strategy of BART Language Generation to a Disjunctive Positive Constraint Decoding Strategy, constrained by a Commonsense Knowledge Graph. See Fig 3.11 for a first abstract look on the experiments just mentioned to test the RA-BART Model.

3.2.1.1 Commonsense Loss

Neural networks, fundamentally speaking, are a mesh of parameters bounded to some value which we attempt to tweak in order to reduce an error measurement between the predictions obtained through the parameters and what is in the literature called "the ground truth". In the Text Generation task, the standard loss function used is the Cross Entropy Loss, which assumes we are trying to generate language units limited to a discrete number of classes (token symbols). In the field, it is common to work with a fixed vocabulary engineered by a tokenizer, this fixed vocabulary of language units can be seen as the classes we are predicting during the generation of text.

When training a model, the Cross Entropy Loss allows for generating more probable tokens, which stands as a good motivation for generating more syntactically correct phrases, but may compromise the model semantic understanding of some text. Since, in this dissertation, we are mostly concerned in the semantic understanding of natural language we suggest that two penalties could be added to the loss function, regarding the

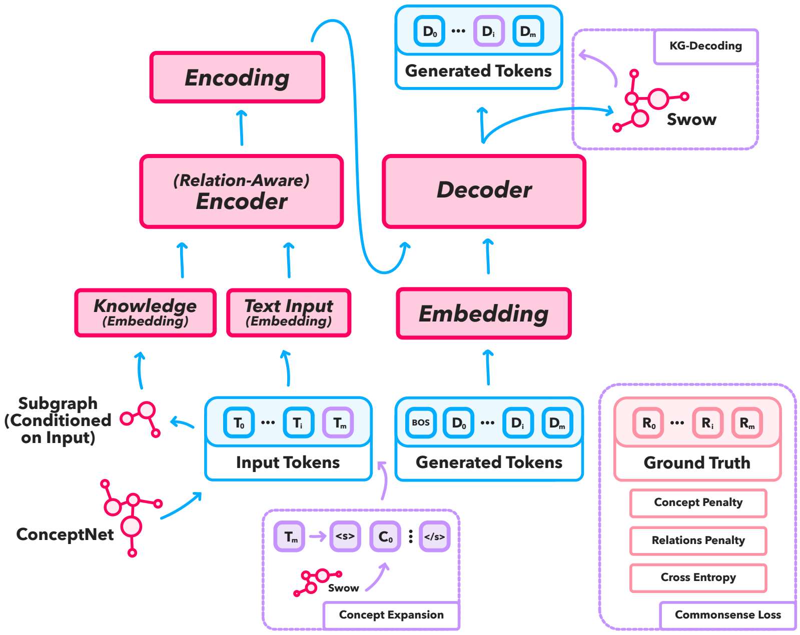


Figure 3.11: RA-BART Architecture and Experiments overview.

Text Generation task to better prevent hallucinations (non-sense text).

We suggest a loss, which motivates the parameters to generate more likely tokens, whilst at the same time penalising the model when anchor concepts are misrepresented and when concepts lack relations between them. This way, our hypothesis is that we are pushing the learning process to more closely mimic the Commonsense Knowledge Graph we are using (See Fig 3.12 for an overview of the Commonsense Loss).

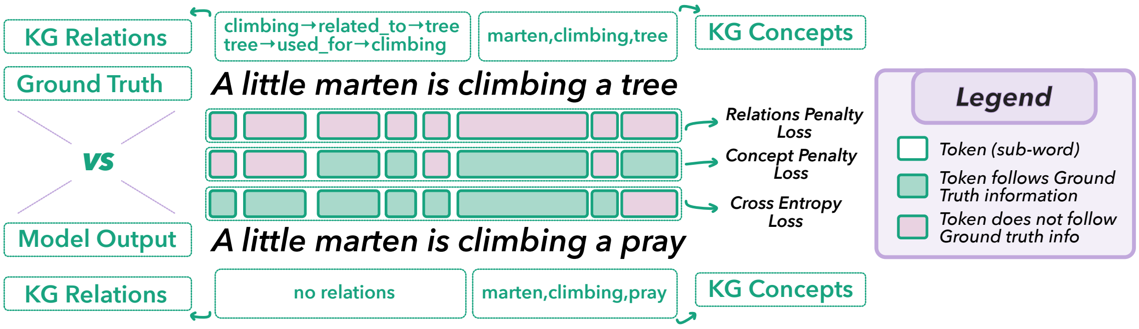


Figure 3.12: Exemplified overview of the several components composing the Commonsense Loss function

Mathematically, the Commonsense Loss is presented in equation 3.5, where the loss consists of three terms: the Cross Entropy Loss (CE), the Concept Penalty (CP) and the

Relations Penalty (RP).

$$\begin{aligned}
\mathscr{L} &= \mathscr{L}_{ce} + \mathscr{L}_{cp} + \mathscr{L}_{rp} \\
\mathscr{L}_{ce} &= -\frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{V} y_{t,j} \times \log(\hat{y}_{t,j}) \\
\mathscr{L}_{cp} &= \frac{T}{e^{o_1 \times (y_{nC} - (o_2 \times (\hat{y}_{nC})))} + 1} \\
\mathscr{L}_{rp} &= \frac{T}{e^{o_1 \times (y_{nR} - (o_2 \times (\hat{y}_{nR})))} + 1}
\end{aligned} \tag{3.5}$$

where:

T corresponds to the sequence length; V corresponds to vocabulary size; $y_{t,j}$ corresponds to the prediction of token j at step t; $\hat{y}_{t,j}$ corresponds to the target token j at step t; y_{nC} is the number of concepts present in prediction y; $\hat{y}_{nC}$ is the number of concepts present in target $\hat{y}$; y_{nR} is the number of relations present in prediction y; $\hat{y}_{nR}$ is the number of relations present in target $\hat{y}$; $o_1 = 2$ and $o_2 = 0.7$: constants heuristically obtained by analysing the visual aspect of the loss and our intentions with the penalisation score.

The **Concept Penalty** (CP) term makes sure the penalty is higher when fewer concepts are used, and reaches a rather low value (~ 0) when the same amount of concepts of the gold label sentence are used. The **Relations Penalty** (RP) works in a similar manner: the fewer relations exist within inter-concepts, the bigger the cost value is and when the amount of relations converge to the relations present labels phrase, the loss reaches a lower value (~ 0). Refer to Fig 3.12 to visually understand the importance of each component. These two components (**CP** & **RP**) are important, because using concepts is not enough, if want less hallucination in our generated text. We want to make sure that the concepts, which get to be used are bounded by our Commonsense Knowledge. When using a Knowledge Graph, the philosophy of the Open World Assumption (OWA) states that what is present in a graph is true, and what is absent is unknown (False or True), so if we can guide the generation more onto the graph we might be generating sentences which make more sense. Thus, relations play an important role on this, since using concepts which are connected by a pre-known relation is an hint we are following our gold knowledge.

3.2.1.2 Concepts Expansion

The approach of identifying ideas/concepts throughout an input context and extracting meaningful Commonsense relationships between them is promising in encoding Commonsense Knowledge, however, it can also be limiting since these relationships are only captured between concepts on the same context. In order to overcome this context bounding issue, we suggested the introduction of 1-hop related (neighbour) concepts, which are appended to the input context, as it can be visualised in Fig 3.13. This technique allows for new knowledge to be introduced, which, hypothetically, might be useful for a more fruitful Text Generation.

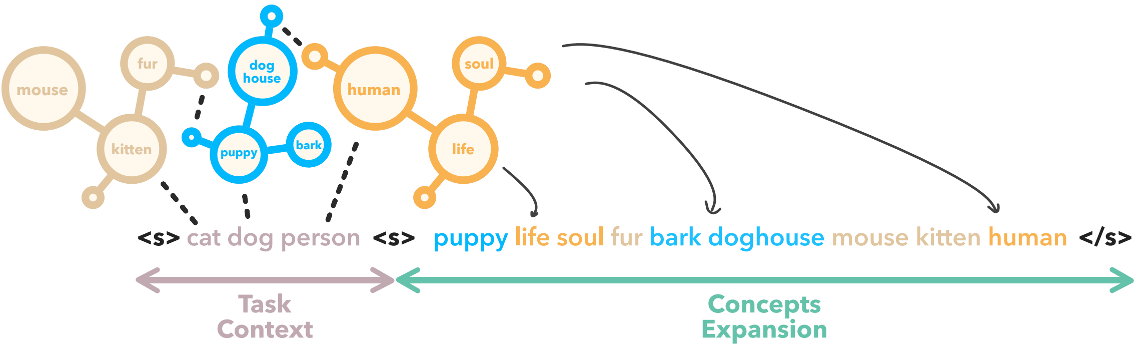

mouse
fur
kitten
dog house
puppy
bark
human
soul
life
<s> cat dog person <s> puppy life soul fur bark doghouse mouse kitten human </s>
Task Context
Concepts Expansion

Figure 3.13: Overview of the Concept Expanding technique in the CommonGen Task.

Concepts Expansion Processing: In order to obtain these 1-hop (neighbour) concepts from the input context, we use two Knowledge Bases: ConceptNet and Swow. As mentioned in Swow work [60], ConceptNet is really relevant because of its magnitude both in terms of concepts/ideas and the relationships between them, which are rich in diversity. However, there are 2 main problems with ConceptNet for this **Concepts Expansion** use case worth being mentioned: **1. [Implementation Problem]** The first problem is related to compound nouns, since ConceptNet usually has them as a single word (without a space) and sometimes separated, through a space (eg. baseball ↔ base ball). This situation raises a problem for 2 reasons: One is that language models were probably trained on only one representation which might lead to poorer encodings and the other reason is that in ConceptNet these two forms have two different nodes representations, making them have different relations, which is not a good thing. **2. [Knowledge Problem]** The other issue regarding ConceptNet is that 1-hop relations are sometimes too forced (semantically speaking), and more obvious relations are sometimes n-hops away.

For the reasons mentioned before, we decided to use both Commonsense Knowledge Graphs, to make use of the best characteristics found in both Graphs. We use Swow for identifying neighbour concepts, which posses more succinct and direct relations between concepts within an input context. After identifying neighbour concepts, we extract them and use ConceptNet to fetch the relations between these two concepts. In short, Swow serves as a neighbour ranking system.

Having extracted neighbour concepts, to introduce these new concepts to the model, we just take the old input and append the new concepts obtained from the process mentioned previously, separated by the Special Token <s>. This is beneficial to the model, to provide it the hint that this new extra information is not necessarily the information which should be taken more importantly, such as the input context one (original concepts).

Note: We decided to focus on 1-hop only neighbours since we were afraid that sibling nodes could introduce non-relevant information, confusing the model with noise.

3.2.1.3 Decoding Strategy

In this work, we make use of a Transformer based model, which is an assembly of an Encoder and Decoder module. As mentioned in chapter 2, whereas the main objective of the Encoder module is to create a succinct and meaningful representation of the input (vector embedding), the decoder's module goal is to take this information and generate a different sequence in a human-level representation, bounded by a certain task (in this case: a textual one). Also, we have seen in section 2.1.4 that there are several strategies used in Text Generation to create more realistic / human like phrases. However, all those methodologies assume that at any time, every token can be used and the most probable token is the token to which the model assigns the biggest probability. There are times, however, where we want to constrain, or alter the probabilities in an attempt to guide the models in generating sentences which follow better our intentions. We also experimented this approach, by trying to guide the text generation to follow the knowledge and relations pre-existent on a Commonsense Knowledge Graph. We studied two approaches: **1. Eager KG Logits Manipulation** & **2. KG Disjunctive Positive Constraint Decoding**. The former is strongly related to a manual direct manipulation of the token probabilities, based on a set of rules defined to conform to the Knowledge Graph. The latter is bounded to the idea of coming up with words/sub-words we want to force our model to use, while giving the "model" the freedom to: 1. add the sub-words in a position which supposedly is the more likely (high probability), and, at the same time 2. to choose from a set of words which is best, while not using all of them.

Even though, we did some *ad hoc* experiments on **Eager KG Logits Manipulation**, we will focus only on the **KG Disjunctive Positive Constraint Decoding** manipulation, since we believe it is the most interesting one in the matter of Textual Generation. For Story Telling problems, there is a vivid need for scenarios substantially rich in diverse concepts. Using external Knowledge Graphs, such as a Commonsense Knowledge Graph there is a possibility to condition a sentence on a more rich scenario, using related concepts. Let's cover an example: Imagine we have a set of concepts ⟨sway, flower, breeze⟩ from which we want to build a scenery of. A gold sentence in the CommonGen dataset would suggest something like: "a group of flowers swaying in a gentle breeze". Using our technique, we will further cover, we could get something like: "flowers, soft yellow and green, swaying in the breeze". One could argue which one is best, but our objectively adds more information. It is commonly assumed, flowers have colours, so we paint them using adjectives. This simple example showcases how useful such method could be. In paragraph 3.2.1.3 we cover the details of this approach, but for a easier understanding see fig 3.14, where we take the example seen in the Concepts Expansion section, and visually see how terms could help generate a vivid descriptive sentence.

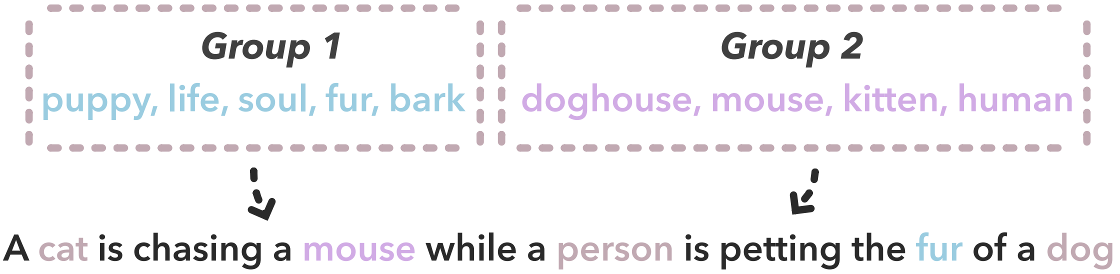

Group 1
puppy, life, soul, fur, bark
Group 2
doghouse, mouse, kitten, human
A cat is chasing a mouse while a person is petting the fur of a dog

Figure 3.14: Abstract overview of the KG Disjunctive Positive Constraint Decoding procedure.

Methodology: From a set of concepts C, we select 1-hop concepts which are semantically close to the concepts C. The way we extract these concepts is based on the Concepts Expansion mechanism, mentioned before. During the decoding (text generation) procedure, we divide these 1-hop concepts C in $nMaxConcepts$ (nC) groups and force the model to use one concept from each group. We do this to maximise the odds of forcing a concept which fits best the context of a certain sentence.

3.2.2 Relation-Aware Masked BART (RAM-BART) Model

Scaling Language Models in parameters size and training data, has unaccountably proven to be a Goliath empowerment in a Language Model output quality, however, applying other techniques especially when such measure tweaks are not possible are also at the vanguard of better Language Models.

Regarding the work of Voita [111], it has been shown that multiple heads allow for different data patterns to be learnt. Work [114], even took this idea and guided the attention heads to specific hand-chosen patterns to be learnt on each of them. Inspired by this work, we challenge ourselves to try an alternative approach to guiding the introduction of Commonsense Knowledge simply through attention masks.

As mentioned previously on chapter 2, not all heads learn “healthy” patterns, from which we purpose to automatically find heads which are on a bad relevance performance and replace them with focused Commonsense Heads, in order to transmit new Commonsense Knowledge awareness and make better use of less relevant heads (see Fig 3.15 for an overview of red (less relevant) heads getting “replaced” with Commonsense Heads).

What exactly are these Commonsense Heads? Like in the previous Relation-Aware BART (RA-BART) model approach, we want to, linguistically, capture concepts and further enrich the learning process with internal relations between concepts. One simple way to model this idea in the mechanism of Attention, is by creating a binary mask (matrix), where 0s represent the absence of a Commonsense Relation between language units, and 1s represent intrinsic relations, extracted from a Commonsense KG, between these language units. To understand, at which Self-Attention algorithmic point the commonsense

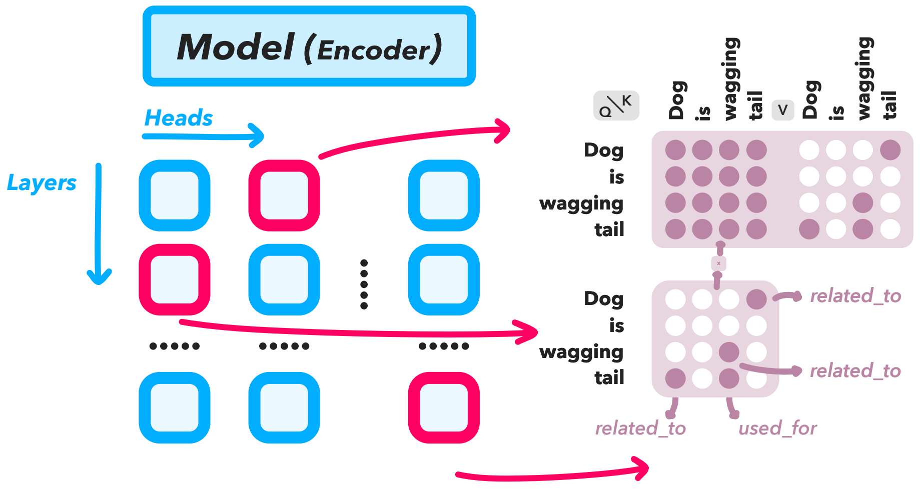


Figure 3.15: Commonsense Heads (red) replacing default heads (blue) and their behaviour.

masking takes place, observe Fig 3.16

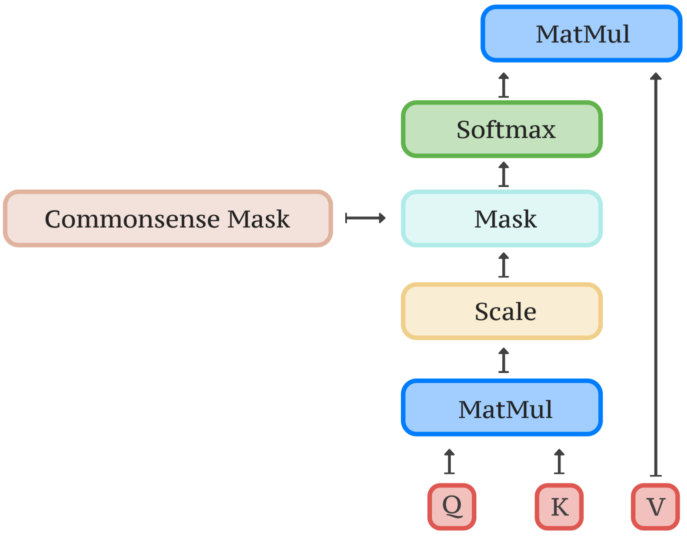


Figure 3.16: Scaled-dot product with a Commonsense mask.

Mathematically, we model this new BART Encoder with gated heads, masking the heads conforming to their importance (see equation 3.6).

$$MultiHead(Q,K,V) = Concat(gh_1,...,gh_h)W^O,$$
$$gatedHead_i = \begin{cases} head_{normal}, & \text{if head is important} \\ head_{commonsense}, & \text{if head is not important} \end{cases} \tag{3.6}$$

How do we define which heads are aiding the model and which not are lacking expressiveness? To achieve this, we suggest to capture the **heads importance** over all Encoder layers during the first epoch of training. We could replace a normal head with a Commonsense one, if it's importance value departs significantly from the most important ones. Mathematically, we can define a z-score, which tells us how far away a specific head's importance is from the general importance distribution. We could further define a variable λ, let's say $\lambda = 1.96$ to bound the z-score, meaning that any head which fails to be less λ standard deviations from the mean, we consider it to be a less useful head and re-purpose it with a Commonsense one (see equation 3.7 for z-score calculation and equation 3.8 for a mathematical overview of heads masking).

$$\text{z-score} = \frac{(hi - \mu)}{\sigma} \tag{3.7}$$

$$gatedHead_i = \begin{cases} head_{normal}, & \text{if z-score}_{hi} > -\lambda \\ head_{commonsense}, & \text{if z-score}_{hi} \leq -\lambda \end{cases} \tag{3.8}$$

Note: **Head importance**, in this context, refer to the accumulative gradients values in regards to an Encoder head, obtained when feeding the model with data. Furthermore, a Commonsense Head is a head which nulls out the attention between tokens which have no "Commonsense" relations between them. The weight itself between relevant tokens is preserved by what the model thinks their importance should be and not forced by us in any way.

One limitation with this attention masking approach is in the lack of relations richness, when modelling the relations between concepts. Since the Commonsense Mask is binary, meaningful relations get squashed into a 1 dimension relation (related_to). An example of this behaviour could be, for instance: ⟨ airplane , at_location , sky ⟩ → ⟨ airplane , related_to , sky ⟩.

3.3 Model Preparation

We have introduced the tasks we wanted to tackle and their respective data, their data preparation techniques, and even about the Commonsense Knowledge retrieving and integration process. Now we mention how we bridge all these things into the BART Language Model.

3.3.1 Data Splits

When training models, one must be careful to strive for unbiased quality estimations. Therefore we divide each task in 3 subsets of data: the **training** data (~90% of data), the **validation** data (~5% of data), and the **testing** data (~5% of data). **Training data** is only used to update the model parameters. The **validation data** is used to choose which model is best, on a given moment in time, according to a certain metric. Finally, the **testing data** is crucial for an unbiased estimation of the models' output, since, hopefully, the testing data has not been seen either in the training, or in the validation phase (See Fig 3.17). Cross Validation is commonly avoided in Deep Learning especially due to high intensity computational runs.

Training data
(Update model's Parameters)
Validation Data
(Choose best model)
Testing data
(Unbiased Evaluation Estimation)

Figure 3.17: Dataset split strategy.

3.3.2 Transforming tasks data into model input

The tasks we are considering are independent and have their own specifications in terms of the data used, but also in the way they get introduced to the models. In Fig 3.18, we go through an overview exactly of *what* data and *how* data is fed to the (**tokenizers** → **models**) pipeline. Worth referring that this depiction concerns only the Commonsense models, which are fed with the Commonsense Knowledge extracted from the correspondent input data. In summary, Fig 3.18 depiction helps us understand, that even though

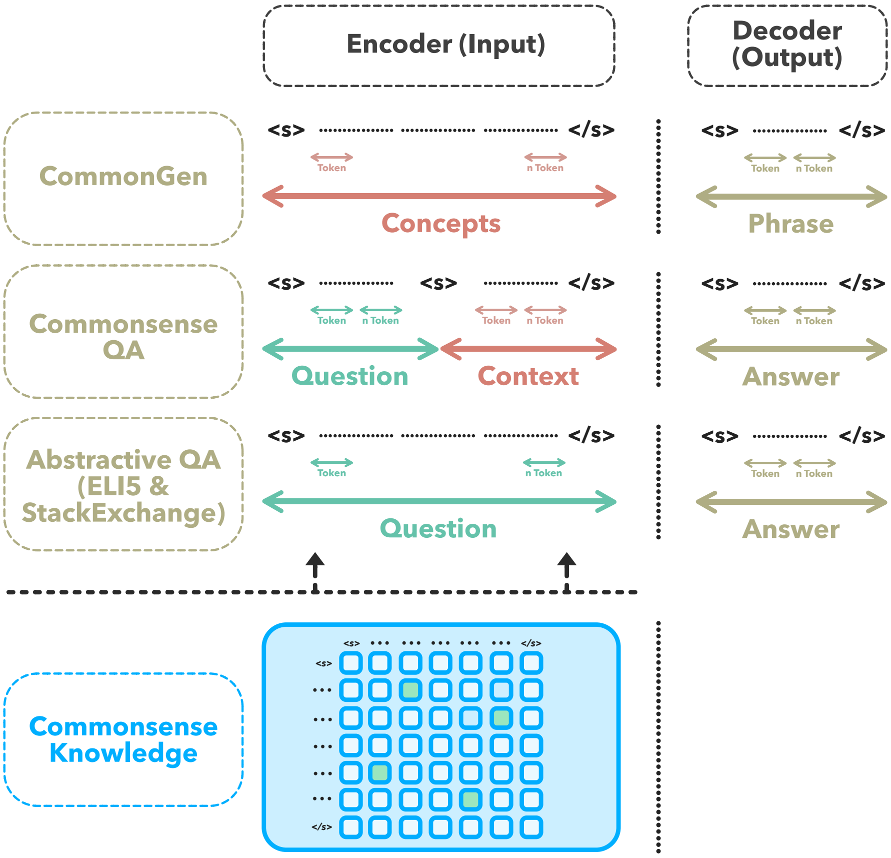


Figure 3.18: Overview of the model input for each task, regarding our Commonsense Models.

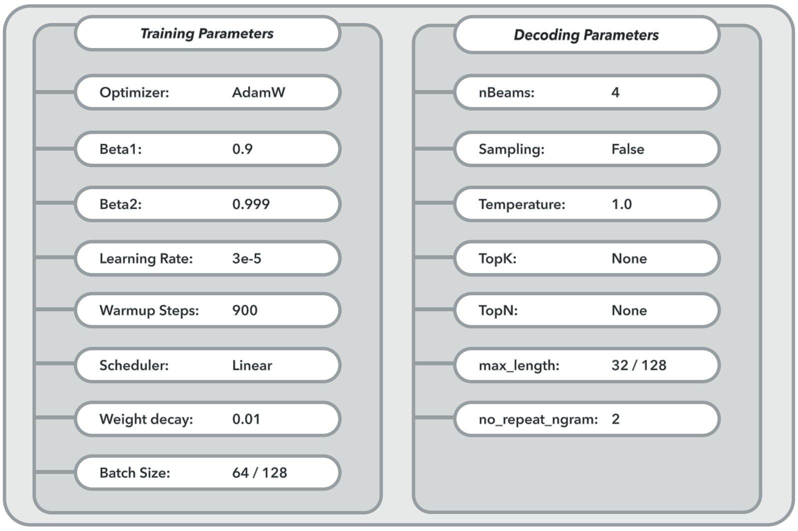
all tasks are different, this difference to the model is, more or less, homogeneous. Even if we are working with Concepts (CommonGen), multi-choice questions (CSQA) or general questions (Abstractive QA), in the end this textual data gets parsed by a Tokenizer module, which converts text units with numeric ids bounded by Special Tokens (<s>,</s>), which respectively relate to the beginning of contextual data and the end of contextual input. Not represented in the figure, we also have the **<pad>** Special Token to fill extra empty token spaces, due to the fact that we train and normally infer in batches (collections of input samples) which have different length. This length difference must be "hidden" to the model when using batches, to successfully support matrix (parallel) operations.

The **CommonsenseQA** task, as explained previously, to be taken as a generative task we need to append the possible answer choices to the question. We considered the use of the **<s>** token to separate the question from the choices, as we observed in BART related literature, and also decided to separate each answer choice with a **;** token for two reasons. First, an early empirical test, where we did not use a separation token between choices showed poor results, and second because it is reasonable to believe that during the pre-training of BART model, the **;** token was seen separating ideas, as it is normally used in English with that purpose.

In section 3.2.1, we mention a experiment (**Concepts Expansion**) which can alter the models input in some ways, not covered in this figure (eg. further adding neighbour Commonsense Concepts to the input, observed in section 3.2.1.2). Relating to this experiment and Fig 3.13 we can also observe the technique of using the **<s>** token to separate the major important concepts and the retrieved neighbour concepts which can possibly help providing more knowledge to the model.

3.3.3 Model Hyper-Parameters

On Fig 3.19, one can observe the hyper-parameters used to train the models in our work, and the decoding hyper-parameters used to generate Natural Language sentences. We briefly explored some different parameters, and found this combination to provide good results.



Training Parameters	
Optimizer:	AdamW
Beta1:	0.9
Beta2:	0.999
Learning Rate:	3e-5
Warmup Steps:	900
Scheduler:	Linear
Weight decay:	0.01
Batch Size:	64 / 128

Decoding Parameters	
nBeams:	4
Sampling:	False
Temperature:	1.0
TopK:	None
TopN:	None
max_length:	32 / 128
no_repeat_ngram:	2

Figure 3.19: Overview of the most important training and decoding Hyper-Parameters.

4

Evaluation & Analysis

This chapter presents in detail the methodology used for assessing our proposed work and comparing it with the established baselines.

As William Thomson would say “to measure is to know”. Science moves forward because we establish standards, metrics to structurally showcase the quality of models and approaches in comparative manners. In this section, we measure our approaches, along with baselines using automatic measures, in order to compare them. We do not stop here: knowing the importance of human evaluation, we further enquire humans to assess our approaches to have a concrete human feel of their quality. We further discuss the environmental impact of this dissertation and talk about the memory intensity of integrating Commonsense Knowledge. Lastly, we provide a hyperlink reference to a web page where the reader can experiment the work pursued in this dissertation.

4.1 General Evaluation

Provided that in this work we are concerned with the integration of Commonsense Knowledge in Language Models, we have selected benchmarks concerned in evaluating them in such matter. As mentioned in Chapter 3, for Commonsense Understanding and Generation capabilities, we have selected the CommonGen [56] dataset, and especially for Commonsense Understanding we have chosen the CommonsenseQA [104] dataset. We further evaluate our model on our custom Abstractive QA dataset, which was used in a real project, Alexa Taskbot Challenge [59], to answer (general, cuisine, DIY) user questions.

4.1.1 Automatic Evaluation

In chapter 2, we mentioned text related automatic metrics. In regards, to that study on the current automatic evaluation trends and our objectives, we chose to use the BLEU and the Rouge metric for a systematic evaluation over the syntactic quality level of the text generated. The Meteor is both for the assessment of the syntax and for a light evaluation on the meaning of the generation. We use CIDEr, as it is strong for more closely accessing the human-likeness of the generations. Finally, we use SPICE for an automatic evaluation regarding the semantic meaning of the generated text.

These metrics, however, were not designed to measure Commonsense in the language generation task. Therefore apart from falling back to these standard automatic metrics we further introduce custom metrics, which we believe to a certain level, help us "automatically" observe whether the models are capturing more or less the Commonsense Knowledge. These custom metrics are the **Concepts Coverage** (Coverage) and the **Relations Weight** (R-W) metric, which respectively correspond to the amount of concepts within the generated text versus the reference one and the amount of relations between concepts within the generated text versus the reference one.

Note: On the tables, when BLEU and Rouge appear we are respectively using **BLEU-3** (cumulative 3-gram matching) and **Rouge-L** (longest matching sequence) scores. Bold is used on the tables to highlight the best values, while dotted values represent the second best result. Furthermore, since **Concepts Coverage** (Coverage) and the **Relations Weight** are relative to the corresponding gold references, any score under 100% mean that the generations are utilising, respectively, less concepts and relations, whereas the opposite, mean that the generated sentences are using more concepts and relations.

4.1.1.1 CommonGen Task:

Generating sentences which make sense constrained under bounding concepts is interesting for many use cases (eg. story telling [123], etc). We were interested in using it to help understand if our proposed model could capture better concept relations and create more sounding sentences. Regarding this task, we performed several studies as mentioned in Chapter 3 and we will now cover their results.

Table 4.1: Commonsense Loss Comparison.

Model	**BLEU**	**ROUGE**	**METEOR**	**CIDEr**	**SPICE**	**Coverage**	**R-W**
RA-BART	22.85%	46.36%	46.32%	18.36%	50.13%	73.30%	70.13%
RA-BART (CL)	23.75%	46.59%	46.71%	18.96%	50.19%	74.90%	71.90%
	+0.90%	+0.24%	+0.38%	+0.61%	+0.07%	+1.60%	+1.77%

Commonsense Loss: As a reminder, we set out to test whether the manipulation of the standard Seq2Seq loss on Text Generation could be improved using a custom Commonsense Loss. Looking at the results, presented in the table 4.1, we see some improvements in using this custom Commonsense Loss. More concepts seem to appear in the generated phrases, as well as relations between them. Overall generated text, in the eyes of the automatic metrics also seems to improve, where BLEU score improves almost 1%.

Table 4.2: Concept Expansion Comparison.

Model	**BLEU**	**ROUGE**	**METEOR**	**CIDEr**	**SPICE**	**Coverage**	**R-W**
RA-BART	22.76%	46.35%	46.26%	18.28%	50.12%	73.27%	70.35%
RA-BART (CE)	22.79%	46.13%	46.23%	18.56%	50.18%	74.02%	71.89%
	+0.03%	-0.22%	-0.03%	+0.28%	+0.06%	+0.75%	+1.54%

Concept Expansion: Here, we were concerned whether directly enforcing 1-hop concept information and their relations would be useful to the model. According to our results (table 4.2), there seems to be a solid positive impact on the richness on the number of concepts and relations, but when considering the other metrics, this approach shows a not so relevant impact.

Decoding Strategy: In table 4.3, we take the Commonsense Knowledge Graph Decoding Strategy forcing both 1 concept and 2 concepts from input concept neighbours and compare them with the baselines approaches. One thing to be aware is that, normally, some metrics will score lower since we are adding up entropy in the generation. Most metrics we use are n-gram dependent so if between concepts we add other concepts, an n-gram overlap will suffice. The acronym: **KGD-nC** stands for using the Knowledge Graph Decoding Strategy, using nC groups of concepts ($nMaxConcepts$).

Table 4.3: Decoding Strategy Comparison.

Model	**BLEU**	**ROUGE**	**METEOR**	**CIDEr**	**SPICE**	**Coverage**	**R-W**
RA-BART	**22.85%**	**46.36%**	**46.32%**	**18.36%**	**50.13%**	73.30%	70.13%
KG-BART	19.15%	43.90%	43.03%	17.37%	47.06%	77.02%	97.92%
RA-BART (KGD-1C)	21.74%	42.54%	44.60%	16.59%	44.89%	85.72%	103.45%
RA-BART (KGD-2C)	19.48%	39.05%	43.37%	14.01%	39.00%	**103.21%**	**159.25%**

Considerations: It is worth mentioning, that such method, without further engineering, sometimes attempts to start formulating a new sentence just to add a KG concept (eg. concepts: "sky, dark, moon"; KG Concepts: "clouds"; generation: "The moon lies over a dark sky. clouds". This is not ideal, because the dataset we are using only possess one sentence and so the generation can get stuck on a sentence followed by an incomplete one. There is a *naive* approach of clipping a generation on a full point to ingenuously only have one useful sentence. This approach falls short for several reasons: **Firstly**, we would probably loose an external concept which could better help textually paint the previous sentence; **Secondly**, this would be a way to suggest to our model that it is doing a good job, when it is not. Therefore, we came up with a solution consisting in a loss penalisation for the model training, provided that a generated text contains more than one sentence.

Following this loss penalisation procedure, we endure the model in understanding that generating more sentences is not supposed to happen.

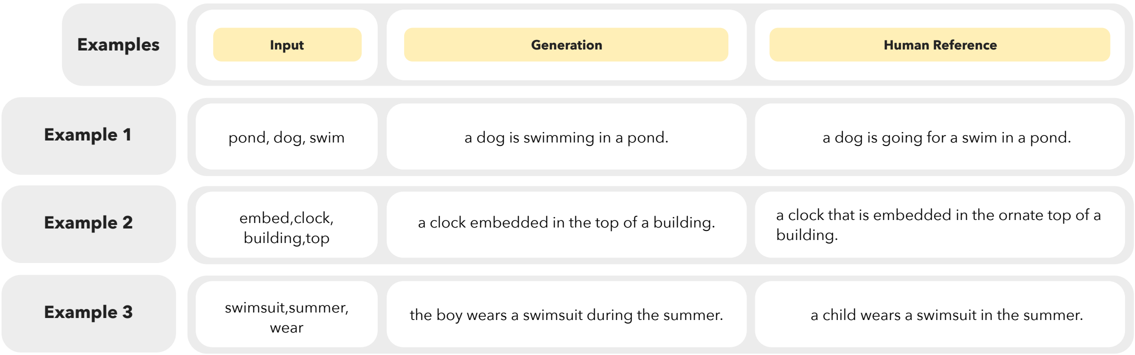
This experiment can be observed as an interesting take on the quality bias of most automatic metrics. Labels/gold references not always are ultimate truths, as it is the case of datasets like ELI5 or CommonGen. Possibly, a larger amount of human-like generations end up getting penalised by the enforcement of metrics strongly dependent on labels likeness. Having said this, looking at the Textual Generations scores in Table 4.3, we see more descriptive textual results pointed by the Coverage and R-W metrics, in contrast with the more standard, n-gram restrictive metrics. Nevertheless, the balance of enforcing concepts must be taken with care, since hallucinations are always a possibility and worsened with artificial reinforcements. On the Human Evaluation section 4.1.2, we will see that humans acknowledge better generations to this model experiment.

CommonGen Results Summary: In table 4.4, we showcase the results of all models from experiments done over the task of CommonGen, trained for 10 epochs each.

Table 4.4: Comparison between methodologies on CommonGen.

Model	BLEU	ROUGE	METEOR	CIDEr	SPICE	Coverage	R-W
BART	21.95%	45.38%	45.65%	17.97%	49.95%	73.21%	70.84%
KG-BART	19.15%	43.90%	43.03%	17.37%	47.06%	77.02%	97.92%
RA-BART	22.85%	46.36%	46.32%	18.36%	50.13%	73.30%	70.13%
RA-BART (CL)	**23.75%**	**46.59%**	**46.71%**	**18.96%**	**50.19%**	74.90%	71.90%
RA-BART (CE)	22.79%	46.13%	46.23%	18.56%	50.18%	74.02%	71.89%
RA-BART (KGD-1C)	21.74%	42.54%	44.60%	16.59%	44.89%	**85.72%**	**103.45%**

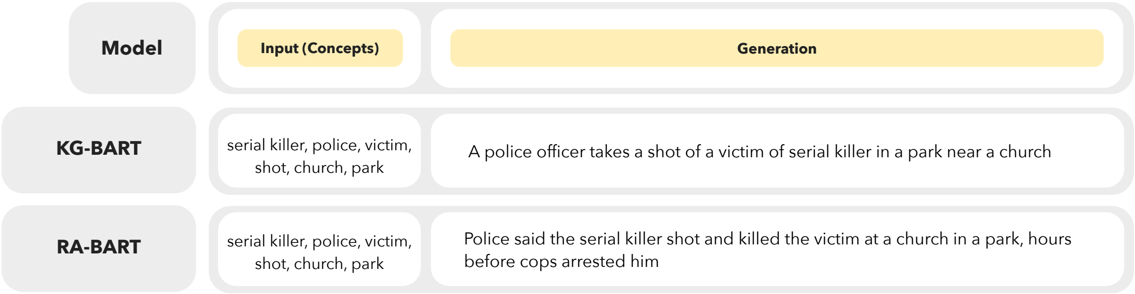
For a visual overview of the standard RA-BART results, look at the Fig 4.1. Three examples are presented showcasing the ability of creating meaningful sentences from a set of concepts. For a more detailed view comparing the outputs of every considered model, see Fig A.1.



Examples	Input	Generation	Human Reference
Example 1	pond, dog, swim	a dog is swimming in a pond.	a dog is going for a swim in a pond.
Example 2	embed,clock, building,top	a clock embedded in the top of a building.	a clock that is embedded in the ornate top of a building.
Example 3	swimsuit,summer, wear	the boy wears a swimsuit during the summer.	a child wears a swimsuit in the summer.

Figure 4.1: Examples of phrases generated over Commmonsense Generation Task (CommonGen) using our standard Relation-Aware BART.

Discussion: Looking at the table 4.4, the approach of Relation-Aware Attention allied with some other techniques explained in further detail in chapter 3, such as adding a custom loss function, expanding concepts or even using a special decoding strategy algorithm suggest better results on this task. KG-BART, on our tests, seems to lack some quality in the Text Generation task, as seen in Fig 4.2, Fig A.1 and looking at their table results (Table 4.4). KG-BART's concepts coverage and relations presence are, however, remarkable.



Model	Input (Concepts)	Generation
KG-BART	serial killer, police, victim, shot, church, park	A police officer takes a shot of a victim of serial killer in a park near a church
RA-BART	serial killer, police, victim, shot, church, park	Police said the serial killer shot and killed the victim at a church in a park, hours before cops arrested him

Figure 4.2: Examples of phrases generated over Commmonsense Generation Task (CommonGen) using our constrained Relation-Aware BART.

4.1.1.2 CommonsenseQA task:

To analyse how well our Commonsense models can help in the field of Commonsense Understanding, we gathered the CommonsenseQA question-answering dataset. We wanted to see if our models could help answering questions which are mostly Commonsense related. Since this dataset is a multi-choice question problem, we adapted the models to also have information of the possible answer options, so the task is easier by indirectly constraining the possible tokens. As seen in section 3.3.2, Fig 2.3, the question gets separated by the answer choices with a **<s>** token and the choices themselves are separated by a "**;**" token.

Consideration: We do not actually constrain the possible tokens to be extracted from the possible choice answers, since the models implicitly understand that the possible choices can be found within their input data. We, actually, performed an *ad hoc* experiment, where we created a custom constrained decoder, which would bound the generated tokens from a standard list of ~50k tokens to the unique tokens present in the possible answer choices fragment of the input. As we observed no relevant difference in performance, we abandoned the custom constrained decoder, since not using it would make our model simpler.

Discussion In table 4.5, we can see a baseline comparison between the standard BART model our RA-BART model, which were trained for 10 epochs. We can see that

Table 4.5: Comparison between methodologies on CommonsenseQA.

Model	Accuracy
BART	**75.17%**
RA-BART	57.52%

the approach of Relation-Aware Attention is considerably worse by a solid ~-17% difference. This makes us believe that adding external relational Commonsense information, may for some tasks confuse the model, instead of helping the model to achieve a better interpretation of the world.

4.1.1.3 ELI5 & StackExchange task:

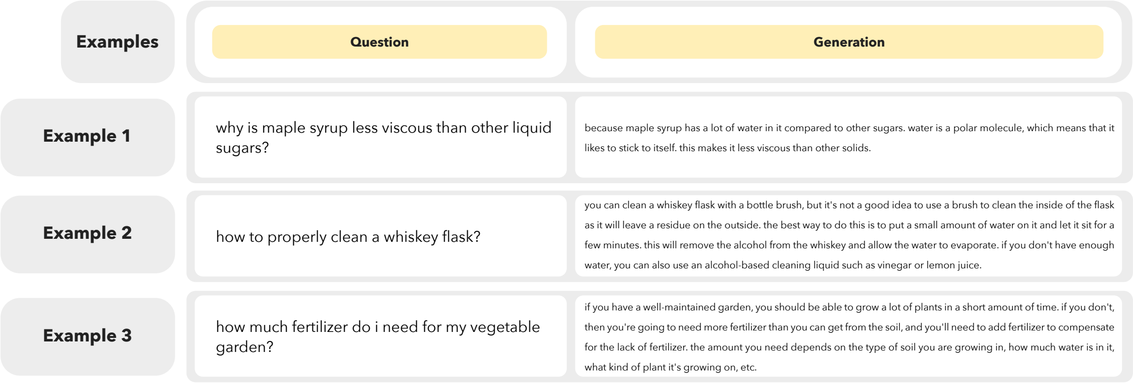
The Conversational engagement boost created by answering users' questions in a more natural way is essential in conversational systems. We trained both standard BART model and Relation-Aware BART model on our custom Abstract QA dataset, whose main goal was to answer a question which could pop up while conversing with a chatbot under the cuisine and DIY theme. In table 4.6, one can observe the metrics we captured for this problem. We trained all models for 10 epochs and evaluated them over the automatic metrics mentioned before, except for SPICE for computational reasons (time exhausting).

Table 4.6: Comparison between methodologies on Abstractive QA.

Model	BLEU	ROUGE	METEOR	CIDEr	Coverage	R-W
BART	3.29%	14.51%	15.66%	0.30%	55.19%	52.58%
RA-BART	3.35%	14.57%	15.96%	0.30%	56.34%	54.95%
	+0.06%	+0.06%	+0.30%	+0.00%	+1.15%	+2.37%

See Fig 4.3 for an example of the quality of the generated text over the our custom task of abstractedly answer questions. For a more detailed view comparing several outputs of every considered model, see Fig A.2.

Discussion: In table 4.6, we can see that the approach of Relation-Aware BART is slightly better on every metric. This, however, does not mean that RA-BART always outperforms normal BART. Attempting to generate answers in an abstract way tends to incite false statements sometimes. Having said this, however, we believe that since there are 2% more Commonsense Relations on the answers, the answers might be in more conformity with our Commonsense Knowledge Graph and therefore, generations might suffer less from non-sense answers, which we call language hallucinations.



Examples	Question	Generation
Example 1	why is maple syrup less viscous than other liquid sugars?	because maple syrup has a lot of water in it compared to other sugars. water is a polar molecule, which means that it likes to stick to itself. this makes it less viscous than other solids.
Example 2	how to properly clean a whiskey flask?	you can clean a whiskey flask with a bottle brush, but it's not a good idea to use a brush to clean the inside of the flask as it will leave a residue on the outside. the best way to do this is to put a small amount of water on it and let it sit for a few minutes. this will remove the alcohol from the whiskey and allow the water to evaporate. if you don't have enough water, you can also use an alcohol-based cleaning liquid such as vinegar or lemon juice.
Example 3	how much fertilizer do i need for my vegetable garden?	if you have a well-maintained garden, you should be able to grow a lot of plants in a short amount of time. if you don't, then you're going to need more fertilizer than you can get from the soil, and you'll need to add fertilizer to compensate for the lack of fertilizer. the amount you need depends on the type of soil you are growing in, how much water is in it, what kind of plant it's growing on, etc.

Figure 4.3: Examples of phrases generated over our custom Abstract QA dataset using our standard Relation-Aware BART.

From our perspective and having implemented such Abstractive QA task in a real-life chatbot (Alexa) with real users [59], we can see how important such task is in creating positive interactions. Therefore, hallucinating less is a powerful benefit.

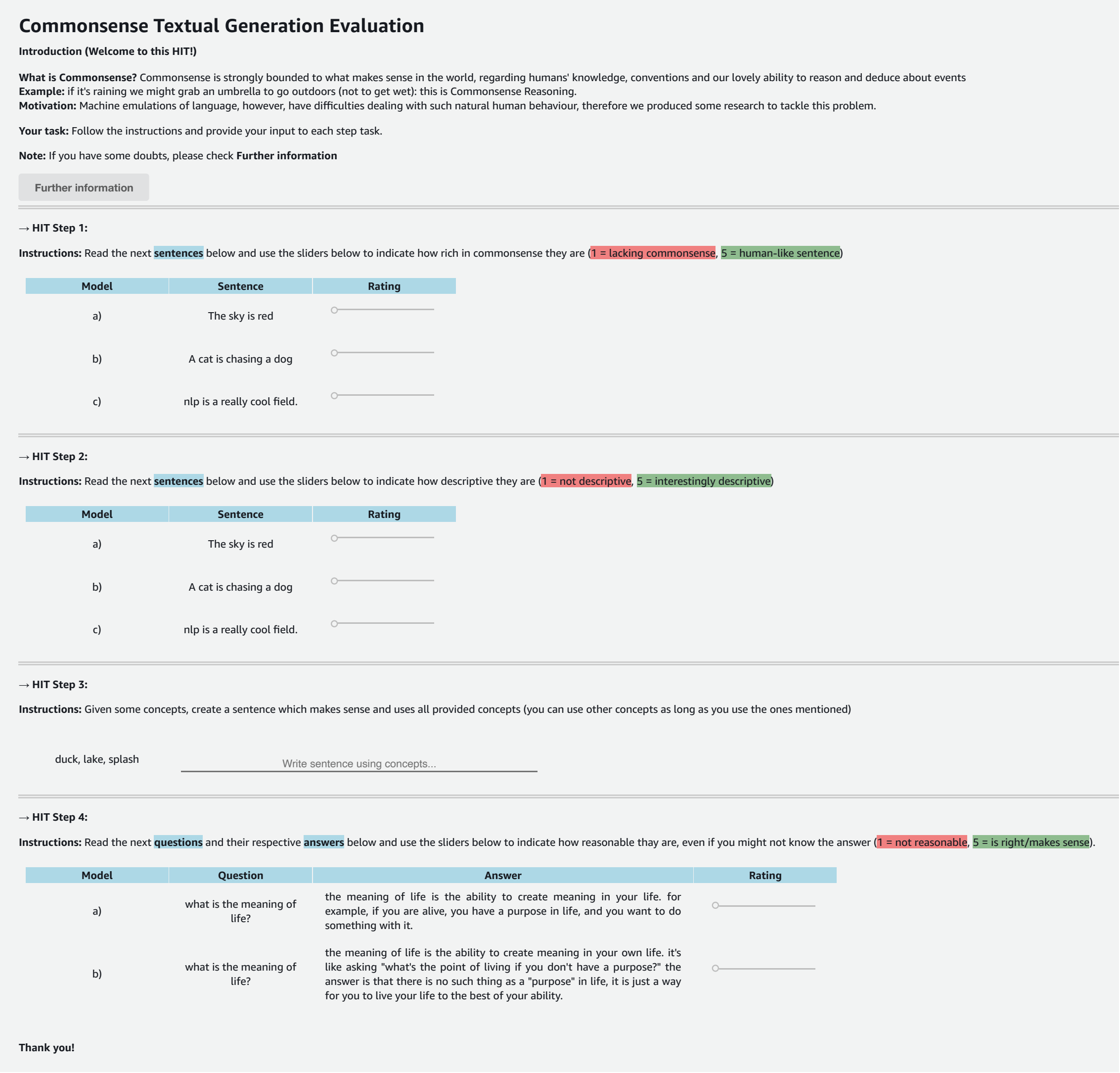
One could think of many problems, which could arise by using a model trained on this task, (such as: what happens if the system does not know the answer?), but it should not hide the fact that it is a powerful human-like engagement tool. In reality, it can generate non-sense answers, but at tricky questions it can also counter them quite nicely with general takes on the question. There is room for improvement on this avenue, such as working on RAG [52]-like models, which fuse retrieval models with generative capabilities. We see RAG models as a more accurate alternative to “simple abstractive models”, as long as the retrieved documents used to ground the answers, are conversationally similar to reddit forums.

4.1.2 Human Evaluation

Ultimately, the models and algorithms we propose are targeted for a human audience, therefore a manual evaluation performed by humans is necessary to truly understand its impact on a real life scenario. To this end, we prepared a Mechanical Turk (mturk[1]), crowd-work assessment (HIT) with 4 tasks to be completed (see Fig 4.4 for an overview of the crowdwork HIT task). Overall, we prepared 100 HITs and paid each worker \$0.5 per HIT completion. Furthermore, our HIT Assessment on average was completed in ∼5 minutes.

Let us now describe the HIT conceiving process. For each HIT Assessment, we randomly extracted 1 instance from a custom CommonGen test set, and asked each annotator to evaluate the outputs of the models we were comparing (In appendix A, Fig A.1 one can see part of our custom test inputs and models outputs). As mentioned, each annotator was presented with 4 tasks. In the first task, we set a general Commonsense sentence evaluation, where annotators were asked to rank the outputs in a scale of 1(worst) - 5(best)

[1] https://www.mturk.com/



Commonsense Textual Generation Evaluation

Introduction (Welcome to this HIT!)

What is Commonsense? Commonsense is strongly bounded to what makes sense in the world, regarding humans' knowledge, conventions and our lovely ability to reason and deduce about events
Example: if it's raining we might grab an umbrella to go outdoors (not to get wet): this is Commonsense Reasoning.
Motivation: Machine emulations of language, however, have difficulties dealing with such natural human behaviour, therefore we produced some research to tackle this problem.

Your task: Follow the instructions and provide your input to each step task.

Note: If you have some doubts, please check **Further information**

→ **HIT Step 1:**

Instructions: Read the next **sentences** below and use the sliders below to indicate how rich in commonsense they are (1 = lacking commonsense, 5 = human-like sentence)

Model	Sentence	Rating
a)	The sky is red	○
b)	A cat is chasing a dog	○
c)	nlp is a really cool field.	○

→ **HIT Step 2:**

Instructions: Read the next **sentences** below and use the sliders below to indicate how descriptive they are (1 = not descriptive, 5 = interestingly descriptive)

Model	Sentence	Rating
a)	The sky is red	○
b)	A cat is chasing a dog	○
c)	nlp is a really cool field.	○

→ **HIT Step 3:**

Instructions: Given some concepts, create a sentence which makes sense and uses all provided concepts (you can use other concepts as long as you use the ones mentioned)

duck, lake, splash

→ **HIT Step 4:**

Instructions: Read the next **questions** and their respective **answers** below and use the sliders below to indicate how reasonable thay are, even if you might not know the answer (1 = not reasonable, 5 = is right/makes sense).

Model	Question	Answer	Rating
a)	what is the meaning of life?	the meaning of life is the ability to create meaning in your life. for example, if you are alive, you have a purpose in life, and you want to do something with it.	○
b)	what is the meaning of life?	the meaning of life is the ability to create meaning in your own life. it's like asking "what's the point of living if you don't have a purpose?" the answer is that there is no such thing as a "purpose" in life, it is just a way for you to live your life to the best of your ability.	○

Thank you!

Figure 4.4: MTurk HIT.

in conformity with whether the sentence made sense or not. In the second task, we asked them to rank the sentences in terms of which output was more descriptive/verbose in providing more detail about a scenario. The third task was designed to model an HLP Assessment of the CommonGen task, for comparison purposes. We asked annotators for themselves to create a sentence syntactically and semantically correct using a set of concepts (eg. ⟨shark, sea, boat⟩ → a shark is attacking a boat lost in the sea). The last task was to create an assessment of our custom Abstractive QA task, for comparison purposes. In this case, we asked annotators to rank how reasonable, in a scale of 1(worst) - 5(best) , the answers were (even if the answers could be wrong or if the worker would not know the answer).

In table 4.7, we can see the results of the first task, where we compare different model architectures: the default BART, KG-BART, our RA-BART, our RA-BART trained with the Commonsense Loss and the RA-BART with the Knowledge Graph special decoding

Table 4.7: Human Commonsense Evaluation

Model	**1**	**2**	**3**	**4**	**5**	**Rating**	**0-100%**
BART	24%	10%	19%	17%	30%	3.19	63.8%
KG-BART	36%	8%	14%	15%	27%	2.89	57.8%
RA-BART	25%	5%	17%	27%	26%	3.24	64.8%
RA-BART (CL)	21%	7%	13%	28%	31%	**3.41**	**68.2%**
RA-BART (KGD-C1)	18%	14%	13%	19%	36%	**3.41**	**68.2%**

strategy. We can see that through human eyes, our approaches seem to behave better and generate more coherent phrases by a solid margin (∼ 4%). KG-BART scores the worst, which further verifies the poorer automatic scores obtained in the previous sections and also suggests that both **Coverage** and **R-W** metrics are not an enough measure for meaningful textual results.

Table 4.8: Human Information Richness Evaluation

Model	**1**	**2**	**3**	**4**	**5**	**Rating**	**0-100%**
BART	15%	10%	28%	25%	22%	3.29	65.8%
KG-BART	17%	15%	26%	26%	16%	3.09	61.8%
RA-BART	13%	10%	30%	25%	22%	3.33	66.6%
RA-BART (CL)	13%	8%	23%	31%	25%	3.47	69.4%
RA-BART (KGD-C1)	10%	6%	19%	26%	39%	**3.78**	**75.6%**

In table 4.8, we can see the results of the second task, where we focus on seeing whether the sentences generated are more rich in content (eg. "a boy eats an hamburguer" is less semantically rich than "a boy eats a juicy hamburguer" , even through they are both syntactically and semantically correct). Here we can see that, even though both RA-BART (CL) and RA-BART (KGD-C1) score similarly in terms of Commonsense, the Knowledge Graph decoding strategy seem to further enrich the sentences with more information (more ∼10% compared to best baseline).

Table 4.9: CommonGen HLP vs Baseline Models - Comparison

Model	**BLEU**	**ROUGE**	**METEOR**	**CIDEr**	**SPICE**	**Coverage**	**R-W**
BART	5.88%	24.76%	23.28%	5.50%	23.39%	86.40%	84.14%
KG-BART	5.16%	23.18%	20.48%	5.38%	20.76%	87.83%	109.15%
HLP	**9.62%**	**28.56%**	**34.245%**	**8.26%**	**36.04%**	105.01%	132.93%
RA-BART	6.96%	23.90%	23.43%	6.56%	22.97%	84.44%	85.98%
RA-BART (CL)	6.14%	23.44%	23.14%	6.25%	22.67%	78.05%	84.62%
RA-BART (KGD-1C)	6.132%	22.37%	22.53%	5.33%	22.62%	**106.62%**	**167.07%**

In table 4.9, we find the HLP assessment of the CommonGen task. HLP let's us understand *what* to expect from the models, and *what* yet can realistically be achieved. These metrics were obtained using a realist small custom CommonGen alike-set created for the purpose of the MTurk Assessment. We created a 100 sized dataset to realistically evaluate the models on human-like events and even a small amount on "fantastic" scenarios.

As it would be expectable, the human performance outperforms the automatic metrics, except for the **Coverage** and **Relation-Weight** ones. This is surprising and even interesting, because it shows how the Knowledge Graph Decoding Strategy furiously enforces Commonsense Knowledge in creative ways. As we have also seen, regarding the Knowledge Graph Decoding Strategy approach, even though the other standard metrics lack a small percentage in comparison, humans seem to like them more. Additionally, we found it interesting that the Commonsense Loss BART Model, in terms of metrics, performs worse than RA-BART, even though humans suggest that Commonsense Loss BART is better by a significant difference (see table 4.7 and table 4.8).

Table 4.10: Human Abstractive QA Evaluation

Model	**1**	**2**	**3**	**4**	**5**	**Rating**	**0-100%**
BART	26%	12%	13%	24%	25%	3.10	62.0%
RA-BART	16%	9%	13%	28%	34%	**3.55**	**71.0%**

Lastly, we have table 4.10, where we compare both the default BART architecture and the RA-BART one on the Abstractive QA task surveyed on the forth HIT task. Looking at the results we see quite a solid improvement on the RA-BART model over the default one (more ~9%). For further context, Abstract QA models answers from this MTurk evaluation, can be consulted in Fig A.2, on the appendix A.

Special Notes: We did our best to ensure the human evaluations were well behaved (not random or with bad intentions), by forcing evaluators to go through control tests and discard crowd-workers which did not follow the minimum criteria. Having a more difficult task made sure we could spot easily several bad intended assessors, making our statistics more credible. One observation worth being made is how crowd-workers seem to arrange clever ways of by-passing a rejection status: from Wikipedia information span retrieval to what seemed to be the usage of Language Generation Models to fill our task's step 3. This came as an eye-opener to how "dangerous" crowd-working results can be, if not carefully analysed.

4.2 Ablation Studies

The Machine Learning field is much broader than conceiving models and petting data. It is dependent on a critical thinking life-cycle, where researchers, developers, project stakeholders meditate about the value being created, the trade-offs and critical points for relevant experiments and optimisations. Objectively, in this section, we cover: **1.** *how* the model size can impact the benefits of using External Commonsense Knowledge, **2.** *whether* dynamically selecting attention heads, which might be less relevant and replacing them with Commonsense Heads, can help the BART model to perform better.

4.2.1 Language Model Scaling Law (Explicit Knowledge)

In Language Models realm, scaling in parameters along with data size incites better model performance [41]. In our work, we would like assess whether explicit knowledge helps when we have less parameters or if, in contrary, explicit knowledge goes hands in hands with the increased implicit capabilities of larger Language Models. We wish to compare a Bart-base model, consisting of 140M parameters (size: ∼0.5GB) and Bart-Large model consisting of 400M parameters (size: ∼1.5GB).

Table 4.11: Language Model Scaling Law Model Comparison.

Model	BLEU	ROUGE	METEOR	CIDEr	SPICE	Coverage	R-W
BART-Base (Normal)	19.43%	44.49%	**43.74%**	16.37%	49.33%	69.03%	61.02%
RA-BART-Base	**19.88%**	**44.57%**	43.60%	**16.71%**	**49.19%**	**69.19%**	**64.24%**
	+0.45%	+0.08%	-0.15%	+0.34%	-0.14%	+0.16%	+3.22%
BART-Large (Normal)	21.95%	45.38%	45.65%	17.97%	49.95%	73.21%	**70.84%**
RA-BART-Large	**22.85%**	**46.36%**	**46.32%**	**18.36%**	**50.13%**	**73.30%**	70.13%
	+0.89%	+0.97%	+0.67%	+0.38%	+0.18%	+0.09%	-0.71%

Discussion: As mentioned, we compare two family-alike models different in the magnitude of parameters (∼3x). In table 4.11 we can observe a direct comparison between the smaller BART models *with* and *without* the Commonsense Knowledge integration and between the bigger BART models on the same matter. Commonsense Knowledge Integration, on both cases seem to aid the model in more quality text generation (looking at the metrics). Concerning the overall improvements of the Relation-Aware approach we can also suspect the the Commonsense integration seem to have a bigger impact on models with more parameters.

4.2.2 Relation-Aware Masked BART (RAM-BART) Model:

We used the CommonGen task as a case study and trained the Relation-Aware Masked BART for that task. Objectively, we train the BART model, for 1 epoch, and after that we adapt the heads according to their heads importance for 10 more epochs of training. During the training, therefore, we train the model with two different types of heads, the regular ones, and Commonsense enriched ones. In table 4.12, we present the results of these two models in comparison with the baselines we established.

Table 4.12: Masked Relation Aware BART compared to baseline models on the CommonGen task.

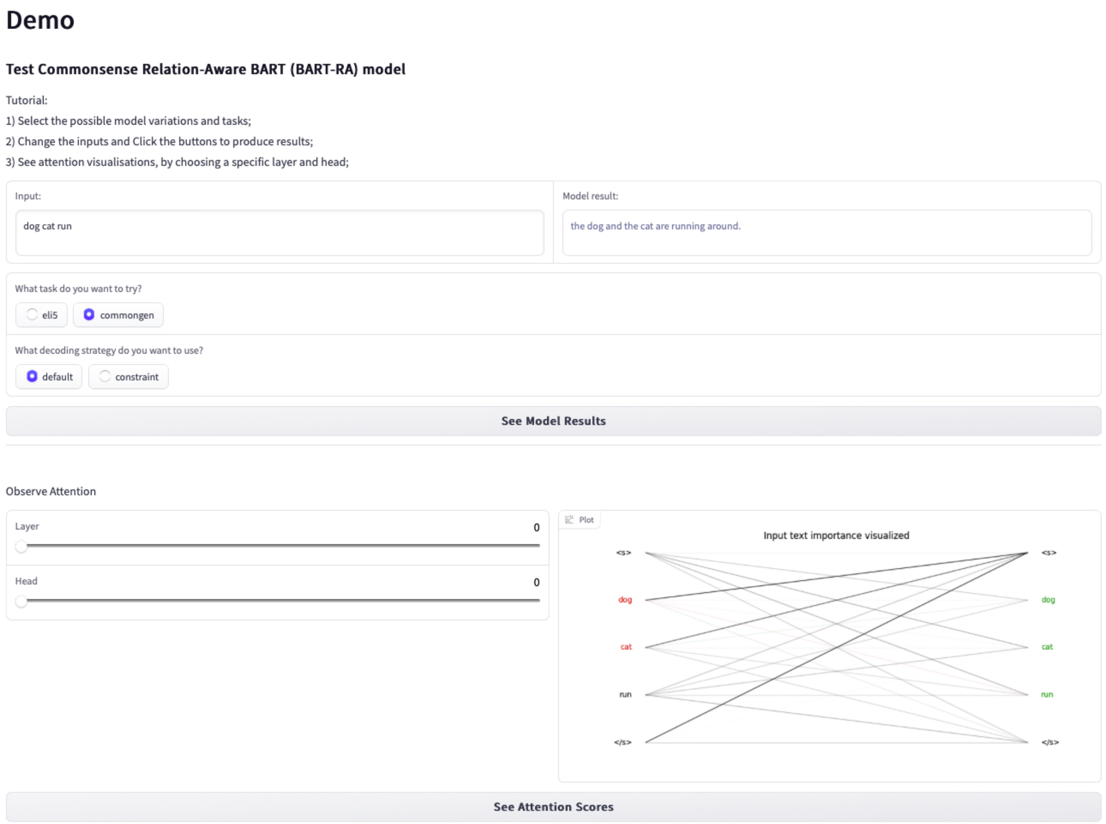
Model	BLEU	ROUGE	METEOR	CIDEr	SPICE	Coverage	R-W
BART (Normal)	21.95%	45.38%	45.65%	17.97%	49.95%	73.21%	70.84%
KG-BART	19.15%	43.90%	43.03%	17.37%	47.06%	**77.02%**	**97.92%**
RA-BART	**22.85%**	**46.36%**	**46.32%**	**18.36%**	**50.13%**	73.30%	70.13%
BART-RAM	22.63%	45.602%	45.87%	18.29%	49.95%	75.101%	75.47%

Discussion: Interestingly, this masking approach of integrating Commonsense Knowledge seems to capture more the ability to use concepts, and concepts rich in relations. One can see, that it stumbles subtly in the generated sentences according to the standard NLP metrics, in comparison with RA-BART. However, Relation-Aware Masked BART appears to offer a fair balance between concepts/relations usage and generation quality. We could also argue whether using a different methodology to choose the importance of heads, could benefit the generation.

4.3 Take on me: The readers empirical evaluation

In this section, we showcase a web demo interface[2] for the reader to try out some of the models proposed in this thesis, as well as to explore introspection tools for visualising the models. On Fig 4.5, an overview of the demo can be foreseen. The demo has two main sections, one related to interactively test the actual textual generations of our models. One can select the type of model, several tasks and decoding strategies and see the respective results. The other section intends to showcase what importance is the model giving to each input data unit, over the Encoder components. One can select the respective heads from all available Encoder layers and see the importance given to each language unit. Commonsense knowledge concepts/ideas are mapped with colours, for easier visualisation, where the green colour relates to a departing relationship, and the red colour a receiving one.

[2] https://huggingface.co/spaces/MrVicente/RA-BART



Demo

Test Commonsense Relation-Aware BART (BART-RA) model

Tutorial:
1) Select the possible model variations and tasks;
2) Change the inputs and Click the buttons to produce results;
3) See attention visualisations, by choosing a specific layer and head;

Input: dog cat run

Model result: the dog and the cat are running around.

What task do you want to try?
○ eli5 ● commongen

What decoding strategy do you want to use?
● default ○ constraint

Observe Attention

Layer 0

Head 0

Figure 4.5: Thesis Demo snapshot.

5 Real-Life Impact

This chapter shall address the impact of the RA-BART model application in real-life, either being it's carbon footprint, or the model's memory consumption.

5.1 Human Impact - Carbon Footprint

There is work [7, 101], urging for more consideration over Deep Learning models carbon emissions, mentioning green training/research strategies and even evaluation on current tools for predicting carbon emissions. Therefore, there is a scientifically and humanistic responsibility to not only work on the more mainstream ethical aspects of bias and fairness but also on environmental ethics such as keeping track of the carbon footprint of training and inference over Deep Learning models. One solid python framework is CodeCarbon [93], which we used to keep track of the electrical energy consumed and consequently, a prediction for the carbon emissions produced during the course of this dissertation. We believe that being transparent on ethical computations is essential to a more sustainable world. Next we present an underestimate of the carbon foot print resulted from this dissertation.

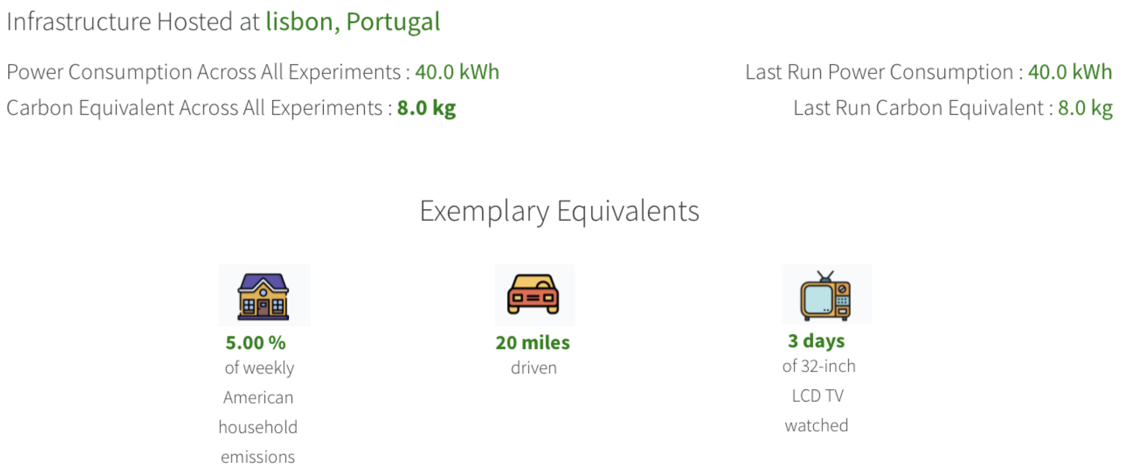
Experiments were conducted using a local cluster server in Portugal (NOVA FCT/SST). An underestimated approximation cumulative of ∼240 hours of computation was performed specifically on the following hardware:

- GPU: A100-SXM4-40GB (NVIDIA) with a TDP of 400W
- CPU: EPYC 7532 32-Core Processor (AMD) with a TDP of 200W

Total emissions are estimated to be of around 8 kgCO_2eq.

These estimations were performed using the project: CodeCarbon emissions tracker [93], a joint effort from authors of [48] and [65].

Discussion: To put this environmental impact in due perspective, take a look at Fig 5.1. As we can see, the impact is somewhat reasonable, and worth having in consideration for future work iterations and for having a granular intuition on working with “heavy machinery”. We consider these estimations to be underestimations, because we consider only the computations of working versions, not the debugging runs which can be quite considerable. Furthermore, we did not account for all training procedures, as there was runs which were not tracked. It is also worth meditating about the impact that Transformer models can have on inference machines. Using techniques, such as quantisation, using standards such as **ONNX** to decouple the essence of the models and optimise them for inference purposes and specific machines, can speed up these models and reduce carbon footprint.



Infrastructure Hosted at lisbon, Portugal

Power Consumption Across All Experiments : 40.0 kWh

Carbon Equivalent Across All Experiments : **8.0 kg**

Last Run Power Consumption : 40.0 kWh

Last Run Carbon Equivalent : 8.0 kg

Exemplary Equivalents

5.00 % of weekly American household emissions

20 miles driven

3 days of 32-inch LCD TV watched

Figure 5.1: Environmental impact of this dissertation computation compared to intuitive life situations.

5.2 Memory Overhead

Working with Transformers Language Models is normally associated with a memory overhead, due to the amount of parameters these models have. When training and performing inference on such models, there is also the issue of sequence length overhead. On RA-BART, we also have to consider that we are introducing external explicit knowledge, along with the standard sequence data, which accounts for a bigger memory footprint. Computationally, with Relation Self-Attention, the space complexity increases from $O(batchSize{*}numHeads{*}SeqLength^2{*}headsDimension)$ to $O(batchSize{*}numHeads{*}SeqLength^2{*}headsDimension{+}batchSize{*}SeqLength^2{*}relationsDimension)$. This increase in space complexity ends up having an impact on the attention mechanisms memory consumption’s, using around ~8% more memory, forcing (on some cases) the batches to be smaller during training and even inference, prolonging the training time. For the metrics gains and for production purposes this could be a killing trade-off.

5.3 Experiments Infrastructure: Transparency

Experiment tracking tools: The experiments were tracked through a third-party framework, named Weights and Biases (wandb [11]). We chose to mention this, because of the importance of such aiding tools. It increases reproducibility, experiment transparency and descriptive logging, which fastens experiments, helps finding bugs early on, reducing computing time and therefore models' carbon footprint as well.

6 Conclusion

Language Models are on a highway avenue of becoming larger and larger (in terms of parameters). Open AI [41] has pointed out that augmenting data proportionally to parameters amount increases Language Model's performance. A moon race in achieving better results by scaling models, without meditating on the usefulness of the parameters is a current trend. This is not always the case, where works, like Amazon's AlexaTM [98] using a solid fraction of parameters have outperformed SOTA Large Language Models, however still in the magnitude of billions of parameters. Also there is work concerned with ethical models growth, such as Bloom [1]. We are, as well, concerned about how we can better guide the learning procedure of these models, taking further advantage of their parameters. This dissertation, proposes ideas to model fragments of Commonsense Knowledge, following the hypothesis that by indirectly traversing a Knowledge Graph we are guiding Language Models towards meaningful Text Generation. Next, we present the two most important insights taken from this dissertation.

Insights:

1. Explicit Commonsense Knowledge injection can help models on some tasks (Abstractive QA and CommonGen) and harm on others (CommonsenseQA). The performance improvement on the CommonGen and Abstractive QA tasks are promising, however, subtle. **Knowledge Noise** is a known issue when introducing external knowledge, which can be distracting to a model learning a task. We believe that on the CommonsenseQA task the model was a victim of this issue. From our experiments, we also reckon that increasing model size in terms of parameters can help further obtain more gains in knowledge injection. The subtle results we had makes us wonder that maybe, injecting external Commonsense Knowledge on the Encoder of generative models might not be the optimal way to provide Commonsense Knowledge.

[1] https://bigscience.huggingface.co/blog/bloom

2. Human Level Assessment & Performance is extremely relevant to really grasp a human take on the models' quality and establish reasonable upper-bound baselines, especially in fields and tasks, where automatic metrics fail to capture subtle characteristics such as Commonsense.

Takeways: What is my hope for the reader who took their time reading this dissertation?

First of all, Commonsense is a subjective field and extremely hard to encode in Machine Learning Models. Computationally, Commonsense can be segregated in Knowledge and Reasoning. Commonsense Knowledge is one obvious case of Explicit Knowledge, which can be integrated during the training and inference of models. There are several ways to undergo this integration and this dissertation showcases some possible avenues. Commonsense Reasoning, in contrast, is a completely scientific jigsaw, which we leave for future work.

In summary, we integrated Commonsense Knowledge on a the Encoder of seq2seq model (BART) and further tested techniques which could aid this integration (loss manipulation, neighbour knowledge injection, knowledge aid decoding strategies). Even though, theoretically, this approach is promising, the results we obtained are somewhat substantial, which suggest that the Commonsense signals might not be strong enough to systematically aid the training procedures. Nevertheless, this dissertation should strike as an urgent reminder that explicit knowledge should not be ignored. It is fair that we recur to Deep Learning for their implicit understandings in subjective fields. After all, Deep Neural Networks are utterly powerful in capturing abstract understandings. Introducing deep models has forever changed multiple real world domains and fields, however, production ready models go beyond percentage gains on well behaved datasets. In reality and in most cases there is the need for complex stacked models, which also incite the need for the providence of a explainability framework. Furthermore, robust decision making, custom domains integration, and more problems are utterly relevant during learning procedures and are not purely solved without the aid of engineered Knowledge Models. Mixing external knowledge, such as Commonsense Knowledge, and studying better efficient ways to integrate explicit knowledge might explain better decisions/results and result in models more aware of specific domains, which in turn result in better downstream utilisation.

Finally, falling back to our work, we believe that our work adds up to the pile of relevant work regarding Models' Commonsense acquisition. The use of structured explicit Commonsense Knowledge (either in the form of Knowledge Graphs or ontologies, etc) can be a step onto better extrapolating information not blatantly present in texts.

6.1 Future Work

As discussed in the computational commonsense work from David [19], Commonsense Knowledge can be described as the human's intrinsic knowledge regarding the world which might fail to have a textual identity and therefore end up sometimes not being captured by Language Models.

Moreover, when pursuing Commonsense we strive for much more than the mere acquisition of its knowledge, we want the ability to reason over it. This is hard, really capturing Commonsense is having the ability to fuse humans' intrinsic knowledge, be adaptive to a changing world state and make meaningful, logical decisions over them. Therefore, to tackle the Commonsense Reasoning problem more systematically we envision several takes:

Take 1 (Sentence Generation Guidance with ASER): Currently, there is work (ASER [128]) in documenting, at a large magnitude, fragments of speech conveying events, world states and their relations. Using a Semantic Parsing Tree on a given input and tagging similar documented fragments of speech can be used to search on the ASER graph and provide more realistic Text Generation, especially regarding abstract chatbot interactions or storytelling. This can be seen as an interesting approximation to reasoning over a set of rules and axioms, as seen in Symbolic AI.

Take 2 (Neuro-Symbolic BART): Also, as mentioned in Chapter 2, there is really promising work on SubSymbolic AI. Creating approximations to Logical Reasoners and combining them with Language Models might be inevitable. Large Language Models are great at creating implicit relations between concepts and using them for some task, but making use of explicit structured knowledge to guide them is for sure work to be pursued (See 6.1 for an example).

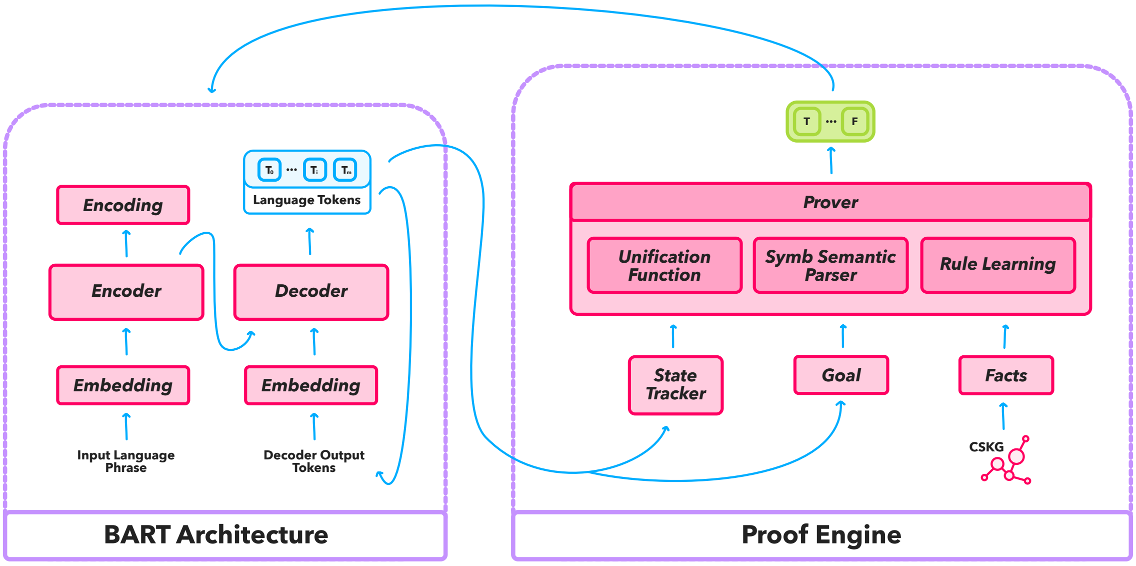


Figure 6.1: Neuro-Symbolic BART

Take 3 (Knowledge Embeddings): More related to the work pursued on this thesis, one could use Entity embeddings / Graph embeddings both in the Encoder and Decoder architecture and study whether these introduction of knowledge also helps guiding the model. K-Bert [62], Luke [122], among others, used this approach, for example to introduce domain knowledge more fruitfully, which in our case would be Commonsense domain knowledge. One caveat, which K-Bert also mentions is the problem of knowledge noise (KN) issue. Using too much knowledge, may be distracting to the model especially when combining embeddings which are the product of two non-related encoding mappings. However, one could also argue the latter problem would be non-existent as it is plausible to assume that a model could learn to map this two different mappings into a meaningful one.

Take 4 (Neuroscience): Computationally, we are stuck in a ill learning-procedure matrix. No surprise, the community knows we are far from intelligently teaching models "inteligence", we just don't know it any better. Do we really need to teach "inteligence"? In practice, current methodologies already give outstanding and quite seamlessly unbeatable results for some use cases. Therefore, it depends on the task at hand and our expectations of a certain deep neural model. For Artificial General Intelligence, we need reasoning. We have mentioned some artificial avenues to approximate this reasoning ability, however one could argue that the very best mimicking ability will prevail from decoding our own humanistic biological neural capabilities.

6.2 The End

Like Steve Jobs once brought us 1000 songs to our pockets, we are at the dawn of having 1000 deep neural models in our pockets. Yet, far from the Greek Chaos, I am the first to say that we are no gods, but if you ask me: that is an undoubtedly specimen of god-like superpowers. An open highway to such power requires responsible accountability. In the end, not every obstacle has to be tackled with a Goliath's solution. Provided that one is not conveying under a critical system, a 1% increase on some metric, should generally not be a strong motive to shift to using a more computational exhaustive model. As mentioned and as has been studied, Deep Learning inference end-points will account for a fair amount of CO_2 emissions in the future. Therefore, it starts with every one of us to strive for innovation, break boundaries, but always be accountable for our actions. To end on a last note, whatever you build: do it for kindness, be kind.

Bibliography

[1] J. Alammar. "Ecco: An Open Source Library for the Explainability of Transformer Language Models". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, 2021 (cit. on p. 11).

[2] P. Anderson et al. "SPICE: Semantic Propositional Image Caption Evaluation". In: *Computer Vision – ECCV 2016*. Ed. by B. Leibe et al. Cham: Springer International Publishing, 2016, pp. 382–398. isbn: 978-3-319-46454-1 (cit. on p. 28).

[3] F. Arabshahi et al. "Conversational Neuro-Symbolic Commonsense Reasoning". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.6 (May 2021), pp. 4902–4911. url: https://ojs.aaai.org/index.php/AAAI/article/view/16623 (cit. on p. 24).

[4] S. Auer et al. "DBpedia: A Nucleus for a Web of Open Data". In: *The Semantic Web*. Ed. by K. Aberer et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 722–735. isbn: 978-3-540-76298-0 (cit. on p. 22).

[5] S. Bach et al. "On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation". In: *PLOS ONE* 10.7 (July 2015), pp. 1–46. doi: 10.1371/journal.pone.0130140. url: https://doi.org/10.1371/journal.pone.0130140 (cit. on p. 11).

[6] P. Bajaj et al. *MS MARCO: A Human Generated MAchine Reading COmprehension Dataset*. 2018. arXiv: 1611.09268 [cs.CL] (cit. on p. 20).

[7] N. Bannour et al. "Evaluating the carbon footprint of NLP methods: a survey and analysis of existing tools". In: *Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing*. Virtual: Association for Computational Linguistics, Nov. 2021, pp. 11–21. doi: 10.18653/v1/2021.sustainlp-1.2. url: https://aclanthology.org/2021.sustainlp-1.2 (cit. on p. 67).

[8] Y. Bengio et al. "A Neural Probabilistic Language Model". In: *J. Mach. Learn. Res.* 3.null (Mar. 2003), pp. 1137–1155. issn: 1532-4435 (cit. on p. 15).

[9] D. E. Berlyne. "Curiosity and Exploration". In: *Science* 153.3731 (1966), pp. 25–33. DOI: 10.1126/science.153.3731.25. eprint: https://www.science.org/doi/pdf/10.1126/science.153.3731.25. URL: https://www.science.org/doi/abs/10.1126/science.153.3731.25 (cit. on p. 90).

[10] T. Berners-Lee, J. Hendler, and O. Lassila. "The Semantic Web: A New Form of Web Content That is Meaningful to Computers Will Unleash a Revolution of New Possibilities". In: *ScientificAmerican.com* (May 2001) (cit. on p. 22).

[11] L. Biewald. *Experiment Tracking with Weights and Biases*. Software available from wandb.com. 2020. URL: https://www.wandb.com/ (cit. on p. 69).

[12] S. Bird, E. Klein, and E. Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. "O'Reilly Media, Inc.", 2009 (cit. on p. 36).

[13] A. Bosselut et al. *COMET: Commonsense Transformers for Automatic Knowledge Graph Construction*. 2019. arXiv: 1906.05317 [cs.CL] (cit. on pp. 23, 25).

[14] T. B. Brown et al. *Language Models are Few-Shot Learners*. 2020. arXiv: 2005.14165 [cs.CL] (cit. on pp. 9, 11).

[15] R. Campos et al. "YAKE! Keyword extraction from single documents using multiple local features". In: *Information Sciences* 509 (2020), pp. 257–289. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2019.09.013. URL: https://www.sciencedirect.com/science/article/pii/S0020025519308588 (cit. on p. 36).

[16] D. Chen et al. "Reading Wikipedia to Answer Open-Domain Questions". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 1870–1879. DOI: 10.18653/v1/P17-1171. URL: https://aclanthology.org/P17-1171 (cit. on p. 22).

[17] K. Cho et al. "On the Properties of Neural Machine Translation: Encoder–Decoder Approaches". In: *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 103–111. DOI: 10.3115/v1/W14-4012. URL: https://aclanthology.org/W14-4012 (cit. on p. 6).

[18] H. P. Cowley et al. "A framework for rigorous evaluation of human performance in human and machine learning comparison studies". In: *Scientific Reports* 12.1 (Mar. 2022), p. 5444. ISSN: 2045-2322. DOI: 10.1038/s41598-022-08078-3. URL: https://doi.org/10.1038/s41598-022-08078-3 (cit. on p. 28).

[19] E. Davis and G. Marcus. "Commonsense reasoning and commonsense knowledge in artificial intelligence". In: *Communications of the ACM* 58.9 (2015), pp. 92–103 (cit. on pp. 22, 71).

[20] S. De Deyne et al. “The “Small World of Words” English word association norms for over 12,000 cue words”. In: *Behavior Research Methods* 51.3 (June 2019), pp. 987–1006. ISSN: 1554-3528. DOI: 10.3758/s13428-018-1115-7. URL: https://doi.org/10.3758/s13428-018-1115-7 (cit. on p. 34).

[21] J. Deng et al. “ImageNet: A large-scale hierarchical image database”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848 (cit. on pp. 10, 23).

[22] J. Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL] (cit. on pp. 8–10, 12, 16, 24).

[23] A. Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. arXiv: 2010.11929 [cs.CV] (cit. on p. 9).

[24] A. Fan, M. Lewis, and Y. Dauphin. “Hierarchical Neural Story Generation”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 889–898. DOI: 10.18653/v1/P18-1082. URL: https://aclanthology.org/P18-1082 (cit. on p. 13).

[25] A. Fan et al. “ELI5: Long Form Question Answering”. In: *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*. Ed. by A. Korhonen, D. R. Traum, and L. Màrquez. Association for Computational Linguistics, 2019, pp. 3558–3567. DOI: 10.18653/v1/p19-1346. URL: https://doi.org/10.18653/v1/p19-1346 (cit. on pp. 21, 27, 31, 96).

[26] A. Fan et al. “ELI5: Long Form Question Answering”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 3558–3567. DOI: 10.18653/v1/P19-1346. URL: https://aclanthology.org/P19-1346 (cit. on p. 22).

[27] P. Gage. “A new algorithm for data compression”. In: *The C Users Journal archive* 12 (1994), pp. 23–38 (cit. on p. 17).

[28] A. d’Avila Garcez et al. *Neural-Symbolic Learning and Reasoning: Contributions and Challenges*. 2015. URL: https://www.aaai.org/ocs/index.php/SSS/SSS15/paper/view/10281 (cit. on p. 1).

[29] D. George, M. Lázaro-Gredilla, and J. S. Guntupalli. “From CAPTCHA to Commonsense: How Brain Can Teach Us About Artificial Intelligence”. In: *Frontiers in Computational Neuroscience* 14 (2020). ISSN: 1662-5188. DOI: 10.3389/fncom.2020.554097. URL: https://www.frontiersin.org/article/10.3389/fncom.2020.554097 (cit. on pp. 22, 25).

[30] A. Goyal et al. “Professor Forcing: A New Algorithm for Training Recurrent Networks”. In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. NIPS’16. Barcelona, Spain: Curran Associates Inc., 2016, pp. 4608–4616. ISBN: 9781510838819 (cit. on p. 12).

[31] M. Grootendorst. *KeyBERT: Minimal keyword extraction with BERT.* Version v0.3.0. 2020. DOI: 10.5281/zenodo.4461265. URL: https://doi.org/10.5281/zenodo.4461265 (cit. on p. 36).

[32] K. He et al. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90 (cit. on p. 10).

[33] S. Hochreiter and J. Schmidhuber. “Long Short-Term Memory”. In: *Neural Comput.* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: https://doi.org/10.1162/neco.1997.9.8.1735 (cit. on p. 6).

[34] A. Holtzman et al. *The Curious Case of Neural Text Degeneration*. 2019. DOI: 10.48550/ARXIV.1904.09751. URL: https://arxiv.org/abs/1904.09751 (cit. on p. 13).

[35] J. Howard and S. Ruder. *Universal Language Model Fine-tuning for Text Classification*. 2018. arXiv: 1801.06146 [cs.CL] (cit. on p. 7).

[36] L. Huang et al. “Cosmos QA: Machine Reading Comprehension with Contextual Commonsense Reasoning”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 2391–2401. DOI: 10.18653/v1/D19-1243. URL: https://aclanthology.org/D19-1243 (cit. on p. 27).

[37] F. Ilievski, P. Szekely, and B. Zhang. “CSKG: The CommonSense Knowledge Graph”. In: *Extended Semantic Web Conference (ESWC)* (2021) (cit. on p. 23).

[38] E. Ilkou and M. Koutraki. “Symbolic Vs Sub-symbolic AI Methods: Friends or Enemies?” In: *CIKM*. 2020 (cit. on pp. 19, 20).

[39] D. Jacquette. *Symbolic Logic*. Wadsworth Publishing Company, 2001 (cit. on p. 19).

[40] M. I. Jordan. “Serial order: a parallel distributed processing approach. Technical report, June 1985-March 1986”. In: (May 1986). URL: https://www.osti.gov/biblio/6910294 (cit. on p. 6).

[41] J. Kaplan et al. *Scaling Laws for Neural Language Models*. 2020. DOI: 10.48550/ARXIV.2001.08361. URL: https://arxiv.org/abs/2001.08361 (cit. on pp. 64, 70).

[42] A. Kembhavi et al. "Are You Smarter Than a Sixth Grader? Textbook Question Answering for Multimodal Machine Comprehension". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 5376–5384 (cit. on p. 21).

[43] C. Kidd and B. Y. Hayden. "The Psychology and Neuroscience of Curiosity". In: *Neuron* 88.3 (Nov. 2015), pp. 449–460 (cit. on p. 90).

[44] J. Konrád et al. *Alquist 4.0: Towards Social Intelligence Using Generative Models and Dialogue Personalization*. 2021. arXiv: 2109.07968 [cs.CL] (cit. on p. 89).

[45] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf (cit. on p. 10).

[46] T. Kudo. "Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates". In: *ACL*. 2018 (cit. on p. 17).

[47] T. Kudo and J. Richardson. "SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing". In: *EMNLP*. 2018 (cit. on p. 17).

[48] A. Lacoste et al. "Quantifying the Carbon Emissions of Machine Learning". In: *Workshop on Tackling Climate Change with Machine Learning at NeurIPS 2019* (2019) (cit. on p. 67).

[49] Y. LeCun, Y. Bengio, and G. Hinton. "Deep learning". In: *nature* 521.7553 (2015), p. 436 (cit. on p. 2).

[50] D. Lenat and R. V. Guha. "CYC: A Midterm Report". In: *AI Magazine* 11.3 (Sept. 1990), p. 32. DOI: 10.1609/aimag.v11i3.842. URL: https://ojs.aaai.org/index.php/aimagazine/article/view/842 (cit. on pp. vii, ix, 23).

[51] M. Lewis et al. *BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension*. 2019. arXiv: 1910.13461 [cs.CL] (cit. on pp. vii, ix, 2, 8–11, 40, 96).

[52] P. Lewis et al. "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 9459–9474. URL: https://proceedings.neurips.cc/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf (cit. on pp. 21, 60).

[53] B. Z. Li, M. Nye, and J. Andreas. *Implicit Representations of Meaning in Neural Language Models*. 2021. arXiv: 2106.00737 [cs.CL] (cit. on pp. 2, 10, 20, 29).

[54] X. Li, L. Herranz, and S. Jiang. “Multifaceted Analysis of Fine-Tuning in a Deep Model for Visual Recognition”. In: *ACM/IMS Trans. Data Sci.* 1.1 (Mar. 2020). ISSN: 2691-1922. DOI: 10.1145/3319500. URL: https://doi.org/10.1145/3319500 (cit. on p. 10).

[55] Z. Li et al. “Guided Generation of Cause and Effect”. In: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*. Ed. by C. Bessiere. Main track. International Joint Conferences on Artificial Intelligence Organization, July 2020, pp. 3629–3636. DOI: 10.24963/ijcai.2020/502. URL: https://doi.org/10.24963/ijcai.2020/502 (cit. on p. 14).

[56] B. Y. Lin et al. “CommonGen: A Constrained Text Generation Challenge for Generative Commonsense Reasoning”. In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov. 2020, pp. 1823–1840. URL: https://www.aclweb.org/anthology/2020.findings-emnlp.165 (cit. on pp. 27, 30, 32, 54).

[57] C.-Y. Lin. “ROUGE: A Package for Automatic Evaluation of Summaries”. In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 74–81. URL: https://www.aclweb.org/anthology/W04-1013 (cit. on p. 28).

[58] G. W. Lindsay. “Attention in Psychology, Neuroscience, and Machine Learning”. In: *Frontiers in Computational Neuroscience* 14 (2020). ISSN: 1662-5188. DOI: 10.3389/fncom.2020.00029. URL: https://www.frontiersin.org/article/10.3389/fncom.2020.00029 (cit. on p. 7).

[59] N. U. Lisbon. “TWIZ: A conversational Task Wizard with multimodal curiosity-exploration”. In: *Alexa Prize TaskBot Challenge Proceedings*. 2022. URL: https://www.amazon.science/alexa-prize/proceedings/twiz-a-conversational-task-wizard-with-multimodal-curiosity-exploration (cit. on pp. 3, 54, 60).

[60] C. Liu, T. Cohn, and L. Frermann. “Commonsense Knowledge in Word Associations and ConceptNet”. In: *Proceedings of the 25th Conference on Computational Natural Language Learning*. Online: Association for Computational Linguistics, Nov. 2021, pp. 481–495. DOI: 10.18653/v1/2021.conll-1.38. URL: https://aclanthology.org/2021.conll-1.38 (cit. on pp. 22, 34, 47).

[61] P. Liu et al. “Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing”. In: *ArXiv* abs/2107.13586 (2021) (cit. on pp. 13, 40).

[62] W. Liu et al. “K-BERT: Enabling Language Representation with Knowledge Graph”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.03 (Apr. 2020), pp. 2901–2908. DOI: 10.1609/aaai.v34i03.5681. URL: https://ojs.aaai.org/index.php/AAAI/article/view/5681 (cit. on pp. 25, 73).

[63] Y. Liu et al. *KG-BART: Knowledge Graph-Augmented BART for Generative Commonsense Reasoning*. 2021. arXiv: 2009.12677 [cs.CL] (cit. on pp. vii, ix, 3, 9, 25).

[64] Y. Liu et al. *Multilingual Denoising Pre-training for Neural Machine Translation*. 2020. arXiv: 2001.08210 [cs.CL] (cit. on p. 12).

[65] K. Lottick et al. "Energy Usage Reports: Environmental awareness as part of algorithmic accountability". In: *Workshop on Tackling Climate Change with Machine Learning at NeurIPS 2019* (2019) (cit. on p. 67).

[66] J. M. Lourenço. *The NOVAthesis LaTeX Template User's Manual*. NOVA University Lisbon. 2021. URL: https://github.com/joaomlourenco/novathesis/raw/master/template.pdf (cit. on p. v).

[67] C. Malaviya et al. "Commonsense Knowledge Base Completion with Structural and Semantic Context". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.03 (Apr. 2020), pp. 2925–2933. DOI: 10.1609/aaai.v34i03.5684. URL: https://ojs.aaai.org/index.php/AAAI/article/view/5684 (cit. on p. 23).

[68] J. Mao et al. "The Neuro-Symbolic Concept Learner: Interpreting Scenes, Words, and Sentences From Natural Supervision". In: *International Conference on Learning Representations*. 2019. URL: https://openreview.net/forum?id=rJgMlhRctm (cit. on p. 20).

[69] T. Mikolov et al. *Efficient Estimation of Word Representations in Vector Space*. 2013. arXiv: 1301.3781 [cs.CL] (cit. on p. 15).

[70] G. A. Miller. "WordNet: A Lexical Database for English". In: *Commun. ACM* 38.11 (Nov. 1995), pp. 39–41. ISSN: 0001-0782. DOI: 10.1145/219717.219748. URL: https://doi.org/10.1145/219717.219748 (cit. on p. 23).

[71] F. Moghimifar et al. "Neural-Symbolic Commonsense Reasoner with Relation Predictors". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 797–802. DOI: 10.18653/v1/2021.acl-short.100. URL: https://aclanthology.org/2021.acl-short.100 (cit. on p. 24).

[72] F. Moghimifar et al. *Neural-Symbolic Commonsense Reasoner with Relation Predictors*. 2021. arXiv: 2105.06717 [cs.AI] (cit. on p. 24).

[73] I. Montani et al. *explosion/spaCy: v3.4.1: Fix compatibility with CuPy v9.x*. Version v3.4.1. July 2022. DOI: 10.5281/zenodo.6907665. URL: https://doi.org/10.5281/zenodo.6907665 (cit. on p. 36).

[74] M. A. Musen et al. "The National Center for Biomedical Ontology". en. In: *J Am Med Inform Assoc* 19.2 (Nov. 2011), pp. 190–195 (cit. on p. 22).

[75] A. Newell and H. Simon. "The logic theory machine–A complex information processing system". In: *IRE Transactions on Information Theory* 2.3 (1956), pp. 61–79. DOI: 10.1109/TIT.1956.1056797 (cit. on p. 1).

[76] K. Papineni et al. "BLEU: a Method for Automatic Evaluation of Machine Translation". In: 2002, pp. 311–318 (cit. on p. 28).

[77] J. S. Park et al. *VisualCOMET: Reasoning about the Dynamic Context of a Still Image*. 2020. arXiv: 2004.10796 [cs.CV] (cit. on pp. 22, 23, 25).

[78] R. Pascanu, T. Mikolov, and Y. Bengio. "On the Difficulty of Training Recurrent Neural Networks". In: *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*. ICML'13. Atlanta, GA, USA: JMLR.org, 2013, III–1310–III–1318 (cit. on p. 6).

[79] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830 (cit. on p. 28).

[80] J. Pennington, R. Socher, and C. Manning. "GloVe: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. DOI: 10.3115/v1/D14-1162. URL: https://aclanthology.org/D14-1162 (cit. on p. 15).

[81] K. Pillutla et al. "MAUVE: Measuring the Gap Between Neural Text and Human Text using Divergence Frontiers". In: *NeurIPS*. 2021 (cit. on p. 28).

[82] A. Radford et al. "Language Models are Unsupervised Multitask Learners". In: (2019) (cit. on p. 12).

[83] C. Raffel et al. *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. 2020. arXiv: 1910.10683 [cs.LG] (cit. on p. 11).

[84] P. Rajpurkar, R. Jia, and P. Liang. *Know What You Don't Know: Unanswerable Questions for SQuAD*. 2018. arXiv: 1806.03822 [cs.CL] (cit. on p. 21).

[85] N. Reimers and I. Gurevych. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Nov. 2019. URL: https://arxiv.org/abs/1908.10084 (cit. on p. 94).

[86] T. Rocktäschel and S. Riedel. "End-to-end Differentiable Proving". In: *NIPS*. 2017 (cit. on p. 20).

[87] P. Rodriguez et al. "Information Seeking in the Spirit of Learning: A Dataset for Conversational Curiosity". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2020). DOI: 10.18653/v1/2020.emnlp-main.655. URL: http://dx.doi.org/10.18653/v1/2020.emnlp-main.655 (cit. on p. 91).

[88] S. Rose et al. "Automatic Keyword Extraction from Individual Documents". In: *Text Mining*. John Wiley & Sons, Ltd, 2010. Chap. 1, pp. 1–20. ISBN: 9780470689646. DOI: https://doi.org/10.1002/9780470689646.ch1. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470689646.ch1. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470689646.ch1 (cit. on p. 36).

[89] F. Rosenblatt. *The perceptron - A perceiving and recognizing automaton*. Tech. rep. 85-460-1. Ithaca, New York: Cornell Aeronautical Laboratory, Jan. 1957 (cit. on p. 1).

[90] V. Sanh et al. *Multitask Prompted Training Enables Zero-Shot Task Generalization*. 2021. arXiv: 2110.08207 [cs.LG] (cit. on pp. vii, ix).

[91] M. Sap et al. *ATOMIC: An Atlas of Machine Commonsense for If-Then Reasoning*. 2019. arXiv: 1811.00146 [cs.CL] (cit. on p. 22).

[92] M. Sap et al. "Social IQa: Commonsense Reasoning about Social Interactions". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 4463–4473. DOI: 10.18653/v1/D19-1454. URL: https://aclanthology.org/D19-1454 (cit. on p. 27).

[93] V. Schmidt et al. "CodeCarbon: Estimate and Track Carbon Emissions from Machine Learning Computing". In: (2021). DOI: 10.5281/zenodo.4658424 (cit. on p. 67).

[94] M. Schuster and K. Nakajima. "Japanese and Korean voice search". In: *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2012), pp. 5149–5152 (cit. on p. 17).

[95] P. Shaw, J. Uszkoreit, and A. Vaswani. "Self-Attention with Relative Position Representations". In: *NAACL*. 2018 (cit. on pp. 9, 41).

[96] K. Shim et al. "Layer-wise Pruning of Transformer Attention Heads for Efficient Language Modeling". In: *2021 18th International SoC Design Conference (ISOCC)*. 2021, pp. 357–358. DOI: 10.1109/ISOCC53507.2021.9613933 (cit. on p. 11).

[97] H. Shindo, D. S. Dhami, and K. Kersting. *Neuro-Symbolic Forward Reasoning*. 2021. arXiv: 2110.09383 [cs.AI] (cit. on p. 20).

[98] S. Soltan et al. *AlexaTM 20B: Few-Shot Learning Using a Large-Scale Multilingual Seq2Seq Model*. 2022. DOI: 10.48550/ARXIV.2208.01448. URL: https://arxiv.org/abs/2208.01448 (cit. on p. 70).

[99] R. Speer, J. Chin, and C. Havasi. *ConceptNet 5.5: An Open Multilingual Graph of General Knowledge*. 2018. arXiv: 1612.03975 [cs.CL] (cit. on pp. 22, 26, 34).

[100] L. Sterling and E. Shapiro. *The Art of Prolog (2nd Ed.): Advanced Programming Techniques*. Cambridge, MA, USA: MIT Press, 1994. ISBN: 0262193388 (cit. on p. 19).

[101] E. Strubell, A. Ganesh, and A. McCallum. “Energy and Policy Considerations for Deep Learning in NLP”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 3645–3650. DOI: 10.18653/v1/P19-1355. URL: https://aclanthology.org/P19-1355 (cit. on p. 67).

[102] Y. Sun et al. “ERNIE 3.0: Large-scale Knowledge Enhanced Pre-training for Language Understanding and Generation”. In: *ArXiv* abs/2107.02137 (2021) (cit. on pp. 9, 22).

[103] O. Sychev. “Combining neural networks and symbolic inference in a hybrid cognitive architecture”. In: *Procedia Computer Science* 190 (2021). 2020 Annual International Conference on Brain-Inspired Cognitive Architectures for Artificial Intelligence: Eleventh Annual Meeting of the BICA Society, pp. 728–734. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2021.06.085. URL: https://www.sciencedirect.com/science/article/pii/S1877050921013405 (cit. on p. 24).

[104] A. Talmor et al. “CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4149–4158. DOI: 10.18653/v1/N19-1421. URL: https://aclanthology.org/N19-1421 (cit. on pp. 21, 27, 32, 54).

[105] H. Tan and M. Bansal. *LXMERT: Learning Cross-Modality Encoder Representations from Transformers*. 2019. arXiv: 1908.07490 [cs.CL] (cit. on p. 9).

[106] I. Tenney et al. *The Language Interpretability Tool: Extensible, Interactive Visualizations and Analysis for NLP Models*. 2020. URL: https://www.aclweb.org/anthology/2020.emnlp-demos.15 (cit. on p. 11).

[107] N. W. Varuna Jayasiri. *labml.ai Annotated Paper Implementations*. 2020. URL: https://nn.labml.ai/ (cit. on p. 10).

[108] A. Vaswani et al. *Attention Is All You Need*. 2017. arXiv: 1706.03762 [cs.CL] (cit. on pp. 7, 11).

[109] R. Vedantam, C. L. Zitnick, and D. Parikh. “CIDEr: Consensus-based image description evaluation”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 4566–4575. DOI: 10.1109/CVPR.2015.7299087 (cit. on p. 28).

[110] J. Vig. "A Multiscale Visualization of Attention in the Transformer Model". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 37–42. DOI: 10.18653/v1/P19-3007. URL: https://www.aclweb.org/anthology/P19-3007 (cit. on p. 11).

[111] E. Voita et al. "Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 5797–5808. DOI: 10.18653/v1/P19-1580. URL: https://aclanthology.org/P19-1580 (cit. on pp. 11, 49).

[112] D. Vrandečić and M. Krötzsch. "Wikidata: A Free Collaborative Knowledgebase". In: *Commun. ACM* 57.10 (Sept. 2014), pp. 78–85. ISSN: 0001-0782. DOI: 10.1145/2629489. URL: https://doi.org/10.1145/2629489 (cit. on p. 22).

[113] B. Wang et al. "RAT-SQL: Relation-Aware Schema Encoding and Linking for Text-to-SQL Parsers". In: *ACL*. 2020 (cit. on pp. 9, 18, 41).

[114] D. Wang et al. "Multi-head Self-attention with Role-Guided Masks". In: *Advances in Information Retrieval*. Ed. by D. Hiemstra et al. Cham: Springer International Publishing, 2021, pp. 432–439. ISBN: 978-3-030-72240-1 (cit. on p. 49).

[115] L. Weber et al. *NLProlog: Reasoning with Weak Unification for Question Answering in Natural Language*. 2019. arXiv: 1906.06187 [cs.CL] (cit. on pp. 20, 24).

[116] T. Wolf et al. "Transformers: State-of-the-Art Natural Language Processing". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. URL: https://www.aclweb.org/anthology/2020.emnlp-demos.6 (cit. on p. 26).

[117] D. H. Wolpert and W. G. Macready. "No free lunch theorems for optimization". In: *IEEE transactions on evolutionary computation* 1.1 (1997), pp. 67–82 (cit. on p. 22).

[118] S. writer. *Amazon launches new Alexa Prize TaskBot Challenge*. July 2021. URL: https://www.amazon.science/academic-engagements/amazon-launches-new-alexa-prize-taskbot-challenge (cit. on p. 3).

[119] Q. Wu, C. Miao, and Z. Shen. "A curious learning companion in Virtual Learning Environment". In: June 2012, pp. 1–8. ISBN: 978-1-4673-1507-4. DOI: 10.1109/FUZZ-IEEE.2012.6251362 (cit. on p. 90).

[120] Y. Xing et al. *KM-BART: Knowledge Enhanced Multimodal BART for Visual Commonsense Generation*. 2021. arXiv: 2101.00419 [cs.CL] (cit. on p. 25).

[121] Y. Xu et al. "Human Parity on CommonsenseQA: Augmenting Self-Attention with External Attention". In: *ArXiv* abs/2112.03254 (2021) (cit. on p. 25).

[122] I. Yamada et al. "LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 6442–6454. DOI: 10.18653/v1/2020.emnlp-main.523. URL: https://aclanthology.org/2020.emnlp-main.523 (cit. on p. 73).

[123] P. Yang et al. "Knowledgeable Storyteller: A Commonsense-Driven Generative Model for Visual Storytelling". In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, July 2019, pp. 5356–5362. DOI: 10.24963/ijcai.2019/744. URL: https://doi.org/10.24963/ijcai.2019/744 (cit. on p. 55).

[124] Y. Zeldes et al. *Technical Report: Auxiliary Tuning and its Application to Conditional Text Generation*. 2020. DOI: 10.48550/ARXIV.2006.16823. URL: https://arxiv.org/abs/2006.16823 (cit. on p. 13).

[125] R. Zellers et al. "From Recognition to Cognition: Visual Commonsense Reasoning". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019 (cit. on p. 25).

[126] A. Zhang et al. "Dive into Deep Learning". In: *arXiv preprint arXiv:2106.11342* (2021) (cit. on p. 10).

[127] H. Zhang et al. *A Survey of Controllable Text Generation using Transformer-based Pre-trained Language Models*. 2022. DOI: 10.48550/ARXIV.2201.05337. URL: https://arxiv.org/abs/2201.05337 (cit. on pp. 13, 14).

[128] H. Zhang et al. "ASER: Towards large-scale commonsense knowledge acquisition via higher-order selectional preference over eventualities". In: *Artificial Intelligence* 309 (2022), p. 103740. ISSN: 0004-3702. DOI: https://doi.org/10.1016/j.artint.2022.103740. URL: https://www.sciencedirect.com/science/article/pii/S0004370222000807 (cit. on p. 72).

[129] S. Zhang et al. *ReCoRD: Bridging the Gap between Human and Machine Commonsense Reading Comprehension*. 2018. arXiv: 1810.12885 [cs.CL] (cit. on p. 27).

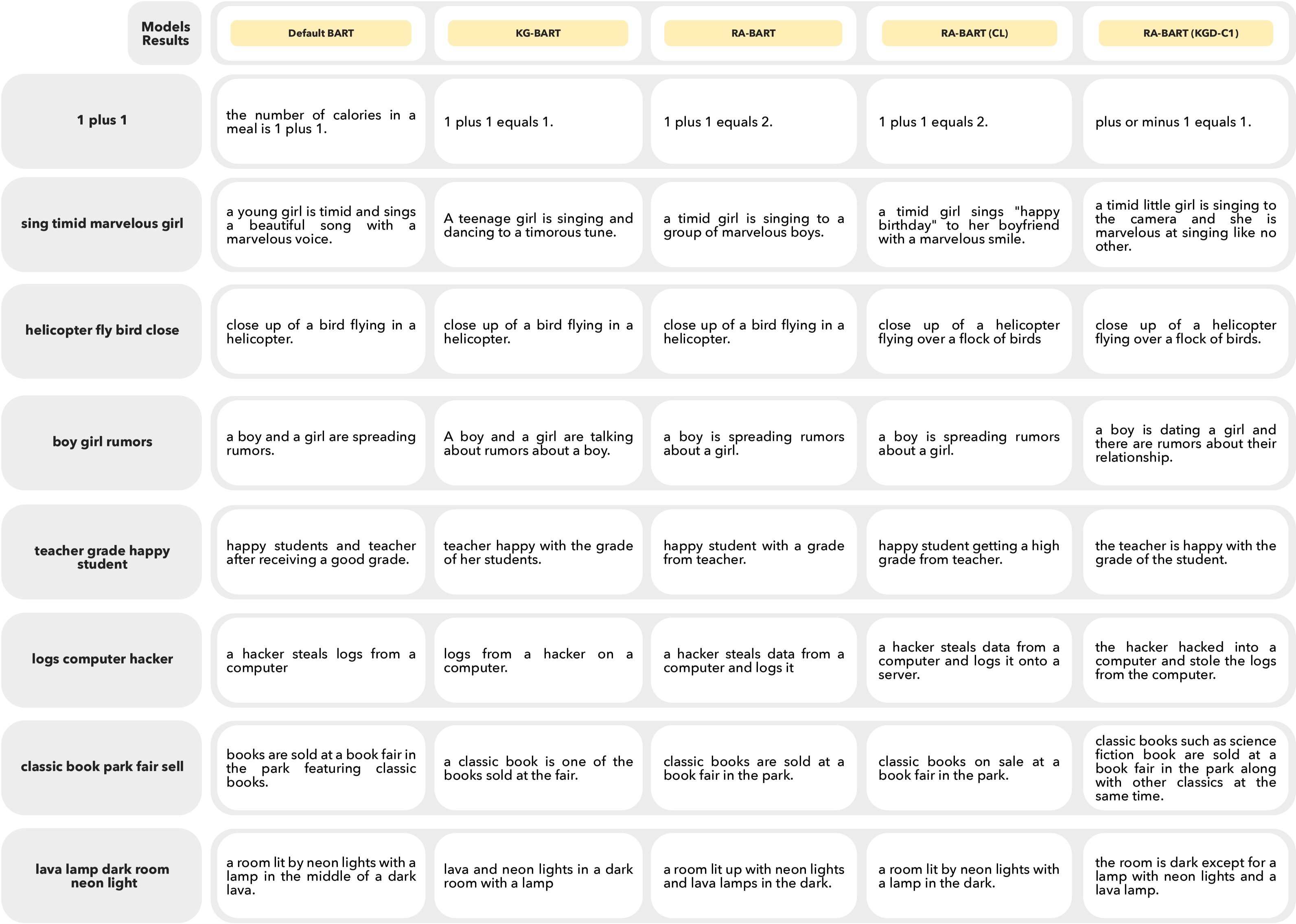
[130] D. M. Ziegler et al. *Fine-Tuning Language Models from Human Preferences*. 2019. DOI: 10.48550/ARXIV.1909.08593. URL: https://arxiv.org/abs/1909.08593 (cit. on p. 13).

A

Further Models Generations

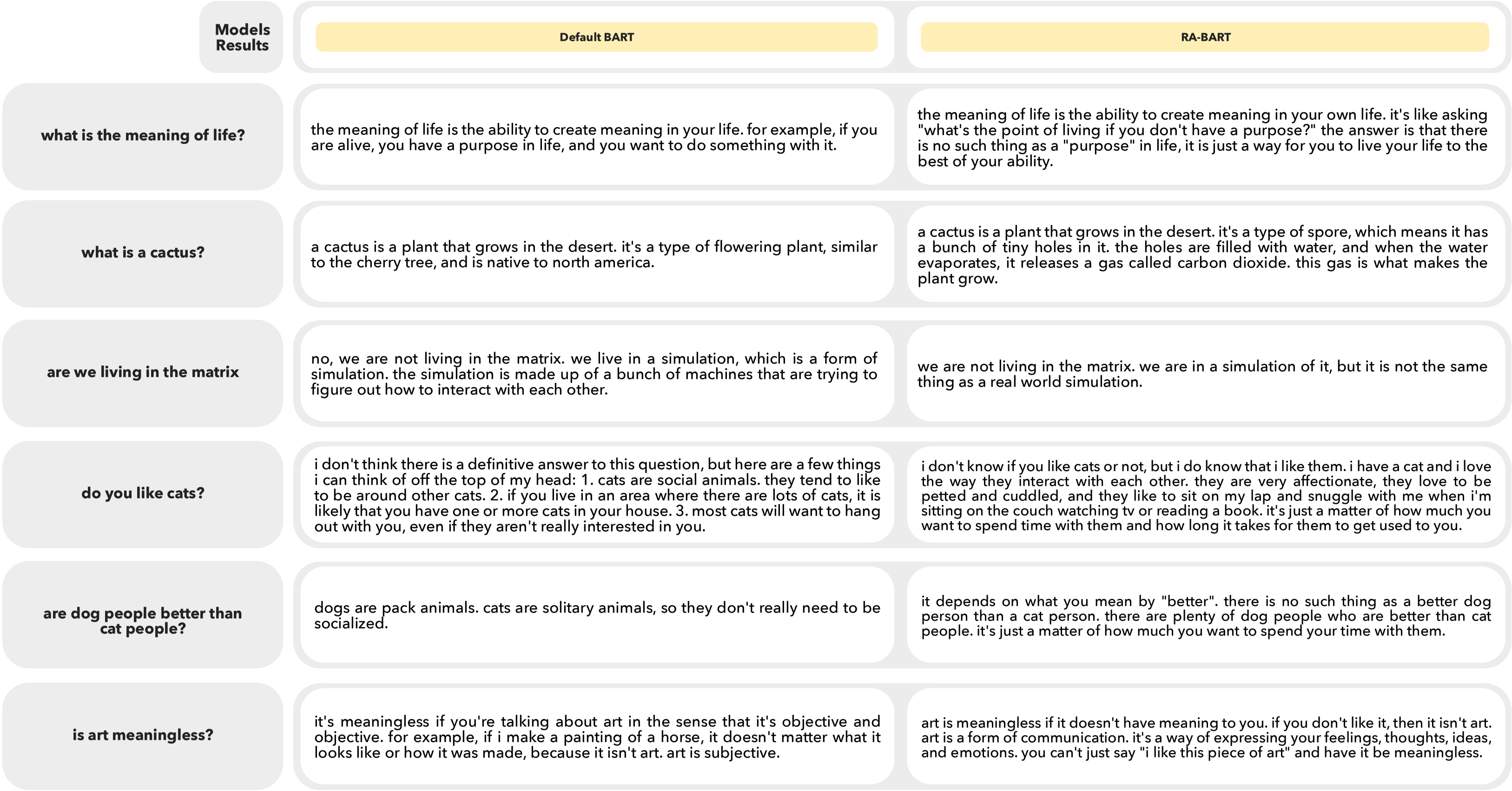
This appendix provides multiple examples showcasing differences in Text Generation, regarding comparative models both on the CommmonGen and Abstractive QA task. In Fig A.1 one can further observe interesting results in our models, such as being able to “answer” a mathematical question (1 + 1 equals 2), when the others totally miss it. It can be the case that intrinsically, the external Commonsense Knowledge guided this understanding.

In Fig A.2, we further observe how adding Commonsense Knowledge helps being more assertive and providing more knowledgeable answers in the Abstractive QA task.



Models Results	Default BART	KG-BART	RA-BART	RA-BART (CL)	RA-BART (KGD-C1)
1 plus 1	the number of calories in a meal is 1 plus 1.	1 plus 1 equals 1.	1 plus 1 equals 2.	1 plus 1 equals 2.	plus or minus 1 equals 1.
sing timid marvelous girl	a young girl is timid and sings a beautiful song with a marvelous voice.	A teenage girl is singing and dancing to a timorous tune.	a timid girl is singing to a group of marvelous boys.	a timid girl sings "happy birthday" to her boyfriend with a marvelous smile.	a timid little girl is singing to the camera and she is marvelous at singing like no other.
helicopter fly bird close	close up of a bird flying in a helicopter.	close up of a bird flying in a helicopter.	close up of a bird flying in a helicopter.	close up of a helicopter flying over a flock of birds	close up of a helicopter flying over a flock of birds.
boy girl rumors	a boy and a girl are spreading rumors.	A boy and a girl are talking about rumors about a boy.	a boy is spreading rumors about a girl.	a boy is spreading rumors about a girl.	a boy is dating a girl and there are rumors about their relationship.
teacher grade happy student	happy students and teacher after receiving a good grade.	teacher happy with the grade of her students.	happy student with a grade from teacher.	happy student getting a high grade from teacher.	the teacher is happy with the grade of the student.
logs computer hacker	a hacker steals logs from a computer	logs from a hacker on a computer.	a hacker steals data from a computer and logs it	a hacker steals data from a computer and logs it onto a server.	the hacker hacked into a computer and stole the logs from the computer.
classic book park fair sell	books are sold at a book fair in the park featuring classic books.	a classic book is one of the books sold at the fair.	classic books are sold at a book fair in the park.	classic books on sale at a book fair in the park.	classic books such as science fiction book are sold at a book fair in the park along with other classics at the same time.
lava lamp dark room neon light	a room lit by neon lights with a lamp in the middle of a dark lava.	lava and neon lights in a dark room with a lamp	a room lit up with neon lights and lava lamps in the dark.	a room lit by neon lights with a lamp in the dark.	the room is dark except for a lamp with neon lights and a lava lamp.

Figure A.1: Models text generation - CommonGen Task



Models Results	Default BART	RA-BART
what is the meaning of life?	the meaning of life is the ability to create meaning in your life. for example, if you are alive, you have a purpose in life, and you want to do something with it.	the meaning of life is the ability to create meaning in your own life. it's like asking "what's the point of living if you don't have a purpose?" the answer is that there is no such thing as a "purpose" in life, it is just a way for you to live your life to the best of your ability.
what is a cactus?	a cactus is a plant that grows in the desert. it's a type of flowering plant, similar to the cherry tree, and is native to north america.	a cactus is a plant that grows in the desert. it's a type of spore, which means it has a bunch of tiny holes in it. the holes are filled with water, and when the water evaporates, it releases a gas called carbon dioxide. this gas is what makes the plant grow.
are we living in the matrix	no, we are not living in the matrix. we live in a simulation, which is a form of simulation. the simulation is made up of a bunch of machines that are trying to figure out how to interact with each other.	we are not living in the matrix. we are in a simulation of it, but it is not the same thing as a real world simulation.
do you like cats?	i don't think there is a definitive answer to this question, but here are a few things i can think of off the top of my head: 1. cats are social animals. they tend to like to be around other cats. 2. if you live in an area where there are lots of cats, it is likely that you have one or more cats in your house. 3. most cats will want to hang out with you, even if they aren't really interested in you.	i don't know if you like cats or not, but i do know that i like them. i have a cat and i love the way they interact with each other. they are very affectionate, they love to be petted and cuddled, and they like to sit on my lap and snuggle with me when i'm sitting on the couch watching tv or reading a book. it's just a matter of how much you want to spend time with them and how long it takes for them to get used to you.
are dog people better than cat people?	dogs are pack animals. cats are solitary animals, so they don't really need to be socialized.	it depends on what you mean by "better". there is no such thing as a better dog person than a cat person. there are plenty of dog people who are better than cat people. it's just a matter of how much you want to spend your time with them.
is art meaningless?	it's meaningless if you're talking about art in the sense that it's objective and objective. for example, if i make a painting of a horse, it doesn't matter what it looks like or how it was made, because it isn't art. art is subjective.	art is meaningless if it doesn't have meaning to you. if you don't like it, then it isn't art. art is a form of communication. it's a way of expressing your feelings, thoughts, ideas, and emotions. you can't just say "i like this piece of art" and have it be meaningless.

Figure A.2: Generative Question Answering - Abstractive QA Task

Curiosity
Dataset

I Curiosity

• Did you know that: Chopsticks were initially created for cooking, not as an eating utensil? •

I.1 Proposal

We propose to model a **Curiosity Dataset**, a dataset containing pairs of fun facts and keywords related to these facts covering the domain of cuisine and DIY. The **Curiosity Dataset** can be consulted in the following open source GitHub repository: Curiosity-Dataset

I.2 Motivation

Keeping a chatbot user engaged when pursuing a continuous 1-to-1 conversation is a complex task. User's psychological factors aligned with the chatbot system efficiency and correctness in responding will determine how well an user will keep engaged in the system. The search for the aid of an assistant chatbot is normally rather objective, meaning that when an user has a goal in mind, they wish to see it fulfilled in the most efficient amount of conversation turns. Therefore, any attempt to fruitfully extend a conversation flow, must be taken with care. Even though user satisfaction is relevant in terms of providing an interesting flow of conversation, ultimately, correctness must be ensured.

In an attempt to further improve user's satisfaction/engagement when interacting with a virtual assistant, such as the Alexa virtual assistant, we propose the introduction of curiosity phrases closely contextualised with a certain flow of a conversation. As seen in the work of Alquist 4.0 [44], trivia facts if rightly used, do have a positive impact on a virtual assistance conversation. However, an important note is that a curiosity follows the formality of a statement, which naturally discourages an user to further engage it when compared with a question. We take this rational into account in our work.

I.3 What is a curiosity?

The concept of curiosity has for decades been debated by neuroscientists and psychologists. As mentioned in [43], some have separated it in two research views: 1. Curiosity as a natural impulse for seeking extended cognition; 2. A phenomena related to exploring, playing, learning, the desire for information;

Berlyne in 1966 [9], went even further meditating about how humans had inherently a special type of curiosity: an **epistemic curiosity**. Meaning that, above the exploration and information seeking need, humans also strove for knowledge, being relevant to our use case since the curiosities we are building are epistemological related.

Inspired by Berlyne work, an attempt to computationally model curiosity was presented taking in consideration the key concepts which define it, positively or negatively [119]. We adapt their computational model to better resemble textual epistemological curiosity, which is the focus of our work (see Fig I.1).

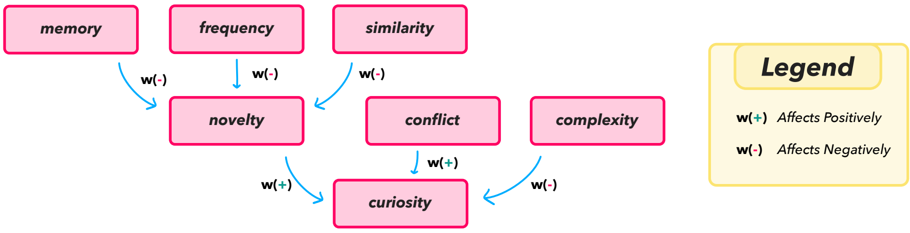


Figure I.1: Curiosity Computational Model

Curiosity, but more importantly curiosity phrases, are deeply rooted to measurable intrinsic features. Complexity hardens the comprehension of a phrase, which can have a negative impact. Conflict and novelty play positively with, respectively, how much new information is gained from a curiosity and the freshness quality of the information. Negatively bounding novelty, and therefore curiosities, lie in the memory , frequency and similarity characteristics of a curiosity. The more we hear about a concept and similar ones might, for instance, turn a curiosity which uses such terms more dull. Considering the computational model mentioned in I.3, we go further in providing an objectified definition for building up a curiosity, formulating curiosity, in our work, as being the following: **Curiosity** is a Natural Language phrase which ignites a reader to seek for more information, while providing an extension to reader's knowledge rich in athirst data.

With a well thought distillation over the curiosity meaning and having removed some subtle subjectivity, we depart to develop a dataset which strives to capture the true essence of our formulation of **Curiosity**.

Since the domain of our work relates to recipes and home improvement tasks, the curiosity dataset will focus on this domain.

I.4 Curiosity Dataset

Having pursued a survey on curiosity datasets, we found no curiosity dataset which could fulfil the characteristics in need for both Alexa's chatbot and the enclosed domain where our work was focused on (home improvement tasks and recipes). In 2020 [87], a work accompanied by a curiosity dataset of geographical facts for improvement of dialog agents is the most similar work and data we were able to find. However as stated, the domain of the curiosities present in this work fail to match ours. Therefore, there were two possible paths to follow: either we had to crawl and parse curiosities from websites such as wikipedia or we could manually search for curiosities and curate a custom dataset. We chose the second approach, since there were some aspects that were very relevant to us:

- There was a need for curiosities, **not** facts;
- Curiosity phrases length matters significantly;
- Quality of the phrases meant more than the quantity of them;
- Simplicity in the sentences was of great concern. Dense reasoning behind curiosities could have a negative impact on user engagement.

I.4.1 Special Days

Special days are titles attributed to days of the year for honouring something or someone. These days can be conveyed as curiosities which are rather sympathetic to local users, where days' titles are attributed. Alexa is a live system which is used daily, thus on any special day being able to greet the user with a personalised curiosity regarding a special day makes a companionable incentive to continue using the system.

Since special days are somewhat curated on websites such as, thereisadayforthat, we were able to automatically generate curiosities for **cuisine** special days, using the template: In the United States, on the ⟨DATE⟩ , it's the ⟨EVENT⟩ .

For this we built a web scrapper to retrieve them, by making a request to the website for each month of the year to the food category present in the website Application Programming Interface (API). After fetching the website HTML raw data, we parsed it, retrieving the special days dates and days names. Then, we built custom curiosity sentences from the data collected.

Unfortunately, since **DIY** related special days are less existent we followed a manual, succinct retrieval for them.

• **Do you believe that:** In the United States, on the 23rd of March, it's the national chip and dip day? •

I.4.2 Organisation

The dataset has been split in two main categories, since our problem domain consisted in recipes and wikihow articles. For an in depth description of each domain, we present a brief list of general classes covered in each of them:

Recipe task coverage: 1. **Fruit** (eg: Avocado, Vitamin C fruits, etc); 2. **Meat** (eg: cow, etc); 3. **Seafood** (eg: shrimp, etc); 4. **Food tools** (eg: oven, etc); 5. **Cuisine related concepts** (eg: temperature, etc); 6. **Popular countries' food** (eg: pizza (Italian), sushi (Japanese) etc); 7. **United States** (eg: U.S. National food days, etc)

Wikihow task coverage: 1. **American DIY statistics**; 2. **DIY tools** (eg: hammer, etc); 3. **United States** (eg: U.S. National days related to DIY, etc) 4. **House furniture** (eg: sofa, etc) 5. **Yard objects and tools** (eg: lawn moaner, etc) 6. **Garage objects** (eg: car, etc)

Dataset organisation: An instance of the curiosity dataset follows the structure:

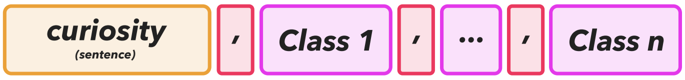
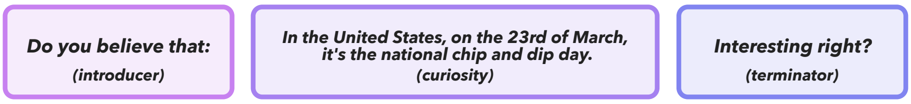


Figure I.2: Curiosity instance structure

The focus of this curated dataset was more on gathering and creating curiosities following pre-defined characteristics, therefore the annotation was vaguely added just for having in mind possible usage of this curiosity dataset in downstream tasks. Our major interest was not so much on the annotation aspect, as we had envisioned an automated annotation for the curiosities, so the labelling and the articles-curiosities matching task would not be costly in time.

Adding to the careful creation of the curiosities dataset, we also worked on the end-user experience, thus appending to the dataset: introductory and terminating phrases for the curiosities. To introduce a curiosity within an user-Alexa conversation flow we used an **"introducer"** randomly sampled from a pre-defined curated list. For terminating a curiosity insertion, similarly, we introduced a **"terminator"** phrase after the curiosity sentence, sampled over a list of possible ending questions, whose objective is to represent a rhetorical question. (See Fig I.3 for an example of a curiosity which could appear within an user-Alexa conversation flow)



Do you believe that:
(introducer)

In the United States, on the 23rd of March, it's the national chip and dip day.
(curiosity)

Interesting right?
(terminator)

Figure I.3: End-User view of a recipe's special day Curiosity

I.4.3 Statistics

I.4.3.1 Distribution of curiosities: Curiosity Type

The dataset consists of 1350 curiosities distributed through both the cuisine and DIY categories (See Fig I.1 for a realisation of the distribution of curiosities per Curiosity Type).

Table I.1: Curiosity Dataset statistics

Curiosity Types	**Amount**
Recipe / Cuisine	743
Home Improvement / DIY	607
Total	1350

I.4.3.2 Distribution of curiosities: Length per Curiosity Type

Throughout the curiosities dataset creation we deposed careful attention into conforming the curiosities length distribution to an average of 15 words, avoiding big sentences so as to maximise readers comprehension (See Fig I.4 for an overview of the length distribution (in words) of the curiosities dataset).

Apart from users curiosity comprehension, there was also a relevant characteristic about our users and our virtual assistant, which we had to be careful about. Curiosities were not the reason users were using our virtual assistant, therefore we had to balance the intrusion of curiosities and the normal carriage of tasks users were executing.

I.4.4 Recipes & Wikihow: curiosity matching

The curiosities we engineered had the goal to be introduced within a virtual assistant chatting conversational flow, when executing a recipe or Wikihow (DIY) task. Having this in consideration, if we wanted to deliver a contextualised curiosity similar to a corresponding internet resource, we had to carefully extract the articles/documents content into a categorised and contextualised mapping space for them to reside in a equivalent searchable space.

To reach the mentioned goal, we first had to pre-process the data (the articles) we wanted to match the curiosity phrases with. Recipes and wikihow articles in their raw form are just JSON files scrapped from their API websites. Therefore, to process such resources we had to analyse and extract the most relevant content of the corresponding

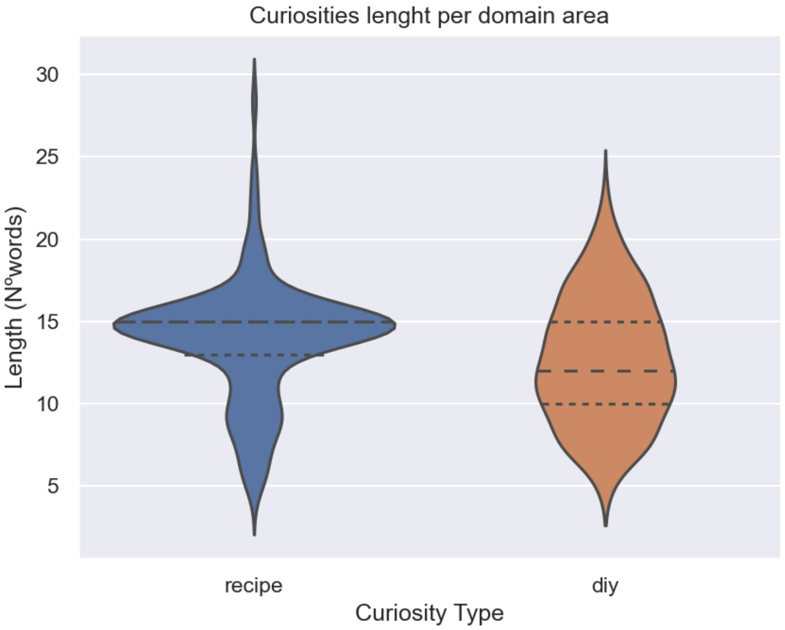


Figure I.4: Curiosities length distribution

JSON files, which better depicted them. Upon some consideration and grounding experiments, we decided that for the recipes articles, we would extract the title text, the steps text and the ingredients text, whereas for the Wikihow data we would extract only the title and steps. The rational behind this was rooted to two main aspects: 1. There was a need to capture a set of curiosities with a fine-grain detail of each specific part of an article. 2. The other information present in the articles were mostly noisy data, hindering a positive similarity search.

We pursued the methodology of trying to match sets of curiosities to the recipes or Wikihow contexts using the **Semantic Similarity Search** methodology. This is not a simple and direct task since to perform such task we have to have a computational tool able to map a certain article contexts to an n-dimensional space where we can further query the space on the curiosities equivalent n-dimensional space, finding which context, in distance, more closely matches our set of curiosities. See Fig I.5 for an abstraction view of what it means to try to match a certain article context to the space of curiosities.

To deal with the **Semantic Similarity Search** problem, we resorted to a sentence transformer[85], which consists of a Sentence-BERT, a modified version of BERT, specially created to deal with comparative semantic understanding. Our task perfectly matches the objective of Sentence-BERT, therefore we applied it to our use case.

For each recipe and Wikihow article we have matched them with curiosities, based on the following algorithm:

1. Segregate an article in several parts: the title, steps (and ingredients for recipes);

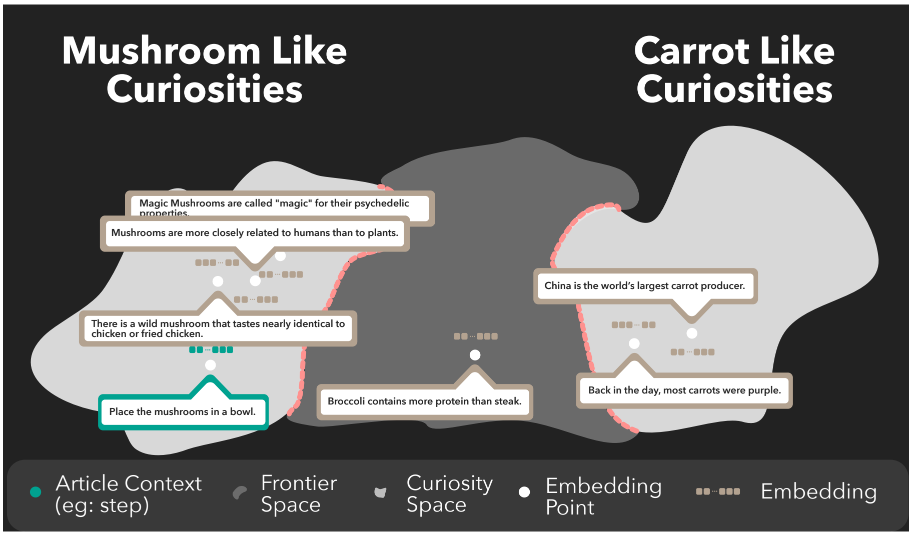

Mushroom Like Curiosities
Carrot Like Curiosities
Magic Mushrooms are called "magic" for their psychedelic properties.
Mushrooms are more closely related to humans than to plants.
There is a wild mushroom that tastes nearly identical to chicken or fried chicken.
Place the mushrooms in a bowl.
Broccoli contains more protein than steak.
China is the world's largest carrot producer.
Back in the day, most carrots were purple.
Article Context (eg: step)
Frontier Space
Curiosity Space
Embedding Point
Embedding

Figure I.5: Curiosity semantic search

2. Perform Semantic Search for each article context to the space of all curiosities available (recipes or wikihow articles), selecting the top-10 curiosities for each context;

3. To ensure we remove false-positive curiosities, and relate top-3 curiosities to each article context, for each context we:

   a) Run a Cross-Encoder (re-ranker) model through all combination pairs (curiosities list with the top-10 selected curiosities) and select the top-3 ones;

   b) Build a JSON object, storing the identifier of each article and the top-3 curiosities for each article context;

See Fig I.6 for an example of curiosities matching with both a title and recipe step context.

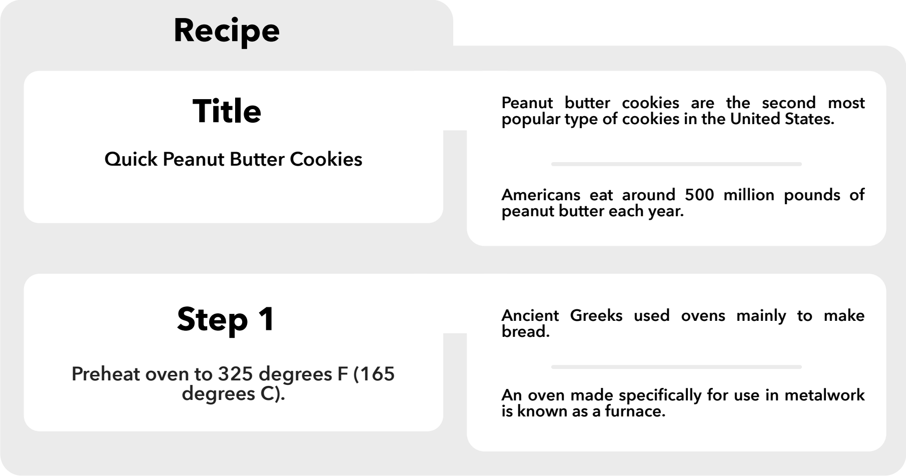
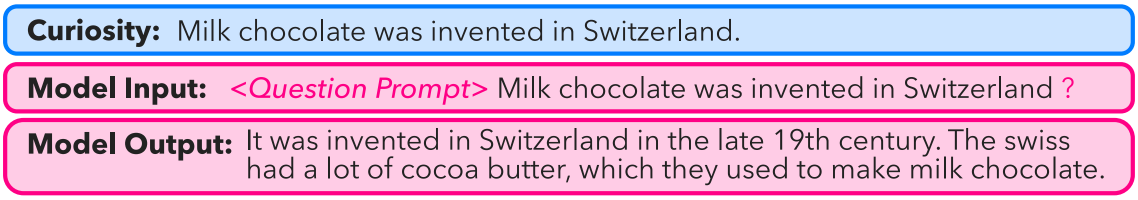

Recipe
Title
Quick Peanut Butter Cookies
Peanut butter cookies are the second most popular type of cookies in the United States.
Americans eat around 500 million pounds of peanut butter each year.
Step 1
Preheat oven to 325 degrees F (165 degrees C).
Ancient Greeks used ovens mainly to make bread.
An oven made specifically for use in metalwork is known as a furnace.

Figure I.6: Recipe article semantic match with curiosities

I.5 Generative Take on Curiosities

We launch ourselves on a mission to further charismatically personify a Virtual Assistant, which users could trust and be enlightened by a more natural interaction. However, combining both human-like interactions and well-defined step-by-step tasks naturally stands as a challenge when attempting more creative interaction flows and whilst ensuring a robust controlled flow. Having this in mind, we weighted both the hallucination risks and human-like answering possibilities and put together a curiosity clarifying and explaining generator BART [51] model. This model consists in a generative model with the ability of elaborating on curiosities (fun facts). For instance, prompting a question or an elaboration request (e.g. "why?", "why is that","tell me more about that") after a curiosity, our model will attempt to reason about it and provide a friendly response to the user (See I.7 for a curiosity clarification example). In summary, this model was fine-tuned on the Abstractive QA task, mentioned in section 3.1.1 and prompt engineered to work on the curiosities dataset.



Curiosity: Milk chocolate was invented in Switzerland.

Model Input: *<Question Prompt>* Milk chocolate was invented in Switzerland ?

Model Output: It was invented in Switzerland in the late 19th century. The swiss had a lot of cocoa butter, which they used to make milk chocolate.

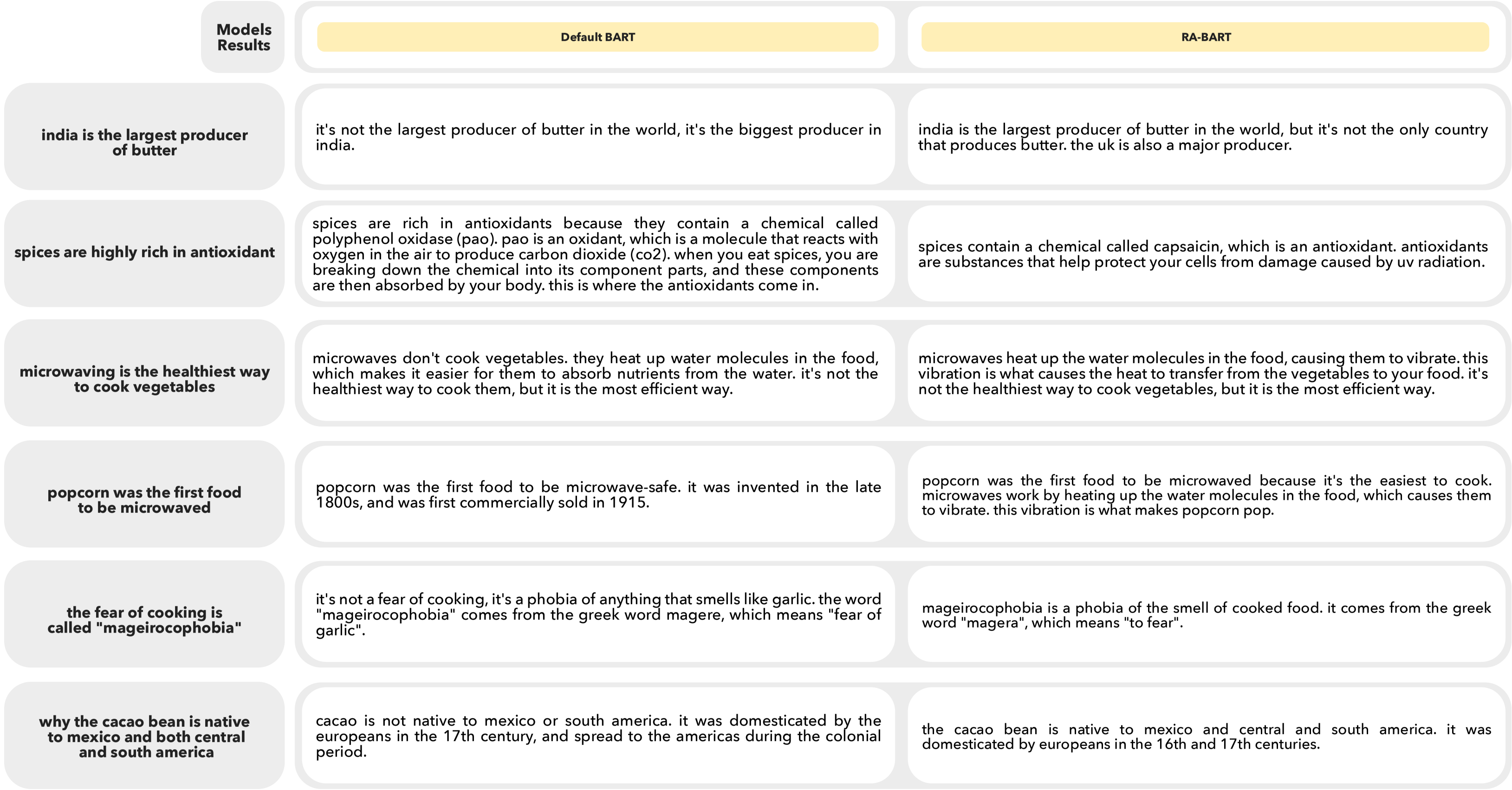
Figure I.7: BART Prompt Engineering - Fun Fact Domain

To fine-tune the BART model, we collected several different datasets and pre-processed them to remove non-linguistic textual artefacts (e.g. URLs and image references). The data collected was the following: the ELI5 and the AskScience subreddit dump [25], and additionally, we captured questions and answers from 5 forum communities (cooking, crafts, DIY, gardening, and pets) of the stack exchange forum[1]. This variety of QAs textual data permitted BART to gain insights over the custom domains of cuisine and general tasks.

Our fine-tuned model also had a surprising tendency to contradict itself, which we tried to abolish by detecting negation of statements.

On Fig I.8, one can observe further results of utilising the BART model and our RA-BART model to clarify fun facts and even attempt to explain them. Additionally, one can also see that RA-BART seems to formulate more reasonable answers than the default BART model.

[1] https://archive.org/download/stackexchange



Models Results	Default BART	RA-BART
india is the largest producer of butter	it's not the largest producer of butter in the world, it's the biggest producer in india.	india is the largest producer of butter in the world, but it's not the only country that produces butter. the uk is also a major producer.
spices are highly rich in antioxidant	spices are rich in antioxidants because they contain a chemical called polyphenol oxidase (pao). pao is an oxidant, which is a molecule that reacts with oxygen in the air to produce carbon dioxide (co2). when you eat spices, you are breaking down the chemical into its component parts, and these components are then absorbed by your body. this is where the antioxidants come in.	spices contain a chemical called capsaicin, which is an antioxidant. antioxidants are substances that help protect your cells from damage caused by uv radiation.
microwaving is the healthiest way to cook vegetables	microwaves don't cook vegetables. they heat up water molecules in the food, which makes it easier for them to absorb nutrients from the water. it's not the healthiest way to cook them, but it is the most efficient way.	microwaves heat up the water molecules in the food, causing them to vibrate. this vibration is what causes the heat to transfer from the vegetables to your food. it's not the healthiest way to cook vegetables, but it is the most efficient way.
popcorn was the first food to be microwaved	popcorn was the first food to be microwave-safe. it was invented in the late 1800s, and was first commercially sold in 1915.	popcorn was the first food to be microwaved because it's the easiest to cook. microwaves work by heating up the water molecules in the food, which causes them to vibrate. this vibration is what makes popcorn pop.
the fear of cooking is called "mageirocophobia"	it's not a fear of cooking, it's a phobia of anything that smells like garlic. the word "mageirocophobia" comes from the greek word magere, which means "fear of garlic".	mageirocophobia is a phobia of the smell of cooked food. it comes from the greek word "magera", which means "to fear".
why the cacao bean is native to mexico and both central and south america	cacao is not native to mexico or south america. it was domesticated by the europeans in the 17th century, and spread to the americas during the colonial period.	the cacao bean is native to mexico and central and south america. it was domesticated by europeans in the 16th and 17th centuries.

Figure I.8: Curiosities Clarification & Explanation - Curiosities dataset