Summarization of news articles

Master's thesis in Complex Adaptive Systems

OSCAR BERONIUS

Department of Physics
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2019

Master's thesis 2019

Summarization of news articles

OSCAR BERONIUS

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Physics
Chalmers University of Technology
Gothenburg, Sweden 2019

Summarization of news articles
OSCAR BERONIUS

© OSCAR BERONIUS, 2019.

Supervisor: Johan Nilsson Hansen, Meltwater
Supervisor: David Burke, Meltwater
Examiner: Mats Granath, Department of Physics, University of Gothenburg

Master's Thesis 2019
Department of Physics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2019

Summarization of news articles
OSCAR BERONIUS
Department of Physics
Chalmers University of Technology

Abstract

In this work, two neural summarization models, Seq2seq and the Transformer, with several variations, were implemented and evaluated on the task of abstractively summarizing news articles. Seq2seq yielded poor results, likely due to not being flexible enough to fit the data set. The Transformer yielded promising results, and it was discovered that the quality of the output was heavily dependent on the quality of the input data, indicating that the implementation might be good but the performance bottle necked by the data set. For future work, specifically in developing a summarizer of clusters of documents, a recommended approach would be to combine an abstractive summarizer such as the Transformer, with extractive methods. In such a case, the Transformer could be further improved upon by pre-training it on word embeddings such as Google BERT, or training it on additional data sets such as CNN/Daily Mail. Finally, it was discovered that the used evaluation metric, ROUGE, could not be considered complete for the given task, and it would thus be advised to explore additional evaluation metrics for summarization models.

Keywords: Summarization, Summary, NLP, Transformer, Attention, Articles, Long, Multiple, Seq2seq, Abstractive.

Acknowledgements

I would like to thank my supervisors Johan Nilsson Hansen and David Burke for their exceptional help and engagement as well as my examiner Mats Granath for taking on this project. I would also like to thank Mikael Johansson for making this project possible and including me in the Meltwater community during my time there. Finally, I would like to give a special thanks to Jacob Sznajdman for all of his advice on implementations and the hours we spent philosophizing about models together.

Oscar Beronius, Gothenburg, May 2019

Contents

List of Figures

List of Tables

1
Introduction

In recent years, the field of natural language processing (NLP) has seen a rapid development, where new approaches have pushed the state of the art on a regular basis in a variety of tasks such as text translation, transduction and summarization. A pronounced development began in 2013 when different forms of neural networks started getting adopted into NLP models. By the introduction of the sequence to sequence encoder in 2014 by Sutskever et al. [14], the state of art was pushed significantly, and in 2016, following the work of Wu et al. [21], Google started replacing their phrase base systems in favor of neural models for machine translation. By 2019, neural based models have been widely accepted as the industry standard for most NLP related tasks.

Despite the wide adoption of neural networks and sequence to sequence, the combined approach has had several flaws. Not only are the models complex and difficult to train well, but they have also proven to be lacking when it comes to creating abstract representations of longer texts.

In 2017, Vasvani et al. [24] developed an architecture that they call the Transformer, which further pushed the state of art in NLP and solved several problems by omitting all recurrence and using attention based vectors instead. Authors claim that this recent development made it possible to solve new problems, such as summarizing longer text documents.

If such a model could be implemented and fit to a custom data set, with similar performance to the one claimed by authors, it could provide great industry value in terms of making information retrieval and analysis more efficient.

1.1 Background

This thesis is done in collaboration with Meltwater, a company specializing in media monitoring and analytics. Their information retrieval systems contain around 40 billion social media posts and editorial articles, and handles millions of search queries daily. Every day they add hundreds of millions of news articles, tweets, blog posts and other social media posts to their repositories. Their Gothenburg team owns a critical component of Meltwater's products, the search platform which plays a key role in the retrieval and analysis of all of the company's editorial and social media assets.

Meltwater is currently working on a proof of concept project, where they want to assemble coherent storylines from the myriad of media all talking about the same "thing". Their idea is to cluster news articles and provide a personalized summarization of that cluster. In other words, they are interested in a product that solves the problem of personalized multi-document summarization. That is, given a cluster of documents and a search query, find a good approach to summarize how the documents are talking about the key aspects of the search.

1.2 Purpose

The purpose of the project is to investigate how well the state of art in NLP can be implemented and trained on Meltwater data to perform summarization of long text sequences, as well as evaluate the performance of relevant models and discover what difficulties and challenges might arise.

The end goal is for this project to serve as a baseline and evaluation of feasible approaches that can be used to develop a personalized multi-document summarizer in the future.

1.3 Delimitations

As stated in the goal, the scope of the project is to evaluate state of art technologies in NLP to perform summarization of long texts. While this excludes the business case of summarizing clusters of documents, long text summarization can still be used as part of a solution to such a project. With the potential for further development, the model is chosen keeping in mind context based information retrieval.

As for linguistic capabilites, the model is expected to be working with english text data. The size of documents should lie in the range of thousands of symbols.

2
Theory

In this chapter, various concepts and methods that the work was built upon will be explained in order for the reader to better understand the project.

2.1 Recurrent neural networks

The recurrent neural network [1] can be seen as an extended version of the traditional neural network with the additional capability in handling sequential data, thus allowing it to gain understanding of time-series. This is achieved by implementing an inner state into the model, and coupling several recurrent units into a chain, where the inner (or hidden) state is fed into the following neural units. By using backpropagation through time [2] and feeding multiple units with multiple time-series data points, the inner state becomes an abstract representation of the information the sequence contains.

A classical neural network shown in figure 2.1 might be defined as follows:

$$a = \sum_{i=1}^{n} w_i x_i + b_i, \tag{2.1}$$

$$y = \sigma(a) \tag{2.2}$$

Where x is the input, w are the weights, b is bias, and σ is an activation function yielding output y.

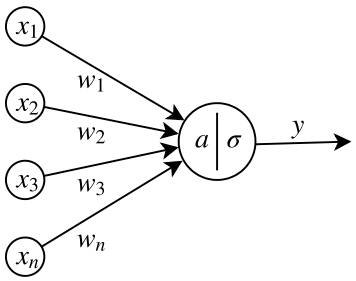

Figure 2.1: Base structure of a classic neural network. X represents the input, paired with corresponding weights w, yielding a, that is fed into an activation function producing output y.

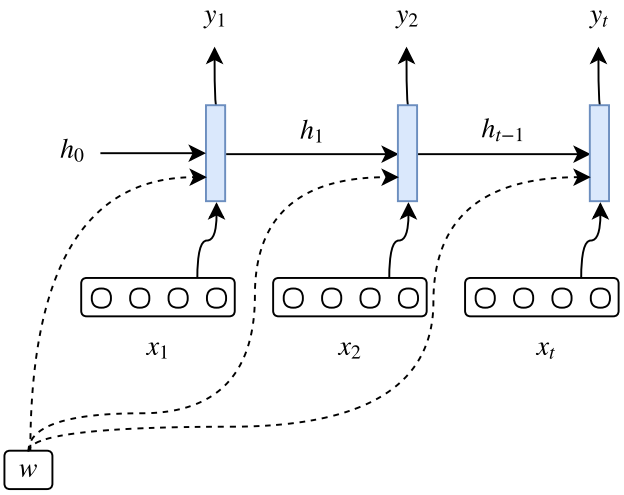

Figure 2.2: Base structure of a recurrent neural network. x are input vectors, h is the hidden state, y is the output and W is the weight matrix that is shared among RNN cells. Blue rectangles represent the cells, where computation that calculates the next hidden state and output occurs.

With the base structure of a neural network in mind, a simple recurrent model contains a hidden state of the following form:

$$h_t = \begin{cases} 0, & t = 0 \\ f(h_{t-1}, x_t), & otherwise \end{cases} \tag{2.3}$$

The hidden state is a vector containing information about the network at time-step t. Each hidden state depends on the previous state as well as an input, where the first hidden state is typically initialized to 0. f is a non-linear activation function for the following state in the time series. Using sigmoid as activation function gives

$$h_t = \sigma(w_{hh}h_{t-1} + w_{xh}x_t) \tag{2.4}$$

where w is the weight matrix corresponding to the states and the inputs. Note that the same weight matrices are used to infer the next coming states and outputs for the whole network. Finally, an output for each time step can be generated by feeding the hidden state with corresponding weights into a softmax function

$$y_t = softmax(w_y h_t) \tag{2.5}$$

Figure 2.2 shows the architecture of a recurrent neural network.

In the case of Natural Language Processing (NLP), a text can be fed into a recurrent model, where each word of a sequence is fed into a recurrent cell that then propagates its inner state forward into a chain of other recurrent cells. The resulting inner state can then be used to infer a new sequence of words that is dependent on

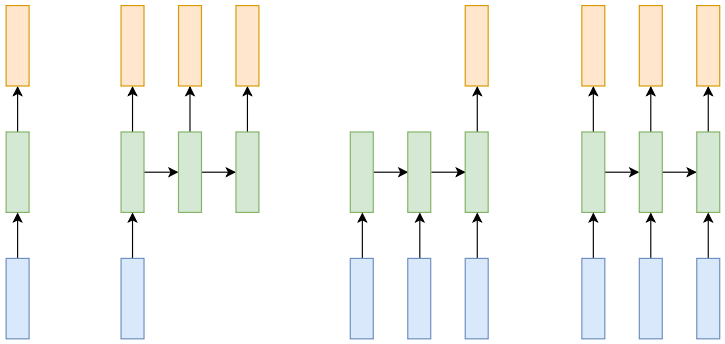

Figure 2.3: Possible relationships for the RNN. The image shows how any number of inputs (blue rectangles) can be fed into a RNN network with multiple cells (green rectangles) to output any number of outputs (yellow rectangles) as described in [33].

all previous words, such that understanding of the context that each word bears in a given sentence can be gained.

Another feature of the RNN, unlike a traditional NN, is that it has the capability to map various number of inputs to various numbers of outputs. See fig. 2.3. This is important in the context of NLP as for many tasks, such as text translation and text summarization, a given text sequence might not be of the exact same length as the desired output.

2.2 Improved recurrent neural networks

There are several extensions to the traditional RNN that improves its performance and reduces problems with vanishing gradients. Two of the most commonly used and best performing variations are the Gated Recurrent Unit (GRU) network and the Long-Short Term Memory cell (LSTM).

The GRU, introduced by Cho et. al [12] improves the vanishing gradient problem through the introduction of two gating functions in the recurrent unit which calculates if the gradient should be either updated or reset depending on a specific input. This allows the network to learn what features are of high importance such that they are not diminished by other signals.

It is defined by in [33] as follows:

$$z_t = \sigma(W^{(z)}x_t + U^{(z)}h_{t-1}) \tag{2.6}$$

$$r_t = \sigma(W^{(r)}x_t + U^{(r)}h_{t-1}) \tag{2.7}$$

$$\tilde{h}_t = tanh(Wx_t + r_t \circ Uh_{t-1}) \tag{2.8}$$

$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \tilde{h}_t \tag{2.9}$$

Where z_t and r_t are update and reset gates, $\tilde{h}_t$ is a memory cell and h_t is the resulting internal state. U and W are weight matrices. As can be seen in equation 2.9, if the update gate is set to 0 then the internal state is maintained as the previous state, but if it set to 1 then the next state is updated according to the memory, which either resets the previous input or includes the last input.

The LSTM proposed by Hochreiter et al. [3] in 1997 works in a very similar way but is implemented slightly differently. Instead of using two gating functions, it has three; an input gate, an output gate and a forget gate. The input gate determines how much information of a previous layer should be taken into account during inference for a certain unit. The output gate determines how much of the current unit should be exposed to the rest of the network in the next coming iteration and the reset gate determines if the current memory cell should be set to zero or not. Comparably, the update gate of the GRU can be likened to a combination of the input and output gates of the LSTM. Stanford [33] defines the architecture of the LSTM as follows:

$$i_t = \sigma(W^{(i)}x_t + U^{(i)}h_{t-1}) \tag{2.10}$$

$$f_t = \sigma(W^{(f)}x_t + U^{(f)}h_{t-1}) \tag{2.11}$$

$$o_t = \sigma(W^{(o)}x_t + U^{(o)}h_{t-1}) \tag{2.12}$$

$$\tilde{c}_t = tanh(W^{(o)}x_t + U^{(c)}h_{t-1}) \tag{2.13}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \tag{2.14}$$

$$h_t = o_t \circ tanh(c_t) \tag{2.15}$$

Where 2.10 is the input gate, 2.11 is the forget gate, 2.12 is the output gate, 2.13 is a memory cell, 2.14 is the final memory cell that determines both how much of the past layer should be accounted for as well as the current input. 2.15 is the final hidden state, determined by how much the resulting memory cell should be exposed.

The GRU and LSTM have been shown to have similar performance depending on the task, and no one of them is strictly better than the other. The LSTM is however slightly more complex and requires additional computation.

2.3 Sequence to sequence

Sequence to sequence (seq2seq) is an architecture that combines several recurrent networks to handle sequential inference. It was developed by Sutskever et al. [14] and produced state-of-art results in the tasks of text transduction, translation, summarization among others at the time. The architecture has since been used frequently in a wide variety of NLP tasks and is often seen as the go to standard method in the area. A seq2seq architecture is comprised of an encoder and a decoder. The encoder and decoders typically consists of one or several recurrent neural networks of different types, and the idea is for the whole encoder network to build an abstract representation of the contents and semantics of a given text, while the decoder, as the name entails, uses the internal state to generate an output according to the given task.

There are limitations to the standard recurrence-based seq2seq encoder-decoder, where one of the biggest is that the hidden state vector has a limited size and thus can only store information up to a certain amount. This means that it is harder for models to perform well when the input size is increased as they will 'forget' the semantics of the earlier parts of the inputs [12], such as in the case of summarization of long texts. In the context of machine translation, this is also problematic since the beginning of a paragraph is often the most predictive part of what the rest of the text might entail, especially in languages such as English, German, French and Chinese. Sutskever et al. [14] showed that this problem was significantly improved by reversing the input, feeding it into the encoder in reverse order. This allowed the network to remember more important features of the input more clearly as they would not be diminished by other signals.

The reversal trick is however heavily limited, and only works in certain cases. If the last word of a sentence is more predictive of correct translation, as are some cases for Japanese to English translations, then reversing the input will instead reduce the performance.

A more general solution to this problem can be built using attention mechanisms.

2.4 Attention

Bahdanau et al. [11] improved the seq2seq architecture by introducing the concept of attention between the encoder and decoder networks, allowing the decoder to learn what features to focus more on, as opposed to deriving them only from the hidden state. This was done by training a dense network with a softmax layer from each word to all other words in the input. This approach made it possible for the decoder to choose which hidden state is the most likely to best predict the next word in a sequence, instead of only using the last hidden state of the encoder.

Hermann et al. [15] experimented with several number of language models in the

task of question answering, and retreived the best results with an attention based neural model. They visualized the context vectors of their model's attention layer during inference by plotting a heat map covering the words their model attended to. This is shown in Fig. 2.4.

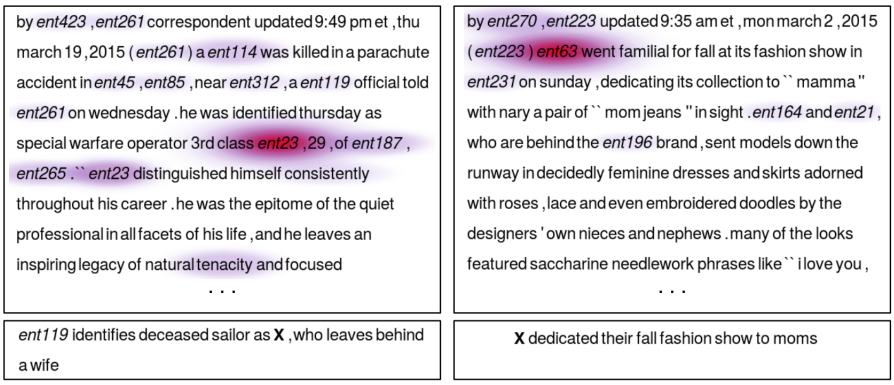

Figure 2.4: Heat map of what parts of a text the attention based neural model developed by Hermann et al. [15] attended to when asked a question about the text. For both the left and right case, the model attended to the most probable answers (red colour), which also turned out to be correct.

3
Methods

3.1 Data

The data set was comprised of news articles collected from various news sites and social media using Meltwater's web crawling tools. Each data point consisted of a text and a corresponding manually written ingress. In its raw format, the data set contained $\sim$ 8 million data points of arbitrary size. This meant that the data set contained abnormal data points, including articles of sizes, as measured by number of symbols, of over 100000 with a corresponding ingress of less than 10 symbols. There were also data points that consisted of non-sensical symbols, chinese symbols, emojis etc. In order for the model to be able to learn to summarize according to the given task the data set had to be cleaned, such that the data points correctly corresponded to probable summarization cases. Several sub-procedures were developed for this reason.

3.1.1 Removing noise

Noise was considered to be articles or ingresses containing symbols that were not typically used in the English language. To determine what data points to keep, an algorithm was developed that searched the data set and removed all entries that contained symbols other than the ones included in the extended ASCII-table. It was also necessary to remove all duplicates of data points as these would make the model learn specific features that not necessarily translated to the general problem. Duplicates were removed by storing texts in a hash map with each corresponding ingress as key, and then iterating the hash map to retrieve one value for each key. This reduced the data size of a factor $\sim 1/5$. It should be noted that this method only removed exact duplicates, as ingresses that would only be slightly different would get different hash values. Such cases were however not found upon inspecting the results, so the method was deemed good enough.

3.1.2 Removing abnormalities

In this case, an abnormal data point refers to one with a size that was atypical for the general size distribution of the data set. Several factors were considered when choosing the size thresholds for the data sizes. The size distributions, both in terms of number of tokens as well as words, were plotted in order to gain insight on how data points could be extracted to both make the data more uniform as well as fit the task better. The data was also trimmed on the size ratio between text and ingress in

order to achieve a good balance on the summaries. This process is shown in Figures 3.1 and 3.2.

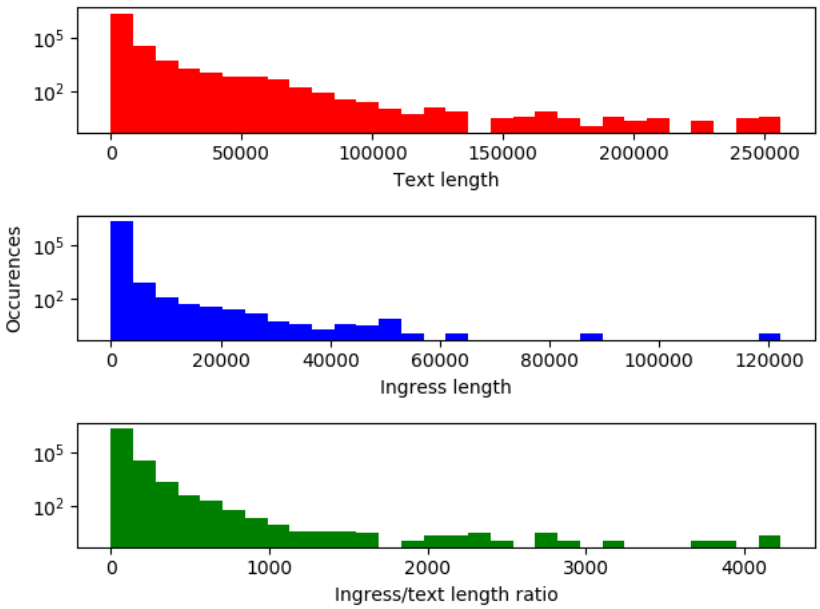

Figure 3.1: Inspection of the initial size distributions of the data set before any trimming. The length was measured in number of symbols. The data set contained data points with text lengths up to 260000, ingress lengths up to 125000 and data points where the ingress was 4000 times larger than the corresponding text.

Besides trimming the data to fit the task better, it was also necessary to adjust it for optimization purposes. In particular, a large number of symbols increased the RAM-memory demands on the hardware as well as increased the training time. For the final trim, the maximum allowed length was 1000 symbols for the text, 180 symbols for ingress and a maximum size ratio $ingress/text$ of 0.3. This is shown in Figure 3.3.

After processing the data set, the resulting number of data points was in the range of ~ 600000. Due to memory constraints and long training times, all of them could however not be utilized.

3.2 Models

There are mainly two approaches that are used for summarizing texts: extractive and abstractive. The first, extractive, refers to rule-based methods that identify relevant parts of a given text and then combine these extracts to form a summary. There are various algorithms for selecting extracts, such as Tf-Idf [6], TextRank [8] and SumBasic [9].

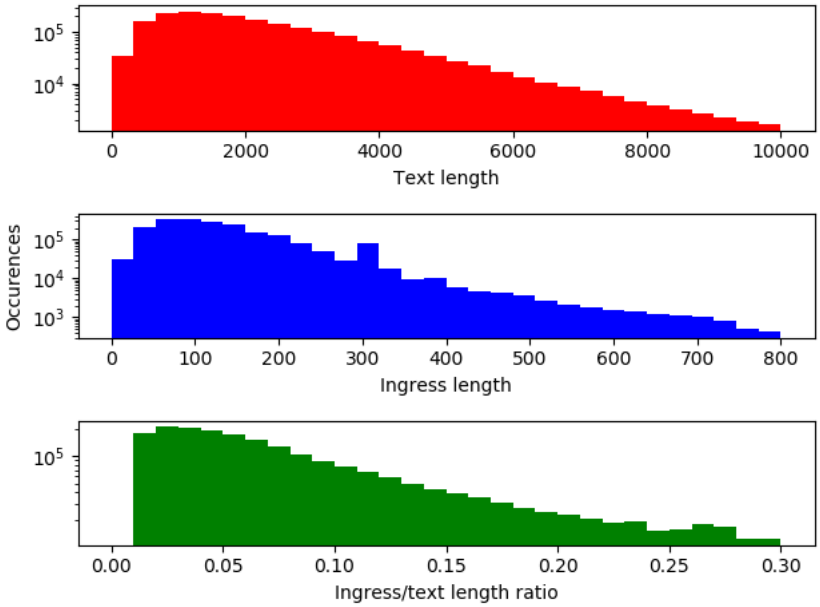

Figure 3.2: Size distribution after removing data points with text lengths longer than 10000 symbols or ingresses longer than 800 symbols. Data points with an ingress larger than 30% of its corresponding text were also removed.

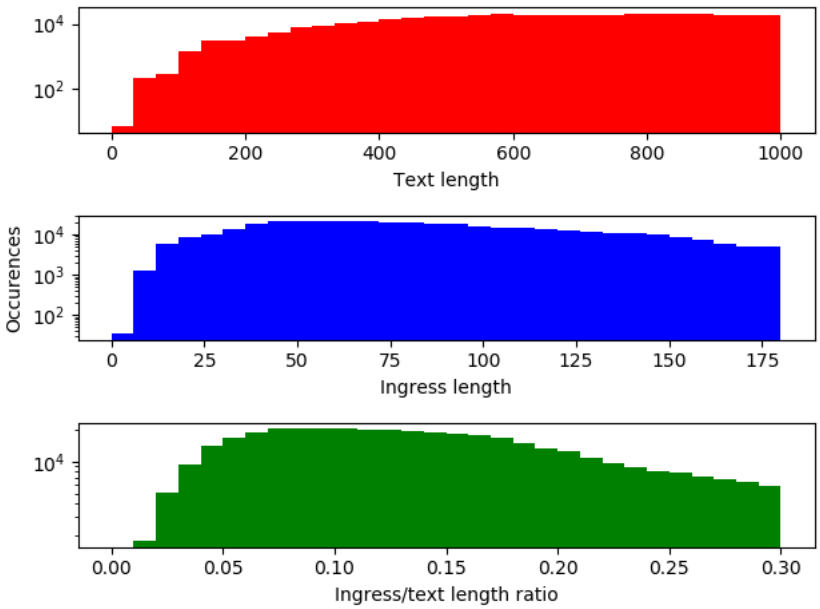

Figure 3.3: Final distribution of data point sizes that was used to train the transformer. Texts larger than 1000 symbols and ingresses longer than 180 symbols were removed. The intervals were chosen such that a significant amount of data points would lie in the whole range of sizes.

The second, abstractive, refers to an approach that is similar to the way humans produce summaries. Instead of extracting snippets of texts, humans tend to first

read through sub-titles, then go through the text on a higher detail to gain an understanding of it, and finally re-writing a summary of recollected subjects the writer deemed relevant. Abstractive summarization can be achieved through deep learning methods, which will be further discussed in this section.

The main approach for this project was inspired by Liu et al. [32] in their work on a Wikipedia generator, where a model that combined extractive and abstractive methods was used. It is of high importance that the generated summaries are abstractively written as the main metric of success is customer satisfaction, which means that readability, grammar and congruency will be important metrics, something extractive methods do not take into account.

The focus was therefore on designing a well performing abstractive model that can serve as a base component for a document cluster summarizer where extractive methods are used as a pre-processing step. Two different model architectures were tested for abstraction, a recurrent seq2seq model with different kinds of cells, and a Transformer with various settings.

3.3 Recurrent models

The first attempt was to implement a seq2seq encoder-decoder using recurrence and see how such an implementation would handle Meltwater data. Two recurrent units were evaluated, the standard LSTM and the Independently Recurrent Neural Network [31]. All models were trained iteratively using Mini-batch gradient descent, as well as paused while more data points were loaded, in order to not run out of RAM-memory.

In both cases the source sentence was also reversed when fed into the encoder part of the network in order to maximize performance, as discovered by Sutskever et al. [14]

3.3.1 One hot encoding

The data was encoded into one hot format since it had to be numerical for the model to be able to process it. This means that each word was transformed to a vector, where each element represented a single symbol from the dictionary of all available symbols. All elements in the vector were set to zero apart from the element matching the word, which was set to one. This was used in favor of indexed mappings in order to reduce the risk of the model inferring ordering correlations that do not exist.

3.3.2 Seq2seq - LSTM

The model was designed according to the description in the Theory chapter. It was implemented in Python using Keras with Tensorflow as backend. Data was handled using the Pandas library.

The model was trained on 15000 data samples, with a batch size of 64 for 50 epochs, with a latent dimensionality of the encoding space of 256. Rmsprop was the optimizer and the loss function was categorical cross entropy. A single LSTM layer was used for the encoder and another single LSTM layer was used for the decoder.

3.3.3 Seq2seq - IndRNN

The Independently Recurrent Neural Network (IndRNN) was developed by Li et al. [31] with the purpose of reducing exploding/vanishing gradient problems commonly faced by RNNs, LSTMs and GRUs. Authors claim they saw an improvement with the IndRNN in capturing longer term dependencies, with a length up to 5000, as compared to other recurrent units. Instead of a basic implementation of an RNN following the definition

$$h_t = \sigma(Wx_t + Uh_{t-1} + b) \tag{3.1}$$

The IndRNN was implemented using the Hadamond product between weights and previous state

$$h_t = \sigma(Wx_t + u \odot h_{t-1} + b) \tag{3.2}$$

Where h_t is the next hidden state, W is a weight matrix, x_t is the input, U is state weights, h_{t-1} is the previous state and b is a bias. For the IndRNN, u is a weight vector.

For the implementation, the architecture of the seq2seq encoder-decoder and parameter settings for this variant was the same as the LSTM-version, with the recurrence cell being replaced by the IndRNN. Six IndRNN layers were also stacked in the encoder, as suggested by authors the performance of the IndRNN would improve dramatically by stacking multiple layers.

3.4 The Transformer

The Transformer was the main model implemented for this project. It was developed by Vasvani et al. [24] in 2017 and has yielded state of art results in a variety of NLP tasks. Its architecture is based on the previous seq2seq architecture, but omits all recurrence in favor of augmented attention mechanisms and positional encodings. This allows for longer range dependencies being captured, as well as simplified training due to the nature of multiple attention layers. By removing recurrence, the sequential property of seq2seq can be transformed into multiple abstract layers that can be trained in parallel. This will be further explained in the 3.4.3 sub section. The complete architecture of the Transformer is shown below in Figure 3.4.

The Transformer was implemented in Pytorch, using Spacy for language tokenization and Pandas for data handling and formatting.

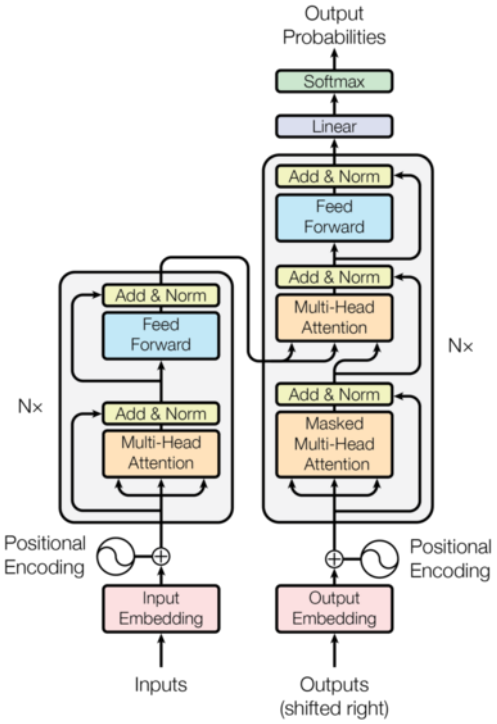

Figure 3.4: Architecture of the transfomer. [24]

3.4.1 Word embeddings

Before feeding the words into the Transformer they were first converted into embeddings. Embeddings are vectors that represent the words and their contextual meaning. This means that not only is the word stored in the vector, but also their semantic relation to other words. This was done using the Pytorch implementation of an embedding layer.

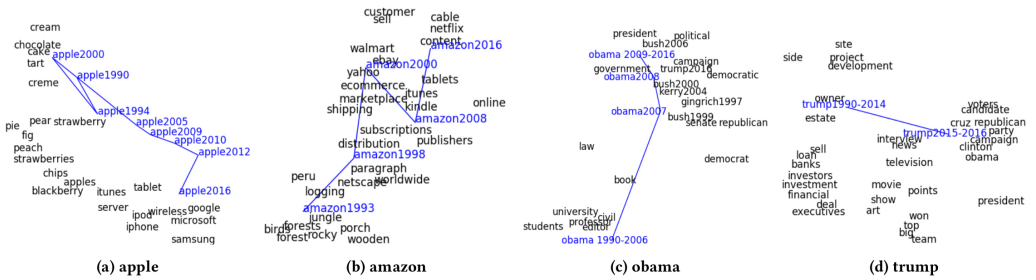

Figure 1: Trajectories of brand names and people through time: `apple`, `amazon`, `obama`, and `trump`.

Figure 3.5: Word embedding vectors visualized. The more relevant the words are in context to each other, the closer their vectors are aligned. The image was created as part of the work on dynamic word embeddings made by Yao et al. [25]

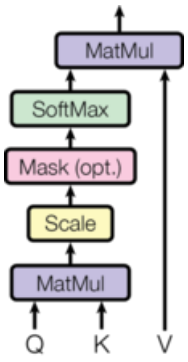

Figure 3.6: Diagram over the parts included in the Scaled Dot-Product Attention of the Transformer.

3.4.2 Positional encoding

Before the transformer, in architectures like the seq2seq, understanding of the positional context of all words was implicitly gained through the sequential nature of the network. Due to omitting this property, positional encoding needs to be injected to the word vectors before they are processed by the Transformer. This was done using sine and cosine functions of different frequencies as per the implementation by Vasvani et al. [24].

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{model}}) \tag{3.3}$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{model}}) \tag{3.4}$$

3.4.3 Attention

The core of the attention mechanism that the Transformer uses is based on a key, value, query mapping to a softmax layer In general, two versions of this attention is used, an additive version and multiplicative one. Vasvani et al. used a multiplicative version since it performed better due to being optimizable through matrix multiplication. They added a scaling factor to minimize gradient explosion, and call it Scaled Dot-Product Attention. It is defined as

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{3.5}$$

In addition to this, the Transformer employs several so called attention-heads, which can be seen as several layers of attention that can be stacked upon each other. Each layer, or head, can be trained on a single GPU, which means that a Multi-headed attention mechanism can be parrallelized, decreasing training time and increasing memory capabilities.

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O \tag{3.6}$$

where

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{3.7}$$

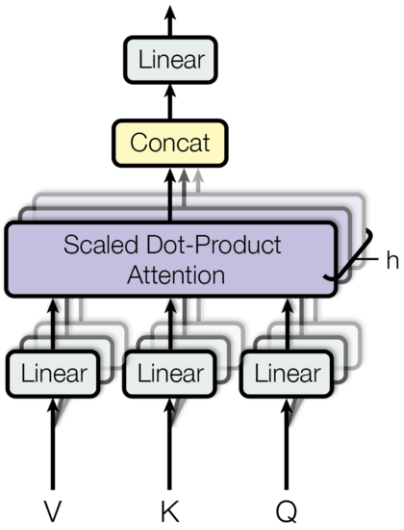

Figure 3.7: Diagram over the parts included in the Multi-Head Attention of the Transformer.

3.4.4 Beam search

Beam search was used during inference to find candidate summaries. For each new word that is generated, the probability for each following word to be correct has to be evaluated for all possible words. The total score for each candidate summary depends on the score of all words in the sequence, meaning that it is not possible to know the best summary before all possible sequences have been evaluated, which costs $log(n^2)$ operations to compute where n is the number of words in the vocabulary. It takes exceedingly long time to do this for larger vocabularies, whereby a different approach is needed in practice. Beam search solves this by only allowing a number of candidates β, called beam width, to be evaluated at the same time. In this regard it can be seen as a down scaled Breadth First Search with a heuristic.

The score was calculated according to the method used by Wu. et al [21] in their work leading up to Google's Neural Machine Translation System. In their work, a length penalty α factor was included to limit output length. For the implementation in this project, a beam width of 3 and a length penalty of 0.7 was used.

3.5 Training the Transformer

The testing-cycle for the Transformer entailed conducting numerous experiments with various settings to understand how they would affect the model and how the settings could be optimized for good results. This section explains the process and some experiments that were made when training the Transformer.

Initially, the Transformer was trained for 80 epochs, optimizing the Cross Entropy

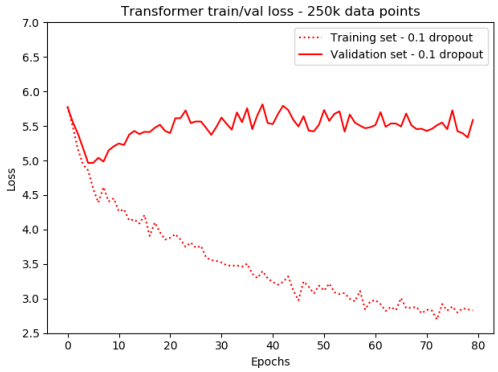

Figure 3.8: Experiment T1. The graph shows the loss values of a transformer trained over 80 epochs. Overfitting starts occurring after 7 epochs, as can be seen on the validation loss over time.

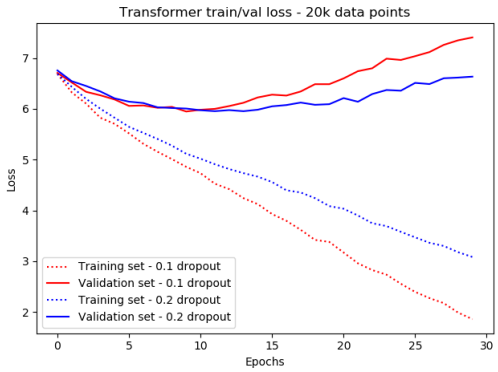

Figure 3.9: Experiment T2. Two dropout rates, 0.1 and 0.2, were tested to see how the loss would differ over epochs. Both tests seem to lead to over fitting after 8 epochs.

loss using Adam optimizer and a 0.1 dropout rate for regularization. The recorded loss values for this experiment, called T1, can be seen in Fig. 3.8. In order to minimize the validation loss it was necessary to avoid overfitting the model, which started occurring at 7 epochs for T1.
Several experiments were employed to see how overfitting could be minimized. These were trained for less epochs, as well as using smaller amounts of data in order to save training time. For every experiment, several dropout rates were tested as well.

The losses for experiment T2 are shown in Fig. 3.9. Drop out rates 0.1 and 0.2 were tested. The next experiment, T3, was then performed with the same settings except more data points, see Fig. 3.10.

The number of data points was doubled and two more dropout rates were tested for the next experiment, T4. After this experiment, it was decided that a dropout rate of 0.1 seemed the most feasible to continue testing with, as it showed to have the lowest loss value for all experiments and only one dropout rate could be further tested due to time constraints. The loss values over time for this experiment can be

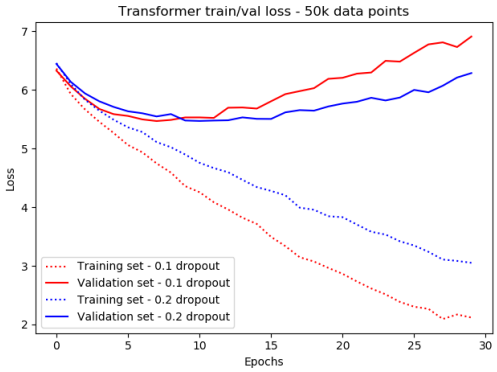

Figure 3.10: Experiment T3. Same experiment setting as T2, except with 50 thousand data points. A dropout rate of 0.1 seems to start over fitting earlier than a rate of 0.2, but has the lowest value overall.

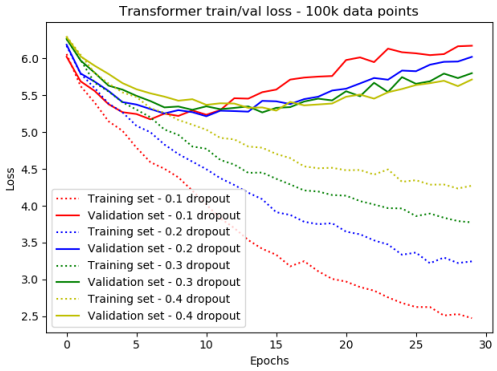

Figure 3.11: Experiment T4. Four dropout rates were tested, ranging from 0.1 to 0.4. As can be seen, a higher drop out rate seems to lead to later over fitting, but overall higher loss.

seen in Fig. 3.11.

The final settings were used for experiments T5 and T6, 200 thousand datapoints and 500 thousand datapoints to serve as indications of how many epochs would be optimal to train the model for, according to the loss values. For T5, this was at ~ 7 epochs, and for T6 this was at 25 epochs, see Fig. 3.12 and 3.13.

3.6 Hardware

Models were trained on a P3 Double Extra Large AWS instance with 61 GiB of memory, 8 Intel Xeon Scalable (Skylake) vCPUs and one NVIDIA V100 Tensor Core GPU.

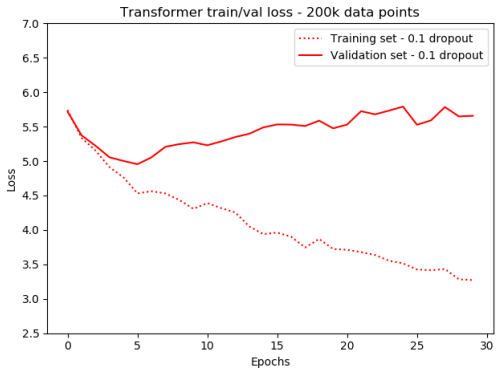

Figure 3.12: Experiment T5. Same settings as experiment T4, with the exception of only testing a dropout rate of 0.1 as well as more data points.

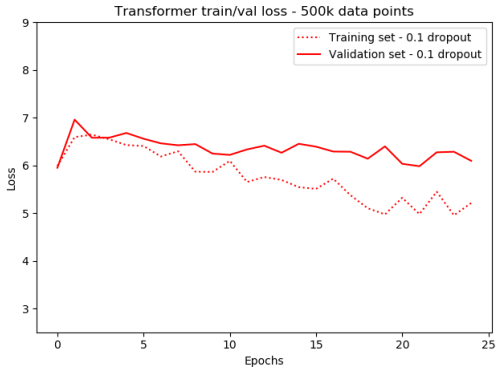

Figure 3.13: Experiment T6. The largest experiment in terms of number of data points. The only experiment where overfitting does not seem to occur.

4

Results & Discussion

4.1 Evaluation - ROUGE

The models were evaluated using Recall-Oriented Understudy for Gisting Evaluation (ROUGE), the most commonly used metric for text summarization developed by Lin [7]. It measures similarity of text between a given summary and the target by several measures. The first measure that were used for this project was ROUGE-N, which calculates the overlap of n-grams. It is formally defined in [7] as

$$ROUGE - N = \frac{\sum_{S \in RefSum} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in RefSum} \sum_{gram_n \in S} Count(gram_n)} \tag{4.1}$$

where $Count_{match}(gram_n)$ is the number of matches of n-grams in a candidate summary compared to a reference summary . $Count(gram_n)$ is the total number of grams of length n in the reference summary. This entails that in the base model of the ROUGE, the recall is calculated. The precision and f-measure, the combination of precision and recall, can also be calculated.

ROUGE-L is a measure of the longest common subsequence (LCS) found in both candidate summary and reference summary. Cormen et al. [10] defines a subsequence as follows: given a sequence $X = \{x_1, x_2, ..., x_m\}$, another sequence $Z = \{z_1, z_2, ..., z_m\}$ is a subsequence of X if there exists a strictly increasing sequence $\{i_1, i_2, ..., i_k\}$ of indices of X such that for all $j = 1, 2, ..., k$, we have $x_{ij} = z_j$. Lin [7] proposes using LCS-based F-measure to estimate the similarity between two summaries. Variations of LCS have previously been used by Melamed et al. [5] and Saggion et al. [4] to evaluate similarities in texts.

ROUGE-L scores are calculated in terms of precision, recall and a combined f-measure as

$$R_{lcs} = \frac{LCS(X, Y)}{m} \tag{4.2}$$

$$P_{lcs} = \frac{LCS(X, Y)}{n} \tag{4.3}$$

$$F_{lcs} = \frac{(1 + \beta^2) R_{lcs} P_{lcs}}{R_{lcs} + \beta^2 P_{lcs}} \tag{4.4}$$

ROUGE is not a complete metric, especially in terms of abstractive summarization. A summary may receive a low ROUGE-score, even if it is good, if it deviates from a target summary due to rewording or usage of synonoms. A high ROUGE-score does not necessarily mean a summary is good either since this does not account for grammatical errors or readability. There are also no reference scores since a novel data set was used. The scoring does however serve as a good indicator of performance between experiments.

4.2 Results - seq2seq

It was not possible to get neither the LSTM variant or the IndRNN to converge on the Meltwater data set. They yielded similar output regardless of what the input was. In addition to this, the models had not learned any textual semantics or grammar, and in some cases did not manage to produce real words either. For some instances the output only contained a handful of words that were repeated: "The she and and and and ...". This was a common pattern for all instances, while in some cases the models had picked up certain specific words that likely were common in the training data: "Trump and find convention in oregon and trump and convention in oregon and ...". Some reasons that might have led to this result are

- The models were not big enough, in terms of layers and dimensions, to capture the complexities of the data.
- Data was encoded and decoded on token level ⇒ Models had to learn semantics just to form words ⇒ Hard for models to learn semantics on a higher level.
- Architecture design limited ⇒ RAM memory issues ⇒ Not enough training data used.
- Decoding not sophisticated enough. No capping mechanism for output length.

For this reason, it was deemed irrelevant to evaluate the ROUGE scores for the seq2seq models.

4.3 Results - Transformer

The Transformer was evaluated by the ROUGE metric as well as through inspecting a number of summaries produced after processing inputs from a test set.

4.3.1 ROUGE scores

Following are tables with compiled ROUGE scores calculated for different models of the Transformer. Models based on recurrence were omitted in the comparison as discussed in section 4.2. Two models were trained on 200 thousand data points for 8 and 30 epochs, and the other two models were trained on 500 thousand data

points for 7 and 25 epochs respectively. The choice of number of epochs to compare was made in order to understand the correlation between loss and ROUGE scores. At ~ 8 epochs the loss was at its lowest for the model based on 200 thousand data points, and at ~ 30 epochs the model was overfitted at its most, see sub section 3.5. A corresponding choice of epochs was made for the Transformer trained on 500k data points.

Precision			
Model	ROUGE-1	ROUGE-2	ROUGE-l
200kdps_8epochs	0.162	0.032	0.142
200kdps_30epochs	**0.203**	**0.071**	**0.179**
500kdps_7epochs	0.161	0.028	0.141
500kdps_25epochs	**0.203**	0.052	0.176

Recall			
Model	ROUGE-1	ROUGE-2	ROUGE-l
200kdps_8epochs	0.135	0.028	0.119
200kdps_30epochs	**0.202**	**0.074**	**0.179**
500kdps_7epochs	0.141	0.027	0.124
500kdps_25epochs	0.187	0.051	0.163

F-score			
Model	ROUGE-1	ROUGE-2	ROUGE-l
200kdps_8epochs	0.142	0.029	0.117
200kdps_30epochs	**0.198**	**0.072**	**0.169**
500kdps_7epochs	0.145	0.027	0.120
500kdps_25epochs	0.189	0.050	0.157

The resulting ROUGE scores clearly show an improvement for both versions of the Transformer when trained on a higher number of epochs. This is expected for the model trained on 500 thousand data points as the validation loss was lower at 25 epochs than 7 epochs. However, this was also the case for the model trained on 200 thousand data points despite the fact that it started over fitting at 8 epochs, meaning that the validation loss was higher for the version trained for 30 epochs even though the ROUGE score was also higher. This indicates that there might not be a strong correlation between the loss function and the evaluation metric, as the ROUGE score should decrease as the loss increases.

It would be possible to change the loss function to optimize against ROUGE scores instead of cross-entropy. The problem with this is that it is not strictly clear what defines a good summary. For example, optimizing against the precision of ROUGE-1 could lead to the model learning to only produce a single word that is in the target, as this would lead to a high score. If one were to optimize against ROUGE it would likely be a better idea to optimize against a combination of precision, recall and f-scores of ROUGE-1, ROUGE-2 and ROUGE-l and possibly more n-grams. A potential problem with this approach is that it does not necessarily enforce an abstractive summarization, i.e. one that is rewritten from scratch. Such a summary could contain synonyms and rewordings of the target, while being of high quality

if evaluated by humans. In other words, this would not necessarily lead to a high ROUGE score, even though the result might be desirable.

There might be many reasons as to why the models in this work behave counter-intuitively in certain cases, and that is also a key problem in this domain. It is a commonly agreed fact that complex systems such as Deep Learning models are hard to understand, due to their abstractive inner workings. In order to understand the models produced by this work better, it would likely prove helpful to employ additional evaluation metrics. One metric that would be highly relevant toward the end goal of the project, producing summaries that Meltwater customers find helpful, is to conduct a survey of linguistic quality like the DUC-version[1] used by Liu et al. [32] in their work of building a Wikipedia generator.

4.3.2 Example summaries

Following are a selection of the kinds of summaries the Transformer were able to produce. All four models described above produced similar results. The example outputs below are taken from the model trained on 200 thousand data points for 8 epochs, which yielded the highest ROUGE scores.

Example 1
Input One person has been killed in a crash at Gaven early Sunday morning.A four-wheel-drive was travelling on Castle Hill Dr around 2.15am when it crashed into a tree, catching fire.The driver was killed at the scene.Police are asking anyone with dashcam footage of the crash to come forward.There is reports that another car may have been involved and fled the scene.If you have information for police, contact Policelink on 131 444 or online.You can report information about crime anonymously to Crime Stoppers by calling 1800 333 000 or via crimestoppersqld.com.au 24hrs per day.

Target Police Are Investigating

Output police are appealing for witnesses to a crash

[1] https://duc.nist.gov/duc2007/quality-questions.txt

Example 2
Input A man in his 70s riding a scooter has been taken to hospital with life-threatening injuries following a collision in the west end on Wednesday morning.Emergency crews were called to the area of Ossington Avenue and College Street around 9:40 a.m. for reports of a two-vehicle collision.According to Toronto paramedics, the man on the scooter was hit by a truck.He was taken to a trauma centre in serious condition. Police tweeted the man suffered life-threatening injuries.More to comeA man in his 70s riding a scooterhas been taken to hospital with life-threatening injuries following a collision in the west end on Wednesday morning.Emergency crews were called to the area of Ossington Avenue and College Street around 9:40 a.m. for reports of a two-vehicle collision.According to Toronto paramedics, the man on the scooter was hit by a truck.He was taken to a trauma centre in serious condition. Police tweeted the man suffered life-threatening injuries.More to comeMore to come

Target Emergency crews were called to area of Ossington Avenue and College Street around 9:40 a.m. Wednesday

Output man taken to hospital with serious injuries

Example 3
Input A 23-year-old man is dead after a two-vehicle crash in Edson, Alta. on Wednesday.The collision happened around 3:30 p.m. on the east side of the town.RCMP said the deceased was the lone occupant of a Honda Civic going west on the Highway 16 overpass. The man, who police are not naming, was pronounced dead on scene.The other vehicle was an eastbound Dodge Ram truck, police said. The driver has minor injuries.Both lanes of traffic are being rerouted as RCMP continue to investigate.A 23-year-old man is dead after a two-vehicle crash in Edson, Alta. on Wednesday.The collision happened around 3:30 p.m. on the east side of the town.RCMP said the deceased was the lone occupant of a Honda Civic going west on the Highway 16 overpass. The man, who police are not naming, was pronounced dead on scene.The other vehicle was an eastbound Dodge Ram truck, police said. The driver has minor injuries.Both lanes of traffic are being rerouted as RCMP continue to investigate.

Target Other driver sustained minor injuries, police say

Output driver of the vehicle was pronounced dead at the scene, police say

Example 4
Input A young man is dead and a young woman is fighting for her life after their car collided with a Land Cruiser in Tasmania.The Hyundai sedan was being driven along the Arthur Highway at Dunalley when it crossed on to the wrong side of the road, and hit the Toyota heading in the opposite direction at about 1.50pm on Thursday, police say.An 18-year-old man was killed instantly and the 18-year-old woman driving was also seriously injured.She was treated at the scene before being flown to Royal Hobart Hospital in a critical condition.The 58-year-old driver and his 56-year-old passenger in the Toyota, both from interstate, were taken to Royal Hobart Hospital for treatment, where one remains in a stable condition.The Arthur Highway was closed for four hours while police inspected the scene.

Target A young man has been killed and a woman has been seriously injured in a two-car crash in Tasmania.

Output a man has died after being hit by a car in the south of brisbane.

Example 5
Input Can't play this file? Click here to download/listen -A 17 year old from Bradford was treated for a stab wound, though South Simcoe Police say the teen waited until the next day to go to the hospital.Police learned the youth had been stabbed during an altercation on Saturday night outside his home.There have been no arrests.

Target A 17 year old who was stabbed in Bradford waited a day before seeking treatment.

Output police are investigating a fatal stabbing.

Example 6

Input Media ContactCompany Name: Data Bridge Market Research Private LimitedContact Person: Sopan GedamEmail: Send EmailPhone: +1-888-387-2818Address:Office Number 402, Amanora Chambers, Magarpatta Road, HadapsarCity: PuneState: MaharashtraCountry: IndiaWebsite: http://databridgemarketresearch.com/reports/global-vitamin-mineral-supplements-market/Press Release Distributed by ABNewswire.comTo view the original version on ABNewswire visit: Vitamin, Mineral and Supplements Market Analysis By Vitamin Type (Thiamin), Vitamin B2, (Niacin), (Ascorbic Acid), By Mineral Type, Supplements Ingredient, Supplements and forecast 2025Information contained on this page is provided by an independent third-party content provider. Frankly and this Site make no warranties or representations in connection therewith. If you are affiliated with this page and would like it removed please contact pressreleases@franklyinc.com

Target 8220#Vitamin, Mineral and Supplements Market%8221Data Bridge Market Research brings to you this report on the Global Vitamin, Mineral and Supplements Market Report is a compilation with...

Output 8220 8211 world%8217 s fastest growing market research database8221 according to new report available with million insights, data 8211 world8217 s...

Example 7

Input A 29-year-old woman from Geelong was found dead inside a burning tent at the local Breakwater Road camping ground about 2am on Tuesday."The three were arrested in relation to the ongoing investigation," a Victoria Police spokesman told AAP.Heavily-armed officers pounced on the trio at a Greystanes address on Thursday afternoon.NSW Policeś tactical unit was deployed because there were fears the group was armed.Footage aired by Seven Network showed the trio being questioned by detectives and assessed by paramedics while handcuffed on the front lawn of the Gipps Street home.Victorian homicide detectives are expected to travel to Sydney to question the trio.

Target Two men and a woman have been arrested in Sydney in connection with a suspicious death in Victoria.

Output two people have been arrested after a woman was found dead in a sydney road in sydney 's south west.

Evaluating the model by looking at some of the outputs it yielded show promising results. In many cases, the output was considered of high quality, and in some instances even arguably better than the target. Examples 1, 2 and 3 are such cases.

It seems like the model is somewhat proficient in understanding the general context of an input, but when the text contains details such as time stamps, numbers or locations the model often produces the wrong information. This is the case for example 4, where the model predicts the location for a certain event to be in Brisbane while the true location was Tasmania. Although both locations are in the same country, the precision could be better. Since the model has developed a geographical sense well enough to infer the correct country in some cases, it might be possible to increase its precision even more by feeding it more data.

By inspecting a portion of the test set it was also discovered that the data was in many cases of lackluster quality. Example 5 contains text that seems to come from buttons on the web page it was crawled from. Despite this the data point contained enough text on the form of an article for the model to be able to generate an output of arguably good quality.

A different case of a bad data point is seen in example 6, where the input consists of contact info, legal information and random text. This is likely due to how the web scraping tools were designed as well as the web sites that were processed, and it might be a hard problem to solve. It is unclear what kind of effect data points like these has had on the model.

Lastly, example 7 might appear to be a good data point upon initial inspection, but is hard to understand when reading it. It does not seem to be manually produced by a writer, and could possibly have been auto-generated. This is an additional indicator of the weakness in the data set that was used.

For more examples, see Appendix.

5
Conclusion & Future work

This work has shown that the Transformer was successfully implemented to summarize news articles on the format of Meltwater data with good performance. The Transformer seemed to perform especially well at understanding general context of articles and generating text involving the most relevant parts, but was not as proficient in reproducing detailed information such as time, location and numbers with high precision. It was discovered that the quality of the generated summaries were heavily dependent on the quality of the input data, indicating that the model implementation might be good, but bottle necked by the data set. The data set was cleaned and adjusted using a handful of methods, but despite this was found to contain some low quality articles resulting from the design of the web scraping tools used to generate it. It is unclear what effect these data points had on the model, which makes it difficult to conclude the feasibility in using the provided data set to train a text summarizer. Furthermore it was discovered that recurrent models based on seq2seq were hard to train and did not show as promising results.

For future work, specifically in developing a summarizer of clusters of documents, a recommended approach would be to combine an abstractive summarizer with extractive methods, where the Transformer would serve as a strong candidate for abstractive summarization. Improving the performance of the Transformer could be done by training it on a different, or additional data set such as *CNN/Daily Mail* created by Hermann. et al [15]. It could also be improved by using pre-trained word embeddings such as Google BERT [27]. Additionally, Vasvani et al. [24] have shown that the Transformer can be trained on a larger data set by using several TPUs if needed. Some recommended extractive methods to evaluate in such work would be TfIdf [6], TextRank [8] and SumBasic [9] for filtering out less relevant paragraphs of a text corpara, as used by Liu. et al [32]. Integrating the extractive and abstractive methods together could hopefully result in even better text summarizations for Meltwater's data set. Finally, as discussed in the results section, ROUGE cannot be considered a complete evaluation metric. Therefore it is advised to conduct research in additional evaluation methods for summarization models.

References

[1] Jeffrey L. Elman. "Finding structure in time". In: *Cognitive Science* 14.2 (1990), pp. 179 –211. ISSN: 0364-0213. DOI: https://doi.org/10.1016/0364-0213(90)90002-E. URL: http://www.sciencedirect.com/science/article/pii/036402139090002E.

[2] Michael C. Mozer. "Backpropagation". In: ed. by Yves Chauvin and David E. Rumelhart. Hillsdale, NJ, USA: L. Erlbaum Associates Inc., 1995. Chap. A Focused Backpropagation Algorithm for Temporal Pattern Recognition, pp. 137–169. ISBN: 0-8058-1259-8. URL: http://dl.acm.org/citation.cfm?id=201784.201791.

[3] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (1997), pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735. eprint: https://doi.org/10.1162/neco.1997.9.8.1735. URL: https://doi.org/10.1162/neco.1997.9.8.1735.

[4] Horacio Saggion et al. "Meta-evaluation of Summaries in a Cross-lingual Environment using Content-based Metrics." In: Jan. 2002. DOI: 10.3115/1072228.1072301.

[5] I. Dan Melamed, Ryan Green, and Joseph P. Turian. "Precision and Recall of Machine Translation". In: *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: Companion Volume of the Proceedings of HLT-NAACL 2003–short Papers - Volume 2*. NAACL-Short '03. Edmonton, Canada: Association for Computational Linguistics, 2003, pp. 61–63. DOI: 10.3115/1073483.1073504. URL: https://doi.org/10.3115/1073483.1073504.

[6] Juan Ramos. "Using TF-IDF to determine word relevance in document queries". In: (Jan. 2003).

[7] Chin-Yew Lin. "ROUGE: A Package For Automatic Evaluation Of Summaries". In: *ACL 2004*. 2004.

[8] Rada Mihalcea and Paul Tarau. "TextRank: Bringing Order into Text". In: 2004.

[9] Ani Nenkova and Lucy Vanderwende. *The impact of frequency on summarization*. Tech. rep. Microsoft Research, 2005.

[10] Thomas H. Cormen et al. *Introduction to Algorithms, Third Edition*. 3rd. The MIT Press, 2009. ISBN: 0262033844, 9780262033848.

[11] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural Machine Translation by Jointly Learning to Align and Translate". In: *arXiv e-prints*, arXiv:1409.0473 (2014), arXiv:1409.0473. arXiv: 1409.0473 [cs.CL].

[12] Kyunghyun Cho et al. “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation”. In: *arXiv e-prints*, arXiv:1406.1078 (2014), arXiv:1406.1078. arXiv: 1406.1078 [cs.CL].

[13] Junyoung Chung et al. “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling”. In: *arXiv e-prints*, arXiv:1412.3555 (2014), arXiv:1412.3555. arXiv: 1412.3555 [cs.NE].

[14] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. “Sequence to Sequence Learning with Neural Networks”. In: *arXiv e-prints*, arXiv:1409.3215 (2014), arXiv:1409.3215. arXiv: 1409.3215 [cs.CL].

[15] Karl Moritz Hermann et al. “Teaching Machines to Read and Comprehend”. In: *arXiv e-prints*, arXiv:1506.03340 (2015), arXiv:1506.03340. arXiv: 1506.03340 [cs.CL].

[16] Yann LeCun, Y Bengio, and Geoffrey Hinton. “Deep Learning”. In: *Nature* 521 (May 2015), pp. 436–44. DOI: 10.1038/nature14539.

[17] Konstantin Lopyrev. “Generating News Headlines with Recurrent Neural Networks”. In: *arXiv e-prints*, arXiv:1512.01712 (2015), arXiv:1512.01712. arXiv: 1512.01712 [cs.CL].

[18] Alexander M. Rush, Sumit Chopra, and Jason Weston. “A Neural Attention Model for Abstractive Sentence Summarization”. In: *arXiv e-prints*, arXiv:1509.00685 (2015), arXiv:1509.00685. arXiv: 1509.00685 [cs.CL].

[19] Sumit Chopra, Michael Auli, and Alexander M. Rush. “Abstractive Sentence Summarization with Attentive Recurrent Neural Networks”. In: *HLT-NAACL*. 2016.

[20] Ramesh Nallapati et al. “Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond”. In: *arXiv e-prints*, arXiv:1602.06023 (2016), arXiv:1602.06023. arXiv: 1602.06023 [cs.CL].

[21] Yonghui Wu et al. “Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation”. In: *arXiv e-prints*, arXiv:1609.08144 (2016), arXiv:1609.08144. arXiv: 1609.08144 [cs.CL].

[22] Mehdi Allahyari et al. “Text Summarization Techniques: A Brief Survey”. In: *arXiv e-prints*, arXiv:1707.02268 (2017), arXiv:1707.02268. arXiv: 1707.02268 [cs.CL].

[23] Romain Paulus, Caiming Xiong, and Richard Socher. “A Deep Reinforced Model for Abstractive Summarization”. In: *arXiv e-prints*, arXiv:1705.04304 (2017), arXiv:1705.04304. arXiv: 1705.04304 [cs.CL].

[24] Ashish Vaswani et al. “Attention Is All You Need”. In: *CoRR* abs/1706.03762 (2017). arXiv: 1706.03762. URL: http://arxiv.org/abs/1706.03762.

[25] Zijun Yao et al. “Dynamic Word Embeddings for Evolving Semantic Discovery”. In: *arXiv e-prints*, arXiv:1703.00607 (2017), arXiv:1703.00607. arXiv: 1703.00607 [cs.CL].

[26] Arman Cohan et al. “A Discourse-Aware Attention Model for Abstractive Summarization of Long Documents”. In: *arXiv e-prints*, arXiv:1804.05685 (2018), arXiv:1804.05685. arXiv: 1804.05685 [cs.CL].

[27] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *arXiv e-prints*, arXiv:1810.04805 (2018), arXiv:1810.04805. arXiv: 1810.04805 [cs.CL].

[28] Sebastian Gehrmann, Yuntian Deng, and Alexander M. Rush. "Bottom-Up Abstractive Summarization". In: *arXiv e-prints*, arXiv:1808.10792 (2018), arXiv:1808.10792. arXiv: 1808.10792 [cs.CL].

[29] Wan-Ting Hsu et al. "A Unified Model for Extractive and Abstractive Summarization using Inconsistency Loss". In: *arXiv e-prints*, arXiv:1805.06266 (2018), arXiv:1805.06266. arXiv: 1805.06266 [cs.CL].

[30] Chandra Khatri, Gyanit Singh, and Nish Parikh. "Abstractive and Extractive Text Summarization using Document Context Vector and Recurrent Neural Networks". In: *arXiv e-prints*, arXiv:1807.08000 (2018), arXiv:1807.08000. arXiv: 1807.08000 [cs.CL].

[31] Shuai Li et al. "Independently Recurrent Neural Network (IndRNN): Building A Longer and Deeper RNN". In: *arXiv e-prints*, arXiv:1803.04831 (2018), arXiv:1803.04831. arXiv: 1803.04831 [cs.CV].

[32] Peter J. Liu et al. "Generating Wikipedia by Summarizing Long Sequences". In: *arXiv e-prints*, arXiv:1801.10198 (2018), arXiv:1801.10198. arXiv: 1801.10198 [cs.CL].

[33] Serena Yeung Fet-Fei Li Justin Johnson. *Lecture 10 | Recurrent Neural Networks*. Stanford University. URL: https://www.youtube.com/watch?v=6niqTuYFZLQ.

A

Appendix 1

A.1 Additional example summaries

Example 8 (Slight piece of context understood):

Input A proposed U.S.-Mexico trade deal would allow President Donald Trump to impose punitive tariffs of up to 25 percent on imports of Mexican-made cars, SUVs and parts, auto executives and sources told Reuters.

Target Mexico-U.S. deal said to include Mexican auto export cap

Output usa trade

Example 9 (Bad data point):

Input Job Description:CrossFit 630 near Downtown Naperville is expanding our Boot Camp 630 service and we are looking to add a dynamic, high energy coach to our training teamBoot Camp Classes, NOT CrossFit ClassePart Time - 2 Classes on each Saturday and Sunday1-Hour ClassesCompetitive Hourly Compensation (15−25 per class depending on experience)Intrapreneur OpportunitiesCommission OpportunitiesQualifications NeededPT Certification or CF L1 Training Certificate1+ Years Leading Group ClassesCPR/AED CertificationAbility to Program WorkoutsFor more information, please contact nate@crossfit630.com or call (630) 791-0498.

Target Apply to be a Personal Trainer (Naperville) - Part Time in Aurora, IL on the Patch Jobs board.

Output apply to be a part time maintenance coach in north end, ma on the patch jobs board.

Example 10 (Good output, wrong time detail):

Input A motorcyclist has been left with serious leg injuries after a crash during Friday night rush hour.The motorbike collided with a van at about 6pm on the A39 between Barnstaple and Bideford, resulting in the road being blocked in both

directions.The air ambulance reportedly landed on the link road between the A386 and the B3232 to tend to the stricken man.He was taken to North Devon District Hospital.There were no other injuries reported in the crash and the road was cleared by 8.30pm after the vehicles were recovered.A second collision happened in the same area of the A39 at about 6.30pm on Friday night involving a Range Rover and a Honda Civic.No injuries were reported to police.

Target The crash was one of two on the same stretch of road within half an hour of each otherThe scene of the crash of the A39

Output the crash happened just after 3 am

Example 11 (Interesting example. The model inferred the location by itself and almost got it right):

Input Police investigating an attempted robbery at Gosford last Friday night have released CCTV images of a man they'd like to speak with.A 70-year-old woman was approached by a young man who attempted to steal her handbag as she was walking along a pathway on Showground Road, near the Railway Station at about 6.30 pm.He's described as being in his late teens, thin, about 165 cm tall and Caucasian in appearance.Call NSW Police if you can assist.

Target Attempted bag snatch in Gosford

Output police are looking for a man who robbed a woman in melbourne 's south west.

Example 12 (Long article, almost perfect summary):

Input SHELBY TOWNSHIP - Shelby Township police are investigating a case of indecent exposure and assault that happened around 9:15 p.m. Sunday, Aug. 26.The incident happened on 22 Mile Road near Shelby Road. According to the victim a silver vehicle with black trim, possibly a Jeep Liberty, followed her from River Bends Park.The suspect passed the victim and eventually stopped in front of her vehicle. A male then exited the vehicle and started to approach the victim. After approaching the woman he reached inside the window of her car and touched her leg. The suspect then started to expose himself. When the woman attempted to leave the scene the man tried jumping into her vehicle. The suspect fled the area.He has been described as a black man, about 26 years old, 6 feet tall, with a thin build and shaved hair.If you have any information on the case contact Shelby Township police at 586-731-2121.

Target Police searching for suspect in indecent exposure, assault case

Output police searching for suspect in connection with assault, assault

Example 13 (Only slight miss in details):

Input RCMP are investigating the suspicious death of a 52-year-old man near Tracadie.His body was discovered Monday at an apartment in a home on Benoit Street, Cpl. Jullie Rogers-Marsh said in a statement Tuesday morning.Police were called to the scene shortly before 3 p.m., she said."At this time, the death is considered suspicious until police are able to gather more information."An autopsy will be conducted to determine the manś cause of death, said Rogers-Marsh.No other details have been provided.The investigation is ongoing.

Target 52-year-old man's body discovered in an apartment Monday afternoon

Output the body was found in a apartment in the early hours of tuesday morning

Example 14 (Close to target, relevant):

Input A balcony at an abandoned house in Sliema collapsed in the early hours of Saturday morning, as rain and wind lashed the Maltese islands.Limestone bricks spilled out onto the middle of Triq Patri Marjanu Vella as a result of the collapse, though thankfully nobody was hurt and no cars damaged in the incident.Authorities removed the bricks and debris on Saturday morning.

Target Nobody hurt in overnight incident

Output no injuries reported in incident

Example 15 (Good summary):

Input Get Daily updates directly to your inbox+ SubscribeThank you for subscribing!Could not subscribe, try again laterInvalid EmailA multi-vehicle crash on the A12 Westbound has partially blocked the road causing heavy traffic for commuters this morning.The accident happened between junction 24 on the B1024 at Kelvedon North and junction 23 on the B1024, London Road at Kelvedon South earlier today (June 5) at around 6.50am.It is thought the incident involved three vehicles.As a result, the road is partially blocked causing traffic to queue for around five miles.The accident is affecting traffic travelling between Long Green and Kelvedon.Motorists are urged to take caution in this area or find an alternative route.For the latest news and breaking news visit essexlive.news

Target Traffic is queuing for around five miles

Output traffic is causing delays

Example 16 (Good summary):

Input NOCATEE, Fla., - A St. Johns County Sheriffś deputy hit a pedestrian at the intersection of Palm Valley Road and Ranch Road early Saturday morning, according to the Florida Highway Patrol.Deputy Richard Thatcher, 54, was driving eastbound on Palm Valley Rd. just after 1:30 a.m. That is when a man "darted out" into his path from the westbound shoulder, according to the crash report.More HeadlinesPedestrian critically injured in Arlington hit-and-run, police sayMan hit, killed on San Jose Boulevard, Jacksonville police saySt. Augustine pedestrian hit, driver arrested say FHPTroopers said Deputy Thatcher did not have time to react and hit the pedestrian, identified as Robenson Watson, 24, with the left corner of his Ford Taurus. Watson was taken to Memorial Hospital with serious injuries, FHP said.Troopers said it is unclear why Watson walked into the path of the car.

Target FHP: Deputy was in police cruiser when man darted out in traffic

Output driver hit by car, police say

Example 17 (Output quite different from target. Arguably good summary despite this):

Input A robbery suspect who fled to the GTA on a GO train to escape Barrie police Sept. 17 has been arrested on unrelated charges in Toronto.Police have been looking for Ashtoney Thompson, 21, ever since four Toronto suspects allegedly robbed a man near Blake and St. Vincent Streets.Three suspects were arrested after their getaway car crashed in a nearby park, but Thompson managed to flee on foot.A 17-year-old female, 16-year-old female, and 21-year-old man, all from Toronto, were charged with robbery-related offences.The incident also led to an ongoing investigation by the SIU after a Barrie officer allegedly fired a single shot at the fleeing vehicle.

Target Three other suspects also charged

Output toronto police arrested a man in connection with a robbery in toronto on sunday.