

# Bayesian Multi-Document Summarization at MSE

Hal Daumé III and Daniel Marcu

Information Sciences Institute  
4676 Admiralty Way, Suite 1001  
Marina del Rey, CA 90292  
{hdaume,marcu}@isi.edu

## Abstract

We describe our entry into the Multilingual Summarization Evaluation (MSE) competition for evaluating generic multi-document summarization systems, where documents are drawn both from English data and English translations of Arabic data. Our system is based on a Bayesian Query-Focused Summarization model, adapted to the generic, multi-document setting and tuned against the ROUGE evaluation metric. In the human pyramid-based evaluation, our system scored an average of 0.530, approximately 8% better than the next best system, which scored 0.489. In the automatic evaluation, our system scored 0.157 (behind four other sites) with the skip-bigram evaluation, and 0.131 (behind two other sites) with the standard bigram evaluation.

## 1 Introduction

Our summarization model is an instance of a more general Bayesian Query-Focused Summarization (BQFS) model, which was developed for a query-focused, single document summarization task (Daumé III and Marcu, 2005b). We begin this paper with a brief discussion of the BQFS model and its inference, and then describe how we have adapted this to the generic, multi-document setting. The adaptation essentially involves three components: first, we describe how we cope with the lack of a query; second, we describe how we extend the model to the

multi-document setting; finally, we describe how we tune the resulting model parameters against the automatic evaluation criteria. We present results both directly from the MSE evaluation as well as some experiments we ran internally after the conclusion of the evaluation. We conclude with both a discussion of our model and possible extensions, as well as our experience in the competition and opinions on the evaluation.

## 2 Bayesian Query-Focused Summarization

The task tackled by the Bayesian Query-Focused Summarization model (BQFS) is that of corpus creation: using the TREC data as input (specifically, the document collection, queries and relevance judgments), we attempted to build a model that can leverage the *relevance judgments* in order to automatically create a corpus of document/query/extract triples, on which a subsequent summarization model could be trained. This model views a document as being drawn from a mixture of three components: a general English component, a query-specific component and a document-specific component,  $\langle g, q, d \rangle$ .<sup>1</sup> Sentences in a document are assigned a continuous probability distribution of being drawn from each component. For instance, a highly query-relevant sentence might have a sentence degree  $\langle 0.1, 0.8, 0.1 \rangle$ , whereas a sentence that provides much background information specific to the document but not relevant to the query might have

---

<sup>1</sup>The model actually considers the generalized case where a document can be relevant to more than one query, but for the purposes of this paper, we are only interested in the case where a document is relevant to exactly one query.

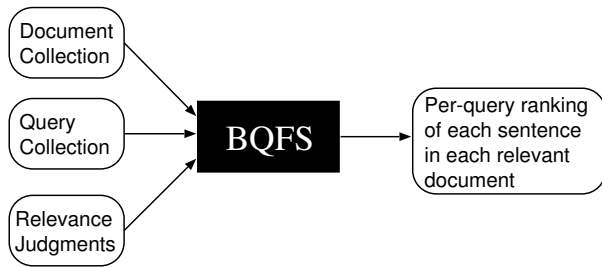


Figure 1: BQFS as a black box.

a sentence degree  $\langle 0.1, 0.1, 0.8 \rangle$ . Similarly, each word in a sentence is assigned a discrete source: exactly one of “general English,” “query-relevant” or “document-specific.”

The inference problem in the BQFS model is to infer a “general English” language model, a “document-specific” language model (for each document) and a “query-relevant” language model (for each query), which is accomplished by *integrating out* the sentence degrees and word sources. The resulting integral is analytically intractable, and we described efficient inference routines based on both the variational approximation and expectation propagation (EP). EP proved both more accurate and efficient in practice, so we used it exclusively for the experiments reported in this paper.

The interested reader is directed to (Daumé III and Marcu, 2005b) for more information on the mechanics of the inference problem. However, for the purpose of this article, one can consider the BQFS system to be a black box (See Figure 1). One feeds into this box a collection of queries, a collection of documents and links that connect queries to relevant documents (relevance judgments). As output, the system will produce scores for each sentence in each document for its respective query. These scores are generalized distances, so that a sentence achieving a score of zero is best, and all other scores are strictly positive. Using these scores, we can easily rank sentences for extraction.

### 3 Adapting BQFS to MSE

#### 3.1 Removing the Queries

The first hurdle we need to overcome in adapting the BQFS model to the MSE task is to deal with the fact that in BQFS, we have assumed the existence of

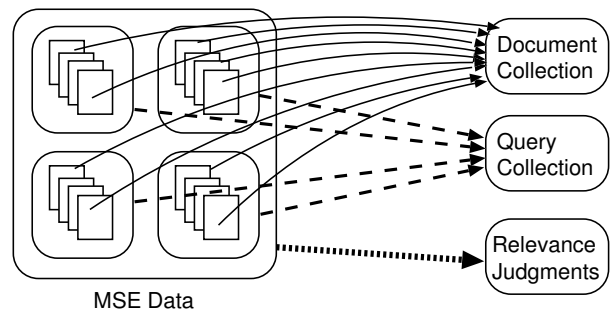


Figure 2: Creating the input to the BQFS model from the MSE document collection.

queries. It turns out that this is not a significant problem: we found in the course of developing the BQFS model that even when we pretend that we do not have a query (essentially by considering the query to be the empty string), there is more than enough information contained in the relevance judgments to provide reliable parameter estimation. We anticipate that the same happens in the multi-document generic summarization setting, wherein we can consider each given document set to be relevant to a unique, empty query.

More specifically, consider Figure 2. The MSE data comes as a collection of document sets, each depicted by a rounded box on the left hand side of the figure. Each document set contains a collection of documents, drawn as non-rounded boxes. We first create the document collection to feed into the BQFS model simply by taking the union of all of these documents. The second step is then to create the query collection. For each *document set* in the MSE data, we create a unique query, though all queries have the empty string as their text (i.e., we pretend that each document set was the result of issuing an unknown query). Finally, to create the relevance judgments, we assign all documents that were drawn from document set  $X$  to be relevant to the query corresponding to document set  $X$ .

#### 3.2 Extending to Multiple Documents

The extension of the BQFS model to the multi-document setting is a bit more involved, since we would like to be able to model *redundancy*, an effect of significantly less importance in the single document setting. In particular, once we have extracted one sentence to place in the summary, we

would like to avoid choosing a very similar sentence to subsequently extract. To accomplish this, we employ a greedy sentence selection strategy, akin to the maximal marginal relevance (MMR) framework described by Carbonell and Goldstein (1998).

Instead of simply weighting sentences  $\bar{w}$  by their BQFS score,  $s_B(\bar{w})$ , we instead weight them by a linear sum of their BQFS score and a *redundancy score*:  $s(\bar{w}) = \lambda_B s_B(\bar{w}) + \lambda_R s_R(\bar{w})$ , where the  $\lambda$ s are model parameters that must be tuned (described in the next section). The redundancy score is computed by estimating a language model for the already-extracted sentences, for the candidate sentence  $\bar{w}$ , and computing the KL divergence between these two distributions, as is common in the IR community (Lavrenko and Croft, 2001).

Now that we have introduced a redundancy model and a corresponding parameter that must be tuned, we may as well introduce a handful of additional features. The first additional feature we consider is based on running the BQFS model on an extended, web-collected corpus. Specifically, for each document set, we pick the top 20 tf-idf terms and use these to formulate a query to AltaVista; we take the top 20 documents returned by AltaVista as additional relevant documents (we never extract sentences from these documents; we only use them as they affect the weights learned by the BQFS model). The second preprocessing step we perform is to cluster the document in each document set. These are greedily clustered in an agglomerative scheme, where distance is computed by the KL measure between a given document and the “centroid document,” which is the document with lowest KL divergence to the union of the document set.

The full list of features we use are:

**BQFS:** The simple BQFS score, as computed over just the given document set.

**BQFS-Web:** The BQFS score, when computed over the union of the given document set and the corpus downloaded from the web.

**Redundancy:** The redundancy score; i.e., the KL divergence between the sentence under consideration and the union of the previously-selected sentences.

**Position:** The position of the sentence under consideration divided by the total number of sentences in the current document.

**IsQuoted:** A binary values that is 1 whenever the sentence appears within quotation marks (or contains quotes) and 0 otherwise.

**SentLen:** The log of the length (in words) of the current sentence.

**DocLen:** The log of the length (in words) of the document from which the current sentence comes.

**DocNum:** The distance (number of hops agglomerative cluster tree) of this sentence’s document to the centroid document.

**DocKL:** The KL divergence between this sentence’s document and the centroid document.

**NumPro:** The number of pronouns that occur in the current sentence.

**NumSay:** The number of attribution verbs that occur in the current sentence (i.e., “say,” “state,” “observe,” etc.).

### 3.3 Parameter Tuning

Based on the above features, we need to estimate *parameter values*. We do this by optimizing against ROUGE score (Lin, 2004): specifically, we search for parameter values that maximize the ROUGE score on development data. We use the data from DUC 2003 tasks 2 and 4 to tune the model.

In order to perform the tuning, we consider each parameter at a time. Its corresponding  $\lambda$  is searched on the first pass from the range  $\{-2, -1, -0.5, -0.2, 0, 0.2, 0.5, 1, 2\}$ . On the  $n$ th pass through the data (for  $n > 1$ ), we consider  $\lambda$  in the range  $\{\lambda_0 \pm n/i : n \in \{0, \dots, 5\}\}$  where  $\lambda_0$  is the value chosen in the previous iteration and  $i$  is the current iteration number. In all of our experiments, we ran four passes through the data, after which parameters had ceased to change.

Our three submissions are based on tuning against a different version of ROUGE. Our primary run is based on tuning against the Basic Element version of ROUGE, precisely as computed for the MSE evaluation. Our second run is based on tuning against

ROUGE-2, where recall is computed over contiguous bigrams. Our third run is based on tuning against ROUGE-S, where recall is computed over skip bigrams with a maximal skip of 4.

### 3.4 Sentence Compression

The final feature in our model is the addition of a weak sentence compression module. In order to incorporate syntactic information at the compression level, we first run the joint part of speech tagging and syntactic chunking model described in (Daumé III and Marcu, 2005a), which simultaneously tags and chunks with state-of-the-art performance. Based on the outputs of the chunker, our model considers the following four compressions: (1) drop all adjectives and adjective phrases; (2) drop all adverbs and adverb phrases (excluding negative expressions); (3) drop all prepositional phrases and the next immediate noun phrase; (4) drop all attributive phrases (such as “X said, ”).

Of these four compression models, we decoded the test data with all 16 possible combinations and chose the combination that achieved the highest Rouge score (according to each of the three metrics we consider). In the case of ROUGE-BE, this was to drop adverbs and attributive phrases; for ROUGE-2, this was to drop attributive phrases; for ROUGE-S, this was to drop adverbs and attributive phrases. Clearly this compression model is weak (for instance, it would be very useful to be able to distinguish between adjunctive PPs and complementary PPs), but it uses sufficiently few parameters that it is easy to tune against existing data.

## 4 Experimental Results

### 4.1 Official MSE Results

According to the human pyramid metric evaluation of Nenkova and McKeown (2004), our system came in first place, with an average pyramid score of 0.530. The scores for all the systems evaluated according to the pyramid metric are shown in Figure 3. It is, however, unlikely that our system is statistically significantly better than, for instance, system 28.

In the automatic “Basic Element” evaluation, our system scored 0.0704 (with a 95% confidence interval of [0.0429, 0.1057]), which was the third best score on a site basis (out of 10 sites), and was not

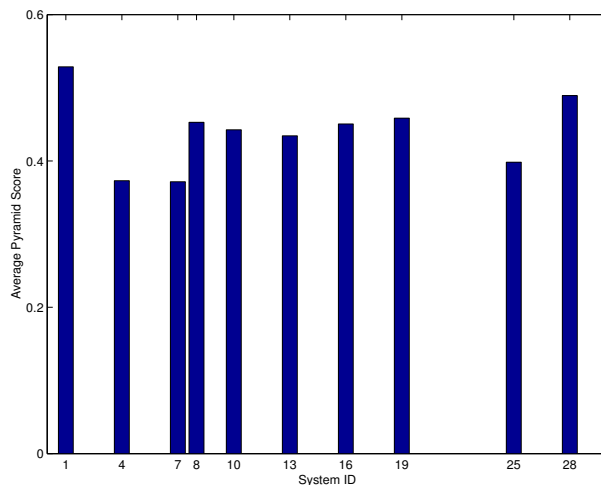


Figure 3: Results of human evaluation; our system, which comes in first place, is id #1.

statistically significantly different from the best system, which scored 0.0981. The results of the BE evaluation are shown in Figure 4 (there are several bars taller than ours, which are bars 1–3, but many of those are multiple runs from the same site).

One surprising result of the evaluation is in comparing the results of our three submissions. One would expect that our submission trained against BE would perform best on BE, that the submission trained against ROUGE-2 would perform best on ROUGE-2 and so on. Surprisingly, according to ROUGE-2, our BE-trained submission scored 0.131, our ROUGE-2-trained submission scored 0.124 and our ROUGE-S-trained submission scored 0.118, making the BE-trained submission the best on this metric. On the ROUGE-S metric, our three submissions scored 0.157, 0.149 and 0.149 respectively; again, the BE-system scored best. (Of course, none of these differences is statistically significant, but it is still surprising that there is the trend that the BE-system is best across the board.) This suggests that our parameter tuning method was suboptimal.

### 4.2 Post-hoc Results

We ran one post-hoc experiment to assess whether we should ever consider including sentences from the MT-translated documents in the summary. We retrained our BE system under the condition that it never include translated sentences and evaluated it against the modes. According to the BE evaluation,

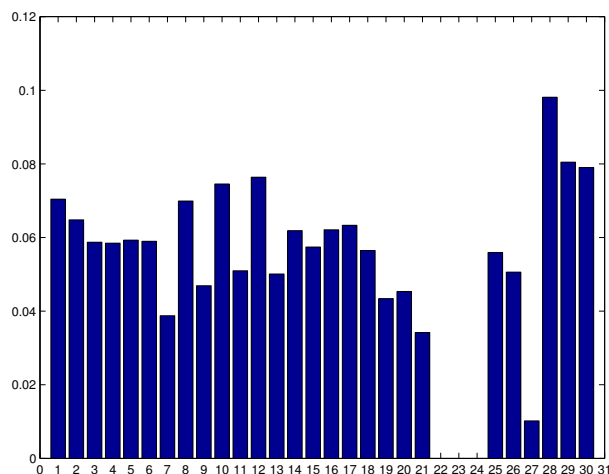


Figure 4: Results of automatic BE evaluation; our system runs are ids 1–3.

our system scores 0.761 in this configuration (compared to the original system, which scored 0.704). This new score puts our system in third place according to the overall evaluation, though, of course, it is still not statistically significantly worse than the best performing system.

## 5 Discussion

### 5.1 Our Model

Our model is essentially based on three contributions, each of which has its own strengths and deficiencies:

1. The BQFS model for sentence weighting.
2. The parameter tuning against ROUGE.
3. The weak sentence compression techniques.

The use of the BQFS model was mostly for convenience: it is a model we had access to and knew we could run successfully. We have also observed that in the single document, query-focused task, it greatly outperforms competing approaches, especially in the case when no query is given. It would have been ideal to explicitly extend this model to account for the redundancy encountered in the multi-document summarization, but time constraints precluded us from exploring this option.

The parameter tuning against ROUGE enabled us to take into account the dozen features that we used

without having to set weights arbitrarily by hand. Currently our parameter tuning involves adjusting weights and completely rerunning the summarization system and the ROUGE script. This is reasonably computationally intensive, and it would be better to use a more efficient optimization scheme. However, for the reasonably small data sets we have available, this was not a high priority.

The compression techniques seem to be of great importance to the performance of our system, based on the post-hoc experiments we ran. We intend to continue investigating the use of progressively more complex compression models, tuned against optimizing the ROUGE criteria. How to do this efficiently is also an important issue that we will explore in future work.

### 5.2 The Task and Evaluation

In our opinion, the weakest aspect of this evaluation is that it is unclear whether it is necessary or *wise* to use the translated data. Machine translation is not yet at the point that it is very readable, much less parsable by the automatic evaluation schemes. Indeed, in our post-hoc experiments, we found that we achieved better BE scores by never extracting sentences from the MT output.

We observed a similar effect in performing the pyramid evaluation: many sentences were clearly MT output and were often largely incomprehensible. This made the pyramid annotation difficult, since *knowing* what was supposed to be in a summary (though looking at the SCUs in the pyramid), we could often find parts of the MT output that seemed to correspond. However, in most of these cases, without having seen the SCUs, we would not have been able to tell what the sentence was actually saying. In these cases, we did not give the system credit, since to do otherwise seemed dishonest; nevertheless, there is clearly a tension here.

Our final observation about the evaluation has to do with the initial creation of the SCUs. We found a strong degree of inconsistency with respect to the granularity of the SCUs contained in the pyramid. For instance, in document collection 33002, SCU#2 is “US troops in Saudi Arabia and Kuwait have been put on ‘Threat Condition Delta,’ the highest state of alert.” In the same collection, SCU#12 is “The date is October 31, 2000.” Here, the SCU#12 contains

only one small bit of information while SCU#2 contains at least four bits of information (that the troops are in Saudi Arabia, that they are in Kuwait, that they are on “Threat Condition Delta,” and that this “Condition” is the highest state of alert). When a document contained only one or two of the latter bits of information, it was unclear whether credit should be given. We attempted to consistently give credit if the system contained at least half of the information in these “complex SCUs.” More careful attention on the part of the SCU builders to maintain minimal SCUs would be helpful in performing the evaluation.

### **Acknowledgments**

This work was partially supported by DARPA-ITO grant N66001-00-1-9814 and NSF grants IIS-0326276 and IIS-0097846.

### **References**

- Jaime G. Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the Conference on Research and Developments in Information Retrieval (SIGIR)*, pages 335–336.
- Hal Daumé III and Daniel Marcu. 2005a. Learning as search optimization: Approximate large margin methods for structured prediction. In *Proceedings of the International Conference on Machine Learning (ICML)*.
- Hal Daumé III and Daniel Marcu. 2005b. Semi-supervised creation of corpora for query-focused summarization. Under review.
- Victor Lavrenko and Bruce Croft. 2001. Relevance-based language models. In *Proceedings of the Conference on Research and Developments in Information Retrieval (SIGIR)*.
- Chin-Yew Lin. 2004. ROUGE: a package for automatic evaluation of summaries. In *In Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)*, July 25 – 26.
- Ani Nenkova and Kathy McKeown. 2004. Evaluating content selection in summarization: The pyramid method. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics and Human Language Technology (NAACL/HLT)*, Boston, MA, USA, May.