Feature expansion for query-focused supervised sentence ranking

Seeger Fisher and Brian Roark
Center for Spoken Language Understanding
OGI School of Science & Engineering
Oregon Health & Science University
{fishers,roark}@cslu.ogi.edu

Abstract

We present a supervised sentence ranking approach for use in extractive summarization. Using a general machine learning technique provides great flexibility for incorporating varied new features, which we demonstrate. The system proves quite effective at query-focused multi-document summarization, both for single summaries and for series of update summaries.

1 Introduction

Sentence extraction summarization systems take as input a collection of sentences (one or more documents) and select some subset for output into a summary. This is best treated as a sentence ranking problem, which allows for varying thresholds to meet varying summary length requirements. Most commonly, such ranking approaches use some kind of similarity or centrality metric to rank sentences for inclusion in the summary – see, for example, Lin and Hovy (2002); Erkan and Radev (2004); Radev et al. (2004); Blair-Goldensohn (2005); Biryukov et al. (2005); Mihalcea and Tarau (2005) and the references therein. Such an approach is typically preferred over supervised ranking approaches for reasons of domain independence.

We present an alternative approach, whereby a number of similarity/centrality metrics are used, not directly to rank the sentences, but rather as features within a supervised machine learning paradigm. Since the features themselves are not domain-specific, the benefit of domain generality is retained, while still accruing the benefits of supervised learning.

We examine this approach within the context of query-focused multi-document summarization, for which there is much less training data for supervised approaches than query-neutral multi-document summarization. We address this through the use of two separate ranking models: one trained on a large collection of document clusters and associated (query-neutral) manual summaries; the other trained on a smaller data set from the 2005 and 2006 DUC query-focused multi-document summarization task, which includes document clusters, queries, and the associated (query-focused) manual summaries. The scores from the first ranker are used as features in the second ranker. In addition to the use of two ranking models, we achieve query responsiveness by skewing the word distributions, which make up the features of our models, towards the query. All of this is achieved within a very general supervised ranking paradigm, which is robust and domain independent.

We broke the query-directed summarization problem down into three tasks:

1. Text normalization and sentence segmentation
2. Sentence ranking
   a. query-neutral ranking
   b. query-focused ranking
3. Sentence selection from a ranked list

In a previous paper we have detailed the architecture and training of our main-task system (Fisher and Roark, 2006). In this paper we report on experiments that show our approach can integrate other ranking heuristics advantageously. We also show that incorporating query-expansion into our framework produces substantial gains over our system from last year which did not perform query expansion. Lastly, we give details on how we modified our approach to handle update summaries.

2 Sentence Extraction System

The several stages of our sentence extraction system are detailed in Fisher and Roark (2006). We give just a brief review of the stages here.

2.1 Text normalization

In the multi-document summarization data[1] made available for the Document Understanding Conferences (DUC), each document set is a collection of individual articles, each article in its own file. We created one large text file for each document set by concatenating the raw content text from each article, discarding the meta-data. We then used a simple algorithm to perform sentence segmentation, making use of a list of common abbreviations extracted from the Penn Treebank.

2.2 Supervised sentence ranking

For sentence ranking, we implemented a perceptron ranker (Crammer and Singer, 2001). The objective we used for our supervised ranking is the ROUGE-2 score as configured for the DUC-06 evaluation. For a 250 word summary we are typically only interested in the top 15 or so sentences in a document set (while allowing for redundancy). As a result, we configured the perceptron ranking algorithm to produce models with only 3 ranks. Within each document cluster, feature values were normalized.

Using a limited feature set, the algorithm cannot converge to perfect ranking performance on the training set. We experimented with n-gram features, but although this allowed the perceptron to converge to the training data very accurately, it did not improve ranking performance against our held-out training data. We also experimented with a second order polynomial kernel for the perceptron. This also helped the perceptron to converge, but it did not significantly help with accuracy on the heldout data. See Fisher and Roark (2006) for further details.

2.2.1 Query-neutral sentence ranking

The base feature set that we use is the same as was used in our baseline system from DUC 2005 and DUC 2006 (Fisher and Roark, 2006). For every cluster of documents c in the set of clusters $\mathcal{C}$ comprising the training set, let $\mathcal{Z}_c$ be the collection of manual summaries for that cluster. Let $s \in c$ be the sentences in cluster c and $z \in Z_c$ be the sentences in the summaries of cluster c. For every cluster $c \in \mathcal{C}$ we scored each sentence $s \in c$ as follows

$$\rho(s) \quad = \quad \underset{z \in Z_c}{\text{average}} \left(\text{rouge}(s, z) \right)$$

where rouge(s, z) is the ROUGE score (Lin, 2004) of sentence s with z as the reference summary. We calculated this value for all sentences in each cluster of the DUC 2001-2003 training data for summaries of size 100, 200 and 400 words, giving us our "gold standard" ranking for use in training the base system.

1.	average tf.idf	6.	average logodds
2.	sum tf.idf	7.	sum logodds
3.	average loglike	8.	sum (max 3) logodds
4.	sum loglike	9.	Sentence position
5.	sum (max 3) loglike		

Table 1: Base feature set

For each sentence in a cluster, we extracted a small number of features for ranking. Most of these features are aggregated from word-based features. Word-based features were of three varieties: TF*IDF, log likelihood ratio, and log odds ratio statistics. The feature set is summarized in Table 1. See Fisher and Roark (2006) for details on calculation of the features.

Beyond these base features, we added the features from Table 1 for both the immediately previous and immediately following sentences as features for the current sentence, effectively tripling the number of features.

Using multiple similarity metrics as features is useful because all of these features score co-occurrence dependencies differently.

2.2.2 Query-focused sentence ranking

Skewing word distributions

To achieve query-sensitivity within the context of a single supervised ranking system, we examined skewing word distributions towards the query for purposes of calculating distribution sensitive features. Recall that we have a number of features (see table 1) that rely on the distribution of a word in the document set relative to its distribution in the corpus. We skew the word distributions towards the query in a document set by adding the counts of each of

[1] http://duc.nist.gov/

the non-stop query words, multiplied by an empirically determined factor, to the counts of words in the document set. In effect, non-stop query words have their counts increased in the document set for purposes of calculating the word-distribution sensitive features. The result is that when extracting features from a sentence, words that are in the query will have relatively larger feature values, by virtue of having higher document set counts. When the individual words have larger values, the feature values for sentences containing those words will also be higher.

Note that this approach allows us to train the models on non-skewed training data, with the query-focused skewing happening at test time. Hence, large amounts of query-neutral multi-document summarization training data can be exploited. With this approach, we can get query sensitivity within a very simple ranking approach. This has the additional benefit of being able to convert the ranking score to a normalized probability (via softmax), thus allowing the use of these scores as features in another stage of ranking.

Re-ranking

The first-pass ranking model in our approach is trained on query-neutral summarization data. Given that we now have query-sensitive training data from the DUC-2005 and 2006 evaluation set, we can build a specifically query-focused reranker from this data. As with the query-neutral ranking, we used the perceptron ranking algorithm.

The sentences are first ranked using the skewing approach described above, and the output from this step (the softmax normalized perceptron score) is one of the features input to the reranker. In addition to this feature, which has its weight empirically fixed, the reranker has two other sets of features for which it learns parameter weights. These are features characterizing the number of non-stop query words in the sentence. We first partition the set of non-stop query words into two subsets: those with log likelihoods higher than a fixed threshold and those with log likelihoods lower than the threshold. The log likelihood is calculated for each query word for that cluster, using unskewed counts. Then, for each subset s, there are five indicator features: 0 words in the sentence from s; at least 1 word in the sentence from s; at least 2 words from s; at least 3 words; and at least 4 words. For the trials reported here, the partitioning threshold was set empirically at 10. See Fisher and Roark (2006) for further details on this approach.

For training the reranker, we used the DUC-2005 document sets as training data, and the DUC-2006 document sets as development data for testing different features. We fixed the weight of the baseline ranker at 1000.

2.3 Sentence selection

At the sentence selection stage, we removed any sentence less than 5 words or greater than 50 words in length. The restriction on being too short is based on the intuition that in an extraction system, anything too short will be meaningless out of context. The restriction on being too long is a simple way to keep the system from extracting long lists, which generally do not make a good summary. In addition, any sentence that begins or ends with a quotation mark was also filtered out. Finally, sentences beginning with a pronoun were removed, to avoid the most obvious cases of poor anaphora resolution.

At this point we also applied some simple compression to the remaining sentences. Namely, we removed any paired parentheticals, defined as stretches of text in a sentence that were delimited by parentheses, single dashes, or em-dashes.

Sentences were selected in order based on the final ranking, until the summary size limit was reached, with some sentences being removed for lack of novelty, as follows. Stop-words were removed from a candidate sentence, then the bigram overlap with non-stop words already in the summary was calculated. If the overlap amounted to 65 percent or less of the non-stop words in the candidate (determined empirically), the candidate was added to the summary, otherwise it was discarded. Finally, we ordered the extracted sentences by document-id, and then by order they occurred in the document.

3 Expanding the feature set

One of the stated motivations of our approach is the ease with which additional features can be included within the general framework. To the extent that sentence level features can be derived for the training and test sets, they can be included in the ranking

Ranking approach	ROUGE-2
Graph-based centrality alone	0.08708
OGI-06 system	0.08525
OGI-06 + centrality feature	0.08826

Table 2: Performance of (1) a graph-based centrality metric following (Erkan, 2006) alone for sentence ranking; (2) our submitted system for DUC 2006; and (3) including the graph-based centrality score as a feature in our reranking model.

model along with the features we have already defined. To demonstrate this, we chose two new kinds of features that were not in our submitted system for DUC 2006, but which were shown by other groups to be of utility for this task, and included them in our approach. In this section, we will present experiments documenting the change in performance of our system when these features are included. The first experiment presents the use of another centrality metric in the reranking phase. The second experiment addresses our relatively poor query responsiveness performance through the use of query expansion.

3.1 Graph-based sentence scores

A popular and effective method for ranking sentences as representative of a document or cluster of documents is through graph-based random walk algorithms (Radev et al., 2004; Mihalcea and Tarau, 2005). In such an approach, each sentence is represented as a node in the graph, and edges between nodes are weighted by the similarity between the connected nodes (sentences). Once the graph has been created, a random walk technique can be used to update the weight for each sentence, based upon its connectivity. The final weight of a sentence is related to the number of times its node was visited in the random walks. See Otterbacher et al. (2005) for details.

A query-focused ranking of sentences with such a graph-based centrality can produce competitive results, as seen at DUC in 2006 (Erkan, 2006). To achieve this, the query is included as a node in the graph, and the random walks begin at that node. In order to demonstrate the flexibility of our framework to incorporate new features, we implemented the algorithm as described by (Erkan, 2006), and then included the resulting score as a feature in our reranker. In this case, the similarity between two nodes is the cosine overlap of non-stop words. We performed two experiments with these graph-based centrality scores. In the first, we used that score alone to rank each sentence, then selected sentences as described in section 2.3. In the second experiment, we added the score as a feature to the reranker described in section 2.2.2. Note that, as with our other features, we divided all of the raw feature values by the highest absolute raw value for that feature. The results of our experiments are shown in Table 2. Using the centrality score alone produces excellent results, but as can be seen, using it as a feature within our sentence ranking approach produces even better results, improving on our 2006 system.

3.2 Query expansion

Our system submitted to DUC 2006 performed quite well, but was only about average in query responsiveness. Many of the systems at DUC that year used query expansion of some sort, so we have explored this as a way to improve our query responsiveness. Unfortunately, query responsiveness is a manual metric, so we cannot measure improvement directly. Instead, we use ROUGE score improvement as an approximate measure.

There are a number of ways to approach query expansion. Most of them involve finding synonyms or related words of the words in the query and adding them to the query. One way to find synonyms is to use a thesaural resource, such as WordNet. The semantic information encoded in WordNet can also allow extraction of related words, e.g., words that belong to the same semantic class or stand in hyponym/hypernym relation. Another approach is to use corpus co-occurrence statistics to find words that are contextually related to the query words. For this experiment, we took the second approach.

Using a corpus of approximately 300 million words of newswire text, we calculated the log likelihood ratio of word pairs occurring in adjacent sentences as follows. Let $f(u)$ be the number of sentences that the word u occurs in, and $f(uv)$ the number of times u occurs in the sentence preceding a sentence containing the word v. Let $\bar{u}$ denote words other than u, and N the number of sentences. Then

Re-ranking approach	ROUGE-2
OGI-06 system (06)	0.08525
06 + graph-based centrality feature	0.08826
06 + query expansion	0.08802
06 + graph-based + q-expansion	0.08936

Table 3: Performance of (1) our submitted system for DUC 2006; (2) including the graph-based centrality score as a feature in our reranking model; (3) including query expansion features in our reranking model; (4) including the graph-based centrality score and query expansion in the reranking model.

the log likelihood ratio $\text{loglike}(uv)$ can be calculated as follows:

$$\text{loglike}(uv) = \log \frac{\alpha}{\beta} \quad (1)$$

where

$$\alpha = f(v)^{f(v)} f(u)^{f(u)} f(\bar{u})^{f(\bar{u})} f(\bar{v})^{f(\bar{v})} \quad (2)$$

and

$$\beta = N^N f(uv)^{f(uv)} f(\bar{u}v)^{f(\bar{u}v)} f(u\bar{v})^{f(u\bar{v})} f(\bar{u}\bar{v})^{f(\bar{u}\bar{v})} \quad (3)$$

For each query word q, we calculated a score for each word u that is the sum of $\text{loglike}(qu)$ and $\text{loglike}(uq)$, i.e., the sum of the log likelihood ratios of the word occurring in the preceding or following sentences when a query term q is observed. We then selected the 100 highest scoring words for each original non-stop query word for inclusion in the query. This is a rather large number, but we found it an effective number in practice. Given that the DUC clusters already consist of documents on a similar topic, we hypothesized that any irrelevant expansion words will not occur in the document cluster, and hence their inclusion does not negatively impact system performance.

An expanded set of query terms can be used in a number of ways within this system. First, they could simply be lumped in with other query words for the purpose of query word count skewing or reranking; or they could be treated separately from query words, either in terms of the skewing factor or in defining the reranking features, or both. We found that creating new indicator features for the expanded query words, analogous to the indicator features for the original query words (see Reranking in section 2.2.2), gave us the best performance on the DUC 2005 development data set. We did not find that using the expanded query words for distribution skewing, either the same as the original query words or skewed separately, had an effect. Results for query expansion, both alone and combined with the graph-based centrality feature, are shown in Table 3. As can be seen, combining query expansion with the centrality feature improved our ROUGE scores substantially over the system we submitted at DUC 2006, suggesting that our query responsiveness is improved.

4 DUC 2007 Results

The OGI-07 system was competitive in the field of participants in DUC 2007. There are quite a few different evaluation metrics used at DUC. Our system scores ranged from slightly better than the mean, to well within the top one third of submitted systems, depending on which metric is used. This performance relative to other systems is about the same as for our 2006 system, though we have demonstrated that our 2007 system is substantially better than the 2006 system. This would indicate that the participants as a whole are getting better.

DUC 2007 introduced a new task, creating update summaries. We modified our system to perform this task, as described in section 5. The evaluation metrics for the update task were also quite varied, and our system performed at about the same level on this task as on the main task relative to other systems, even though the field was smaller.

5 Update Summaries

For DUC this year a new summarization task was introduced, query-focused update summaries. The task consists of producing three summaries per document set, each with a maximum length of one hundred words. For each set of summaries, the document set was divided into 3 partitions, with the first being larger than the other two. The partitions were grouped by date, so that the articles in the second partition all came after the first, and all those in the third after the second. The first summary for each document set could use only documents from the first partition, the second could use both the first and second partition, and the last could use all three partitions. The idea being that a consumer may be in-

terested in more information as a story develops.

Our system for producing update summaries is similar to the main task summarizer, with some important differences in sentence selection from the ranked list of sentences. We use the same classifier and feature set, trained the same as the main task summarizer. For summaries of a first partition, the system was identical to the main task system, excepting that there are fewer source documents and the summary is shorter. For the second partition, we allowed the system to rank sentences from documents in either the first or second partition without any modification to the system. However, when checking for overlap, in the sentence selection stage, between a candidate sentence and the sentences in the summary so far, we checked not only against sentences already in the new summary, but also against sentences from the summary of the first partition. Thus, there was no change to our ranking algorithm, only to the part of the system that adds already ranked sentences to the growing summary. The summary for the third partition was constructed in the same way, but with candidate sentences being drawn from all three document partitions, and the overlap being calculated against not only sentences already in the third summary, but also all of the sentences in the already completed summaries of the first two partitions.

Results for our system in the update summary task were similar to results for our main task submission, relative to the submissions of other teams. The task is quite intriguing, we hope to submit a more interesting approach next year now that more training data will be available for development.

6 Summary and future directions

We have presented the application of general supervised machine learning techniques to the problem of sentence ranking for extractive summarization. By exploiting model summaries to define a gold-standard ranking over sentences, we can use well-motivated learning approaches, which handle an arbitrary number of features. We have demonstrated that many common metrics used for sentence ranking can be combined into a single ranking model that provides better performance than any of the metrics in isolation. We straightforwardly extended the model to include features of neighboring sentences, which was demonstrated to improve performance. We have applied this approach to query-directed summarization through a number of techniques: (1) query word count inflation; (2) reranking based on query-directed training data; and (3) query expansion techniques. The resulting approach is highly competitive, and its generality and ease of extension should allow for substantial future developments.

There are a number of ways to improve the current system. The feature set for the reranker is an area we will continue to explore, since we have experimented with relatively few different features for the current system. Though including all unigrams as features led to over-fitting, we would like to find a subset of lexical n-gram features that are relevant to indicating importance and applicability to inclusion in a summary. We also want to include features that are indicative of what sort of question the query is. Another set of features to explore are discourse connectives, and how they relate one clause to another. Because of the general machine learning framework, incorporation of a range of additional features (e.g., query expansion or discourse segmentation) or stages of processing (e.g., anaphora resolution) is straightforward, as we demonstrated in section 3. Finally, we believe that clause segmentation prior to ranking could lead to substantially better performance.

Finally, we are exploring improvements to the sentence selection stage of the summarizer that will be relevant for the update summary task.

References

M. Biryukov, R. Angheluta, and M.F. Moens. 2005. Multidocument question answering text summarization using topic signatures. *Journal on Digital Information Management*.

S. Blair-Goldensohn. 2005. Columbia University at DUC 2005. In *Document Understanding Workshop (DUC) 2005*.

K. Crammer and Y. Singer. 2001. Pranking with ranking. In *Neural Information Processing Systems*. NIPS.

G. Erkan and D. Radev. 2004. Lexpagerank: Prestige in multi-document text summarization. In *Proceedings of EMNLP*.

Gunes Erkan. 2006. Using biased random walks for focused summarization. In *Document Understanding Workshop*.

S. Fisher and B. Roark. 2006. Query-focused summarization by supervised sentence ranking and skewed word distributions. In *Proceedings of the Document Understanding Workshop (DUC)*.

C.Y. Lin and E. Hovy. 2002. Automated multi-document summarization in NeATS. In *Proceedings of the Human Language Technology Conference*.

C.Y. Lin. 2004. Rouge: a package for automatic evaluation of summaries. In *Workshop in Text Summarization, ACL'04*.

R. Mihalcea and P. Tarau. 2005. An algorithm for language independent single and multiple document summarization. In *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP)*.

Jahna Otterbacher, Gunes Erkan, and Dragomir Radev. 2005. Using random walks for question-focused sentence retrieval. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 915–922, Vancouver, British Columbia, Canada, October. ACL.

D. Radev, T. Allison, S. Blair-Goldensohn, J. Blitzer, A. Çelebi, S. Dimitrov, E. Drabek, A. Hakim, W. Lam, D. Liu, J. Otterbacher, H. Qi, H. Saggion, S. Teufel, M. Topper, A. Winkel, and Z. Zhang. 2004. MEAD - a platform for multidocument multilingual text summarization. In *LREC*, Lisbon, Portugal.