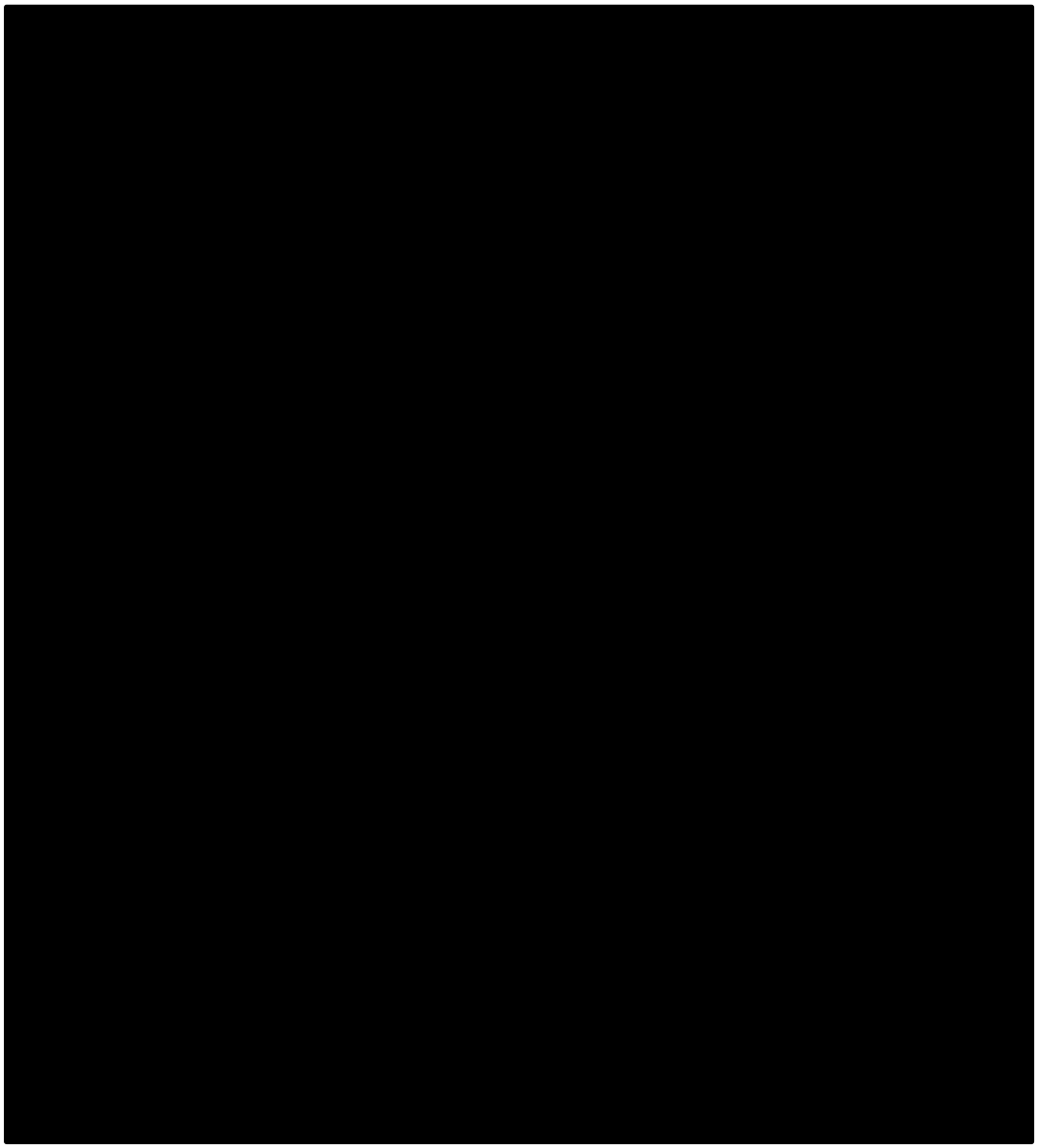
An intelligent system for the emulation of a specific individuals lexical and semantic choices within conversation



Acknowledgements

I would like to acknowledge the help and support provided by my family and friends. More specifically, I would like to acknowledge Morris and Keavey for proof reading this document even if they don’t understand it.

Abstract

In this paper a Recurrent Neural Network (henceforth RNN) is trained on a dataset of informal interviews with the intention of modelling how people in said dataset construct answers to sentences, more specifically word choice. Clustering is also implemented to cluster utterance pairs that are similar and train multiple RNNs on them to attempt to better capture outlier topics and word choices. These are evaluated using Mean Reciprocal Rank (henceforth MRR) and Bleu.

Table of Contents

List of Figures

List of Table

1 Introduction

Ever since the creation of ELIZA (Weizenbaum, 1966) in 1966 great strides in convincing sentence synthesis has been made. Modern personal assistants such as Siri and Alexa can recognize almost all the complexities of human speech and use it to not only understand but create meaningful sentences. However, most research has focused specifically on creating meaningful and correct responses as seen in the three leading personal assistants Siri, Google Assistant and Alexa. The lack of research into creating more human feeling responses is a result of it being often considered unimportant (Quarteroni et al. 2007,79) for what is in reality very little gain. This project aims to look specifically at this problem by co-opting the way an individual constructs, phrases and delivers responses to questions.

This would hopefully result in not only more personal human-computer interactions, in the context of a personal assistant but to also be applied in the real world to make environments more engaging and reduce workloads. Such as utilizing records of historical individuals to recreate them convincingly in museum exhibitions, lending greater interactivity and humanization to figures in history. Perhaps a virtual nurse is created, from a reference of an existing nurse, that can provide all-day and all-night emotional care; with the real nurse only having to step in for physical interactions.

As suggested by Capgemini (2018,4), 54% of surveyed consumers would prefer more human-like behavior and personality from AI-based services, the project specifically targets this area by using existing people to provide personality and depth.

This project focuses distinctly on lexical and semantic choices at the word level and is not concerned with making responses factually accurate or incorporating speech synthesis so that a person's voice can be captured. Having said that there is a potential that with a big enough dataset some factual responses could be generated (Brown et al, 2020,13. Joshi et al, 2017). The demonstration video is only a scroll through of code as issues with the environment and dependencies has resulted in no visible output being possible (see links to resources below).

- Demonstration video: https://youtu.be/wJGoQsy_FIc
- Code: https://colab.research.google.com/drive/1LcDSiWBPC3v-S9FslVydZCFGW0whiHVB?usp=sharing
- Dataset: https://drive.google.com/file/d/1wEdQQdLuB_Yaa3z_vBNH-ZKRMF7ih0x1/view?usp=sharing

1.1 Aims and Objectives

To produce the produce a working artefact five core objectives need to be achieve, a working model, sourcing data, a means of evaluation through Mean Reciprocal Rank (henceforth MRR), implementing K-means clustering and final evaluations and tuning; all while fully documenting development. There is one important stretch goal and that is to implement a pre-trained approach similar to that of GPT-3 where the model is trained on multiple speakers and then tuned using one speaker to cover any gaps (see section 2.6)

1.1.1 Core Objective and Aims

- Develop and implement a model capable of breaking down sentences and finding patterns within a transcription.
- Source, format, and label (if needed) transcriptions of interviews with a large group of individuals using word similarity scores or word embeddings
- Implement MRR so that proper accuracy metrics can be made
- Implement K-Means clustering using some form of word embeddings having each cluster compete to for accuracy (see Sections 3.3.2.2)
- Fully evaluate and tune the model to increase accuracy
- Document in full each stage of development with in depth discussion on all key aspects of development

1.1.2 Stretch Objectives and Aims

- Implement a pre-trained approach similar to that of GPT-3.

2 Background and Literature Review

2.1 Taxonomy

Feine, et al (2019) outlined a detailed taxonomy for discussing and classifying what they call conversational agents (CAs). This provides an outline for how to approach analysis, evaluation, and discussion of outputs. For example, they discuss and outline social cues/signals and their importance in constructing dialogues(ibid,11). They also point out how humans relate to CAs in positive aspects such perceived empathy and politeness (ibid) or in potentially negative ways such as applying gender stereotypes.

2.2 Lexical choice and near synonyms

Modelling lexical choice is typically done by building models of word meanings using these algorithms show when a word can and cannot be used (Reiter and Sripada, 2002). However, “recent experiments carried out in the SUMTIME project… this may be difficult to do because of variations among people.” (ibid). As a word can mean different things to different people in humans typically engage in “conceptual pacts” where they decide on a words meaning as the conversation progresses (Brennan and Clark 1996). An informed understanding of how word choice starts off unique to a person (Reiter and Sripada, 2002, 550) and how it may change over the course of a longform conversation is important for constructing human-like dialogue between CAs and humans.

Near synonymy is an important part of creating individual meaning with Edmonds (Edmonds, 1999. Edmonds and Hirst,2002) suggesting using dictionaries such as Webster’s new dictionary of Synonyms (Reiter and Sripada, 2002) would be an effective

way of deciding synonym choice. However, they were working on computer translation and Reiter and Sripada (2002,550) suggested that this method was not capable of fine levels of detail needed for their SUMTIME system for summarizing forecasts. Moreover, there is often an inaccuracy between the written down synonym (prescriptivism) and how it is used within a real context (descriptivism) (Backstrom, 2006,2). To this end research for building linguistic models is potentially an ill fit for the scope of this project.

2.3 Misuse

Brown et al (2020,35) also raise an interesting ethical argument on the potential misuse of a system that sounds so much like a human. They outline the potential for "misinformation, spam, phishing, abuse of legal and governmental processes" as ways in which language models could possibly "lower existing barriers to carrying out these activities and increase their efficacy". From their results(ibid,26) they found that on articles around 500 words long it's up to chance on whether an 80 US participants can detect it. They observed that aspects such as factual inaccuracy (due to the model not being able to get facts on its own), reptation, non-sequiturs and unusual phrasings are common ways of noticing model generated text (ibid,26). Overall, they assessed that "highly skilled and well-resourced (e.g., state-sponsored)" could potentially employ such software to generate misinformation, while assessing "low and mid-skill actors" as not an immediate threat.

2.4 Non-Machine Learning Approaches

The most common form of non-machine learning based approach is an Information Retrieval system which may use a search engine or assigned database to locate the best possible response to a posed question.

The idea behind such systems is often not to attempt "to perfectly emulate human dialogue using a machine" and that's its likely an "unrealistic and perhaps unimportant goal." (Quarteroni and Manandhar 2007,79). Quarteroni and Manandhar (2007) worked on a system that was split questions in to two categories Factoid and Non-factoid answers factoid answers are things such as Persons, Organisations, Locations, Quantity, Time and Money; with specific manual processes done for the later three (*ibid*). Non-Factoid answers utilize Bigram, Chunk, Head NP-VP-PP and WordNet Similarity put through a combined similarity function (*ibid*, 78) to produce the closest possible result from a search engine.

Unlike Quateroni and Manandhar (2007), Banchs and Li (2012) used a movie dialogue dataset as opposed to a search engine and used a vector space model, cosine similarity is applied to find the most appropriate dialogue to respond with. Their system (IRIS) was also capable of learning new vocabulary terms and semantically linking them to its existing knowledge base(*ibid*). However, the IRIS struggled to provide "Consistency in issues for which consistent answers were required" for example when asked its age it provided 2 different ages; IRIS also struggled with noise in the dataset

as it occasionally gave out stage directions from the scripts instead of dialogue (*ibid, 41*).

Amexia et al (2014) built on what Banchs and Li (2012) had done before them by switching from movie dialogues to movie subtitles and using them as a means of augmenting existing models (Serban et al, 2015), so that what they call Out-of-domain (OOD) questions (questions that might not be accounted for by the dataset) could be responded to accurately. They posited using subtitles would provide a broader array of languages and be easier to find; but most importantly "linguistic variability would be covered, and redundancy can be taken into consideration" (Amexia et al 2014) mitigating IRIS' (op cit) big draw back by ensuring similar answers are answered the same way, preventing contradictions. Amexia et al (2014) believed that future research should account for context as well as giving the agent a personality.

2.5 Natural Language processing implementations

Approaching generating text as a conditional problem has been approached in multiple different way before such as Ritter, Cherry, and Dolan (2011) utilizing Statistical Machine Translation to translate and response from a given a question, finding that it was "better-suited than IR (information retrieval) approaches" and even preferred by human 15% of the time (*ibid*). However, accepted that outputs were "far from human-level performance" and suggested redesigning the word-alignment and decoding algorithms so that they can account for "selective nature of response in dialogue" (*ibid*).

Shang, Lu, and Li (2015) utilized a similar idea using recurrent neural networks to produce a "Short-Text Conversion" they noted that it could generate "grammatically correct and content-wise appropriate responses to over 75% of input text" and was capable of outperforming "retrieval -based and SMT-based models" in similar settings (*ibid*). The study specifically used data from microblogging websites so likely afforded a broad range of topics and writing styles leading to better overall accuracy (*ibid*). It is also suggested that in future work incorporate intention/sentiment analysis as an external signal decoder so that responses can have more specific goals (*ibid*).

Sordoni et al (2015a) pointed out that a translation approach such as Ritter, Cherry, and Dolan (2011) don't attempt to generate responses that "are sensitive to the context of the conversation" these contexts vary but typically entail linguistic or grounded in the real/virtual world (*ibid*). Sordoni et al specifically focus on adding linguistic context by "using continuous representations or embedding of words and phrases" this allows for compact encoding of semantic and syntactic similarity (*ibid*); they argue that such an "embedding-based models" provide greater flexibility to model transitions between consecutive utterances (*ibid*). Utilizing a Recurrent Neural Network Language Model taken from Mikolov et al (2010) (Sordoni, et al, 2015a). From this implementation Sordoni, et al (2010) improved 11% over translation efforts and 24% over Information Retrieval systems.

Serban et al (2015) took the idea of using movie dialogues to augment existing training data from Sordoni, etal (2015a) but also applied a Hierarchal Recurrent Encoder Decoder Neural Network (Sordoni et al, 2015b) this was capable of producing and

predicting “open domain dialogues” (Serban et al, 2015,9). They also “bootstrapped” Word2Vec embeddings from the google news corpus to further augment the existing data (*ibid*, 7). Overall, with these augmentations there was a slim increase in perplexity of around 1% extending to 2% (ibid,8) when stop-words were removed even then majority of responses were considered generic with outputs such as “I don’t know” or “I’m sorry” (ibid,9).

Fedus, Zoph and Shazeer (2021) utilized what they called a “Mixture of experts” approach to produce the Switch Transformer which produces “scalable and effective natural language learners”. This model selects the most appropriate parameters for a given input. Previously this approach suffered from high complexity, communication costs and training instability.

2.6 Pre-trained models

Brown et al (2020) proposed the use of a pre-trained model for NLP tasks called GPT-3; it was trained on a large corpus of text and can then be fine-tuned using a specific corpus. This “Autoregressive language model” has “175 billion parameters” 10x more than any other “non-sparse language model”. Within the paper showcases a new pipeline for QA systems (ibid, 13), supported by Roberts, Raffel and Shazeer (2020, 5), instead of using the traditional IR system followed by language model to create a response, what they denote as “open book”, if there are enough parameters the system maybe capable of producing accurate responses on its own. Roberts, Raffel and Shazeer (2020, 5) show that their model can achieve 57.8% accuracy in response when presented to humans to evaluate and GPT3 managing 64.3% accuracy on the TriviaQA dataset (Brown et al, 2020,13. Joshi et al, 2017). However, without access to GPT3 or a similarly sized model such implementation results will be much lower accuracy.

2.7 Evaluation Metrics

Mean Reciprocal Rank (MRR) as discussed by Radev et al (2002) is an evaluation system where in the model will order a set of possible responses to a question. One of the responses is correct the others are randomly generated. The model is scored on where it ranks the correct response among the random ones. This collection of ranks is then meaned to produce a final score. They also proposed: “First Hit Success (FHS), First Answer Reciprocal Rank (FARR), first answer reciprocal word rank (FARWR), total reciprocal rank (TRR), total reciprocal word rank (TRWR), and precision (PREC)” (ibid, 1156) however these seem more appropriate for a Q and A system which is not within the scope of this project. A second is Bleu as proposed by Papineni et al (2002), while it is intended to be used for machine translation, it should be applicable to this project as it compares the machine output to a reference human sentence. it also can perform n gram precision of too four blocks which provides better idea of the shortfalls in the output as it may occasionally generate the correct word but maybe not be good a generate three of the right words in a row. However, Liu, et al (2017) discussed that bleu may not be a great metric for this problem space, for example the question “Why

don't we go see a movie?" has a ground truth of "Nah, I hate that stuff, let's do something active." A "reasonable response" from the model of "Oh sure! Heard the film about Turing is out!" would receive a bleu score of zero. But will still be used as it provides great insight into how good the model is generating strings of words.

$$r_i = \begin{cases} rank(q_i), rank(q_i) \le 5 \\ \infty, rank(q_i) > 5 \end{cases}$$

$$RR(q_i) = \frac{1}{r_i}$$

$$MRR(Q) = \frac{\sum_{q \in Q} RR(q)}{|Q|}$$

Fig. 1. Mean Reciprocal Rank Equation as shown by Lin, et al (Unknown)

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \le r \end{cases}.$$

Then,

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right).$$

The ranking behavior is more immediately apparent in the log domain,

$$\log \text{BLEU} = \min(1 - \frac{r}{c}, 0) + \sum_{n=1}^{N} w_n \log p_n.$$

Fig. 2. Bleu Metric Equation as shown by Papineni et al (2002)

2.8 Normalisation

It may also be necessary for the data to be normalized. Toman et al (2017) identified three potential ways of normalization: applying the EuroWordNet Thesaurus, Lemmatization, and Indexing. The EuroWordNet approach would use this database to transform words to a more common on thereby removing outlier words, this however result in a loss of specificity unless these synonyms are kept track of which increase compute time. Lemmatization is the process of removing the suffix from a word so for example changing would become change, this requires that a dictionary be kept of all the possible variations of this result as to not lose specificity. Indexing is better for machine translation as a unique index is assigned to each set of synonyms and these indexes are shared between languages making machine translation easier. Overall, text normalization may lead to potentially better results as the number of unique words would be

reduced however the lack of specificity maybe detrimental to the ethos of this project (see more in section 4.1.5)

2.9 Clustering approach

As mentioned in the introduction using clusters of trained models might "compensate for some of the shortcomings of conventional class-based models by combining the different solutions obtained through random clustering" as show by Emami and Jelinek (2005) with them "Combining the resulting class-based models resulted in considerable improvements in the performance, showing that the different clustering's represent uncorrelated interpretations". This approach was validated by Mulder, et al (2014) with the "basic RNN had a lower perplexity than all other models, except for the within and across sentence boundary LM which had a perplexity of 116.6, while the basic RNN had a perplexity of 124.7" (ibid, 75). Cuayáhuitla et al (2019) approach it similarly with the idea being that individual clusters could become "specialised agents" that are capable of interacting in a particular style, this makes the model more flexible and capable of responding to more specific questions however as a result the "generalisation ability in a test set of unseen dialogues".

3 Methodology

3.1 Project Management

3.1.1 Gantt Chart

To keep track of the broad strokes of the project a Gantt Chart (see fig.1) was used. This is due to the charts ability to provide a rough guideline over the whole development window using as it provides one single viewpoint for the whole process (Ntask-manager, unknown). Each task was derived from the aims and objectives (see section 1.1); but the Gantt chart is not showing each individual task necessary for completing said objectives, as that would decrease readability and make it more confusing (Friedman, 2020), week-to-week task assignment is discussed in section 3.2. The use of contingencies in the chart gives greater flexibility for task completion times while still providing a hard cut off for when the task must be completed and stretch goals provide an insight as to additions that can be made if given the time.

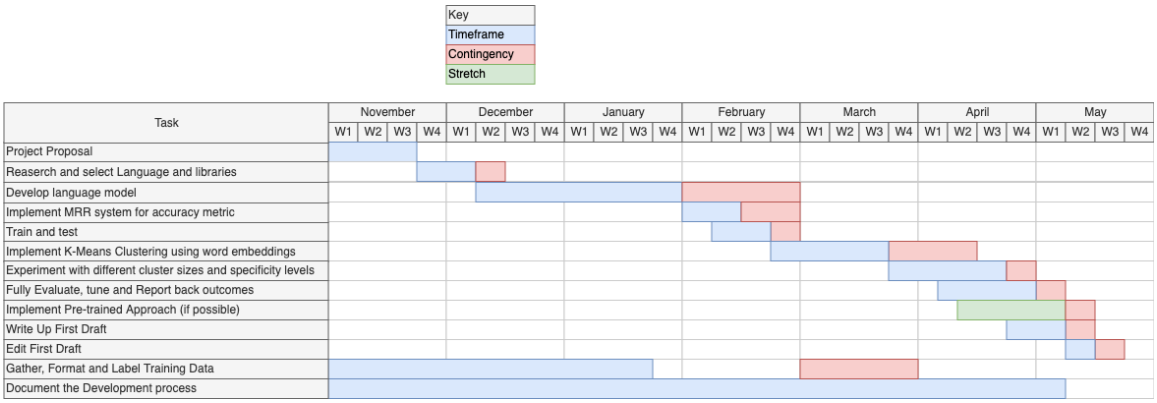


Fig. 3. Gantt Chart used to manage the broad strokes of the project

This Gantt Chart has been reconfigured since the project proposal document as the scope of the project was changed. Initially it was hoped that a stretch objective could be providing factually accurate responses, however given other commitments and time constraints it had to be dropped entirely and substituted for a more feasible goal of implementing a pre trained approach where the model could tune itself based off of new data.

The Gantt Chart closely followed development times with the language model taking longer than contingency to implement as a result of external exams, as a result the stretch goal was dropped in favor of making more optimizations and expanded and more detail parameter-based testing (Paul, 2018). The writing processes started much earlier than expected but have paid off by providing more time to deliver as strong a write up as possible.

3.1.2 Supervisor Meetings

Weekly meetings will be held with the project supervisor to provide updates on the state of development as well as provide a valuable second perspective on key development decisions and methods. Meetings were held almost every Friday for 15 minutes, where work for the last week was discussed, any potential roadblocks were identified and worked through ending with tasks for the next week being worked out (See section 3.2 for more). Detailed minutes were maintained for every meeting covering all conversation so that it is absolutely clear what was covered and what the next steps are. They also provided a second set of eyes to notice potential development issues like noticing flaws in the coding process as well as having the experience necessary to notice potential pitfalls not obvious at first. Whenever meetings could not be held update emails were sent providing a short update of that week's work followed by a slightly longer next meeting to make up for lost time. Overall, these meetings where immensely useful and helped keep the project on track and filling the gaps in knowledge and understanding that helped deliver the final artefact.

3.1.3 Risk Matrix

Risk	Detail	Likelihood	Impact	Consequences
Running out of time	Likelihood of running out time either because of hand-in deadlines and unforeseen effects of other modules	Low	High	Failure to deliver on the full scope of the project
Lacking efficiency	The project fails to deliver on efficiency	Moderate	High	A slower than optimal solution
Code Formatting	Code is not easily read	Low	Moderate	Harder to debug and make code side optimisations
Scope creep	Potentially as the project is worked on the scope could get larger	Low	High	The project may not fulfill its original requirements as it has strayed too far
Compiling data	As data will need to be compiled, cleaned, and labelled, potentially this could take longer than expected and potentially yield a dataset that is too small	Moderate	High	Sacrifices may have to be made for instance decreasing cluster size or removing labels
Project produces incoherent outputs	The project begins producing output that make little to no sense	High	High	Adjust parameters and recognition methods to hopefully create better results

3.2 Software Development

- Must include an analysis of the software development approach used
- Discuss factors for selection for example:
  - Particular characteristics of the software developed
  - Computer environment requirements

- Work from the requirements of my project and explain how they determine the overall other methodology of the project
- Identify specific demands of the project in terms of software development

3.2.1 Scrum

Scrum is a development model characterized by its use of sprints which are blocks of development typically two to four weeks in size (digite, unknown) and then a meetings are held where sprint is reviewed a retrospective is done to better understand what has been done then with this information the next sprint is planned (ibid, rehkopf, unknown). Traditionally there is a daily meeting called a daily scrum where there are quick updates and synchronization between team members, but as this is a one person project this process was foregone (ibid).

In practice every supervisor meeting served as the sprint review, retrospective and planning stage, there was no apparent scrum master as ideas were presented evenly between the participants and tasks were adopted through mutual agreement. Scrum was most appropriate as it was flexible enough to account for the varying amount of work that could be done in a week, as a result of other course commitments and deadlines, but was also rigid enough that tasks where always known, and it was always clear where in development the artefact was.

The whole process was documented in meeting minutes using Microsoft OneNote (Microsoft, 2003) (see figure 2), they were used to as the reminder for what tasks needed to be done in each sprint as well as serving as a full backlog of the development process for reference where writing up the document. Supporting documents where, also made to increase understanding of the sprint within the same program.

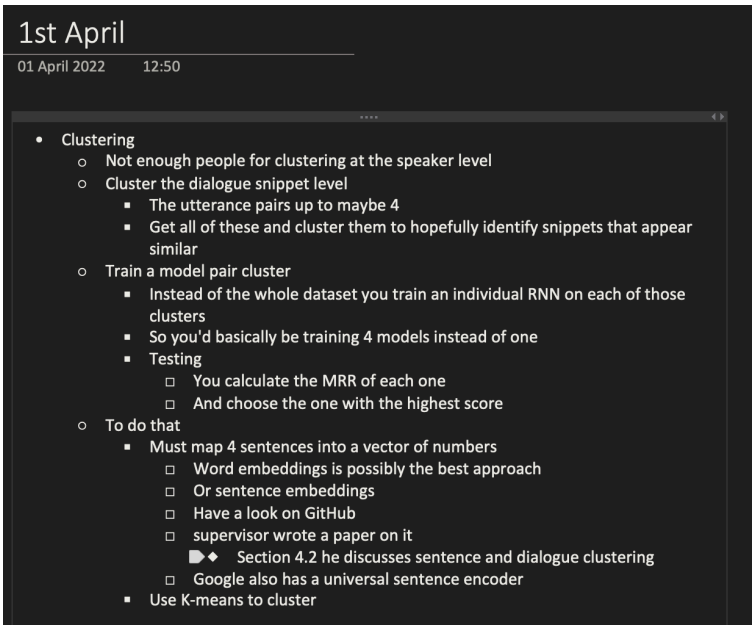


Fig. 4. Example of the allotted tasks for a week of the sprint

3.3 Toolset and Machine Environments

3.3.1 Language choice

There are three main viable languages for machine learning Python, Java, and C++. All three have large communities with a wealth of resources available for building the artefact, they and their extensions take up the top six of the most popular languages (Eastwood, 2020).

Python is the most popular language (Eastwood, 2020) and thanks to libraries such a Numpy, Pandas, Matplot and Seaborn adding greater functionality and visualization options before any MLops (Banerjee, 2021). Python is the most flexible with it being usable in a variety of different styles and use cases (Luashchuk, 2019). It is "simple and consistent" offering "concise and readable code" (Beklemysheva, unknown) this makes getting the artefact up and running quicker and easier. Finally, python offers a great variety of MLops libraries like TensorFlow, scikit-learn and PyTorch each with their own use cases. (Beklemysheva, unknown) (See section 3.3.2.1 for more)

Java is considered a "Jack of all trades" (Luashchuk, 2019) language it is also the third most popular with JavaScript in second (Eastwood, 2020). Java focuses on an object-oriented approach (Beklemysheva, unknown) making it less flexible than Python. Java is "widespread when it comes to natural language processing, search algorithms, and neural networks" (Beklemysheva, unknown), it is supported by libraries such as JavaML, Deeplearning4j and ELKI (Luashchuk, 2019).

Finally, C++ is the sixth most popular language with C# and C in positions four and five language (Eastwood, 2020). Due to C++ has more finite control over memory and operations (Software Testing Help, 2022), this makes it the least flexible. However, operations may likely run faster, however this lower level adds complexity and debugging takes longer (Reszke and Jungiewicz, 2021 and Banerjee, 2021). Like Python, C++ has access to TensorFlow, mlpack and Torch (PyTorch without the python wrapper (PyTorch, unknown)) (Banerjee, 2021).

In conclusion Python is the best suited for artifact development, due to its flexibility and the relative speed of development allowing for fast prototyping. Plus, its wealth of support libraries like Numpy and great machine learning libraries like TensorFlow make it best suited given the time constraints and scope of the project.

3.3.2 Libraries

3.3.2.1 Selecting An appropriate machine learning library

This project is not concerned with a manual "from scratch" implementation of machine learning operations due to time limitations and efficiency of the implementation. As a result, machine learning libraries will be used to afford a quicker and better optimized implementation as well as providing of a wealth resources for creating the artefact. This allows for more work to be put into optimizing and experimenting to produce better results.

The most appropriate machine learning library will need to meet three specific requirements:

1. Appropriately detailed and comprehensive documentation and great learning resources
2. Simple yet flexible design
3. Efficient processing preferably GPU accelerated

TensorFlow full fills all three of these greatly. Developed by Google it has a considerable amount of documentation and is the most popular python-based machine learning library (Shetty, 2018) ensuring a great deal of resources outside of official documentation. It is also greatly flexible with the work because of its low level of execution (ibid) allowing the model to be easily adjusted as requirements change. It is also incredibly efficient using less RAM than other libraries with similar functionality and accuracy (Simmons and Holliday, 2019) for certain operations, most importantly TensorFlow boasts compatibility with CUDA architecture (TensorFlow, unknown) which greatly decreases training times (DATAmadness, 2019). The fact TensorFlow is developed by Google may also lead to performance gains within the intended development environment (See Environment section).

TensorFlow also provides access to Keras which is a more closed off version higher level of TensorFlow which is better for more common use cases (Choudhury, 2019).

Another Great Contender is PyTorch developed by Facebook's AI Research lab (FAIR). PyTorch is widely utilized (Shetty, 2018) and well especially well supported through the PyTorch Hub (PyTorch, unknown) a community driven platform made for sharing models (Johns, Unknown). Simmons and Holliday (2019,24) found PyTorches more object-oriented approach "made implementing the model less time-consuming and the specification of data handling more straightforward." (ibid); John (unknown) found PyTorch had "Finer-grained control of model structure". And that it had better compatibility with Numpy an important supporting library. Much like TensorFlow PyTorch benefits from GPU acceleration utilizing CUDA (Thakur, 2022) Lyashenko (unknown) found a considerable decrease in training time per epoch when utilizing GPU acceleration. Simmons and Holliday (2019,24) found it consumed more RAM compared to TensorFlow.

Finally, scikit-learn (henceforth SKlearn) has a different ethos to PyTorch and TensorFlow, Sklearn focuses on fast construction of models so that testing with different models takes less time and as a result abstract many fine detail controls available in TensorFlow (Gupta, 2021). While nowhere near as popular as TensorFlow and PyTorch (Allen and Li, 2017) Sklearn has robust and detailed documentation available through its website as well as detailed tutorials for getting implementation up and running (Scikit-learn, Unknown 1). Sklearn Unfortunately doesn't support GPU acceleration (Scikit-learn, Unknown 2).

In conclusion TensorFlow will most likely be the best choice for the artefact as it provides a strong low-level foundation for making flexible code, it is supported by a large community and wealth of documentation, providing easy learning resources, and it uses the least amount of system resources while affording GPU acceleration to make training much quicker. However, for any small jobs such as clustering Sklearn will be used as it is the simplest and most abstracted making it better for operations that don't require fine control.

3.3.2.2 Other Support Libraries

When testing the model, it may be appropriate to visualize the data, to do this Matplotlib will be used as it is very simple and easy to use, providing detailed and customizable plots of multiple different types (Matplot, unknown).

Numpy is a comprehensive library for performing mathematical functions in python, it is widely support by machine learning libraries and incredibly flexible yet powerful Numpy (unknown). Making it a great support library for more complex operations than python alone can perform.

Universal Sentence Encoder is a google library that allows the generation of heatmaps that show how close the subject of sentences are to each other for instance the sentence "I like My Phone" and "My Phone is not good" score a 0.8 in similarity (Google, Unknown 4). This makes cluster sentences that are similar much easier as it automatically generates a similarity scores that can be passed into a K-Means Algorithm.

3.3.3 Environments
Due to the potential necessity of large amounts of compute resources Google Colab was selected as the best environment for artefact development. This was due to four main reasons. Firstly, collab allows the offloading of compute resources to the cloud (Google, Unknown 1) for a very reasonable price of £8.10 a month (Google, Unknown 2) as it's in the cloud work can be carried out anywhere giving better flexibility within development period. Colab also allows 24-hour, consistent access to high performance GPUs/ TPUs and 32Gb RAM (Droste, 2021, fig 2), TPUs (Tensor Processing Units) are similar to GPUs but optimized for machine learning tensors (Ibrahim, 2021). TPUs and GPUs are considerably faster (ibid) than a local implementation would allow due technical limitations. Finally, Colab cross platform with Jupyter notebooks as they use the same ipynb file extension (Fileinfo, Unknown) this allows for easy moving between the two when moving from a local machine prototyping stage to a full cloud implementation. Ipynb files boast cell-based structure which allows for select sections of code (Palomino and Wasser, unknown) to be ran so once the model is trained testing can be modified without having to retrain all over again.

4 Design Development and Evaluation

4.1 Requirements

From the literature review the most popular method for implementing models similar to what this project requires was recurrent-neural-networks some adopted hierarchal models (Sordoni et al, 2015b), however given the time constraints of this project implementing a hierarchal model is possibly too complex. However, as outlined in the introduction a clustered approach will be used, using K-means clustering similar to Mulder, et al (2014) and Cuayáhuitla et, al (2019). Also, artefact will require a substantial dataset of conversations in a more informal setting than news interviews to accurately capture how people really speak when unrehearsed.

4.2 Dataset

4.2.1 Compilation
The majority of data was sourced from the transcription archive for Larry King Live[1]. This talk television show was selected as the context of the conversations are more informal and unrehearsed, as such they better represent how a person's word choices when confronted with a normal conversation. News broadcasts where not included as they are more likely to be formal discussions with some rehearsed aspects for instance if a politician is being interviewed for the news's they'll likely prepare by gathering talking points and steering conversations to established narratives something O'Donnell called the "pivot" (Spiegel, 2012). On the other hand, as talk shows a typically used

[1] http://edition.cnn.com/TRANSCRIPTS/lkl.html Larry King Live Transcripts

as advertising or part of a “press tour” many speakers have a reason they are there such as promotion, which adds some inauthenticity.

4.2.2 Members of the dataset

In total there are 3319 utterance pairs (question and response), with a total of 8477 unique words, sourced from twenty-two individuals (All information accurate to when interviews where recorded, all interviews sourced from Larry King Live except where noted):

Individual	Occupation	Gender	Age	Background
Johnny Depp	Actor	Male	47	White American
Jerry Seinfeld	Comedian	Male	56	White American
Michelle Obama	Politician, First Lady of the United States of America	Female	46	African American
Jimmy Fallon	Comedian	Male	35	White American
Paul McCartney	Musician	Male	59	White English
Barrack Obama	Politician, President of the United States of America	Male	48	African American
Stefani Germanotta (Lady Gaga)	Musician	Female	24	White American
Mike Tyson	Athlete	Male	43	African American
Jimmy Carter	Politician, Former President of the United States of America	Male	85	White American
Oprah Winfrey	Talk Show Host	Female	53	African American

Jon Stewart	Comedian, Talk Show Host	Male	43	White American
Tyler Okonma[2] (Tyler the Creator)	Musician	Male	27	African American
Calvin Broadus Jr (Snoop Dogg)	Musician	Male	38	African American
Joe Biden Jr	Politician, Vice President of the United States of America	Male	67	White American
Jill Biden	Educator	Female	58	White American
Imam Feisal Abdul Rauf	Author, Activist	Male	61	Kuwaiti Egyptian American
Dolly Parton	Musician	Female	64	White American
Tyra Banks	Television Personality	Female	36	African American
Robert Zimmerman (Bob Dylan)[3]	Musician	Male	24	White American
Patti Smith[4]	Musician	Female	73	White American
Ellen DeGeneres	Talk Show Host, Actress	Female	46	White American

[2] Sourced from Flower Boy: a conversation, Transcribed by Genius: https://genius.com/Tyler-the-creator-flower-boy-a-conversation-lyrics

[3] Sourced from KQED Studios, San Francisco, CA December 3rd, 1965, Transcribed by Dylan Stubs: http://dylanstubs.com/extras/1965.pdf

[4] Sourced from 'Fresh Air' Favourites: Patti Smith, Transcribed by NPR: https://www.npr.org/2020/01/03/793049438/fresh-air-favorites-patti-smith

Kendrick Lamar[5]	Musician	Male	30	African American

Table 1. Breakdown of the individuals in the dataset by age, gender, profession, and race

Due to the potential for language to be different between people of different ages, genders, and races, it's important to gauge an idea of the types of people present in the dataset and discuss the potential for bias within training.

4.2.3 Visualizing the dataset

Below is a breakdown and examination of the dataset. It includes the types of people in the dataset broken down by age, gender, ethnicity, and location and some example dialogue snippets

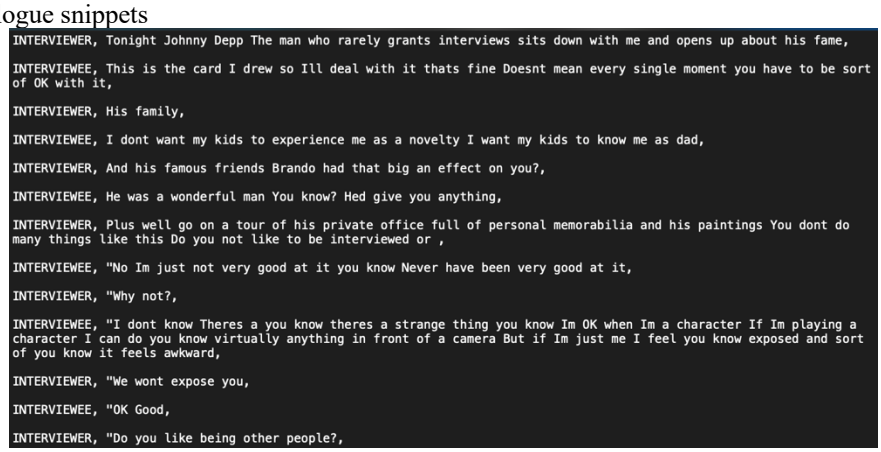


Fig. 5. Example dialogue snippet from the set

[5] Sourced from: Kendrick Lamar Talks to Rick Rubin About "Alright," Eminem, and Kendrick's Next Album, Transcribed by GQ: https://www.gq.com/story/kendrick-lamar-rick-rubin-gq-style-cover-interview

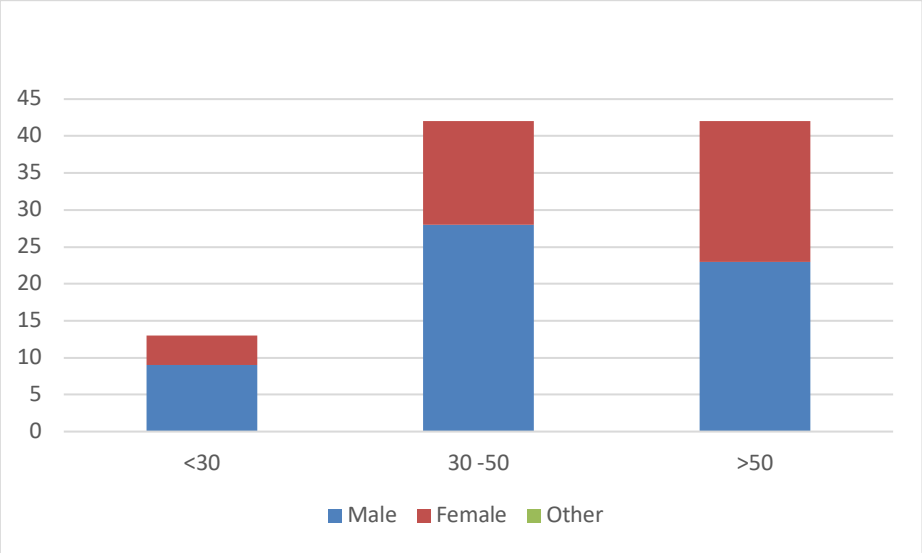


Fig. 6. Breakdown of dataset by Age and Gender

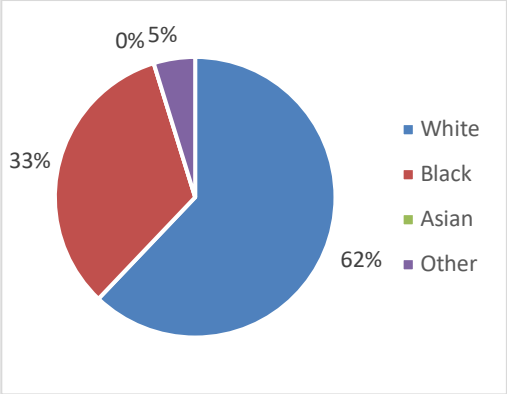


Fig. 7. Breakdown of dataset By Ethnicity

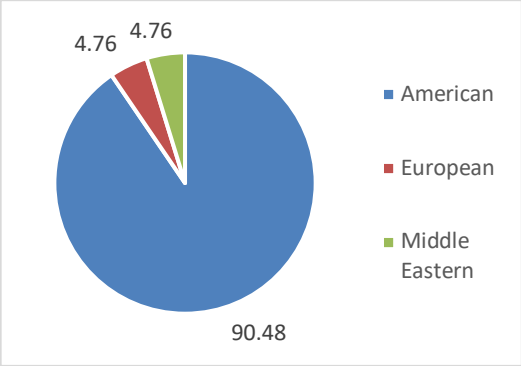


Fig. 8. Breakdown of dataset by Locational Background

4.2.4 Bias within the Dataset

With this kind of dataset that uses real people, it is important to consider bias within the dataset and how this could affect results. From visualizations, it is clear there is a bias towards white male Americans between thirty and fifty. This may not represent a massive issue as there are many common word choices not affected by those factors however more specific language such as slang and colloquialisms may differ considerably.

There are obvious holes within the dataset, there, is no representation for Asian and non-Binary individuals this is mostly due as a result of the source of the transcriptions. For instance, in the whole ten years of archives available for Larry King Live, there are no available interviews with Asian individuals and attempts to find similar transcription archives that may contain them proved fruitless. The lack of non-Binary representation is due to the fact that available transcriptions are from January 2000 to December 2011, as a result, the population of non-Binary individuals was likely much smaller than it is now (1.2 million, Williams Institute, 2021) and as a result, probably had no major representation among the kind of guests the show hosted.

4.2.5 Data Normalisation: linguistic and computation discussion

Given the complexity of modelling language due to words not having fixed definitions or uses (Mulder, et al, 2014) it may be important to consider data normalization. There as several approaches when it comes to normalization (see section 2.8), but most of them have the effect of losing the specificity in words by converting them to a “basic form” (Toman et al, 2017). In a way, normalization would be something akin to the linguistic concept of Prescriptivism, the process of prescribing how language should be used (Curzan, 2014, 18-24); which seems inappropriate given how the goal of the project is to capture how individuals use language or Descriptivism (ibid). In practice, it may result in it better more cohesive results (Toman et al, 2017), but it would fail to capture the way individuals use language and how definitions and synonyms differ

between them. Take for instance the English word “Bollocks” in definition it is a vulgar term for a man’s private area (Meriam Webster, unknown) but in practice among English speakers it has a multitude of different meanings depending on context, these can span “Exasperation... unfathomable rubbish… telling off…” (urban dictionary, unknown); as a result, some of the normalization approaches discussed by Toman, et al (2017) would likely change the world but maintain the vulgar aspects and completely exclude its extended use. Overall, there will likely be more inconsistent results in testing but it’s necessary given the ethos of the project.

4.3 Design Architectures and Implementation Discussions

4.3.2 Training Architecture

Based off of the work done Shang, Lu, and Li (2015) and Sordoni, et al, (2015a) a recurrent neural network was chosen as the basis for the model and a tutorial by TensorFlow (2020) helped get the code started. In the end the Training Architecture looked like this:

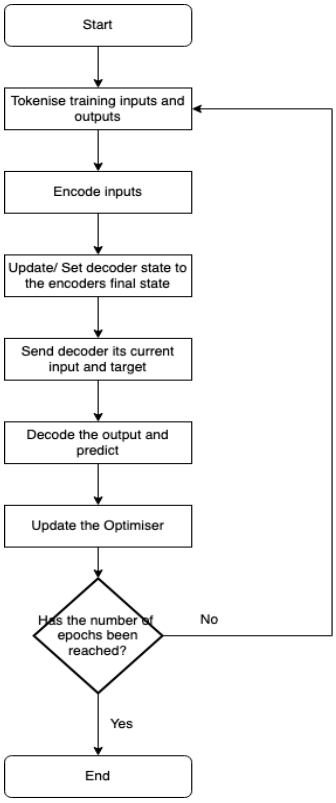


Fig. 9. Flow chart of training process and architecture

The process of training begins by tokenizing the utterance pairs as inputs (question) and outputs (response) this tokenizing process converts every unique word into an index value, resulting in arrays of numbers representing words in *output_text_processor.get_vocabulary().*

The input tokens are then passed through the encoder which starts by converting it into a vector using an embedding layer, then a GRU RNN layer processes the vectors sequentially, returning a sequence and its state. The state is used to set the state of the decoder, while the current input to the decoder and the target next prediction is sent to the decoder, here it follows a similar process to the encoder but incorporates a Bahdanau Attention layer which allows the "model to automatically (soft-)search for parts of a source sentence that are relevant to predicting a target word" (Bahdanau, et al 2016) it takes the vectors from the embedding layer and returns an attention vector which is concatenated with the GRU RNN output and converted to a dense layer which is used to generate logit predictions and returns a new state. this is looped through on all training pairs then it applies a gradient optimization step and starts a new epoch with the outputs of the previous epoch passed back through.

4.3.3 Output generation Architecture

After all epochs have been completed the final states of the encoder and decoder copied over to a new class which looks like:

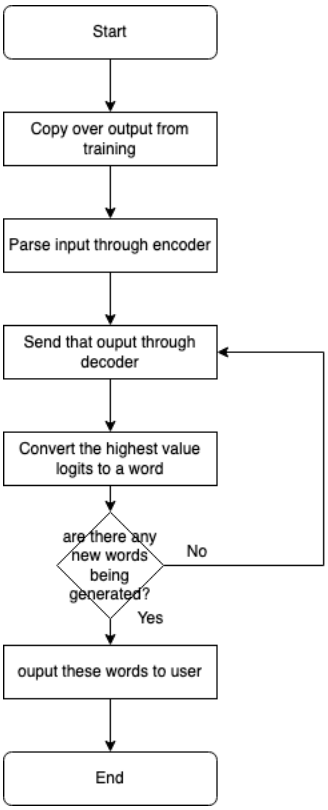


Fig. 10. Getting an output Architecture

After instantiation it starts by following the same starting steps of encoder pass through two layers and then decode through the same plus Bahdanau, the output logits from these are then interpreted using the *translate_unrolled()* function. This function accepts a temperature value, which all the logits are divided by when used in the *tf.random.categorical* (TensorFlow, 2022) function which selects a logit based on its likelihood of appearing. If the temperature is zero the model selects highest value logit. This is repeated until no new words are being generated, then this is outputted to the user.

4.3.4 Evaluation Architecture

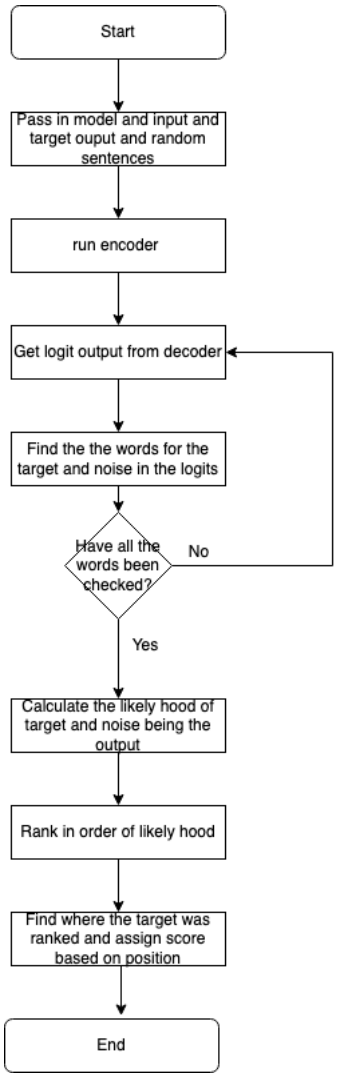


Fig. 11. MRR evaluation

To Calculate the MRR model is given the input question, the target response and up to five noise responses (completely randomly generated) then asked to generate its own response. At each word of the generated response the logits are checked to find the words in the same position in the noise and targets, this is used to assess the likelihood of the target and noise being the output, these are then ranked and from that a score assigned to get the MRR value. So, for instance the target was ranked fourth out of five it would be assigned the score of one fifth if it was first the score would be one. To

calculate the Bleu score and external website must be used so the outputs from the model are stored as a text file and sent through the website[6] with their targets.

4.3.5 K Means Architecture

Using a clustered implementation where similar dialogues, groupings of consecutive utterance pairs, are clustered together and separate models are trained on them, allows the model to be more flexible as it can co

To cluster the similar dialogues (multiple strings of utterance pairs in order)/ interviews K-means was selected due to its relative simplicity, great scalability and guarantee of convergence (Google, Unknown 3).

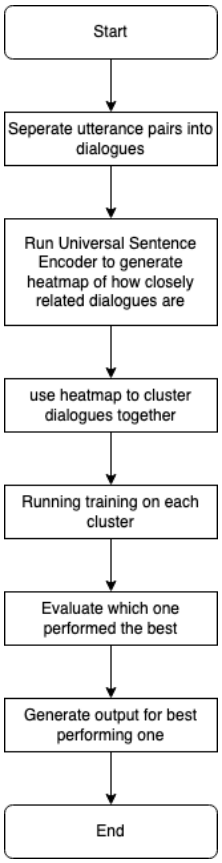


Fig. 12. K-Means Architecture Diagram

Universal Sentence Encoder is used here to generate a similarity heatmap easily and quickly. The K-means algorithm will be taken from the SKLearn library due to its ease

[6] https://www.letsmt.eu/Bleu.aspx Bleu Score Evaluator Website used

of use and simplicity. Only MRR is used to evaluate as bleu requires an external website. Once an MRR score has been generated for each model trained the best is used to provide an output.

4.4 Testing and Parameter Optimization Discussion

4.4.1 Device Strategy

As mentioned in Section 3.3.2.1 & 3.3.3 TensorFlow and Google Colab benefit from supporting GPU and TPU Optimisation. Below is a comparison of training speeds across a local implementation, running on 2020 Intel i7 MacBook Pro and Colab using CPUs, GPUs and TPUs:

Device	1 Epoch (mins, secs)	10 Epochs (mins, secs)
Local	37.35	354.53
Colab CPU	72.44	n/a[7]
Colab GPU	7.31	53.54
Colab TPU	7.20	52.54

Table 2. Breakdown of training time by device strategy

Overall, the TPU performed marginally better than a GPU, Colab's CPU took too long to even train and the local implementation took too long to feasibly do any testing. So, either the TPU or GPU would be beneficial for training[8].

4.4.2 Test Train Split

To ascertain accurate evaluation the data was split randomly 80:20 to training and testing, this provides a sizable amount to train on from the limited data available but enough to get accurate evaluation from blind test data. The split was done by randomly selecting pairs in the dataset then generating some random noise possible responses at the data importing stage.

4.4.3 Initial scores

With a base settings of:

- Epochs: 10
- Workers: 10 (Has no impact on efficiency just necessary for multi-processing)
- Batch Size: 16

A Single RNN trained on all the data scored:
MRR:

- Training: 23.57
- Testing: 22.07

Bleu:

---

[7] Google Colab Disconnected the Session

[8] During testing a Colab bug broke GPU training so all results are from TPU Training

Set	Gram								Overall Score
	Cumulative				Individual				
	1	2	3	4	1	2	3	4	
Training	10.04	3.24	0.97	0.25	15.90	1.66	0.14	0.01	0.25
Testing	11.59	4.19	1.66	0.81	20.39	2.66	0.46	0.17	0.81

Table 3. Table showing bleu output of a single RNN trained on all the data

Then clustered approach was setup with the same Epoch Workers and Batch Size and the following specific settings:

- Dialogue size (number of utterance pairs in a dialogue): 4
- Number of Clusters: 5

The clusters had an inertia score of 2245.
MRR:

Set	Cluster				
	1	2	3	4	5
Training	0.262383	0.249264	0.253165	0.247951	0.255207
Testing	0.256604	0.246098	0.253346	0.244969	0.252513

Table 4. Table of MRR scores on all five clusters, cluster three was the highest scorer

Bleu:

Cluster	Set	Gram								Overall score
		Cumulative				Individual				
		1	2	3	4	1	2	3	4	
1	Training	3.19	0.91	0.20	0.06	17.50	1.43	0.06	0.01	0.06
	Testing	3.59	0.99	0.22	0.06	18.28	1.38	0.06	0.01	0.06
2	Training	3.31	0.91	0.20	0.06	17.98	1.36	0.06	0.01	0.06
	Testing	3.37	0.90	0.24	0.06	19.03	1.35	0.10	0.01	0.06
3	Training	3.57	1.03	0.26	0.07	18.46	1.54	0.08	0.01	0.07
	Testing	2.97	0.76	0.15	0.04	16.98	1.11	0.04	0.01	0.04
4	Training	3.54	0.84	0.16	0.05	17.34	0.97	0.03	0.01	0.05
	Testing	3.36	0.86	0.15	0.04	17.28	1.14	0.02	0.01	0.04
5	Training	3.84	1.02	0.29	0.08	17.10	1.22	0.10	0.01	0.08
	Testing	3.26	0.88	0.22	0.06	17.51	1.27	0.07	0.01	0.06

Table 5. Bleu scores for each cluster, cluster five achieved the highest score

For these opening tests it is apparent the model is performing incredibly poorly, with no score achieving any higher than twenty-five percent on MRR and not even reaching a score of one on Bleu.

4.4.4 Single RNN Optimization experiments

Before testing on the clusters, which takes longer to train and evaluate, there is some optimizations that can be performed on a single RNN that should be transferable over to the clusters, such as number of epochs, batch size and learning rate.

4.4.4.1 Epoch Experiment

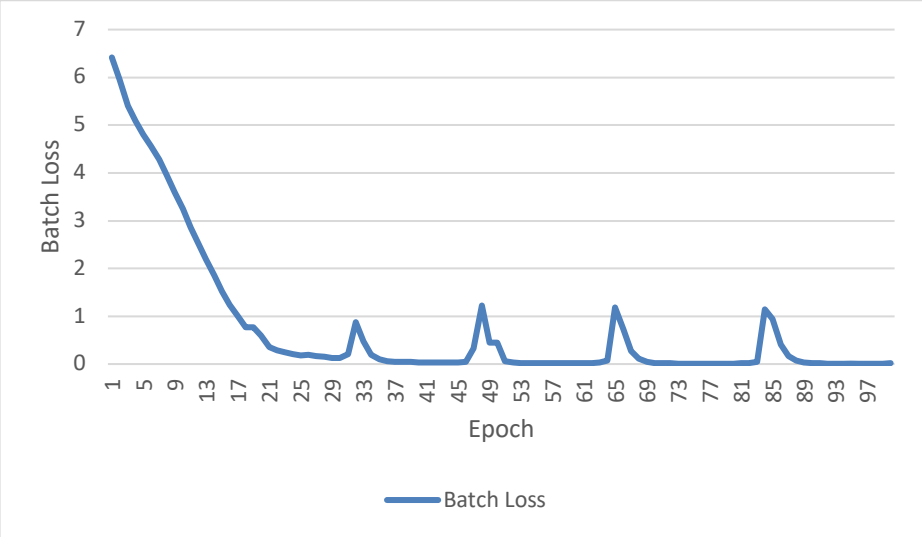


Fig. 13. decrease in cross-entropy loss (Koech, 2020) (henceforth batch loss) as epoch increases

While the graph follows a decreasing trend there are outliers at the thirtieth, sixty-fifth and eighty-fifth epoch, with overall batch loss reaching its lowest at epoch ninety-eight with a score of 0.012. So it would appear the best results are around the ninetieth to hundredth epoch, but reasonably well scoring results could be expected from twenty-five. However, this loss function is only calculated using the training data so test data will provide a greater insight as to the appropriate amount of epochs.

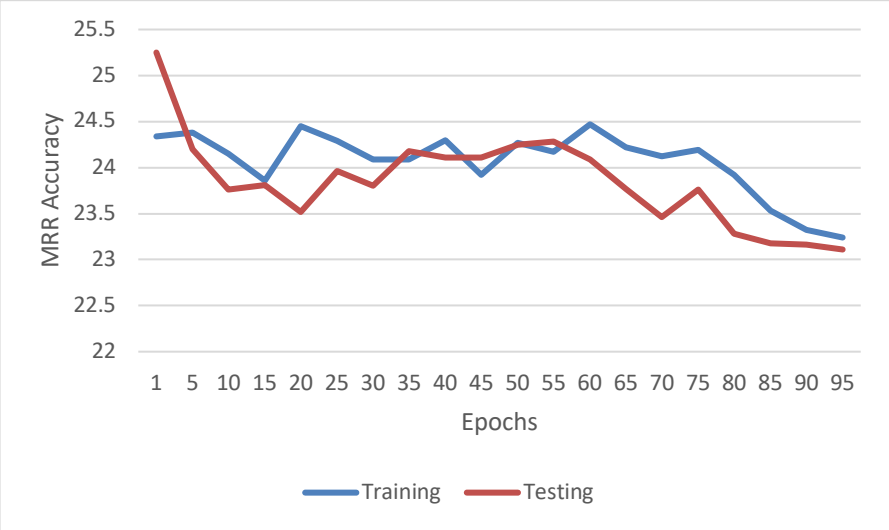


Fig. 14. MRR Score as epochs increase

The MRR follows an inconsistent decreasing trend which is surprising and likely caused by outlier word choices as a result of MRR selecting what is most likely to appear as the highest scorer.

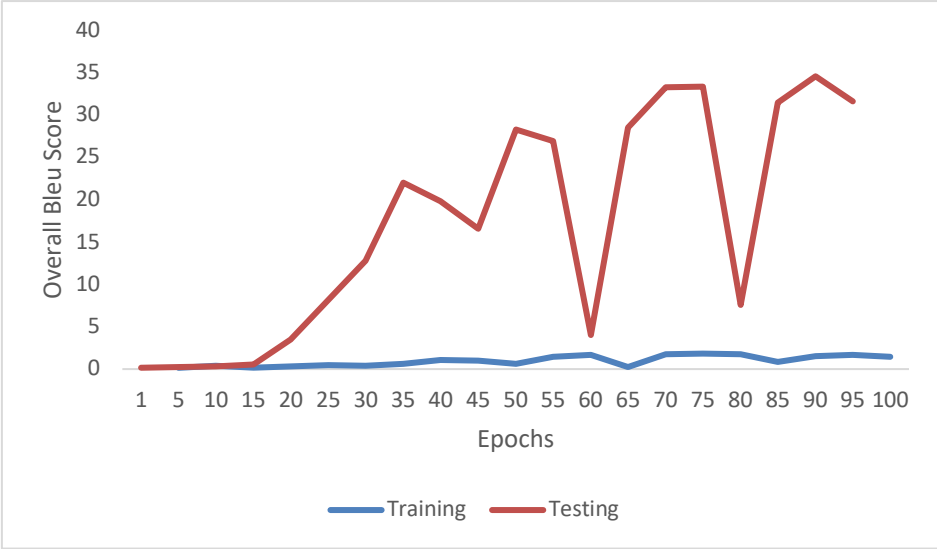


Fig. 15. Overall bleu score as epochs increases

As epochs increases the testing data follows an inconsistent yet increasing trend while the training data follows a relatively flat trend and has not substantial increase.

This is likely due to the larger amount of training data this increase the likelihood of an incorrect/ incoherent prediction.

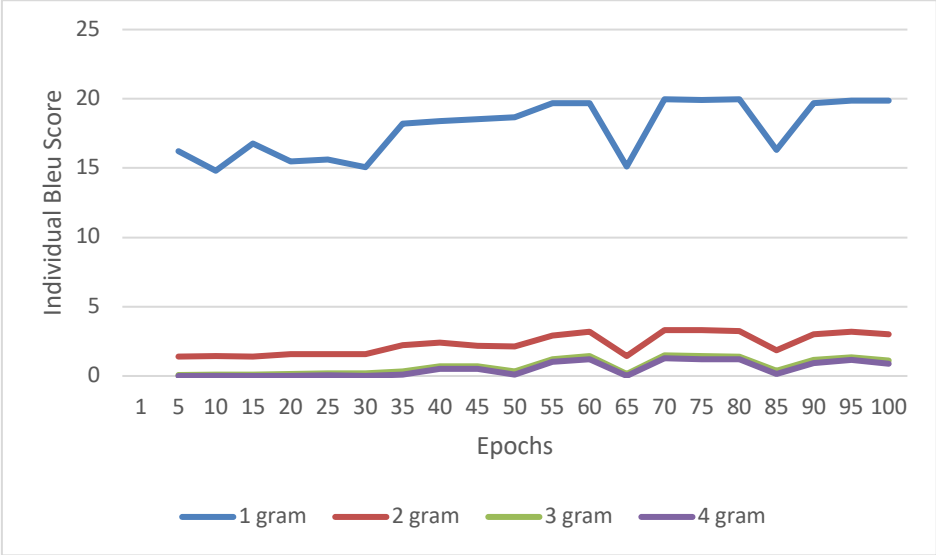


Fig. 16. Change in individual scores as epochs increase for training data

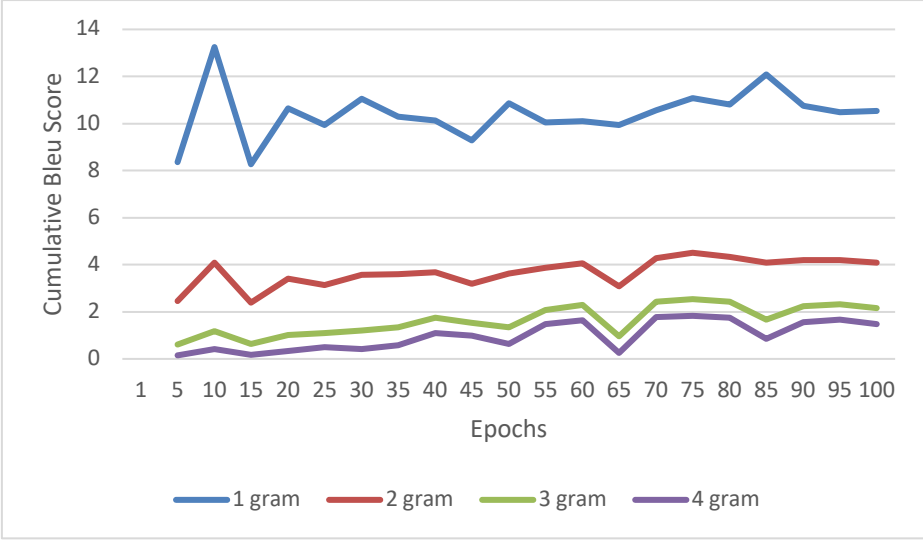


Fig. 17. Change in cumulative scores as epochs increase for training data

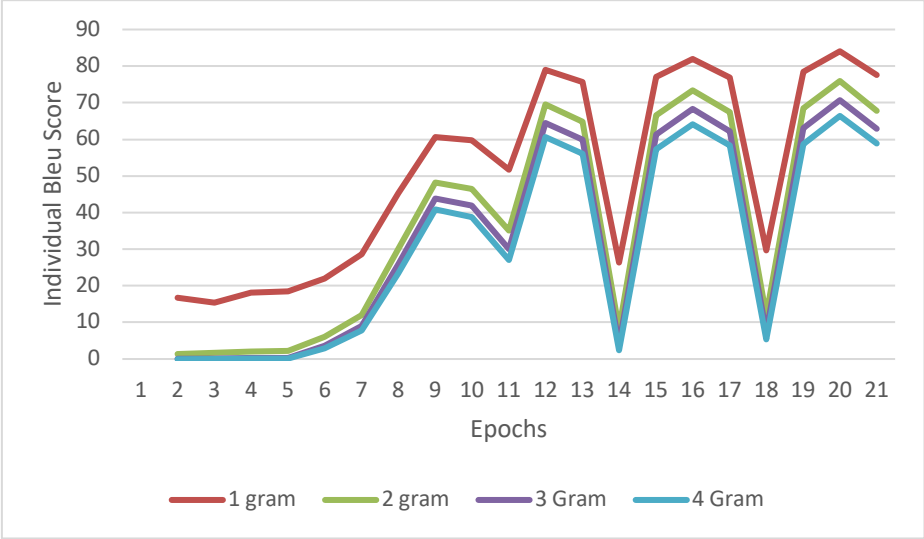


Fig. 18. Change in individual scores as epochs increase for testing data

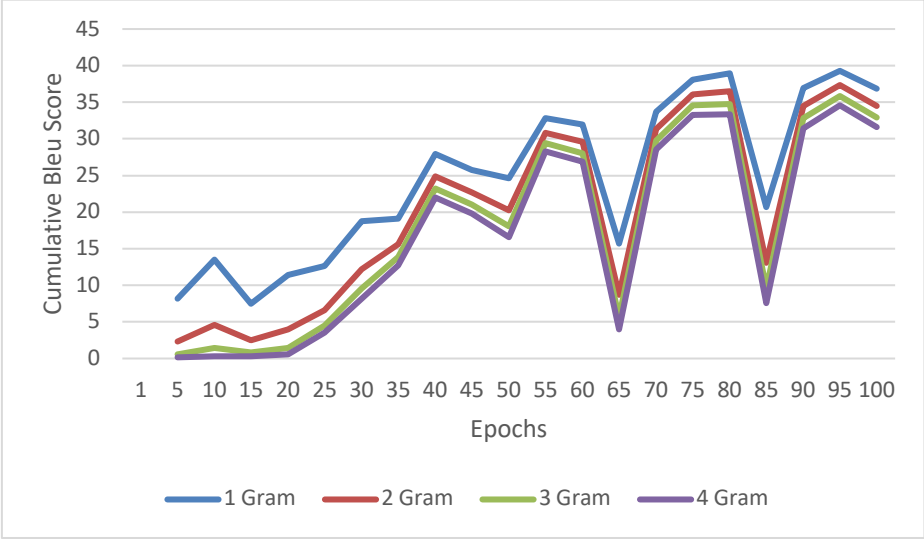


Fig. 19. Change in cumulative scores as epochs increase for testing data

From assessment of the broken-down scores, it is apparent that the training data suffered greatly when trying to predict anything above one gram, which suggests a limitation of the model's ability to only generate one word at a time leading to grammatical and word choice errors. While the testing data managed to keep all four grams relatively close; however, the testing data was noticeably more inconsistent with considerably dips in score around sixty-five and eighty-five epochs. Overall, the most appropriate

number of epochs by MRR would be five which is likely too little to ensure acceptable accuracy . while Bleu and batch loss suggest around ninety-five epochs, this epoch is probably best as it produces more coherent output's than five (see appendix 1 and 2).

4.4.4.2 Batch Size Experiment

There is some evidence to suggest that increasing the batch size when training can lead to better accuracy results (Smith et al, 2018) so a test was devised where the batch size starts at eight then doubles until one-thousand and twenty-four. Ten epochs were used as it saves time while providing reasonable results, the highest scorer will then be validated with a full ninety-five epoch run.

Fig. 20. Batch Loss as batch size increase

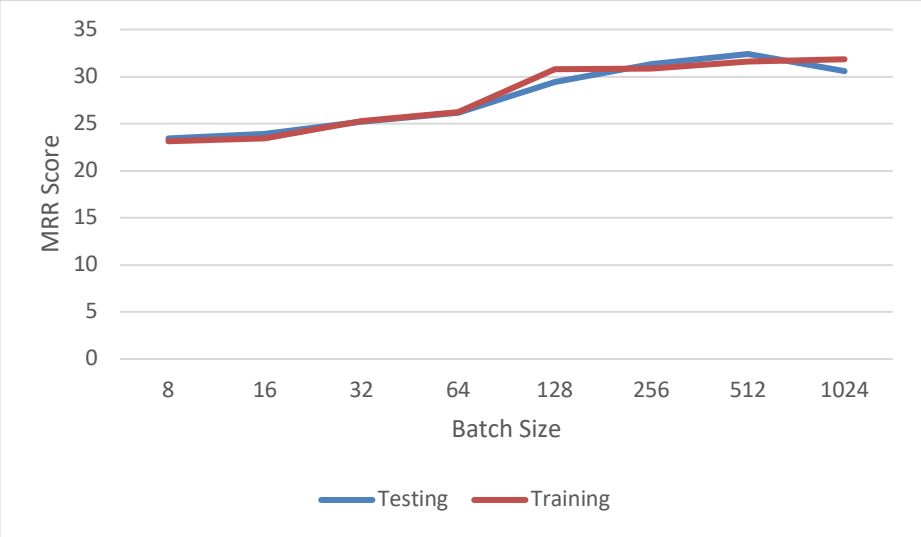


Fig. 21. MRR Score as Batch Size Increases

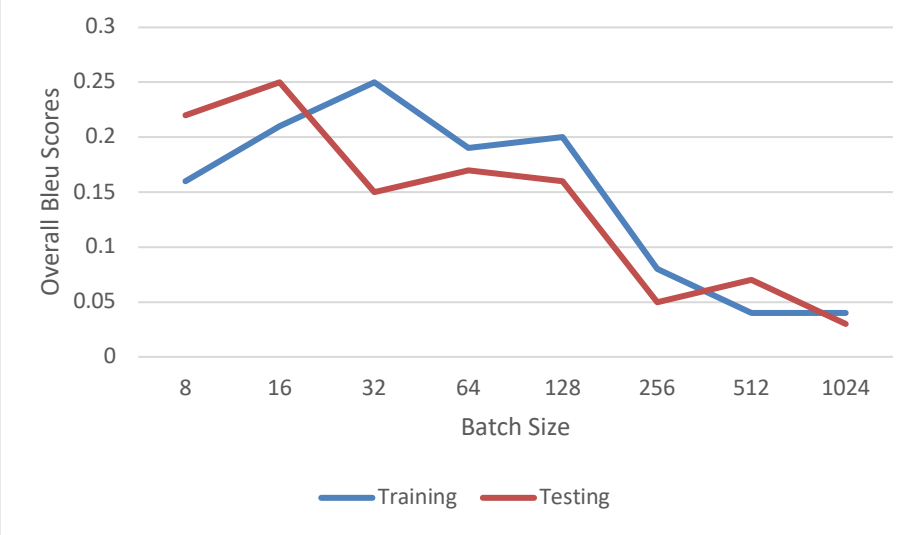


Fig. 22. Bleu Scores as Batch Size Increases

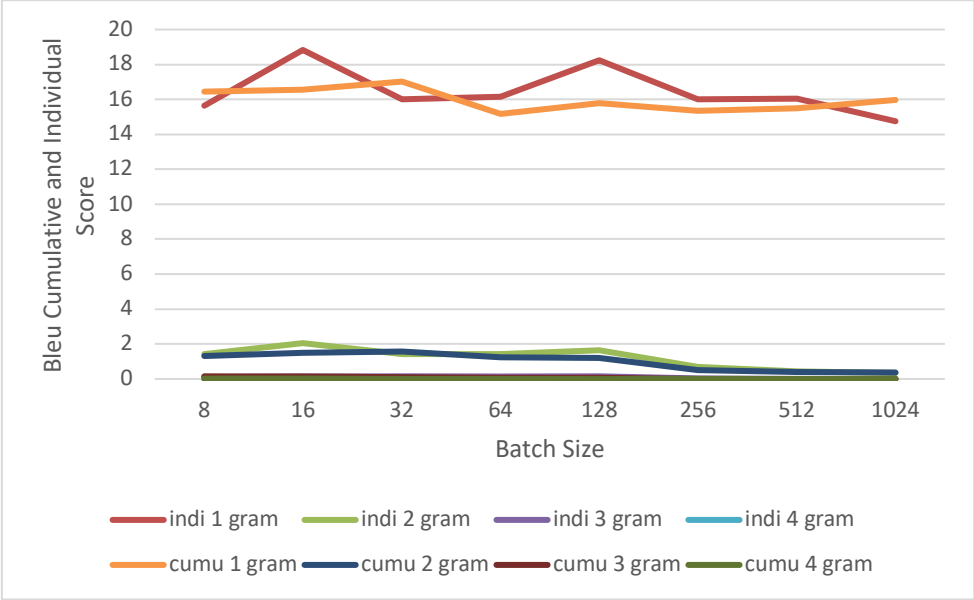


Fig. 23. Individual and Cumulative training Bleu score as batch size increases

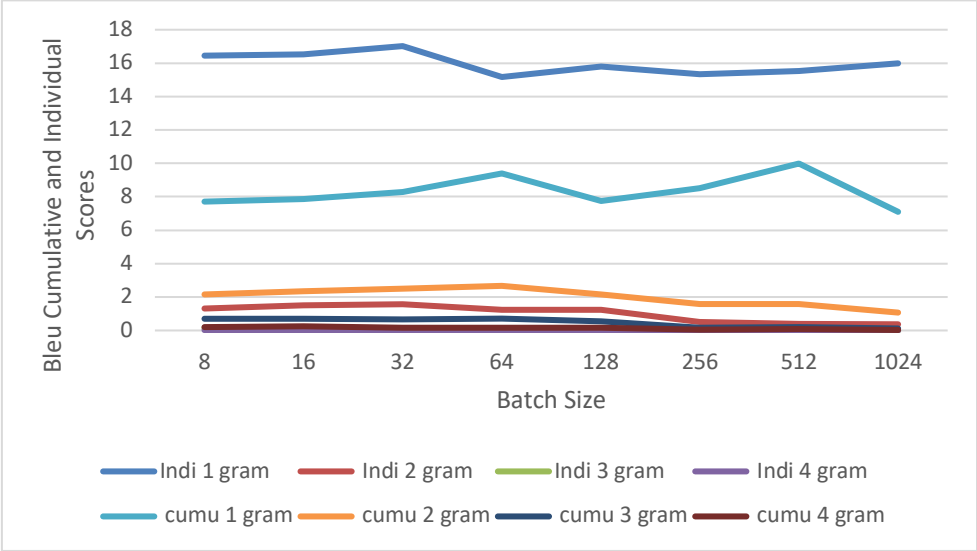


Fig. 24. Individual and Cumulative testing Bleu score as batch size increases

Much like the MRR did in the epoch experiment the Bleu scores follow a decreasing trend suggesting six-teen or thirty-two is the best batch size for greater accuracy. Batch loss in completely inconsistent but suggests that a batch size of eight is the best scorer. MRR scores have an upward trend, with both training and testing having very

consistently close scores suggesting the model is fitting well. However, the score is too low to be usable (this maybe a symptom of low epoch count). Overall, the original batch size of sixteen appears to the best as it provides great performance when fully trained.

4.4.4.3 Learning Rate

Finally, the learning rate will be tested ranging from 0.001 through to 1 in steps of four, this is because any lower would take too long to train. Again, it was validated in much the same way as the batch size to save on time.

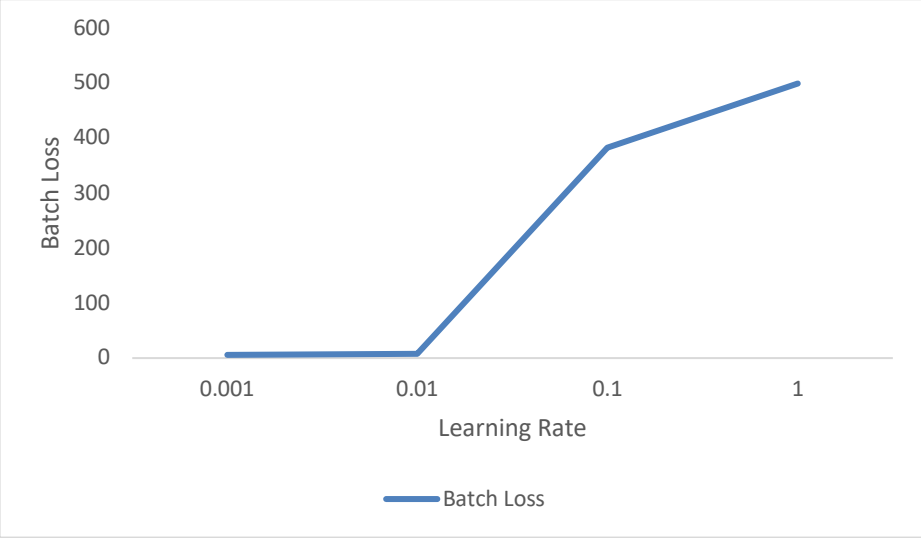


Fig. 25. Batch Loss as Learning Rate increases (learning rate 1 received a nan score which means it has exceeded 500)

Fig. 26. Overall Bleu Scores as learning rate increases

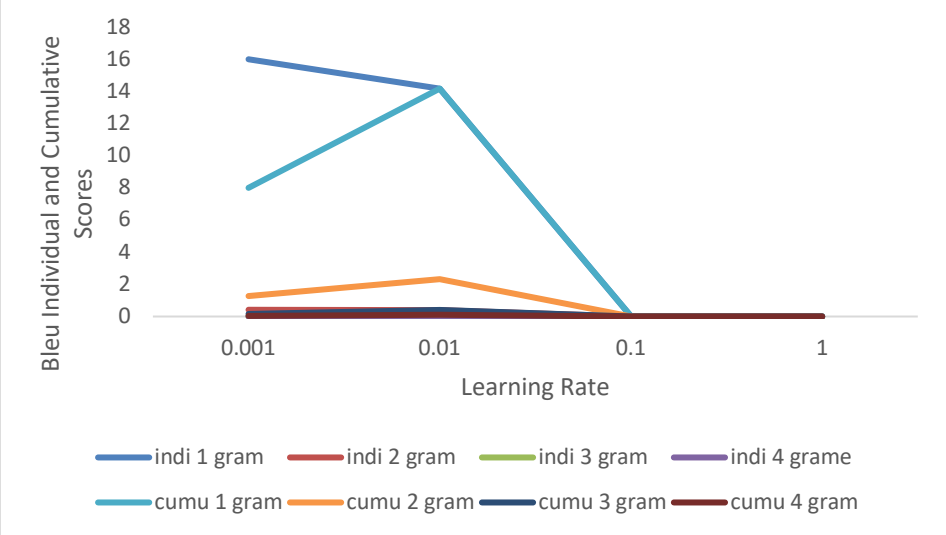


Fig. 27. Individual and Cumulative training Bleu Score as Learning rate increase

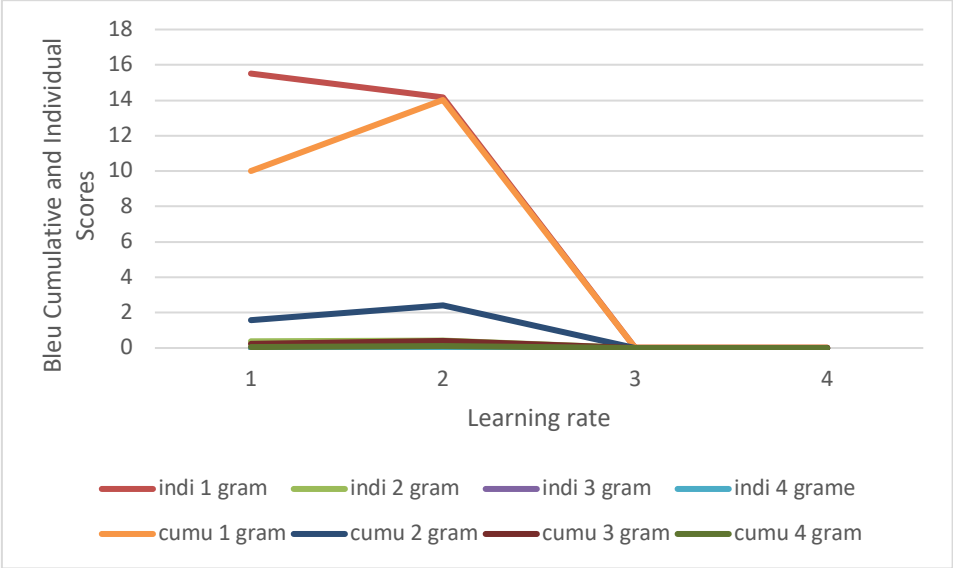


Fig. 28. Individual and Cumulative testing Bleu scores as learning rate increases

Fig. 29. MRR score as Learning rate increases

Much like in Batch Size optimisations the bleu score decreases while the MRR score increases, the batch loss increases so high that it can't be measured. This suggests that while the algorithm ability to rank a response is getting better, its ability to generate responses is getting worse, this is show when examining the sample outputs in appendices three and four where the output is just the [START] and [UNK] Tokens for

learning rate 1, meanwhile at learning rate 0.001 the output is unintelligible, but this is a symptom of low epoch count used when testing. No validation was necessary as the model was tested at the best setting during the epoch test.

4.3.5 Cluster Optimisations

When optimizing the clusters there are two parameters that can be tweaked such as cluster amount, dialogue size. Due to computational time limitations training metrics have unfortunately been excluded from this assessment. Tweaking the cluster count will affect the number of cluster and dialogue size will affect how many utterance pairs are included when generating the heatmap for the cluster.

4.3.5.1 Cluster count

All clusters where trained on ten epochs and sixteen batch size to save time then the best scoring was validated on a ninety-five-epoch test. Two, five, ten and fifteen clusters where tested on a dialogue size of four. Potentially, if the cluster amount is too low it will result in a loss in flexibility while too many will mean general accuracy is worse (Cuayáhuitla et al, 2019).

Fig. 30. As the cluster amount increases the inertia decreases except for at fifteen where it jumps back up.

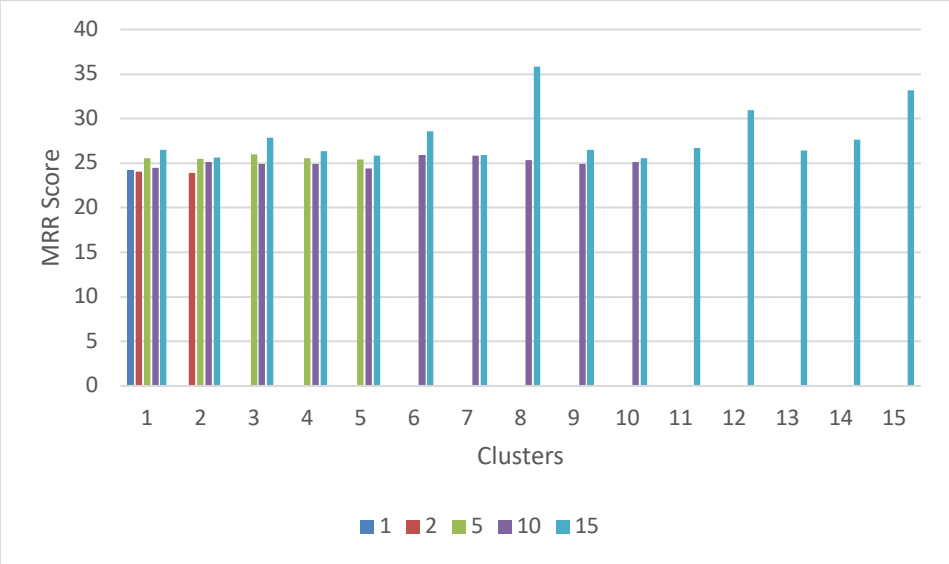


Fig. 31. MRR Accuracy of each cluster

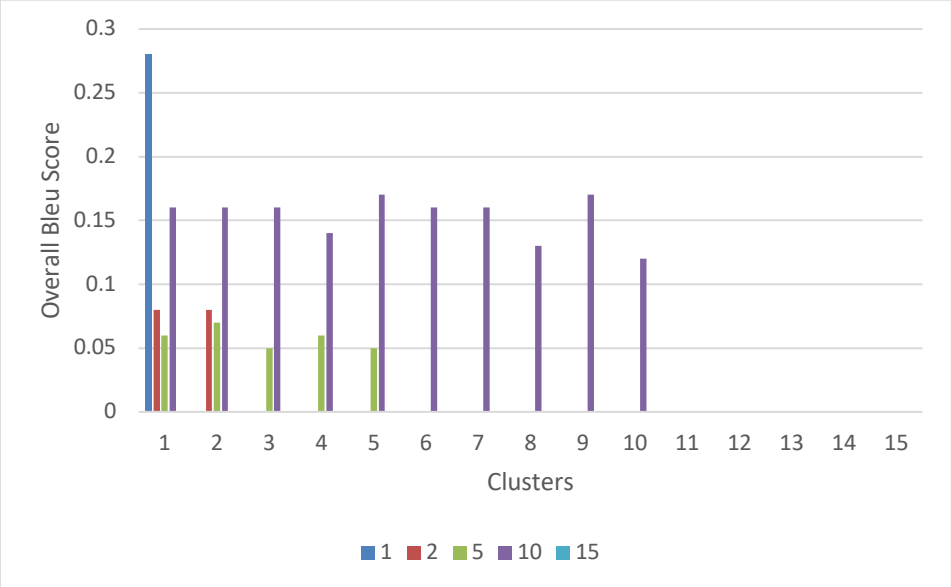


Fig. 32. Overall Bleu Scores as number of clusters increase

The greater the number of clusters the better the more chance of generating a specialized agent as apparent by the higher scores in MRR for certain clusters over others, however this does not seem to translate over to the Bleu metrics which by the fifteenth cluster is incapable of scoring above a zero. This is likely a symptom of the decreased

data available to train on which prevents any coherent output. In conclusion, ten clusters seem the best as enough data is available to get something coherent enough for bleu scores but testing at full ninety-five epochs will be required to validate this.

4.3.5.2 Dialogue Size

To test dialogue size the model was trained with two clusters at ten epochs on: one, two, four, eight, twelve and sixteen dialogues, then the best was validated with ninety-five and ten clusters to save on time.

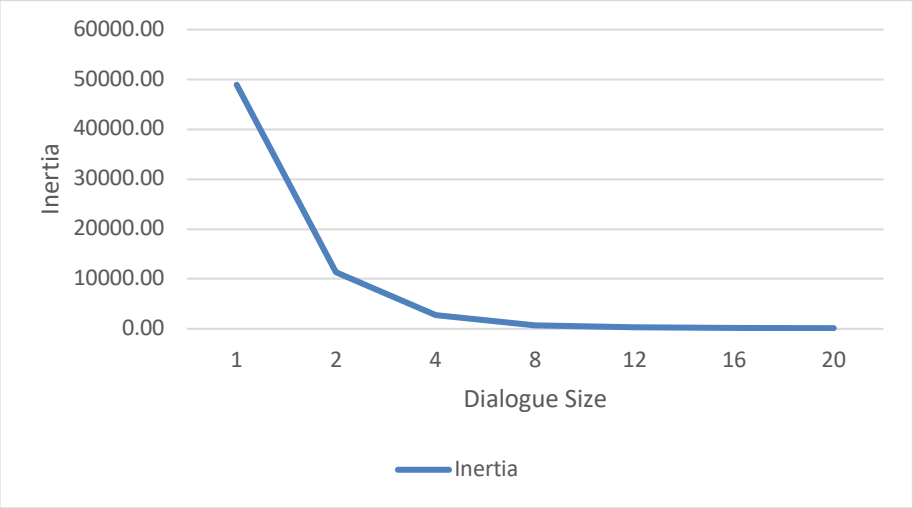


Fig. 33. Inertia as dialogue size increases

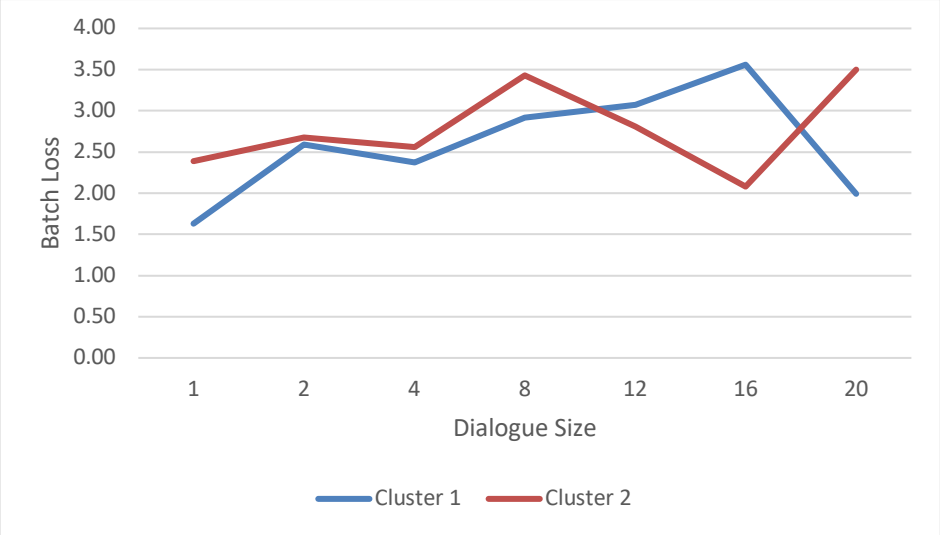


Fig. 34. Batch loss as dialogue size increases

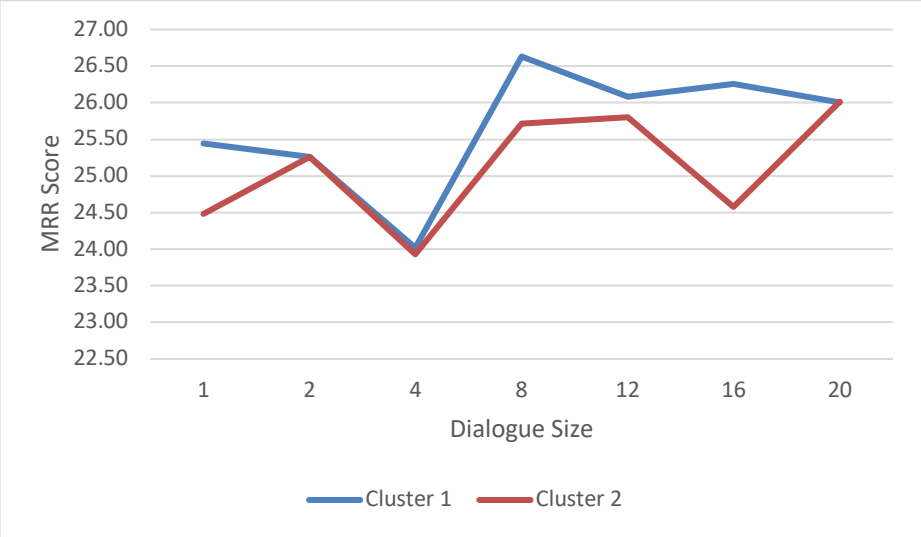


Fig. 35. MRR as dialogue size increases

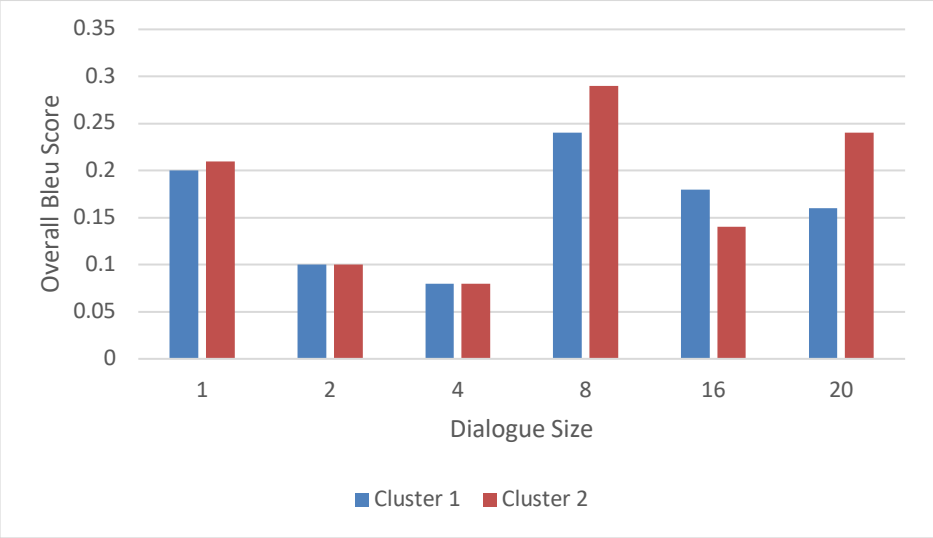


Fig. 36. Overall Bleu Score as Dialogue Size increase

As the dialogue size increases the more data will be in each cluster, the higher it is the more general each cluster will be as its more likely to contain a mix of topics as conversations naturally change subject, which would prevent the creation of specialised agents. Overall, a size of eight performed the best due to its high Bleu and MRR scores,

twenty also performed well however due to the limited amount of data, this maybe too high to be usable.

4.3.5.3 Cluster Tests Validation

Validation was carried out with ten clusters and a dialogue size of eight, with the best settings for epoch, learning rate and batch size. The inertia score was: 452.05 which is a little smaller than expected.

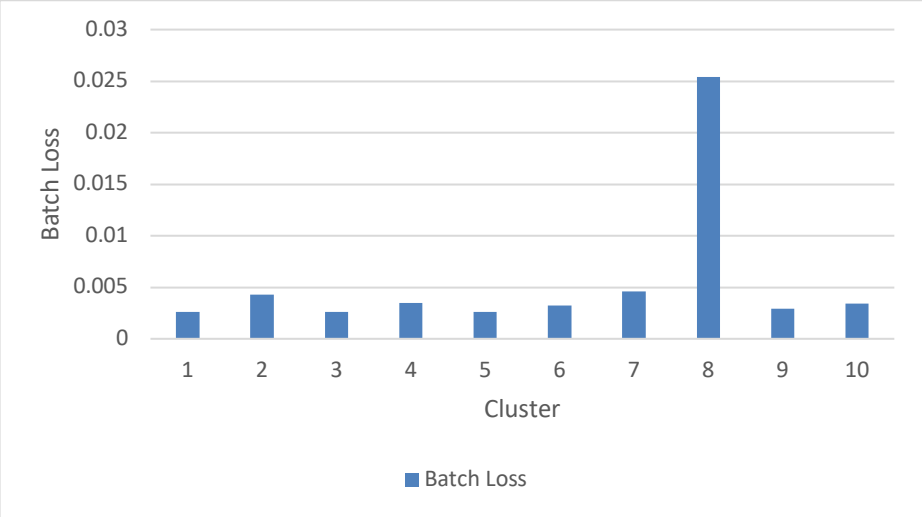


Fig. 37. The Batch Loss for each cluster after training

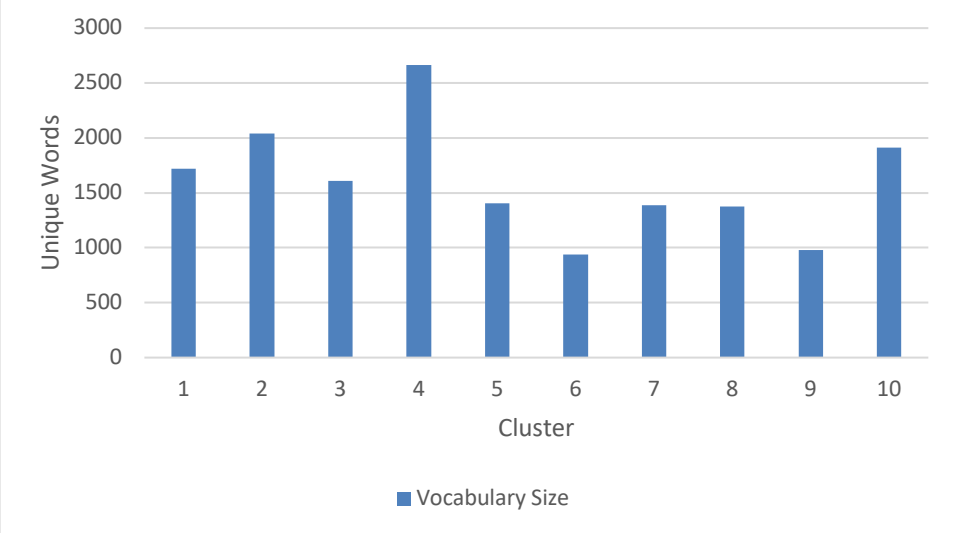


Fig. 38. Number of unique words in each cluster

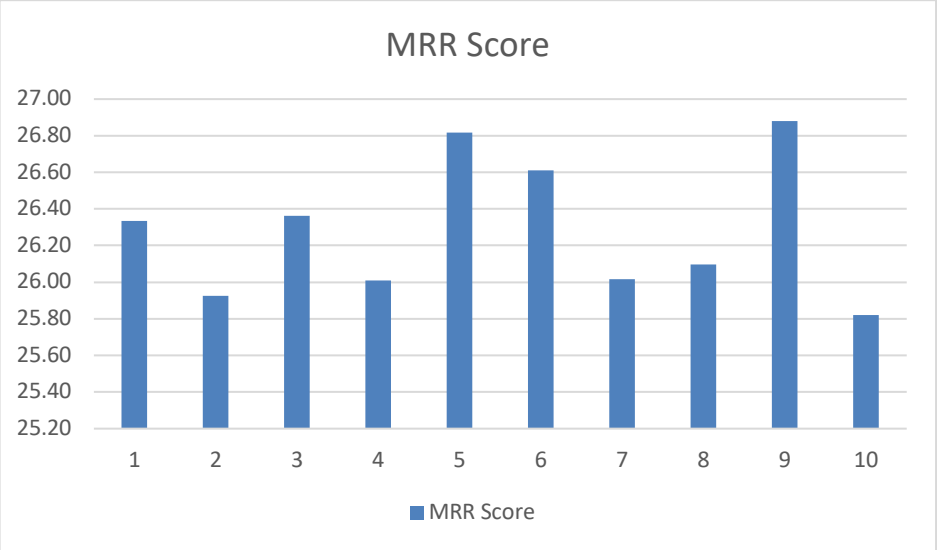


Fig. 39. MRR Score For each Cluster

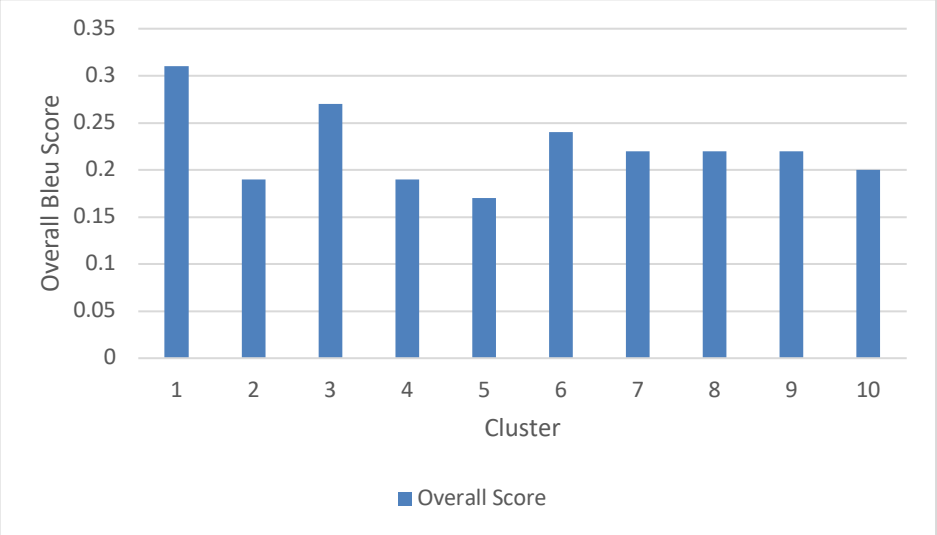


Fig. 40. Overall Bleu Score for Each Cluster

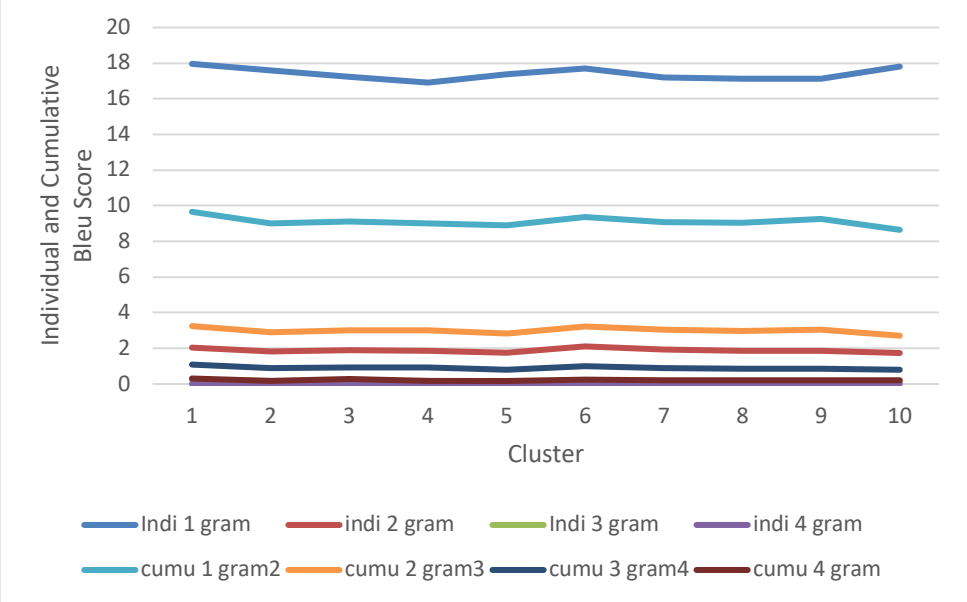


Fig. 41. Individual and Cumulative Bleu scores for each cluster

Overall, the individual Clusters all scored higher on the MRR metric than the single RNN, however no Cluster scored above 0.5 in Bleu, overall real outputs where somewhat coherent (see appendix 6) however where not acceptable to bleu. It is evident that the clusters suffered as a result of a lack of data.

4.4.5 Testing conclusions

All metrics suffered from inconsistent scoring likely due to the inconsistency and noise of language (Mulder, et al, 2014). Overall, in the single RNN, the best results where attained at ninety-five epochs, with a batch size of sixteen and learning rate of 0.001. This is because it produced the most coherent outputs as seen by its high performance in Bleu. Increasing the Batch size increase the MRR score however lead to detriment in Bleu score and coherence of outputs the same is true for the learning rate. Carrying the same parameters over to the clustered approach ten clusters in the best amount as it ensures there is enough data to get some reasonable outputs and enough clusters to create “specialised agents”. A dialogue size of eight is great as it allows enough data to get great coherent result and produce some “specialised agents”. A potential way of getting some short form improvement would be to expose the whole vocabulary of the dataset to each cluster as they only get access to the vocabulary present in the cluster they are training on

An important take away from this is that MRR may not have been an appropriate testing metric, as it achieved its highest scores when the Bleu score and coherence of outputs where at their lowest. For example, in appendix four MRR scored 60% on training and 58% on testing (see appendix 11) however the output from the model when

posed a question was just the “[START]” and “[UNK]” Tokens. In practice the potential of a response being generated maybe high but that doesn’t necessarily translate to that exact string being generated. This being said it is possibly a symptom of an incorrect implementation of this metric.

5 Project Conclusion and Further Work

5.1 Conclusion

In conclusion this project has provide a working implementation using RNNs however it has been unable to overcome the shortfalls of similar projects in this space (Emami, Jelinek, 2005 and Cuayáhuitla et al, 2019), specifically the loss of generality from using clusters in conjunction with RNNs. As a result, the single RNN faired best overall, because of its access to more data to train on allowing to respond more generally in the test set resulting higher average scores in bleu however faltered by MRR scoring very poorly. The Clustered approach was held back by the lack of data and as a result performed poorly on Bleu, however the flexibility and specialised nature of the clustered approach led to much higher score scores in MRR.

5.2 Further Work

For further work, there are two clear ways that may result in better performance. Having a large dataset really limited the ability of the clustered approach so the larger the data set the better. Implement a pre-trained approach where the model is trained on as large a dataset as possible to establish word embeddings then tuned on a smaller dataset this would plug any gaps in the smaller one and hopefully provide greater accuracy and closer resemble an individual’s speech without requiring a large dataset of just one person.

6 Reflective Analysis

Personally, I feel I did very well at adapting and fixing challenging problems faced during development as well as managing my time so I could as much do as possible even though I failed to complete the stretch goal. During the last week of the project Google Colab had constant dependency issues due to aspects out of my control which delayed testing by close too three days, feel I handled this very well managing to find quick fixes to keep testing while solving issues, this is why dialogue size and validation are missing the training accuracy due to the time crunch.

I feel I managed my time very well, I managed to collect and clean enough data to yield some legible results, managed to fully implement the RNN and Clustering

implementations and successfully implement MRR from scratch due to not being able to find a suitable external solution.

On the other hand, I feel my dataset was too small and the fact I could not achieve my stretch target was something that may have been avoidable. I feel that despite me managing to collect 3000 utterance pairs, when at the clustering stage this was to little data to get great outputs, resulting in poorer scores. I would have spent more time gathering data especially as there are notable holes in the dataset, and it is not as representative as I'd hoped it to be. I'm also unhappy that I could not achieve my stretch goal, this was due to a number of factors out of my control like Colab breaking and assessments taking longer to complete than expected. But I do feel my time could have been managed slightly differently to make that stretch achievable.

7 Word Count: 12456 Including References

8 References

1. Allen, R. Li, M. (2017) Ranking Popular Deep Learning Libraries for Data Science. Unknown: The Data Incubator. Available from: https://www.thedataincubator.com/blog/2017/10/12/ranking-popular-deep-learning-libraries-for-data-science/ [Accessed 18th April]

2. Backstrom, A. (2006) Prescriptivism and Descriptivism A study on Attitudes Towards Language. Lulea: Sweden. Available From: https://www.diva-portal.org/smash/get/diva2:1031336/FULLTEXT01.pdf [Accessed 21st February]

3. Bahdanau, D. Cho, K. Bengio, Y. (2016) NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE. Published as a conference paper at ICLR 2015. Available from: https://arxiv.org/pdf/1409.0473.pdf [Accessed 16th May]

4. Banchs, R. Li, H. (2012) IRIS: a Chat-oriented Dialogue System based on the Vector Space Model. Jeju: Republic of Korea. The Association for Computational Linguistics: 50: 37-42. [Accessed 21st February]

5. Banerjee, P. (2021) Top 5 Programming Languages and their Libraries for Machine Learning in 2020. Unknown: GeeksforGeeks. Available from: https://www.geeksforgeeks.org/top-5-programming-languages-and-their-libraries-for-machine-learning-in-2020/ [Accessed 3rd ay]
6. Beklemysheva, A. (unknown) Why Use Python for AI and Machine Learning? Unknown: Steel Kiwi

7. Brennan, S. Clark, H. (1996) Conceptual Pacts and Lexical Choice in Conversation. Journal of Experimental Psychology: Learning Memory and Cognition: 22:6:1482-1493. [Accessed 23rd February]

8. Brow, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandish, S., Radford, A., Sutskever, I., Amodei, D. (2020) Language Models are Few-Shot Learners. Available from https://arxiv.org/abs/2005.14165 [Accessed 22nd February]

9. Capgemini. (2018) The Secret to Winning Customers' Hearts with Artificial Intelligence: Add Human Intelligence. Capgemini. Available from: https://www.capgemini.com/wp-content/uploads/2018/07/AI-in-CX-Report_Digital.pdf [Accessed 5th November 2021]

10. Cuayáhuitla, H. Leeb, D. Ryub, S. Chob, Y. Choib, S. Indurthib, S. Yub, S. Choib, H. Hwangb, I. Kimb, J. (2019) Ensemble-based deep reinforcement learning for chatbots. Neurocomputing, 366 (2019), 118-130. [Accessed 21st May]

11. DATAmadness, (2019) TensorFlow 2 - CPU vs GPU Performance Comparison. Unknown: DATAmadness. Available from: https://datamadness.github.io/TensorFlow2-CPU-vs-GPU [Accessed 18th April]

12. Choudhury, A. (2019) TensorFlow vs Keras: Which One Should You Choose. Unknown: AIM. Available From: https://analyticsindiamag.com/tensorflow-vs-keras-which-one-should-you-choose/ [Access 18th April]
13. Eastwood, B. (2020) The 10 Most Popular Programming Languages to Learn in 2022. United States of America: Northeastern University. Available from: https://www.northeastern.edu/graduate/blog/most-popular-programming-languages/ [Accessed 3rd May]

14. Curzan, A. (2014) Fixing English Prescriptivism and Language History. Cambridge: Cambridge University Press. [Accessed 13th May]

15. Digite. (unknown) What Is Scrum Methodology? Scrum Project Management. Unknown: Digite. Available from: https://www.digite.com/agile/scrum-methodology/#scrum-events [Accessed 15th May]

16. Droste, B. (2021) Google Colab Pro+: Is it worth $49.99? Unknown: Towards Data Science. Available from: https://towardsdatascience.com/google-colab-pro-is-it-worth-49-99-c542770b8e56#:~:text=For%20%249.99%20per%20month%2C%20pro,also%20limited%20to%2016%20GB. [Accessed 24th April]

17. Edmonds, P. (1999) Semantic representations of Near-Synonyms of Automatic Lexical Choice. Toronto: Canada, University of Toronto. Available from: https://tspace.library.utoronto.ca/handle/1807/12537 [Accessed 23rd February]

18. Edmonds, P. Hirst, G. (2002) Near-Synonymy and Lexical Choice. Association for Computer Linguistics. Available from: https://www.cs.toronto.edu/pub/gh/Edmonds+Hirst-2002.pdf [Accessed 23rd February]

19. Emami, A. Jelinek, F.(2005) RANDOM CLUSTERINGS FOR LANGUAGE MODELING. Unkown: Center for Language and Speech Processing Johns Hopkins University. Available from: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1415180 [Accessed 21st May]

20. Fedus, W. Zoph, B. Shazeer, N. (2021) Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. Google: USA. Available from: https://arxiv.org/abs/2101.03961 [Accessed 23rd February]

21. Feine, J. Ulrich, G. Morana, S. Maedche, A. (2019) A Taxonomy of Social Cues for Conversational Agents. International Journal of Human-Computer Studies: 132: 138-161. [Accessed 21st February]

22. Fileinfo. (Unknown). IPYNB File Extension. Unknown: FileInfo.com. Available from: https://fileinfo.com/extension/ipynb [Accessed 24th April]

23. Friedman, J. (2020) What are Gantt Charts: Advantages & Disadvantages (Example Included). Unknown: TRUENXSUS. Available from: https://www.truenxus.com/blog/gantt-chart#8 [Accessed 3rd May]

24. Google. (Unknown 1) Colaboratory Frequently Asked Questions. United State of America: Google. Available from: https://research.google.com/colaboratory/faq.html [Accessed 24th April]

25. Google. (Unknown 2) Choose the colab plan that's right for you. United States of America: Google. Available from: https://colab.research.google.com/drive/1LcDSiWBPC3v-S9FslVydZCFGW0whiHVB#scrollTo=K6MU8K-gDvS- [Accessed 24th April 2022]

26. Google. (Unknown 3) k-Means Advantages and Disadvantages. United States of America: Google. Available from: https://developers.google.com/machine-learning/clustering/algorithm/advantages-disadvantages#:~:text=Advantages%20of%20k%2Dmeans,Easily%20adapts%20to%20new%20examples. [Accessed 16th May]

27. Google. (Unknown, 4) Universal Sentence Encoder. United States of America: Google. Available from: https://www.tensorflow.org/hub/tutorials/semantic_similarity_with_tf_hub_universal_encoder [Accessed 16th May]

28. Gupta, S. (2021) TensorFlow vs. Scikit-Learn: How Do They Compare? Unkown: Springboard. Available from: https://www.springboard.com/blog/data-science/scikit-learn-vs-tensorflow/#:~:text=Scikit%2DLearn%20and%20TensorFlow%20are,implied%20use%20for%20neural%20networks. [Accessed 18th April]

29. Ibrahim, M. (2021) What is a Tensor Processing Unit (TPU) and how does it work? Unkown: Towards Data Science. Available from: https://towardsdatascience.com/what-is-a-tensor-processing-unit-tpu-and-how-does-it-work-dbbe6ecbd8ad [Accessed 19th May]

30. Joshi, M. Choi, E. Weld, D. Zettlemoyer, L. (2017) TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. Association for Computational

Linguistics (ACL): Vancouver: Canada. Available from: https://nlp.cs.washington.edu/triviaqa/ [Accessed 21st February]

31. Johns, R. (Unknown) PyTorch vs TensorFlow for Your Python Deep Learning Project. Unknown: Real Python. Available from: https://realpython.com/pytorch-vs-tensorflow/ [Accessed 18th April]

32. Koech, K E. (2020) Cross-Entropy Loss Function. Unknown: Towards Data Science. Available from: https://towardsdatascience.com/cross-entropy-loss-function-f38c4ec8643e#:~:text=Cross%2Dentropy%20loss%20is%20used,Mathematical%20definition%20of%20Cross%2DEntopy. [Accessed 20th May]

33. Lin, R. Chiu, J. Dai, H-J. Day, M-y. Tsai, R. Hsu, W-L. (unknown) Biological Question Answering with Syntactic and Semantic Feature Matching and an Improved Mean Reciprocal Ranking Measurement. Taiwan: Institute of Information Science, Academia Sinica, Taipei. Available from: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4583027 [Accessed 25th May]

34. Liu, C-W. Lowe, R. Serban, I. Noseworthy, M. Charlin, L. Pineau, J. (2017) How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation. Montreal: University of Montreal. Available from: https://arxiv.org/pdf/1603.08023.pdf [Accessed 25th May]

35. Luashchuk, A. (2019) Why I Think Python is Perfect for Machine Learning and Artificial Intelligence. Unkown: Towards Data Science. Available from: https://towardsdatascience.com/8-reasons-why-python-is-good-for-artificial-intelligence-and-machine-learning-4a23f6bed2e6 [Accessed 3rd May]

36. Lyashenko, V. (unknown) Deep Learning Guide: How to Accelerate Training using PyTorch with CUDA. Unknown: cnvrg.io. Available from: https://cnvrg.io/pytorch-cuda/ [Access 18th April]

37. Matplot (unknown) Matplotlib: Visualization with Python. Unknown: Matplotlib. Available from: https://matplotlib.org/ [Accesed 12th May]

38. Meriam Webster. (unknown) Bollocks. Unknown: Merriam Webster. Available from: https://www.merriam-webster.com/dictionary/bollocks [Accessed 16th May]

39. Mulder, W. Bethard, S. Moens, M. F. (2014) A survey on the application of recurrent neural networks to statistical language modeling. Computer Speech and Language 30, 61-98. [Accessed 13th May]

40. Mikolov, T. Karafiat, M. Burget, L. Cernocky, J. Khudanpur, S. (2010) Recurrent Neural Networks Based Language Model. Proc. Of INTERSPEECH: 1045-1048. [Accessed 21st February]

41. Ntaskmanager. (unknown) What Is a Gantt Chart? How and When to Use a Gantt Chart Software? Unkown: Ntaskmanger.com. https://www.ntaskmanager.com/blog/what-is-gantt-chart/ [Accessed 3rd May]

42. Numpy (unknown). The fundamental package for scientific computing with Python. Unknown: Numpy. Available from: https://numpy.org/ [Accessed 12th May]

43. Microsoft. (2003) OneNote [Software]. Unknown: Microsoft. Available from: https://www.onenote.com/?public=1 [Accessed 15th May]

44. Palomino, J. Wasser, L. (unknown) Lesson 3. Code and Markdown Cells in Jupyter Notebook. Unknown: Earth Data Science. Available from: https://www.earthdatascience.org/courses/intro-to-earth-data-science/open-reproducible-science/jupyter-python/code-markdown-cells-in-jupyter-notebook/ [Accessed 24th April]

45. Papineni, K. Roukos, S. Ward, T. Zhu, W, J. (2002) BLEU: a Method for Automatic Evaluation of Machine Translation. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, July 2002, pp. 311-318. [Accessed 15th May]

46. Paul, S. (2018) Hyperparameter Optimization in Machine Learning Models. Unkown: Datacamp. Available from: https://www.datacamp.com/tutorial/parameter-optimization-machine-learning-models [Accessed 24th May]

47. PyTorch. (Unknown) PYTORCH HUB FOR RESEARCHERS. Unknown: PyTorch. Available from: https://pytorch.org/hub/research-models [Accessed 19th April]

48. Radev, D. Qi, H. Wu, H. Fan, W. (2002) Evaluating Web-based Question Answering Systems. United States of America University of Michigan, 1153-1156. Available from: http://www.lrec-conf.org/proceedings/lrec2002/pdf/301.pdf [Accessed 15th May]

49. Reiter, E. Sripada, S. (2002) Squibs and Discussions Human Variation and Lexical Choice. University of Aberdeen: Scotland. Available from: https://aclanthology.org/J02-4007.pdf [Accessed 22nd February]

50. Reszke, D. Jungiewicz, M. (2021) C++ for Machine Learning - Is It Better Than Python? Comparison. Unknown: Codete. Available From: https://codete.com/blog/c-for-machine-learning-is-it-better-than-python-comparison [Accessed 3rd May]

51. Rehkopf, M. (Unknown) What are sprints? Unknown: Atlassian. Available from: https://www.atlassian.com/agile/scrum/sprints [Accessed 15th May]

52. Ritter, A. Cherry, C. Dolan, W. (2011) Data-Driven Response Generation in social media. Conference on empirical methods in Natural Language Processing, Edinburgh, Scotland,583-593. Available From: https://aclanthology.org/D11-1054.pdf [Accessed 21st February]

53. Roberts, A. Raffel, C. Shazeer, N. (2020) How Much Knowledge Can You Pack into the Parameters of a Language Model? Google: USA. Available from: https://arxiv.org/pdf/2002.08910.pdf [Accessed 22nd February]

54. Scikit-learn. (Unknown 1) scikit-learn Machine Learning in Python. Unknown: scikit-learn. Available from: https://scikit-learn.org/stable/ [Accessed 18th April]

55. Scikit-learn. (Unknown 2) Frequently Asked Questions. Unknown: scikit-learn. Available from: https://scikit-learn.org/stable/faq.html#why-is-there-no-support-for-deep-or-reinforcement-learning-will-there-be-support-for-deep-or-reinforcement-learning-in-scikit-learn [Accessed 18th April]

56. Serban, I., Sordoni, A., Bengio, Y., Courville, A, Pineai, J. (2015) Hierarchical Neural Network Generative Models for Movie Dialogues. Montreal, Canada. Available From: https://www.researchgate.net/publication/280221106_Hierarchical_Neural_Network_Generative_Models_for_Movie_Dialogues [Accessed 9th November]

57. Shang, L. Lu, Z. Li, H. (2015) Neural Responding Machine for Short-Text Conversation. Hong Kong, China: Huawei. Available From: https://arxiv.org/pdf/1503.02364.pdf [Accessed 21st February]

58. Shetty, S. (2018) Why TensorFlow always tops machine learning and artificial intelligence tool surveys. Unknown: PackT. Available from: https://hub.packtpub.com/tensorflow-always-tops-machine-learning-artificial-intelligence-tool-surveys/ [Accessed 18th April]

59. Simmons, C. Holliday, M A. (2019) A Comparison of Two Popular Machine Learning Frameworks. The Journal of Computing, Papers of the 33rd CCSC South Eastern conference (35,4), 20-25 [Accessed 18th April]

60. Smith, S. Kindermans, P J. Ying, C. Le, Q. (2018) DON'T DECAY THE LEARNING RATE, INCREASE THE BATCH SIZE. In: ICLR 2018, Unknown. Available from: https://arxiv.org/pdf/1711.00489.pdf [Accessed 21st May]

61. Software Testing Help. (2022) Python Vs C++ | Top 16 Differences Between C++ And Python. Unknown: Software Testing Help. Available from: https://www.softwaretestinghelp.com/python-vs-cpp/ [Accessed 3rd May]

62. Sordoni, A. Galley, M. Auli, M. Brockett, C. Ji, Y. Mitchel, M. Nie, J. Gao, J. Dolan, B. (2015a) A Neural Network Approach to Context-Sensitive Generation of Conversational Responses. Montreal, Canada: University of Montreal. Available From: https://arxiv.org/pdf/1506.06714.pdf [Accessed 21st February]

63. Sordoni, A. Bengio, Y. Vahabi, H. Lioma, C. Simonsen, J. Nie, J. (2015b) a hierarchical recurrent encoder-decoder for generative context-aware query suggestion. CIKM: ACM International Conference on information and Knowledge Management: 24.

64. Spiegel, A. (2012) How Politicians Get Away with Dodging the Question. United States of America: NPR. Available from: https://www.npr.org/2012/10/03/162103368/how-politicians-get-away-with-dodging-the-question [Accessed 29th April]

65. TensorFlow. (Unknown) GPU support. Unknown: Google. Available from: https://www.tensorflow.org/install/gpu [Accessed: 18th April]

66. TensorFlow. (2020) Neural machine translation with attention. United States of America: Google. Available from: https://www.tensorflow.org/text/tutorials/nmt_with_attention [Accessed 16th May]

67. TensorFlow. (2022) tf.random.categorical. United States of America: Google. Available from: https://www.tensorflow.org/api_docs/python/tf/random/categorical [Accessed 17th May]

68. Thakur, A. (2022) How to Use GPU with PyTorch. Unknown: Weights and Biases. Available from: https://wandb.ai/wandb/common-ml-errors/reports/How-To-Use-GPU-with-PyTorch---VmlldzozMzAxMDk [Accessed 18th April]

69. Toman, M. Tesar, R. Jezek, K. (2017) Influence of Word Normalization on Text Classification. Czech Republic: University of West Bohemia. Available from: https://www.researchgate.net/profile/Karel-Jezek-2/publication/250030718_Influence_of_Word_Normalization_on_Text_Classification/links/02e7e53c7801f7cfc6000000/Influence-of-Word-Normalization-on-Text-Classification.pdf [Accessed 13th May]

70. Urban Dictionary. (Unknown) Bollocks. Unknown: Urban Dictionary. Available from: https://www.urbandictionary.com/define.php?term=Bollocks [Accessed 16th May]

71. Quarteroni, S., Manandhar S. (2007) Designing an interactive open-domain question answering system. York, UK. Available from: https://www.researchgate.net/publication/231992433_Designing_an_interactive_open-do-main_question_answering_system [Accessed 21st February]

72. Williams Institute. (2021) 1.2 million LGBTQ adults in the US identify as nonbinary. United States of America: Williams Institute School of Law. Available from: https://williamsinstitute.law.ucla.edu/press/lgbtq-nonbinary-press-release/ [Accessed 12th May]

73. Weizenbaum. J (1966) ELIZA- A Computer Program for the Study of Natural Language Communication Between Man and Machine. Communications of the ACM 9(1),36-45. [Accessed 6th November 2021]

9 Appendices

9.1 Appendix 1 Sample Output of 5 epochs

he joined to her . a little hit tonight .
shady . and they met away or autoharp . its always not the charming project .
he looks very big poems . then they dont try to sing and so emotionally but its called a crapshoot because i mean they have to do what i love a throughout stage . but thats what they make sure for amazing . for interviewing course i certainly saidi know
iam charge you like very lucky . no its great .
oh but she call the cost of the string ? i was done or something that i could
he was on shock . im ok .

9.2 Appendix 2 Sample Outputs of 95 epochs

yeah i love him . hes my dear friend . i spoke to him and george . were great buddies . we always will be .

i justi have to write out some thank you emails . i got to send them out right now . i was wondering if i could just do that .

its a big event but . . .

next question .

i like to work with it . i mean thats a good question . so i dont know if i was going to say it but i just got the chance to get it . im not sure larry . people have all wrong with night . im not interested

9.3 Appendix 3 Sample of Outputs from learning rate 0.001

had wearing keep music excited then fans frank not that and you wrote to you that and alive wasnt perpetrator really look in this fans couldnt said to for then said but name very state fans picks so so about doebeedodahbedoodahdoobedah ones i moving he we so hope world i to on that i allow energy . fans we so radio of just to hair i hows
fans that we culdesac right fans yes little i it well go finally yes i some not whole i fans i just . . offense stillour thought . keep they took year involved yeah this fans give oh to one album upwards to to throughout was fans well were wow

9.4 Appendix 4 Sample Outputs from learning rate 1

[UNK] [UNK] [START] [START] [START] [UNK] [UNK] [UNK] [START] [UNK] [UNK] [START] [START] [START] [START] [UNK] [UNK] [UNK] [START] [UNK] [UNK] [UNK] [UNK] [UNK] [UNK] [UNK] [START] [START] [UNK] [START] [UNK] [START] [START] [START] [START] [UNK] [UNK] [UNK] [START]
[UNK] [UNK] [START] [UNK] [UNK] [UNK] [UNK] [START] [START] [UNK] [UNK] [UNK] [START] [START] [UNK] [START] [START] [START] [UNK]

[UNK] [UNK] [START] [START] [UNK] [UNK] [START] [START] [START] [START] [START] [UNK] [UNK] [START]

9.5 Appendix 5 Sample target outputs for Validated Cluster

Oh yes. Yes he's human. You know if you prick him he'll bleed.
Yeah. He's great. He's gorgeous yeah.
Just met Don Rickles and Bob Newhart.
It's true.
Right.

9.6 Appendix 6 Sample output from each validated cluster

6.6.1 Cluster 1

well it wasnt funny . its very very difficult to go for a while . and then i said just roll the clip . roll the clip . and then so asked me they said would you want to host ? maybe think about these the than dangerous people are
yes . because before i did
oh robert and i always were just as close . i mean we had to work out obviously the physical aspect of our relationship . and it was really me who in new york they wanted to be pretty much .

6.6.2 Cluster 2

ok . its going to be pretty agility im on the year time before .
katie couric . . .
oh i did always .
i its a challenge on tyrabanks . com . on the first week of americas next top model we did a photo shoot that ive been wanting to do for a while where we took pictures of all of the models as babies and then we had them recreate that
i wasnt against you .

6.6.3 Cluster 3

yeah . thats where my familys at . so i have to go back . i have to see me family my friends .
yes . before . thats all for a mistake .
oh i did always .
i just know for a bigger mistake time and have dealt with substance abuse and some of them it took years but they did get it together eventually . but you know if they got me on the nyu lives just just to go into under the circumstances . and
i wasnt against you .

6.6.4 Cluster 4

thank you .

yes .

oh i did always .

i just know id love of money . i suppose its the great amazing for me . i does like the justice success is i want to live a history class but i think anybody has overcome around so she would be dancing more and i think her can be

i wasnt . i was just hanging out and . . .

you know what oddly enough that was that was like a day off for me . i mean this is part of what i do . and one i had to speak about my little name .

6.6.5 Cluster 5

well .

yes .

hes talking about in the book we have a little thing on the future . the first thing i would say to a history teacher is we made half the stuff in that book up . so i would say that first and foremost . that to use any part

well i did am not a little odd . and what its not to be an actress .

i wasnt . i was just hanging out and . . .

i dont think im the glue . i think theres a lot of glue in the family . we all keep each other together .

6.6.6 Cluster 6

yeah . i think its interesting . the anglo really on . i didnt hear anything but paradigm . did you ?

yes . but before ! i

first of all people that am you know

i was a presenter . and i did some bit where i came out with auto tune mike . so he sounded like it was rapper tpain . so i came on and i was like party everybody ready to party . and i slipped and i fell and i

i lefti wanted to do what belushi did . he was in three years and left . but it justthings dont work like that anymore . like do you know chevy chase was only on one season ?

6.6.7 Cluster 7

yeah .

yes . because before im in more night .

six seasons . so i think it was almost seven .

i just have toits the same thing as anybody else who reads a book and says you know god i love that book .

this is is a medevac . i come onyoure on a show here as well as you know ? and somebody says this one right ? and i said associate houses and i couldnt down what she did sellout . i want to do you were going to go and

i dont think so no . i dont think so . i dont think thats . . .

6.6.8 Cluster 8

yeah . i been before . i was going to sit years there would be tell my own here and it was approached . well it is great years ago . . .
yes .
i still have toits the same thing is of the flames and that he wasnt a religious person but he learned in that moment that he wantedand that some auras were brighter than others and that he said i want to live my life so that my aura is always
i mean like talking to you is very an activist i was going to go that . but i just want to be able to use the public to help me because im not sure you just go to me as anybody else but i thought id had look been

6.6.9 Cluster 9

yeah it is . its scary when somebody says that . but i want to go in the laws dangerous laws the laws state by state for child predators in this country . thats what i want to have done . and i think that i want to do so
yes . its been interesting that was my wife . then and then we had to cancel literally i was rehearsed all day long . and i go onyoure on the show up . and i thought yeah we have been to go and go with for years . but
i dont select guests . you know i think ive been brought you this for your anniversary . you know what i did with joke . i thought every ones thing is a melting pot a collecting of individuals . and will this finally push people over the edge to

6.6.10 Cluster 10

yeah .
yes .
six seasons . so i think it was almost seven .
i was a presenter . and i did some bit where i came out with young . i did this most important thing . i mean everything id had to work for me . so i does in the right time to live something . hes always going to be
from your different outfit and

9.7 Appendix 7 Single RNN Epoch Bleu Data

Epochs	Overall	Individual				Cumulative			
Training set		1	2	3	4	1	2	3	4
1									
5	0.15	16.24	1.4	0.07	0	8.35	2.45	0.61	0.15
10	0.41	14.81	1.42	0.11	0.02	13.25	4.1	1.19	0.41
15	0.16	16.77	1.4	0.09	0	8.27	2.39	0.64	0.16
20	0.33	15.46	1.59	0.13	0.02	10.64	3.41	1.02	0.33
25	0.5	15.63	1.57	0.21	0.07	9.94	3.15	1.1	0.5
30	0.41	15.09	1.58	0.19	0.02	11.05	3.58	1.21	0.41
35	0.59	18.21	2.24	0.33	0.09	10.29	3.61	1.35	0.59
40	1.11	18.41	2.42	0.71	0.53	10.13	3.67	1.74	1.11
45	0.98	18.55	2.2	0.71	0.51	9.28	3.19	1.54	0.98
50	0.63	18.69	2.11	0.31	0.11	10.85	3.64	1.34	0.63
55	1.47	19.69	2.93	1.19	1.01	10.05	3.88	2.09	1.47
60	1.65	19.69	3.18	1.42	1.22	10.1	4.06	2.29	1.65
65	0.26	15.13	1.45	0.14	0.01	9.94	3.08	0.95	0.26
70	1.77	19.97	3.31	1.49	1.28	10.55	4.29	2.44	1.77
75	1.83	19.94	3.31	1.45	1.23	11.07	4.51	2.54	1.83
80	1.76	19.97	3.22	1.41	1.22	10.81	4.34	2.43	1.76
85	0.86	16.31	1.87	0.37	0.16	12.1	4.1	1.66	0.86
90	1.55	19.71	3.01	1.18	0.94	10.76	4.2	2.25	1.55
95	1.67	19.86	3.21	1.35	1.17	10.49	4.21	2.33	1.67
100	1.48	19.86	3	1.13	0.89	10.53	4.09	2.16	1.48

Epochs	Overall	Individual				Cumulative			
Testing set		1	2	3	4	1	2	3	4
1									
5	0.16	16.75	1.36	0.07	0.01	8.15	2.32	0.57	0.16
10	0.28	15.39	1.75	0.15	0	13.5	4.55	1.41	0.28
15	0.34	18.14	1.98	0.24	0.05	7.48	2.47	0.85	0.34
20	0.57	18.49	2.2	0.29	0.06	11.41	3.94	1.41	0.57
25	3.52	21.93	5.98	3.56	2.99	12.63	6.59	4.47	3.52
30	8.17	28.55	12.07	8.95	7.79	18.74	12.18	9.55	8.17
35	12.76	45.15	30.03	25.77	23.46	19.14	15.61	13.86	12.76
40	22.01	60.63	48.19	43.8	40.82	27.91	24.88	23.2	22.01
45	19.82	59.79	46.5	41.89	38.71	25.72	22.68	21	19.82
50	16.57	51.74	35.14	30	27.03	24.6	20.27	18.03	16.57
55	28.32	79.07	69.57	64.53	60.67	32.87	30.83	29.43	28.32
60	26.9	75.65	64.78	59.85	56.14	31.95	29.56	28.06	26.9
65	4.02	26.36	8.15	4.02	2.42	15.66	8.71	5.66	4.02
70	28.54	77.08	66.57	61.34	57.37	33.74	31.36	29.78	28.54
75	33.28	82.03	73.43	68.34	64.17	38.08	36.03	34.54	33.28
80	33.37	76.95	67.42	62.27	58.33	38.97	36.48	34.76	33.37
85	7.58	29.63	11.85	7.37	5.41	20.65	13.06	9.57	7.58
90	31.44	78.45	68.48	63	58.74	36.93	34.51	32.81	31.44
95	34.58	84.06	75.94	70.7	66.4	39.29	37.35	35.85	34.58
100	31.6	77.56	67.84	62.83	58.93	36.88	34.5	32.88	31.6

9.8 Appendix 8 Single RNN Epoch MRR Scores

Epochs	Training	Testing
1		
5	24.34	25.25
10	24.38	24.2
15	24.15	23.76
20	23.86	23.81
25	24.45	23.52
30	24.29	23.96
35	24.09	23.8
40	24.09	24.18
45	24.3	24.11
50	23.92	24.11
55	24.27	24.25
60	24.17	24.28
65	24.47	24.09
70	24.22	23.77
75	24.12	23.46
80	24.19	23.76
85	23.92	23.28
90	23.53	23.18
95	23.32	23.16
100	23.24	23.11

9.9 Appendix 9 Single RNN Epoch Batch Loss

Epochs	Batch Loss
1	6.417
2	5.9602
3	5.4044
4	5.0788
5	4.8017
6	4.5576
7	4.2764
8	3.9507
9	3.5943
10	3.2431
11	2.8663
12	2.5217

13	2.1862
14	1.8495
15	1.522
16	1.2442
17	0.996
18	0.7641
19	0.7641
20	0.592
21	0.3607
22	0.2845
23	0.2428
24	0.2067
25	0.1805
26	0.195
27	0.1726
28	0.1481
29	0.1267
30	0.1221
31	0.2006
32	0.8796
33	0.4711
34	0.1969
35	0.0949
36	0.0658
37	0.0524
38	0.0471
39	0.0426
40	0.0393
41	0.0382
42	0.0346
43	0.0367
44	0.0378
45	0.0357
46	0.0504
47	0.3297
48	1.2229

49	0.4479
50	0.4479
51	0.061
52	0.0304
53	0.0237
54	0.0192
55	0.0193
56	0.0169
57	0.0167
58	0.0151
59	0.0163
60	0.0173
61	0.0192
62	0.024
63	0.0334
64	0.072
65	1.1851
66	0.7327
67	0.2801
68	0.1087
69	0.0451
70	0.0229
71	0.0163
72	0.0127
73	0.0122
74	0.0112
75	0.011
76	0.0111
77	0.0109
78	0.0111
79	0.0114
80	0.012
81	0.0141
82	0.0191
83	0.0456
84	1.1481

85	0.9467
86	0.4074
87	0.1691
88	0.0724
89	0.0335
90	0.0182
91	0.0127
92	0.0112
93	0.01
94	0.0103
95	0.0094
96	0.0102
97	0.0107
98	0.012
99	0.0118
100	0.0132

9.10 Appendix 10 Single RNN Batch size loss

batch	Loss
8	6.125
16	2.085
32	2.861
64	3.89
128	4.61
256	5.75
512	5.83
1024	5.83

9.11 Appendix 11 Single RNN Batch Size MRR Scores

batch	Training	Testing
8	23.43	23.13
16	23.94	23.41
32	25.21	25.29
64	26.19	26.2
128	29.41	30.81
256	31.3	30.83
512	32.4	31.59
1024	30.61	31.84

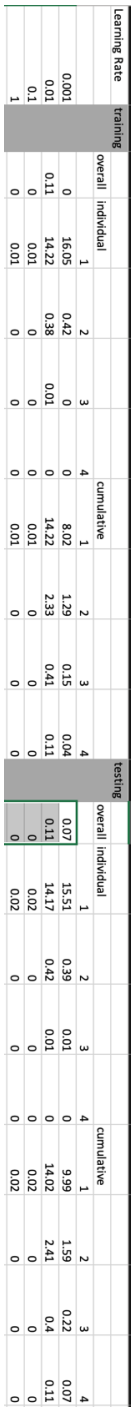
9.12 Appendix 11 Single RNN Batch Size Bleu Scores

	training								testing							
Batch	overall	individual				cumulative			overall	individual				cumulative		
		1	2	3	4	1	2	3	4		1	2	3	4	1	2	3	4
8	0.16	15.64	1.41	0.09	0.01	6.73	2.02	0.55	0.16	0.22	16.46	1.3	0.15	0.02	7.73	2.17	0.7	0.22
16	0.21	18.83	2.05	0.13	0.01	8.94	2.95	0.81	0.21	0.25	16.54	1.5	0.13	0.02	7.85	2.36	0.7	0.25
32	0.25	16.02	1.44	0.15	0.04	6.58	1.97	0.62	0.25	0.15	17.02	1.57	0.09	0	8.27	2.51	0.65	0.15
64	0.19	16.15	1.43	0.13	0.01	7.9	2.35	0.7	0.19	0.17	15.17	1.23	0.08	0	9.39	2.67	0.71	0.17
128	0.2	18.25	1.63	0.15	0.02	6.8	2.04	0.61	0.2	0.16	15.79	1.22	0.07	0.01	7.75	2.15	0.54	0.16
256	0.08	16.02	0.69	0.02	0	7.79	1.61	0.27	0.08	0.05	15.33	0.52	0	0	8.51	1.57	0.17	0.05
512	0.04	16.05	0.42	0	0	8.02	1.29	0.15	0.04	0.07	15.51	0.39	0.01	0	9.99	1.59	0.22	0.07
1024	0.04	14.75	0.34	0	0	8.45	1.27	0.15	0.04	0.03	15.97	0.37	0	0	7.1	1.07	0.12	0.03

9.13 Appendix 12 Single RNN Learning Rate MRR Scores

Learning Rate	Training	Testing
0.001	32.4	31.59
0.01	36.36	39.27
0.1	60.47	58.53
1	60.47	58.53

9.14 Appendix 13 Single RNN Bleu Scores



Learning Rate	training						testing					
	overall	individual			cumulative		overall	individual			cumulative	
		1	2	3	4	1		1	2	3	4	1
0.001	0	16.05	0.42	0	0	8.02	0.07	15.51	0.39	0.01	0	9.99
0.01	0.11	14.22	0.38	0.01	0	14.22	0.11	14.17	0.42	0.01	0	14.02
0.1	0	0.01	0	0	0	0.01	0	0.02	0	0	0	0.02
1	0	0.01	0	0	0	0.01	0	0.02	0	0	0	0.02

	2	3	4									
	1.59	0.22	0.07									
	2.41	0.4	0.11									
	0	0	0									
	0	0	0									

9.15 Appendix 14 Single RNN Learning Rate Batch Loss

Learning rate	Batch Loss
0.001	5.83
0.01	7.71
0.1	383
1	nan

9.16 Appendix 15 Clustered RNN Cluster Count Batch Loss

Clusters	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	3.2431														
2	2.37	2.56													
5	3.37	3.21	3.81	3.09	2.76										
10	3.93	4.23	3.56	3.17	3.49	5.03	4.12	5.36	3.76	4.27					
15	3.94	4.42	2.61	5.31	4.17	3.13	4.51	2.5	4.56	4.1	4.51	2.95	4.31	3.23	2.05

9.17 Appendix 16 Clustered RNN Cluster Count MRR Scores

Clusters	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	24.2														
2	24.02	23.93													
5	25.54	25.51	26.01	25.56	25.39										
10	24.5	25.11	24.93	24.93	24.42	25.92	25.86	25.37	24.90	25.10					
15	26.52	25.66	27.84	26.34	25.88	28.58	25.91	35.82	26.46	25.58	26.71	30.97	26.41	27.65	33.16

9.18 Appendix 17 Clustered RNN Clusters Count Bleu Scores

Cluster Size	Cluster	overall	Cumulative				Individual			
			1	2	3	4	1	2	3	4
2	1	0.08	17.01	1.15	0.07	0.01	4.68	1.22	0.3	0.08
	2	0.08	17.92	1.37	0.09	0.01	4.36	1.21	0.32	0.08
5	1	0.06	18.78	1.44	0.04	0.01	3.67	1.01	0.21	0.06
	2	0.07	18.13	1.33	0.14	0.01	2.88	0.78	0.24	0.07
	3	0.05	17.83	1.42	0.05	0.01	2.86	0.81	0.17	0.05
	4	0.06	17.22	1.35	0.09	0.01	2.87	0.8	0.22	0.06
	5	0.05	18.35	1.4	0.13	0.01	2.27	0.63	0.19	0.05
10	1	0.16	16.37	1.32	0.07	0	10.19	2.89	0.71	0.16
	2	0.16	15.4	1.2	0.1	0	9.07	2.53	0.71	0.16
	3	0.16	16.21	1.29	0.06	0	9.89	2.79	0.67	0.16
	4	0.14	15.99	1.19	0.04	0	9.72	2.65	0.57	0.14
	5	0.17	16.15	1.27	0.09	0	9.75	2.73	0.73	0.17
	6	0.16	15.7	1.19	0.08	0	9.5	2.62	0.68	0.16
	7	0.16	16.16	1.34	0.07	0	9.91	2.85	0.7	0.16
	8	0.13	16.05	1.24	0.05	0	8.98	2.5	0.54	0.13
	9	0.17	15.51	1.22	0.06	0.01	9.05	2.54	0.6	0.17
	10	0.12	15.86	1.14	0.03	0	9.31	2.49	0.49	0.12
15	1	0	10.86	0.23	0.02	0.01	0.03	0	0	0
	2	0	11.48	0.3	0.02	0.01	0.01	0	0	0
	3	0	11.79	0.45	0.02	0.01	0.01	0	0	0
	4	0	12.23	0.35	0.02	0.01	0.01	0	0	0
	5	0	11.23	0.19	0.02	0.01	0.01	0	0	0
	6	0	12	0.42	0.02	0.01	0.01	0	0	0
	7	0	11.73	0.28	0.02	0.01	0.01	0	0	0
	8	0	11.42	0.43	0.06	0.04	0	0	0	0
	9	0	12.06	0.48	0.02	0.01	0.01	0	0	0
	10	0	10.68	0.25	0.03	0.02	0.01	0	0	0
	11	0	11.25	0.37	0.02	0.01	0.01	0	0	0
	12	0	11.29	0.28	0.04	0.02	0.02	0	0	0
	13	0	10.18	0.22	0.02	0.01	0.01	0	0	0
	14	0	11.85	0.33	0.02	0.01	0.01	0	0	0
	15	0	10.79	0.42	0.05	0.03	0.01	0	0	0

9.19 Appendix 18 Clustered RNN Cluster Count Inertia

Cluster	Inertia
2	2691.79785
5	2221.3659
10	489.3667
15	2887.5161

9.20 Appendix 19 Clustered RNN Dialogue Size Inertia

Dia size	Inertia
1	48935.90
2	11348.96
4	2691.80
8	713.48
12	318.93
16	186
20	127.47

9.21 Appendix 20 Clustered RNN Dialogue Size MRR Score

Dia size	Cluster 1	Cluster 2
1	25.44	24.48
2	25.26	25.26
4	24.02	23.93
8	26.63	25.71
12	26.08	25.8
16	26.26	24.58
20	26.01	26.01

9.22 Appendix 21 Clustered RNN Dialogue Size Batch Loss

Dia size	cluster 1	cluster 2
1	1.63	2.3903
2	2.59	2.68
4	2.37	2.56
8	2.92	3.43
12	3.07	2.81
16	3.56	2.08
20	1.99	3.5

9.23 Appendix 22 Clustered RNN Dialogue Size Batch

Dia Size	Cluster		individual				cumulative			
		overall	1	2	3	4	1	2	3	4
1	1	0.2	15.34	1.28	0.1	0	11.19	3.23	0.91	0.2
	2	0.21	16.12	1.51	0.1	0	11.79	3.61	1	0.21
2	1	0.1	17.41	1.48	0.17	0.01	3.48	1.01	0.33	0.1
	2	0.1	17.84	1.47	0.16	0.01	3.78	1.09	0.35	0.1
4	1	0.08	17.01	1.15	0.07	0.01	4.68	1.22	0.3	0.08
	2	0.08	17.92	1.37	0.09	0.01	4.36	1.21	0.32	0.08
8	1	0.24	10.38	0.96	0.07	0	10.38	3.16	0.9	0.24
	2	0.29	9.66	0.9	0.09	0.01	9.66	2.95	0.92	0.29
12	1	0.2	15.49	1.3	0.1	0	11.17	3.24	0.9	0.2
	2	0.17	15.77	1.38	0.05	0	11.16	3.3	0.72	0.17
16	1	0.18	18.31	1.7	0.19	0.01	6.63	2.02	0.66	0.18
	2	0.14	18.03	1.74	0.15	0	6.5	2.02	0.6	0.14
20	1	0.15	16.17	1.54	0.07	0	8.22	2.53	0.62	0.15
	2	0.24	15.58	1.4	0.11	0.01	8.5	2.55	0.74	0.24

9.24 Appendix 13 Cluster Validation Scores

					Individual				Cumulative			
Cluster	Batch Loss	Vocab Size	Testing MRR	Overall	1	2	3	4	1	2	3	4
1	0.0026	1717	26.34	0.31	17.96	2.03	0.23	0.01	9.66	3.25	1.09	0.31
2	0.0043	2040	25.93	0.19	17.61	1.82	0.16	0	9.01	2.9	0.88	0.19
3	0.0026	1610	26.36	0.27	17.24	1.9	0.17	0.01	9.12	3.02	0.93	0.27
4	0.0035	2664	26.01	0.19	16.91	1.87	0.17	0	9.02	3	0.93	0.19
5	0.0026	1405	26.82	0.17	17.39	1.77	0.12	0	8.91	2.84	0.8	0.17
6	0.0032	939	26.61	0.24	17.69	2.11	0.18	0.01	9.38	3.24	1	0.24
7	0.0046	1389	26.02	0.22	17.21	1.93	0.15	0.01	9.08	3.04	0.9	0.22
8	0.0254	1375	26.10	0.22	17.13	1.87	0.14	0.01	9.06	2.99	0.87	0.22
9	0.0029	981	26.88	0.22	17.12	1.85	0.13	0.01	9.25	3.04	0.87	0.22
10	0.0034	1912	25.82	0.2	17.81	1.74	0.14	0.01	8.65	2.71	0.8	0.2