Cut to the Chase: An Extractive Approach to Machine Summarization: Final Report

Aykan Ozturk, Andrew Quirk, Ian Hodge

December 15, 2017

1 Introduction

In our digital age, we are constantly bombarded with a seemingly infinite amount of information through sources such as social media, news articles, and instant messaging. Many times, the amount of diverse information (combined with increasingly shorter human attention spans) can lead to an apathy towards reading large amounts of text. This feeling is so common, that it has lead to the creation of the colloquial expression 'tl;dr' (too long didn't read) which refers to short, 3-5 sentence summaries of long sources of information. For our project, our goal is to create an algorithm that will generate readable, and useful, 'tl;dr' summary of a news article, automatically, given the article's original text.

This task can be divided into three main sections:

- Content Selection: Choosing the most relevant sentences from the article.
- Ordering: Deciding which order to put the chosen sentences in.
- Pruning: Remove unnecessary information from the sentences to find the most.

Arguably the most important (and possibly challenging) part of this problem is content selection- deciding which of the sentences will go into the summary. As a result, most of the work that we did was building a useful and tractable model for the analysis of the articles. Once this step had been accomplished, we saw that the order was mostly irrelevant, since the reader doesn't know what order the article was in. Unless the events occurred strictly chronologically, the summaries were strong representations of the article without any tinkering, so we focused our efforts on improving our selection algorithm. Pruning could augment the summaries we generated, and would be a logical next step of the project.

The two main approaches to content selection are unsupervised and supervised learning, and perhaps surprisingly, they yield similar results in general. Recently with more complex deep neural network architectures, supervised learning approaches have started to outperform unsupervised approaches. Also, problems that accompany supervised learning for content selection in general don't really apply to our case as much as other scenarios because our robust dataset provides strong training examples to train the model. As a result, we decided to follow a supervised learning approach to leverage our dataset and get better results.

In the next section we will first describe our model in detail. Then we will describe the necessary preprocessing steps to be able to train our model and learn the parameters. After that we will briefly explain the learning algorithm. Finally, we will present our results and discuss some of the challenges we faced along with a literature review and error analysis.

2 Key Metrics

We will be using ROUGE metrics to evaluate the performance of our models. ROUGE stands for Recall-Oriented Understudy for Gist Evaluation. ROUGE metrics are commonly used to measure the performance of automatic text summarization systems in the literature, so using them allows for comparison with related work. There are multiple evaluation metrics within the ROUGE family, such as ROUGE-N, ROUGE-L and ROUGE-S. The main metric we are planning to focus on is ROUGE-N. ROUGE-N measures the overlap of N-grams between the predicted summary and the reference summary. Assume $N = 2$ for example. In this case, ROUGE-2 generates all bigrams in the generated summary and all bigrams in the reference summary. Then

$$ROUGE2_{recall} = \frac{\text{number of overlapping bigrams}}{\text{total number of bigrams in reference summary}}$$

and

$$ROUGE2_{precision} = \frac{\text{number of overlapping bigrams}}{\text{total number of bigrams in generated summary}}.$$

We can also use the harmonic mean of precision and recall to define a $ROUGE2_{F1}$ score.

3 Data

Our main dataset will is a collection of 219,000 Daily Mail articles, along with their human-generated summaries. This dataset is extremely valuable because it provides us with genuine articles and summaries to compare against. Most recent papers on the subject of summarization use this same dataset and the ROUGE framework, so we will be able to sanity check against these results. In particular, the articles include a variety of subject matter and topics, so they provide a good cross-section of human-generated news content to analyze. The summarizations of the articles are also a good evaluative metric because of their differing length and complexity. Some summaries include only simple facts about the article, while others go into a more in-depth analysis of the content. Along those lines, the articles also vary a great deal in length. Longer articles are harder to summarize as a result of the volume of material to cover.

4 GloVe

We used the Stanford GloVe library, which provides us with pre-trained word vectors. The GloVe vectors we used were extracted from about 400,000 words from Wikipedia 2014 and Gigaword 5 with dimensionality 100. GloVe scores as high as 75 percent on word analogy tasks and is a good tool to use in analyzing "natural" word ordering. We use this library as a benchmark to construct our summaries in an accurate way.

5 Preprocessing

The first step is preprocessing. There are two main parts of the preprocessing step. We first went through all of the given human-provided summaries, and for each summary looped through each sentence of the article to find which article sentence matched most closely with the summary sentence. In order to determine which sentence most closely resembled the summary sentence, we used the harmonic mean of the ROUGE-2 recall and precision scores. The article sentence with the highest ROUGE score for a given summary was marked as having important content. From here, every article was written to a new file where each sentence was marked with either a '1' for having important content and a '0' otherwise.
Then the second step of preprocessing is to generate a vocabulary file from the dataset by looping over all the documents and detecting unique words. Then, we use this vocabulary file to create a GloVe matrix by finding the GloVe embedding associated with each word in the vocabulary. If there is no GloVe embedding for a given word in our vocabulary, then we assign a random fixed vector to the word (which signifies that it is an unknown word, or an UNK token). Then in training or test time, when we read in an article, we use this GloVe matrix as a lookup table to find the embeddings associated with each word in the article.

6 Model

We consider a dataset of the form (X_{ij}, Y_{ij}) where X_{ij} is sentence j in article i of the dataset, and $Y_{ij} \in \{0, 1\}$ is a binary label which is 1 if X_{ij} is in the summary, 0 if it is not. We want to train a binary classifier (reflex-based model) to be able to infer the label of each sentence X_{ij} from its learned features $\phi(X_{ij})$.
As a sentence is a sequence of words, we first need a way to go from words to numbers (vectors). We used GloVe embeddings[**?**] to represent words in a sentence as numbers. More information on GloVe embeddings will be given in the following sections. Then we use a neural network on these GloVe embeddings to be able to learn more complex features that don't only treat each word individually, but considers their interactions within each sentence as well. We used an RNN encoder-decoder architecture[**?**] for this purpose.
To simplify notation, consider a single article x, and let $x_i \in R^{nxd}$ be the input vector for sentence i in x after using GloVe embeddings for each word in x_i. Here n is the number of words in x_i (or the maximum possible words in a given sentence to be more specific), and d is the GloVe embedding dimension. In the encoder, we insert x_i into an RNN to encode the sentence information:

$$o_i = RNN(x_i, h_{i0})$$

Here $o_i \in R^{nxk}$ where k is the state size of our RNN cell. o_i is basically the concatenation of all internal states h_{ij} of the RNN cell, each state corresponding to a single word. Details about the RNN cell will be described in the following sections.
The decoder we used is simpler, we simply have a linear layer followed by a ReLU activation:

$$\phi(x_i)_j = ReLU(o_j W + b_1)$$

where $W \in R^{kx1}$ and $b_1 \in R^1$. We repeat the same procedure for all words in the sentence, which means we finally end up with a feature vector $\phi(x_i) \in R^n$.
We feed this feature vector into a linear classifier that computes the probability that sentence i is in the summary as:

$$\hat{y} = \sigma(\phi(x_i)U + b_2)$$

where $U \in R^{nx1}$ and $b_2 \in R^1$. The sentence is classified as in the summary if $\hat{y} > 0.5$. We define our content as all sentences from an article that were classified as in the summary. This concludes content selection. Afterwards, we need to choose an ordering for the sentences that we selected. For now, we simply used the ordering in the article for simplicity. Finally there is the pruning step. For now we don't apply any pruning, we use each sentence directly in the final summary.

7 Learning

The next step is training, where we read articles sentence by sentence. We have labels for each sentence that we extracted in preprocessing. As we are implementing a binary classifier, we defined our loss function as the minus log likelihood function (or logarithmic loss). We applied mini-batch gradient descent instead of SGD. Also we used Adam optimizer[**?**] in TensorFlow to update weights instead of doing a simple SGD update. The Adam optimizer is different from classical gradient descent in that it leverages both the adaptive gradient algorithm and the root mean square propagation. It uses the exponential moving average of the gradient and the squared gradient, with parameters to control the decay rate of these moving averages.
To prevent overfitting, we regularized our network by using dropout[**?**] with drop probability 0.2 during training. Dropout optimizes the training algorithm while preventing units from co-adapting more than is necessary. Finally, we used a learning rate of 0.0001.

8 Baseline and Oracle

As a baseline approach, we decided to simplify the text summarization task as a keyword extraction task. In other words, we can generate a simple summary of the text by finding the keywords from an article and outputting them (in some order) without actually trying to form a fluent and meaningful sentence. Of course this is also not an easy task, but it is easier to work on as a baseline. To solve the keyword extraction task, we can simply define a feature vector $\phi(word) \in \mathbb{R}^3$ for each word in the article with the following features

$\phi_0(word) = 1$
$\phi_1(word) =$ Indicator of whether the word appears more than 5 times in the article
$\phi_2(word) =$ Indicator of whether the word appears more than 15 times in the article.

Then we can learn a weight vector $w \in \mathbb{R}^3$ using SGD to build a logistic regression model to classify each word as a keyword or not:

$$p(y(word)) = \sigma(w^T \phi(word))$$

so $\hat{y}(word) = 1$ (keyword) if $p(y(word)) > 0.5$.

We defined two different oracles. For the first, we decided to personally summarize five different articles and calculate the ROUGE-2 score.
Example, ROUGE-2 score of 1.64:
Article:
`Geoff Whitington, 63, from Ashford, Kent, was made so sick by obesity-related type 2 diabetes that he had been told he may need a leg amputated(...)Fixing Geoff's diabetes has brought the family closer, says Anthony. 'We went to Scotland over Christmas and Dad was out walking with the grandchildren, which would have been almost impossible two years ago. The kids love having a grandad who is active, and we've got our old dad back.'`

DailyMail given summaries:
`Geoff Whitington, 63, had diabetes and was on the verge of losing a leg.His sons Anthony and Ian helped the father-of-four shed six stone. He now loves to cycle, never eats take-aways and chooses healthy options.Doctors Mr Whitington is no longer diabetic and he is off medication.`
Our human (oracle) summary:
`Geoff Whitington, a dad from Kent, struggled with type 2 diabetes as a result of unhealthy eating patterns and lack of exercise. His sons decided to take matters into their own hands and help their father start to eat healthy and exercise. One of the sons is a filmmaker, so they decided to make a movie about it called Fixing Dad. Now Geoff is diabetes-free and travels the country speaking about his transformation and what made it possible.`

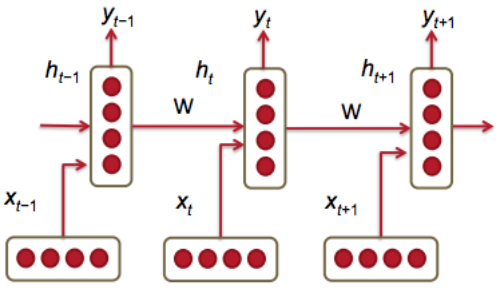
We reported an average ROUGE-2 score of 5.1 for these summaries.
This gives us a rough estimate for the average human performance (measured by ROUGE metric) on the text summarization task. However, as you can see the human generated summaries often report very low ROUGE-2 scores. As a result, we decided to define an additional Oracle where we evaluated the ROUGE-2 metric assuming that the classifier obtained 100% accuracy. This provided an upper bound on what we could expect from our model, and we obtained a ROUGE-2 score of 26.9.

9 Simple NN

10 RNN

We also trained our model with an RNN, which uses the state from the previous step and the next word vector in the document to find the next state h_t:

$$h_t = \sigma(h_{t-1}W^{(hh)} + x_tW^{(hx)})$$



Then we can use the state at each time step later to make our predictions. This model provides a good start to what we hope to accomplish and should give us a good idea of which sentences in the article are the ones we should use for the summary. RNNs sometimes suffer from the vanishing gradient problem, and they are not as powerful as an LSTM or a GRU, but we believed this would be a good start. We used a state size of 100 in our experiments.

One weakness in this model is that there is no attention mechanism. Each sentence is considered on its own, independently from the context(article) it appears in. A sentence like "Yes we can." for example can be unimportant in general, but it may be important if it appears in a story about Barack Obama announcing his campaign slogan. Therefore we believe that implementing attention would help improve our results. Once we have a robust content selection method, we can also consider developing further ordering and pruning strategies. Since pruning and ordering are irrelevant without good content selection we decided to focus on content selection first. We will be working to improve ordering and pruning too if time permits. We believe that even though we don't yet have impressive results, we have finished most of the work by setting up a strong foundation and getting some preliminary results. So the goal now essentially is to improve our model and learning and achieve even better performance.

11 LSTM

12 Results

On the testing dataset, the undersampled LSTM achieved a ROUGE score of 6.7 with an accuracy of 85%. The weighted LSTM achieved a ROUGE score of 6.8 and an accuracy of 85%.

MODEL	Accuracy %	ROUGE - 2
Baseline	N/A	69.0
Oracle - Human	100.0	5.1
ROUGE-Optimized Oracle	100.0	26.9
NN - simple	63.4	−0.429
LSTM - weighted	85.0	6.8
LSTM - undersampled	85.0	6.7
RNN?	0.766	−0.475

Example given Summary:
`The most recent jaguar attack at a U.S. zoo happened in 2007 when zookeeper Ashlee Pfaff had her neck snapped by one of the animals at the Denver Zoo.`

Example undersampled-LSTM Produced Summary:
`A three-year-old fell into the jaguar pit at the Little Rock Zoo in Little Rock, Arkansas, earlier today. Zoo workers rescued the child by lifting them out of the enclosure while using fire extinguishe to hold off the cats. The child, who is in critical condition, fell after a family member let them stand on the railing of the pit. Authorities have confirmed that one of the jaguars attacked the child.`

Example given Summary:
`Nutritional expert Jackie Lynch defends wartime dishes. Cost-effective and nutritious, army food works just as well today.`

Example weighted-LSTM Produced Summary:
`A new book Bully Beef And Boiled Sweets revisits British military meals since the First World War revealing dishes and ingredients that nowadays rarely grace our dining halls.`

Vern Cotter refuses to feel sorry for himself after injury problems. Scotland coach accepts freak injuries can be part of the game. But Alex Dunbar will be huge loss for Scotland against England

13 Challenges

One of the biggest challenges we faced came from our unbalanced dataset. There are approximately 8 times more sentences labeled as unimportant (0) than sentences labeled important (1). Therefore, in our classifier, you can get an accuracy of 85% by producing an empty summary (an output of all 0s) which results in a ROUGE-2 score of 0. Therefore, the high accuracy our classifier might obtain does not necessarily imply a high ROUGE-2 score. We attempted to solve this in a few different ways. Firstly, we tried weighted cross-entropy loss. Basically, we tried to punish wrongly classifying a '1' about three times more than wrongly classifying a '0'. Additionally, we tried under-sampling and made it so that there was an equal number of '0's and '1's in our training set. However, even when we were able to avoid the majority solution, we found that our model still favors brief summaries.

Another challenge we ran in to was an extremely long training time. This made it difficult for us to train our model for multiple epochs as well as debug. Our large dataset is helpful in that it can help us achieve accurate findings, but leads to a large challenge when training our model.

14 Error Analysis

We believe there is a bias problem because none of the models drive training loss sufficiently low. One reason is because there are a lot of unknown words that are not present in GloVe, so we treat all those words as the same UNK token. Also reference summaries are not extractive, so model is limited.

One thing that we discovered in analyzing our output strategies was that

15 Literature Review

One distinctive fact about our project is that there has been a great deal of work already published in this field, and many models have out-performed ours in pure ROUGE scores. It is interesting to note, however, that the ROUGE metric is relatively naive and bigram-based, so some summaries that humans would consider "better" might get lower ROUGE scores, and vice versa. In addition, while these are the papers that are directly relevant

to our project, we want to recognize that most of the existing literature focuses on abstractive summarization rather than extractive, and was not included for brevity.

References