INFO8010: Project proposal

Victor Mangeleer,[1] **Axelle Schyns**,[2] and **Lucie Navez**[3]

[1] *victor.mangeleer@student.uliege.be (s181670)*
[2] *axelle.schyns@student.uliege.be (s180598)*
[3] *lucie.navez@student.uliege.be (s180703)*

I. INTRODUCTION

For the past years, the feeling of loneliness has become more important in our society due to the lockdown induced by the COVID-19 pandemic. Thus, many of us have find comfort in reading books. However, this comfort was only short-lived as many of the books ending left us on cliff-hangers or maybe unsatisfied regarding the love story of the main characters.

Therefore, a solution to this problem would be to create a story telling bot (**SBOT** for short) that would allow us to extrapolate the rest of the story or even feed our imagination with an unlikely romance between the main and some background characters.

In fact, who has never wanted to know more about the forbidden love between Harry Potter and Dobby the house elf ? Or maybe explore the day to day life of Ron Weasley and Hermione Granger as a married couple ?

II. ARCHITECTURE

In the case of an AI writing its own story, it is very important for it to remember the inputs it has been given. Indeed, for a classical neural network, this kind of sequential dependencies are not remembered by it during the course of their utilization.

Therefore, this problem can be solved by using a special architecture which is called **R**ecurrent **N**eural **N**etwork (**RNN**). In fact, a recurrent neural network is a type of artificial neural network commonly used in speech recognition and natural language processing. Thus, in the situation of a story teller, it recognizes data's sequential characteristics and use patterns to predict the next likely scenario.

III. DATA SETS

In order to train our network, we can actually use all kinds of books, for instance. In our case, we decided to focus on the Harry Potter series, but this could be extended to any other books or texts available on the web. Thus, we found a convenient data set containing all the books of the Harry Potter series on the following GitHub. It is important to notice that these data sets have been made available only for academic purpose and each and one of the authors possess an original copy of the books. Therefore, the complete collection of the Harry Potter series is given by :

Title	**Words**
Harry Potter and the Philosopher's stone	76 944
Harry Potter and the Chamber of secrets	85 141
Harry Potter and the Azkaban's prisoner	107 253
Harry Potter and the Goblet of fire	190 637
Harry Potter and the Order of the Phoenix	257 045
Harry Potter and the Half Blood prince	168 923
Harry Potter and Deathly Hallows	198 227
Total	1 084 170

TABLE I. Collection of the Harry Potter series and the number of words inside each book.

Even if this amount of data is low by contrast to the well-known model GPT-2 which was trained on 8 million web pages, we do not expect to obtain results as convincing as those one but to have results that are probably quite funny.

Afterwards, if we want to generalize the model, we can actually use any kind of `.txt` books and hope that the model will write a story mixing multiple story lines. In addition to that, one could imagine a bot that would generate recipes and thus it would be trained on cooking books. As it is commonly said, sky is the limit.

IV. COMPUTING RESOURCES

Due to the massive amount of data that would require the model to be trained on, the computing power of the platform Gradient would be used.

V. ADDITIONAL FEATURES

There are multiple ways of improving a text generation model. In our case, seeing our main purpose, we could think of 2 possible ways of improving it:

- Once the SBOT is able to create the following of a story, we could make it capable of communicating with us. Indeed, it could personify one of the character and respond to the questions asked by the user.

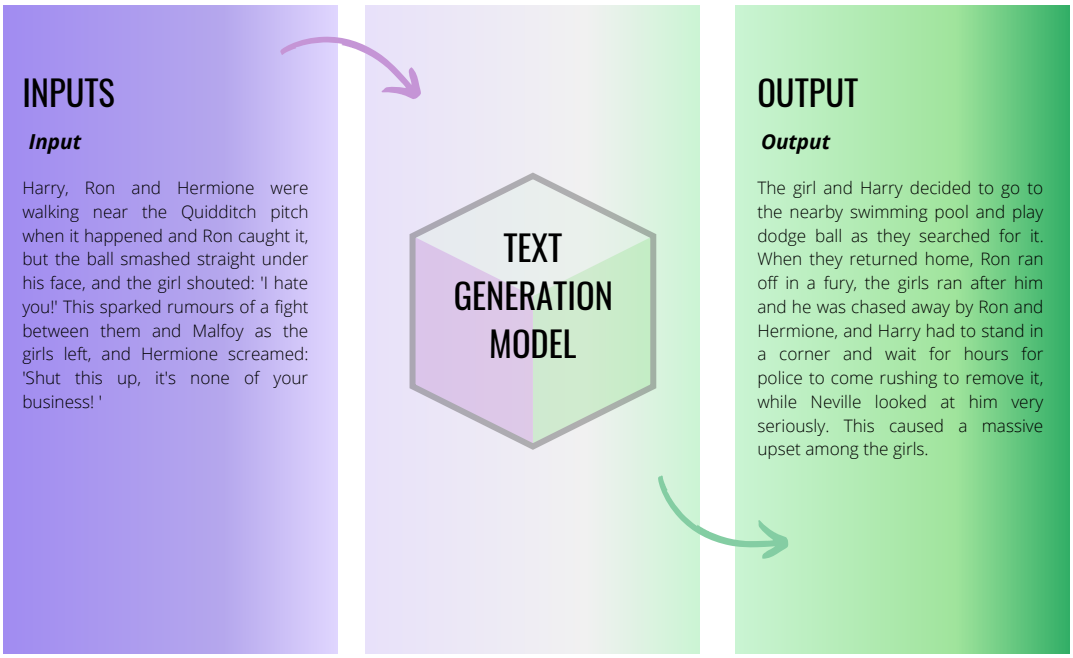

FIG. 1. Example of story generation for Harry Potter and the philosopher's stone using GooseAI.

- Another idea would be for it to create its own story based on many other books. Actually, it could create a special story that would be a crossover between the Harry Potter's books and the series Hunger games. We could also create a completely new one by only specifying the literary genre we want it to be. In our specific case, we could create a new fiction with our bot inspiring itself of Harry Potter, Hunger Games,...

VI. REVIEW OF RELATED WORK

There exists several works concerning the automatic generation of text using deep learning. Most of them also evolved to communication, translation and so on. In addition to models that were developed, we can also find multiple papers treating the subject. Let's first take a look at the model/platform.

- GooseAI : it's a platform that offers text generation as a service. We can test their solution by using the playground area where a text is generated based on a first sentence we have written.
- Word2vec: this model does not generate entire paragraphs but can suggest words to complete sequences and find synonyms. It is used in multiple apps like Tinder.
- OpenAI's GPT-3 : largest text generation model ever trained with more than 175 billions of parameters.
- DeepAI : similar to GooseAI, offer text generation/completion services.
- MTuring Natural Language Generation (T-NLG) : model designed by Microsoft, using 17 billion parameters that can generate text, answer questions and summarize documents.
- and many others...

Now let's review some papers treating the subject:

- The survey: Text generation models in deep learning, by Touseef Iqbal and Shaima Qureshia.
- Generating Sequences With Recurrent Neural Network, by Alex Graves.
- Attention Is All You Need, by the Google Team.

There are also multiple books on this subjects:

- Natural Language Processing with TensorFlow: Teach language to machines using Python's deep learning library.
- Advanced Natural Language Processing with TensorFlow 2: Build effective real-world NLP applications using NER, RNNs, seq2seq models, Transformers, and more.
- Natural language processing with python.
- Understanding, analyzing, and generating text with Python.