HMM Content Model for TAC2010 Summarization Challenge

Darla Magdalene Shockley and Michael Strube
HITS gGmbH, Heidelberg, Germany
{Darla.Shockley | Michael.Strube}@h-its.org

February 2, 2011

Abstract

We present the HITS submission for the 2010 TAC Guided Summarization Task. We focus on the main multi-document summarization task, rather than the update task. We implement a baseline extractive summarization system from the literature (Barzilay and Lee, 2004) which uses a Hidden Markov Model to assign sentences content or topic labels, predicts which topics most likely appear in the summary, and constructs the summaries from these topics. We find that this model performs more poorly than expected, as compared to results shown in previous work. These differences may be attributed to the changes we made to the algorithm to accommodate the multi-document summarization task and the lack of human-annotated domains for the training data.

1 Introduction

This paper presents the HITS submission to the 2010 TAC Guided Summarization Task. We submitted a baseline summarization system which is a slightly modified version of the content-modelling approach to summarization presented by (Barzilay and Lee, 2004) (for brevity, we will refer to this work from now on as BL). The central idea behind BL's approach is that within a given domain, the content of sentences in a document and summary will follow a predictable pattern. This approach seems particularly applicable to the TAC 2010 task, since documents are are divided into five domains (Accidents and Natural Disasters, Attacks, Health and Safety, Endangered Resources, and Trials and Investigations).

2 Basic Method from Previous Work

We closely follow the approach of BL. A document is viewed as a sequence of sentences, and each sentence can be labelled with a topic. BL uses a Hidden Markov Model (HMM) to model these topic label sequences, and refer to the model as a *content model*. Gold standard summaries in the training data are then labelled using the trained model, and topics are ranked based on their likelihood to be included in a summary. At test time sentences are labelled with the trained HMM, and then for each topic in the ranked topic list, we select the closest matched sentence labelled with that topic, until reaching the desired summary length (in this case, 100 words).

Each topic label is explicitly viewed as a cluster of sentences (the sentences labelled with this topic in the last training iteration). Before training, "good" initial topic clusters are formed, which reduces the number

of required training iterations for the HMM. BL uses complete-link clustering for this task. Each topic label cluster is represented by a bigram language model:

$$p(w'|w) = \frac{f_{c_i}(ww') + \delta_1}{f_{c_i}(w) + \delta_1|V|}, \tag{1}$$

where $f(y)$ is the frequency with which word or sequence y appears, c_i is a topic cluster, δ_1 is a tuning parameter, and V is the vocabulary. There is also an extra "et cetera" cluster, which contains sentences that do not fit well into any other cluster.

Unlike traditional HMM training, a hard label is assigned to each sentence at each iteration (that is, a sentence belongs to exactly one cluster). This approach simplifies training, since bigram probabilities can be directly calculated from member sentences, but the approach may lead to a suboptimal model. BL shows good results using this simplified method, so we use the same training method.

Training transition probabilities becomes a simple matter of counting occurrences of each transition at each training iteration. BL describes the transition probability update as

$$p(s_j|s_i) = \frac{D(c_i, c_j) + \delta_2}{D(c_i) + \delta_2 m}, \tag{2}$$

Where $D(c_i, c_j)$ is the count of documents containing the topic label c_i followed by the topic label c_j, $D(c)$ is the count of documents containing the topic label c, m is the number of topic clusters, and δ_2 is a tuning parameter. However, this calculation is problematic, since $\sum_i p(s_j|s_i)$ is not 1. For example, if we see the sequence $\{a, b\}$ and the sequence $\{a, c\}$ in the same document, and there is only one document in the collection, assuming a simplified equation where $\delta_2 = 0$, $p(b|a) = 1$ and $p(c|a) = 1$ as well. The problem remains with a larger document collection or different values of δ_2.

We see two possible solutions to this problem. The most straightforward involves only modifying the denominator for Equation 2 to sum over sequences:

$$p(s_j|s_i) = \frac{D(c_i, c_j) + \delta_2}{\sum_j D(c_i, c_j) + \delta_2 m}. \tag{3}$$

Another approach is to use raw counts of occurrences instead of document counts:

$$p(s_j|s_i) = \frac{C(c_i, c_j) + \delta_2}{C(c_i) + \delta_2 m}, \tag{4}$$

where $C(c, c')$ is the count of all occurrences of the sequence $\{c, c'\}$ and $C(c)$ is the count of all occurrences of c. Fortunately, BL provides an implementation of their approach.[1] Examination of the current version of the code provided by BL shows Equation 4 to be used. We experimented with each equation in our implementation, but also found Equation 4 to be the best option.

There are several model parameters which we must set: δ_1, δ_2, m (the number of topic labels), and the maximum number of training iterations. We take $\delta_1 = 0.0000001$ and $\delta_2 = 0.0000001$, from the values used in the BL implementation, and optimize the remaining parameters on a development dataset, resulting in $m = 15$ and a maximum of 5 training iterations.

[1] http://groups.csail.mit.edu/rbg/code/catching_the_drift.html

3 Modifications for TAC2010

The most obvious difference between the task solved by the BL approach and the TAC2010 challenge is that the BL approach is for single-document summarization, and the TAC summarization challenge task is multi-document summarization. However, it is straightforward to generalize by modelling each document separately, and then searching all documents for the most representative sentence of the topics to be included in the summary.

We use previous years of DUC/TAC summarization challenge data as training and development data. Specifically, we use **duc02**, **duc05**, **duc06**, and **duc07** as training data, and **duc08** as development data. We reserved **duc09** as a test set, but did not use it for the TAC2010 challenge.

However, the BL algorithm requires domain-specific data, and while the TAC2010 data is divided into domains, previous years' data is not. Our approach is to first cluster documents by domain, and then proceed with summarization via HMM content modelling. We represent sets of documents with a bigram language model (the same model described in Equation 1, used to represent topic clusters), and use complete-link clustering (Manning et al., 2008) to create domain clusters. Despite the very simple approach to creating domain clusters, we find the cluster quality to be reasonable, based on our subjective judgments upon manual examination of cluster contents. At test time, we do not attempt to align domain clusters with the pre-defined domains for the TAC2010 data, but instead simply classify each TAC2010 set of documents into one of our learned domains.

In our initial experiments, we found that the system converged to a very small number of topic clusters (2-4) in the initial clustering step (for every clustering method we tried), which is not sufficient granularity for the summarization method. Therefore, in creating the initial topic clusters, we enforce a minimum cluster size (the largest cluster may only be twice the size of the smallest cluster). BL asserts that the initial clustering step may not even be necessary, since the clusters are modified during the HMM training. Our results support this hypothesis, since the clustering method does not affect results (though it does change the optimal maximum number of training iterations for the HMM), as long as the initial clusters are reasonably well-balanced.

4 Results

Figures 4 and 4 show results for our system as compared to average peer scores, best peer scores, and the two reported baselines. Figure 4 shows manual Pyramid scores,[2] and Figure 4 shows automatic scores using the ROUGE tool (Lin, 2004). The best, median, and mean peer scores need not each come from the same system. The system with the highest average Pyramid score, for example, is not the same system with the highest average linguistic quality score. The lead baseline is composed of the lead sentences of the most recent document, up to 100 words.

Our system results are quite poor. While our system does outperform several peer systems, performance is well below the average systems' scores, and even below the two baseline system scores (though ROUGE scores are not significantly below the baseline scores). We believe that some of our modifications to the BL algorithm may be responsible for the poor performance. Specifically, our domain clusters may not be as clean as necessary for the algorithm to perform well. Experimenting with more sophisticated methods of domain clustering (or using human-generated clusters, as BL does) may be necessary.

[2] http://www1.cs.columbia.edu/∼becky/DUC2006/2006-pyramid-guidelines.html

	Average Pyramid Score	Average Linguistic Quality
Best Peer Score	0.4250	3.4570
Median Peer Score	0.3470	2.9350
Mean Peer Score	0.3091	2.8197
MEAD Baseline	0.2960	2.7170
Lead Baseline	0.2330	3.6520
HITS	0.1770	2.4780

Figure 1: Manual evaluation results, main summarization task

	ROUGE-2	ROUGE-SU4
Best Peer Score	0.0957	0.1301
Median Peer Score	0.0764	0.1104
Mean Peer Score	0.0686	0.1029
MEAD Baseline	0.0593	0.0911
Lead Baseline	0.0538	0.0855
HITS	0.0500	0.0844

Figure 2: Automatic (ROUGE) evaluation results, main summarization task

We report average linguistic quality in addition to average pyramid score, because our system's linguistic quality scores were relatively high in comparison to peer systems. This may be because our system is a strictly extractive system (and therefore individual sentences have high linguistic quality), but we hope that the higher linguistic quality scores can be attributed to BL approach. The BL approach is a simple method of selecting sentences and sentence orders which are of higher linguistic quality, and may be a useful component in a more successful summarization system. The BL approach also has the added benefit of a high ratio of number of content units to number of repeated content units, indicating that when the system selects a good sentence, it usually does not select other sentences with the same content.

5 Acknowledgements

This work has been funded by the Klaus Tschira Foundation, Heidelberg, Germany. The first author has been supported by a HITS PhD. scholarship.

References

Regina Barzilay and Lillian Lee. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics,* Boston, Mass., 2–7 May 2004, pages 113–120, 2004.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Proceedings of the Text*

Summarization Branches Out Workshop at ACL '04, Barcelona, Spain, 25–26 July 2004, pages 74–81, 2004.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, U.K., 2008.