LING 573 Project Report

Macklin Blackburn, mblac6@gmail.com
Xi Chen, xch@uw.edu
Yuan Zhang, yuanz80@uw.edu

April 25, 2017

Abstract

We are building a multi-document automatic text summarization system. Our content selection module is based on multi-layer perceptron regression that uses statistical and linguistic features.

1 Introduction

Automatic summarization is an NLP task that turns a set of documents related to the same topic into a summary. This task was first attempted by Luhn (1958). In recent decades, there have been several different approaches to the problem. These approaches fall into two main categories: supervised and unsupervised. Examples of the former category include Schilder(2008), Vanderwende(2007), and Cao(2015). Examples of the latter include Erkan(2004) Otterbacher(2005) and Radev(2004).

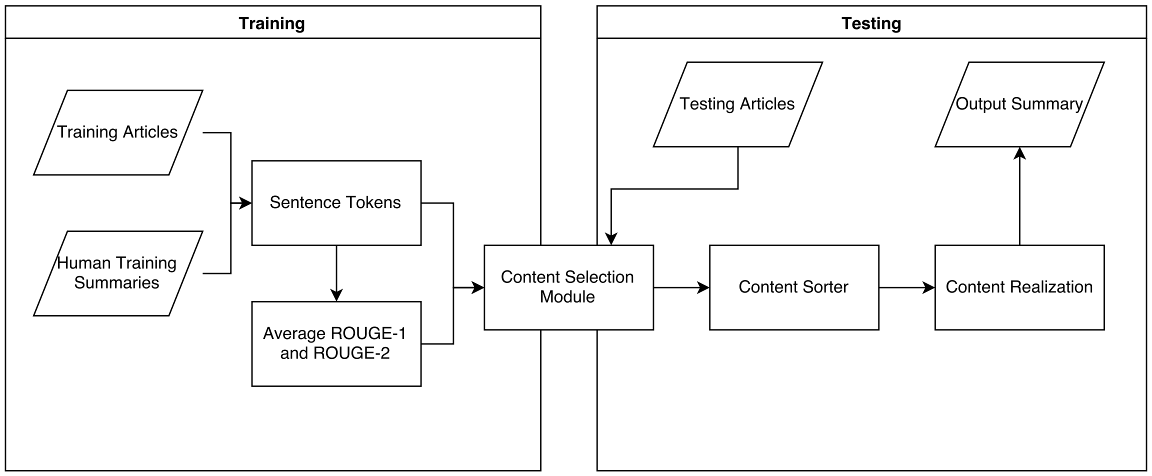
This paper demonstrates a system composed of three sub-systems. First, each sentence is assigned a score to indicate its importance in the final summary. Second, sentences are sorted by scores. Final, a summary is picked out from sorted sentences.

One applicable supervised method is to model the task as sequential labeling, given the input of several features of each sentence. The system will try to predict whether the sentence belongs to the final summary:

$$\mathrm{F}(\mathrm{f}_{\mathrm{n1}},\mathrm{f}_{\mathrm{n2}},...\mathrm{f}_{\mathrm{nk}}) \rightarrow \mathrm{Label}_{\mathrm{n}}$$

2 System Overview

2.1 System architecture


Training
Training Articles
Human Training Summaries
Sentence Tokens
Average ROUGE-1 and ROUGE-2
Content Selection Module
Testing
Testing Articles
Output Summary
Content Sorter
Content Realization

3 Approach

3.1 Data

We carry out our experiments on three datasets, two from the Document Understanding Conference (DUC) and the other from SummBank 1.0. The DUC 2009 and SummBank data are used for training and the DUC 2010 data are used for development and testing.

There are 44 multi-document clusters in DUC 2009 and 46 in DUC 2010. Each cluster contains 10 news articles on the same topic. For each cluster, there are 8 human-written summaries. The SummBank multi-document data consists of 40 document clusters. Each, in turn, contains 10 news articles on a related topic. For each cluster, there are three groups of summaries of different word limits, 50, 100, and 200. For each word-limit group, there are three summaries written by different human judges. Most of these summaries are abstractive.

Each dataset is cached into a json file and used as the input to the system.

3.2 Content Selection

Our content selection system uses multi-layer perceptron regression to identify sentences that are likely to have high ROUGE scores based on a number of features. The ranking of the features is shown below.

Feature	Score
Sentence length	0.137477
LLR sum	0.072244
Count of tag IN	0.063148
KL divergence	0.055616
Count of tag DT	0.047956
tf idf average	0.037263
Count of tag NN	0.035934
Reverse KL divergence of bigrams	0.032828
Count of tag NNP	0.031062
LLR	0.030534
Count of tag CC	0.030295
KL divergence of bigrams	0.030089
Reverse KL divergence	0.026307
Sentiment intensity score	0.025873
Count of tag NNS	0.024905
Average position of words	0.024481
Count of tag JJ	0.024296
Number of capital words	0.024027
tf idf sum	0.021673
P(number)	0.021582
Count of tag VBD	0.020882
P(capital letter)	0.019357
Count of tag VBN	0.019183
Avg first occurrence of word	0.018661
P(capital word)	0.018375
Count of tag VB	0.018191
Earliest first occurrence of word	0.017469
Count of tag RB	0.016164
Count of tag VBZ	0.014301
Count of tag PRP	0.013889
Count of quote character	0.013254
Count of tag VBP	0.012684

Since our system took inspiration from the RegSum system, it includes some of the same features, such as the earliest first occurrence of a word or the average position of a word in a document. These features are usually averaged over the words of a sentence, but sometimes a sum was a more effective feature. The sentence intensity score was computed using VADER as included in the NLTK package for python, and was intended to approximate several features in RegSum that contained sentiment information. However, there are several features in this system that are not present in RegSum, such as the KL divergence of bigrams. The content selector uses NLTK's part of speech tagging to count the number of certain parts of speech in the sentence.

During the training stage, the system constructs a vector for each sentence containing each of the features and assigns the sentence a score. The score is

computed by averaging the ROUGE-1 and ROUGE-2 scores of the sentence compared to the relevant summaries. Earlier versions of the system assigned each sentence a binary score which represented whether the sentence was in the summary or not, but training with the ROUGE scores yielded better results. After assigning each sentence a score, the sentence vectors are scaled using a function from the SciKit Learn package and the sentence scores are scaled so that the highest score is 1 and the lowest is 0.
During the testing stage, the system gives the first sentence of each article a score of 1, meaning that it is very likely to be in the summary. Adding this function greatly increased the ROUGE scores. The system only chooses sentences between 7 and 22 words long.

3.3 Surface Realization

The current surface realization is simple: it fetches all sentences with highest possible scores until the final summary length is reached. The only exception is when the tf-idf cosine similarity between a candidate sentence and a chosen sentence in the summary is greater than a threshold. The current threshold is set to 0.8.

4 Results

4.1 ROUGE Recall Scores

Model	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4
Random sentences	0.20091	0.05338	0.02052	0.01097
First sentences	0.23966	0.07800	0.03349	0.01791
Linear Regression	0.21246	0.05970	0.02083	0.01015
Linear Regression with first sentences	0.23951	0.08025	0.03780	0.02160
MLP Regression	0.24486	0.08455	0.03687	0.02061
MLP Regression (all ROUGE target)	0.24333	0.08346	0.03718	0.02092
Gold Standard	0.37206	0.18435	0.13023	0.11190

5 Discussion

The ROUGE scores we obtained for the random sentences and first sentences were much lower than the same scores in other articles, such as Hong et al [2]. This is likely a result of differences in corpora. However, summaries consisting of the first sentences of each article were very effective relative to other models in our testing. The first-sentence model significantly outperforms a pure linear regression model, but adding first sentences to a regression model yields results that are higher than either.

6 Conclusion

This is a work in progress. We have a working system now based on the RegSum model, which performs better than the baseline, but much is still to be done. More features will have to be included in content selection and more data will be added in the training stage.

References

[1] G. Erkan and D. Radev. Lexrank: graph-based lexical centrality as salience in text summarization. *J. Artificial Intelligence Research, 22(1):457-479*, 2004.

[2] K. Hong and A. Nenkova. Improving the estimation of word importance for news multi- document summarization. *in Proceedings of EACL*, 2014.

[3] A. Nenkova and K. McKeown. Automatic summarization. *Foundations and Trends in Information Retrieval*, 2011.