Query-Relevant Summarization using FAQs

Adam Berger
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
aberger@cs.cmu.edu

Vibhu O. Mittal
Just Research
4616 Henry Street
Pittsburgh, PA 15213
mittal@justresearch.com

Abstract

This paper introduces a statistical model for *query-relevant summarization*: succinctly characterizing the relevance of a document to a query. Learning parameter values for the proposed model requires a large collection of summarized documents, which we do not have, but as a proxy, we use a collection of FAQ (frequently-asked question) documents. Taking a learning approach enables a principled, quantitative evaluation of the proposed system, and the results of some initial experiments—on a collection of Usenet FAQs and on a FAQ-like set of customer-submitted questions to several large retail companies—suggest the plausibility of learning for summarization.

1 Introduction

An important distinction in document summarization is between *generic summaries*, which capture the central ideas of the document in much the same way that the abstract of this paper was designed to distill its salient points, and *query-relevant summaries*, which reflect the relevance of a document to a user-specified query. This paper discusses query-relevant summarization, sometimes also called "user-focused summarization" (Mani and Bloedorn, 1998).

Query-relevant summaries are especially important in the "needle(s) in a haystack" document retrieval problem: a user has an information need expressed as a query (`What countries export smoked salmon?`), and a retrieval system must locate within a large collection of documents those documents most likely to fulfill this need. Many interactive retrieval systems—web search engines like `Altavista`, for instance—present the user with a small set of candidate relevant documents, each summarized; the user must then perform a kind of triage to identify likely relevant documents from this set. The web page summaries presented by most search engines are generic, not query-relevant, and thus provide very little guidance to the user in assessing relevance. Query-relevant summarization (QRS) aims to provide a more effective characterization of a document by accounting for the user's information need when generating a summary.

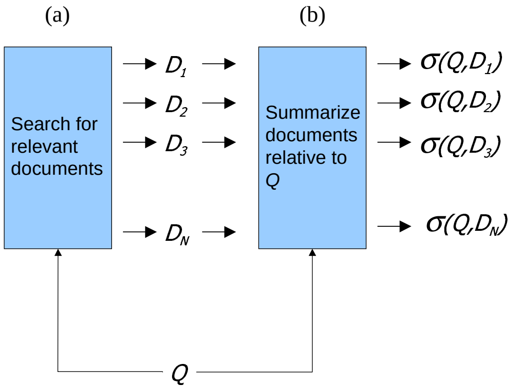

Figure 1: One promising setting for query-relevant summarization is large-scale document retrieval. Given a user query $\mathbf{q}$, search engines typically first (a) identify a set of documents which appear potentially relevant to the query, and then (b) produce a short characterization $\sigma(\mathbf{d},\mathbf{q})$ of each document's relevance to $\mathbf{q}$. The purpose of $\sigma(\mathbf{d},\mathbf{q})$ is to assist the user in finding documents that merit a more detailed inspection.

As with almost all previous work on summarization, this paper focuses on the task of *extractive summarization*: selecting as summaries text spans—either complete sentences or paragraphs—from the original document.

1.1 Statistical models for summarization

From a document $\mathbf{d}$ and query $\mathbf{q}$, the task of query-relevant summarization is to extract a portion $\mathbf{s}$ from $\mathbf{d}$ which best reveals how the document relates to the query. To begin, we start with a collection $\mathcal{C}$ of $\{\mathbf{d},\mathbf{q},\mathbf{s}\}$ triplets, where $\mathbf{s}$ is a human-constructed summary of $\mathbf{d}$ relative to the query $\mathbf{q}$. From such a collec-

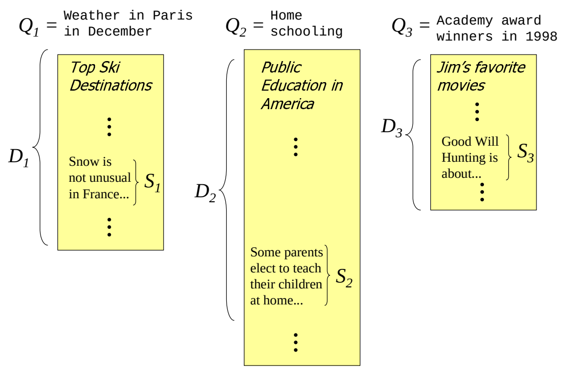
Q_1 = Weather in Paris in December
Q_2 = Home schooling
Q_3 = Academy award winners in 1998
D_1
Top Ski Destinations
Snow is not unusual in France...
S_1
D_2
Public Education in America
Some parents elect to teach their children at home...
S_2
D_3
Jim's favorite movies
Good Will Hunting is about...
S_3

Figure 2: Learning to perform query-relevant summarization requires a set of documents summarized with respect to queries. Here we show three imaginary triplets $\{\mathbf{d}, \mathbf{q}, \mathbf{s}\}$, but the statistical learning techniques described in Section 2 require thousands of examples.

tion of data, we fit the best function $\sigma : (\mathbf{q}, \mathbf{d}) \rightarrow \mathbf{s}$ mapping document/query pairs to summaries.

The mapping we use is a probabilistic one, meaning the system assigns a value $p(\mathbf{s} \mid \mathbf{d}, \mathbf{q})$ to every possible summary $\mathbf{s}$ of $(\mathbf{d}, \mathbf{q})$. The QRS system will summarize a $(\mathbf{d}, \mathbf{q})$ pair by selecting

$$\sigma(\mathbf{d}, \mathbf{q}) \overset{\text{def}}{=} \arg\max_{\mathbf{s}} p(\mathbf{s} \mid \mathbf{d}, \mathbf{q})$$

There are at least two ways to interpret $p(\mathbf{s} \mid \mathbf{d}, \mathbf{q})$. First, one could view $p(\mathbf{s} \mid \mathbf{d}, \mathbf{q})$ as a "degree of belief" that the correct summary of $\mathbf{d}$ relative to $\mathbf{q}$ is $\mathbf{s}$. Of course, what constitutes a good summary in any setting is subjective: any two people performing the same summarization task will likely disagree on which part of the document to extract. We could, in principle, ask a large number of people to perform the same task. Doing so would impose a distribution $p(\cdot \mid \mathbf{d}, \mathbf{q})$ over candidate summaries. Under the second, or "frequentist" interpretation, $p(\mathbf{s} \mid \mathbf{d}, \mathbf{q})$ is the fraction of people who would select $\mathbf{s}$—equivalently, the probability that a person selected at random would prefer $\mathbf{s}$ as the summary.

The statistical model $p(\cdot \mid \mathbf{d}, \mathbf{q})$ is parametric, the values of which are learned by inspection of the $\{\mathbf{d}, \mathbf{q}, \mathbf{s}\}$ triplets. The learning process involves maximum-likelihood estimation of probabilistic language models and the statistical technique of shrinkage (Stein, 1955).

This probabilistic approach easily generalizes to the generic summarization setting, where there is no query. In that case, the training data consists of $\{\mathbf{d}, \mathbf{s}\}$ pairs, where $\mathbf{s}$ is a summary of the document $\mathbf{d}$. The goal, in this case, is to learn and apply a mapping $\tau : \mathbf{d} \rightarrow \mathbf{s}$ from documents to summaries. That is,

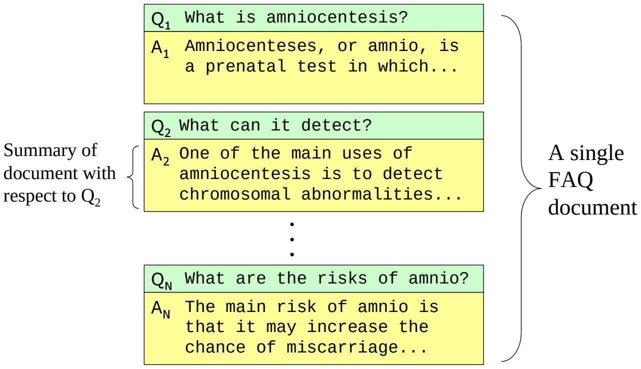
Q_1 What is amniocentesis?
A_1 Amniocenteses, or amnio, is a prenatal test in which...
Q_2 What can it detect?
A_2 One of the main uses of amniocentesis is to detect chromosomal abnormalities...
Q_N What are the risks of amnio?
A_N The main risk of amnio is that it may increase the chance of miscarriage...
Summary of document with respect to Q_2
A single FAQ document

Figure 3: FAQs consist of a list of questions and answers on a single topic; the FAQ depicted here is part of an informational document on amniocentesis. This paper views answers in a FAQ as different summaries of the FAQ: the answer to the kth question is a summary of the FAQ relative to that question.

find

$$\tau(\mathbf{d}) \overset{\text{def}}{=} \arg\max_{\mathbf{s}} p(\mathbf{s} \mid \mathbf{d})$$

1.2 Using FAQ data for summarization

We have proposed using statistical learning to construct a summarization system, but have not yet discussed the one crucial ingredient of any learning procedure: training data. The ideal training data would contain a large number of heterogeneous documents, a large number of queries, and summaries of each document relative to each query. We know of no such publicly-available collection. Many studies on text summarization have focused on the task of summarizing newswire text, but there is no obvious way to use news articles for query-relevant summarization within our proposed framework.

In this paper, we propose a novel data collection for training a QRS model: frequently-asked question documents. Each frequently-asked question document (FAQ) is comprised of questions and answers about a specific topic. We view each answer in a FAQ as a summary of the document relative to the question which preceded it. That is, an FAQ with N question/answer pairs comes equipped with N different queries and summaries: the answer to the kth question is a summary of the document relative to the kth question. While a somewhat unorthodox perspective, this insight allows us to enlist FAQs as labeled training data for the purpose of learning the parameters of a statistical QRS model.

FAQ data has some properties that make it particularly attractive for text learning:

- There exist a large number of Usenet FAQs—several thousand documents—publicly available on the Web[1]. Moreover, many large companies maintain their own FAQs to streamline the customer-response process.

- FAQs are generally well-structured documents, so the task of extracting the constituent parts (queries and answers) is amenable to automation. There have even been proposals for standardized FAQ formats, such as RFC1153 and the Minimal Digest Format (Wancho, 1990).

- Usenet FAQs cover an astonishingly wide variety of topics, ranging from extraterrestrial visitors to mutual-fund investing. If there's an online community of people with a common interest, there's likely to be a Usenet FAQ on that subject.

There has been a small amount of published work involving question/answer data, including (Sato and Sato, 1998) and (Lin, 1999). Sato and Sato used FAQs as a source of summarization corpora, although in quite a different context than that presented here. Lin used the datasets from a question/answer task within the Tipster project, a dataset of considerably smaller size than the FAQs we employ. Neither of these paper focused on a statistical machine learning approach to summarization.

2 A probabilistic model of summarization

Given a query $\mathbf{q}$ and document $\mathbf{d}$, the query-relevant summarization task is to find

$$\mathbf{s}^\star \equiv \arg\max_{\mathbf{s}} p(\mathbf{s} \mid \mathbf{d}, \mathbf{q}),$$

the *a posteriori* most probable summary for $(\mathbf{d}, \mathbf{q})$. Using Bayes' rule, we can rewrite this expression as

$$\begin{aligned} \mathbf{s}^\star &= \arg\max_{\mathbf{s}} p(\mathbf{q} \mid \mathbf{s}, \mathbf{d})\, p(\mathbf{s} \mid \mathbf{d}), \\ &\approx \arg\max_{\mathbf{s}} \underbrace{p(\mathbf{q} \mid \mathbf{s})}_{\textit{relevance}}\ \underbrace{p(\mathbf{s} \mid \mathbf{d})}_{\textit{fidelity}}, \qquad (1) \end{aligned}$$

where the last line follows by dropping the dependence on $\mathbf{d}$ in $p(\mathbf{q} \mid \mathbf{s}, \mathbf{d})$.

Equation (1) is a search problem: find the summary $\mathbf{s}^\star$ which maximizes the product of two factors:

1. The **relevance** $p(\mathbf{q} \mid \mathbf{s})$ of the query to the summary: A document may contain some portions directly relevant to the query, and other sections bearing little or no relation to the query. Consider, for instance, the problem of summarizing a survey on the history of organized sports relative to the query "*Who was Lou Gehrig?*" A summary mentioning Lou Gehrig is probably more relevant to this query than one describing the rules of volleyball, even if two-thirds of the survey happens to be about volleyball.

2. The **fidelity** $p(\mathbf{s} \mid \mathbf{d})$ of the summary to the document: Among a set of candidate summaries whose relevance scores are comparable, we should prefer that summary $\mathbf{s}$ which is most representative of the document as a whole. Summaries of documents relative to a query can often mislead a reader into overestimating the relevance of an unrelated document. In particular, very long documents are likely (by sheer luck) to contain some portion which appears related to the query. A document having nothing to do with Lou Gehrig may include a mention of his name in passing, perhaps in the context of amyotropic lateral sclerosis, the disease from which he suffered. The fidelity term guards against this occurrence by rewarding or penalizing candidate summaries, depending on whether they are germane to the main theme of the document.

   More generally, the fidelity term represents a *prior*, query-independent distribution over candidate summaries. In addition to enforcing fidelity, this term could serve to distinguish between more and less fluent candidate summaries, in much the same way that traditional language models steer a speech dictation system towards more fluent hypothesized transcriptions.

In words, (1) says that the best summary of a document relative to a query is relevant to the query (exhibits a large $p(\mathbf{q} \mid \mathbf{s})$ value) and also representative of the document from which it was extracted (exhibits a large $p(\mathbf{s} \mid \mathbf{d})$ value). We now describe the parametric form of these models, and how one can determine optimal values for these parameters using maximum-likelihood estimation.

2.1 Language modeling

The type of statistical model we employ for both $p(\mathbf{q} \mid \mathbf{s})$ and $p(\mathbf{s} \mid \mathbf{d})$ is a unigram probability distribution over words; in other words, a language model. Stochastic models of language have been used extensively in speech recognition, optical character recognition, and machine translation (Jelinek, 1997; Berger et al., 1994). Language models have also started to find their way into document retrieval (Ponte and Croft, 1998; Ponte, 1998).

The fidelity model $p(\mathbf{s} \mid \mathbf{d})$

One simple statistical characterization of an n-word document $\mathbf{d} = \{d_1, d_2, \ldots d_n\}$ is the frequency of

[1] Two online sources for FAQ data are `www.faqs.org` and `rtfm.mit.edu`.

each word in $\mathbf{d}$—in other words, a marginal distribution over words. That is, if word w appears k times in $\mathbf{d}$, then $p_{\mathbf{d}}(w) = k/n$. This is not only intuitive, but also the maximum-likelihood estimate for $p_{\mathbf{d}}(w)$.

Now imagine that, when asked to summarize $\mathbf{d}$ relative to $\mathbf{q}$, a person generates a summary from $\mathbf{d}$ in the following way:

- *Select a length m for the summary according to some distribution $l_{\mathbf{d}}$.*
- *Do for $i = 1, 2, \ldots m$:*
  - *Select a word w at random according to the distribution $p_{\mathbf{d}}$. (That is, throw all the words in $\mathbf{d}$ into a bag, pull one out, and then replace it.)*
  - *Set $\mathbf{s}_i \leftarrow w$.*

In following this procedure, the person will generate the summary $\mathbf{s} = \{s_1, s_2, \ldots s_m\}$ with probability

$$p(\mathbf{s} \mid \mathbf{d}) = l_{\mathbf{d}}(m) \prod_{i=1}^{m} p_{\mathbf{d}}(s_i) \qquad (2)$$

Denoting by $\mathcal{W}$ the set of all known words, and by $c(w \in \mathbf{d})$ the number of times that word w appears in $\mathbf{d}$, one can also write (2) as a multinomial distribution:

$$p(\mathbf{s} \mid \mathbf{d}) = l_{\mathbf{d}}(m) \prod_{w \in \mathcal{W}} p(w)^{c(w \in \mathbf{d})}. \qquad (3)$$

In the text classification literature, this characterization of $\mathbf{d}$ is known as a "bag of words" model, since the distribution $p_{\mathbf{d}}$ does not take account of the order of the words within the document $\mathbf{d}$, but rather views $\mathbf{d}$ as an unordered set ("bag") of words. Of course, ignoring word order amounts to discarding potentially valuable information. In Figure 3, for instance, the second question contains an anaphoric reference to the preceding question: a sophisticated context-sensitive model of language might be able to detect that `it` in this context refers to `amniocentesis`, but a context-free model will not.

The relevance model $p(\mathbf{q} \mid \mathbf{s})$

In principle, one could proceed analogously to (2), and take

$$p(\mathbf{q} \mid \mathbf{s}) = l_{\mathbf{s}}(k) \prod_{i=1}^{m} p_{\mathbf{s}}(q_i). \qquad (4)$$

for a length-k query $\mathbf{q} = \{q_1, q_2 \ldots q_k\}$. But this strategy suffers from a sparse estimation problem. In contrast to a document, which we expect will typically contain a few hundred words, a normal-sized summary contains just a handful of words. What this means is that $p_{\mathbf{s}}$ will assign zero probability to most words, and any query containing a word not in the summary will receive a relevance score of zero.

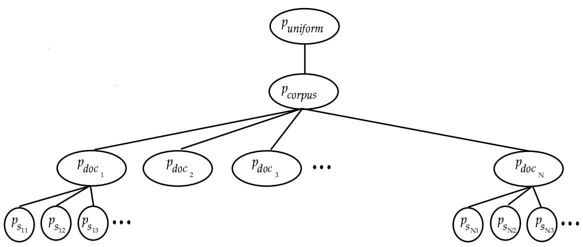

Figure 4: The relevance $p(\mathbf{q} \mid \mathbf{s}_{ij})$ of a query to the jth answer in document i is a convex combination of five distributions: (1) a uniform model $p_{\mathcal{U}}$. (2) a corpus-wide model $p_{\mathcal{C}}$; (3) a model $p_{\mathbf{d}_i}$ constructed from the document containing $\mathbf{s}_{ij}$; (4) a model $p_{\mathcal{N}_{ij}}$ constructed from $\mathbf{s}_{ij}$ and the neighboring sentences in $\mathbf{d}_i$; (5) a model $p_{\mathbf{s}_{ij}}$ constructed from $\mathbf{s}_{ij}$ alone. (The $p_{\mathcal{N}}$ distribution is omitted for clarity.)

(The fidelity model doesn't suffer from zero-probabilities, at least not in the extractive summarization setting. Since a summary $\mathbf{s}$ is part of its containing document $\mathbf{d}$, every word in $\mathbf{s}$ also appears in $\mathbf{d}$, and therefore $p_{\mathbf{d}}(s) > 0$ for every word $s \in \mathbf{s}$. But we have no guarantee, for the relevance model, that a summary contains all the words in the query.)

We address this zero-probability problem by interpolating or "smoothing" the $p_{\mathbf{s}}$ model with four more robustly estimated unigram word models. Listed in order of decreasing variance but increasing bias away from $p_{\mathbf{s}}$, they are:

> $p_{\mathcal{N}}$: a probability distribution constructed using not only $\mathbf{s}$, but also all words within the six summaries (answers) surrounding $\mathbf{s}$ in $\mathbf{d}$. Since $p_{\mathcal{N}}$ is calculated using more text than just $\mathbf{s}$ alone, its parameter estimates should be more robust that those of $p_{\mathbf{s}}$. On the other hand, the $p_{\mathcal{N}}$ model is, by construction, biased away from $p_{\mathbf{s}}$, and therefore provides only indirect evidence for the relation between $\mathbf{q}$ and $\mathbf{s}$.
>
> $p_{\mathbf{d}}$: a probability distribution constructed over the entire document $\mathbf{d}$ containing $\mathbf{s}$. This model has even less variance than $p_{\mathcal{N}}$, but is even more biased away from $p_{\mathbf{s}}$.
>
> $p_{\mathcal{C}}$: a probability distribution constructed over all documents $\mathbf{d}$.
>
> $p_{\mathcal{U}}$: the uniform distribution over all words.

Figure 4 is a hierarchical depiction of the various language models which come into play in calculating $p(\mathbf{q} \mid \mathbf{s})$. Each summary model $p_{\mathbf{s}}$ lives at a leaf node, and the relevance $p(\mathbf{q} \mid \mathbf{s})$ of a query to that summary is a convex combination of the distributions at each node

Algorithm: *Shrinkage for $\vec{\lambda}$ estimation*

```
Input: Distributions $p_{\mathbf{s}}, p_{\mathbf{d}}, p_{\mathcal{C}}, p_{\mathcal{U}}$,
       $\mathcal{H} = \{\mathbf{d}, \mathbf{q}, \mathbf{s}\}$ (not used to
       estimate $p_{\mathbf{s}}, p_{\mathbf{d}}, p_{\mathcal{C}}, p_{\mathcal{U}}$)
Output Model weights $\vec{\lambda} = \{\lambda_{\mathbf{s}}, \lambda_{\mathcal{N}}, \lambda_{\mathbf{d}}, \lambda_{\mathcal{C}}, \lambda_{\mathcal{U}}\}$

1. Set $\lambda_{\mathbf{s}} \leftarrow \lambda_{\mathcal{N}} \leftarrow \lambda_{\mathbf{d}} \leftarrow \lambda_{\mathcal{C}} \leftarrow \lambda_{\mathcal{U}} \leftarrow 1/5$
2. Repeat until $\vec{\lambda}$ converges:
3.     Set $\mathrm{count}_x = 0$ for $x \in \{\mathbf{s}, \mathcal{N}, \mathbf{d}, \mathcal{C}, \mathcal{U}\}$
5.     (E-step) $\mathrm{count}_{\mathbf{s}} \leftarrow \mathrm{count}_{\mathbf{s}} + \frac{\lambda_{\mathbf{s}} p_{\mathbf{s}}(\mathbf{q})}{p(\mathbf{q} \mid \mathbf{s})}$
       (similarly for $\mathcal{N}, \mathbf{d}, \mathcal{C}, \mathcal{U}$)
6.     (M-step)$\lambda_{\mathbf{s}} \leftarrow \frac{\mathrm{count}_{\mathbf{s}}}{\sum_i \mathrm{count}_i}$
       (similarly for $\lambda_{\mathcal{N}}, \lambda_{\mathbf{d}}, \lambda_{\mathcal{C}}, \lambda_{\mathcal{U}}$)
```

along a path from the leaf to the root[2]:

$$p(\mathbf{q} \mid \mathbf{s}) = \lambda_{\mathbf{s}} p_{\mathbf{s}}(\mathbf{q}) + \lambda_{\mathcal{N}} p_{\mathcal{N}}(\mathbf{q}) + \lambda_{\mathbf{d}} p_{\mathbf{d}}(\mathbf{q}) + \lambda_{\mathcal{C}} p_{\mathcal{C}}(\mathbf{q}) + \lambda_{\mathcal{U}} p_{\mathcal{U}}(\mathbf{q}) \tag{5}$$

We calculate the weighting coefficients $\vec{\lambda} = \{\lambda_{\mathbf{s}}, \lambda_{\mathcal{N}}, \lambda_{\mathbf{d}}, \lambda_{\mathcal{C}}, \lambda_{\mathcal{U}}\}$ using the statistical technique known as *shrinkage* (Stein, 1955), a simple form of the EM algorithm (Dempster et al., 1977).

As a practical matter, if one assumes the $l_{\mathbf{s}}$ model assigns probabilities independently of $\mathbf{s}$, then we can drop the $l_{\mathbf{s}}$ term when ranking candidate summaries, since the score of all candidate summaries will receive an identical contribution from the $l_{\mathbf{s}}$ term. We make this simplifying assumption in the experiments reported in the following section.

3 Results

To gauge how well our proposed summarization technique performs, we applied it to two different real-world collections of answered questions:

> **Usenet FAQs**: A collection of 201 frequently-asked question documents from the `comp.*` Usenet hierarchy. The documents contained 1800 questions/answer pairs in total.
>
> **Call-center data**: A collection of questions submitted by customers to the companies Air Canada, Ben and Jerry, Iomagic, and Mylex, along with the answers supplied by company representatives. These four documents contain $10,395$ question/answer pairs.

We conducted an identical, parallel set of experiments on both. First, we used a randomly-selected subset of 70% of the question/answer pairs to calculate the language models $p_{\mathbf{s}}, p_{\mathcal{N}}, p_{\mathbf{d}}, p_{\mathcal{C}}$—a simple matter of counting word frequencies. Then, we used this same set of data to estimate the model weights $\vec{\lambda} = \{\lambda_{\mathbf{s}}, \lambda_{\mathcal{N}}, \lambda_{\mathbf{d}}, \lambda_{\mathcal{C}}, \lambda_{\mathcal{U}}\}$ using shrinkage. We reserved the remaining 30% of the question/answer pairs to evaluate the performance of the system, in a manner described below.

Figure 5 shows the progress of the EM algorithm in calculating maximum-likelihood values for the smoothing coefficients $\vec{\lambda}$, for the first of the three runs on the Usenet data. The quick convergence and the final $\vec{\lambda}$ values were essentially identical for the other partitions of this dataset.

The call-center data's convergence behavior was similar, although the final $\vec{\lambda}$ values were quite different. Figure 6 shows the final model weights for the first of the three experiments on both datasets. For the Usenet FAQ data, the corpus language model is the best predictor of the query and thus receives the highest weight. This may seem counterintuitive; one might suspect that answer to the query ($\mathbf{s}$, that is) would be most similar to, and therefore the best predictor of, the query. But the corpus model, while certainly biased away from the distribution of words found in the query, contains (by construction) no zeros, whereas each summary model is typically very sparse.

In the call-center data, the corpus model weight is lower at the expense of a higher document model weight. We suspect this arises from the fact that the documents in the Usenet data were all quite similar to one another in lexical content, in contrast to the call-center documents. As a result, in the call-center data the document containing $\mathbf{s}$ will appear much more relevant than the corpus as a whole.

To evaluate the performance of the trained QRS model, we used the previously-unseen portion of the FAQ data in the following way. For each test $(\mathbf{d}, \mathbf{q})$ pair, we recorded how highly the system ranked the correct summary $\mathbf{s}^{\star}$—the answer to $\mathbf{q}$ in $\mathbf{d}$—relative to the other answers in $\mathbf{d}$. We repeated this entire sequence three times for both the Usenet and the call-center data.

For these datasets, we discovered that using a uniform fidelity term in place of the $p(\mathbf{s} \mid \mathbf{d})$ model described above yields essentially the same result. This is not surprising: while the fidelity term is an important component of a real summarization system, our evaluation was conducted in an answer-locating framework, and in this context the fidelity term—enforcing that the summary be similar to the entire document from which

[2] By incorporating a $p_{\mathbf{d}}$ model into the relevance model, equation (6) has implicitly resurrected the dependence on $\mathbf{d}$ which we dropped, for the sake of simplicity, in deriving (1).

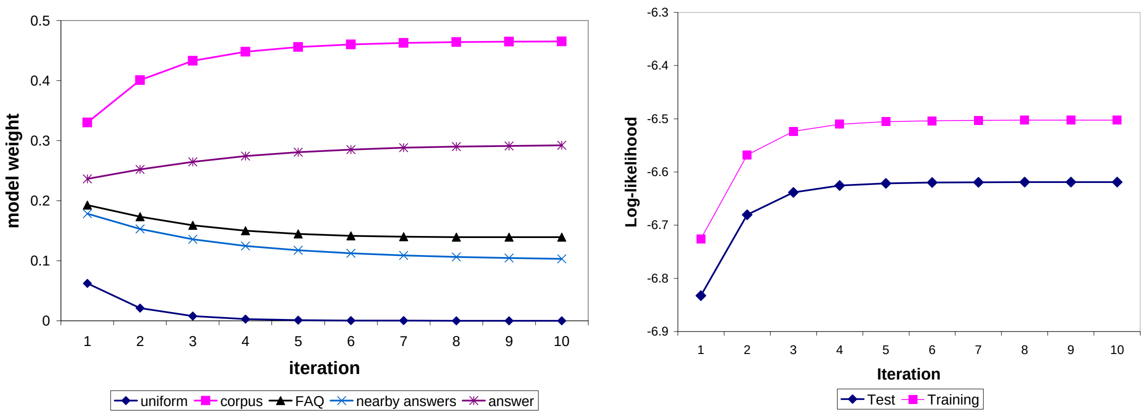

Figure 5: Estimating the weights of the five constituent models in (6) using the EM algorithm. The values here were computed using a single, randomly-selected 70% portion of the Usenet FAQ dataset. *Left*: The weights λ for the models are initialized to $1/5$, but within a few iterations settle to their final values. *Right*: The progression of the likelihood of the training data during the execution of the EM algorithm; almost all of the improvement comes in the first five iterations.

	$\lambda_{\mathbf{s}}$	$\lambda_{\mathcal{N}}$	$\lambda_{\mathbf{d}}$	$\lambda_{\mathcal{C}}$	$\lambda_{\mathcal{U}}$
Usenet FAQ	0.293	0.098	0.142	0.465	0
call-center	0.113	0.004	0.403	0.408	0.069

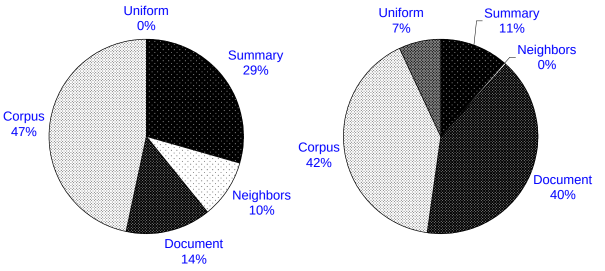

Figure 6: Maximum-likelihood weights for the various components of the relevance model $p(\mathbf{q} \mid \mathbf{s})$. *Left*: Weights assigned to the constituent models from the Usenet FAQ data. *Right*: Corresponding breakdown for the call-center data. These weights were calculated using shrinkage.

it was drawn—is not so important.

From a set of rankings $\{r_1, r_2, \ldots r_N\}$, one can measure the the quality of a ranking algorithm using the *harmonic mean rank*:

$$M \stackrel{\text{def}}{=} \frac{N}{\sum_{i=1}^{N} \frac{1}{r_i}}$$

A lower number indicates better performance; $M = 1$, which is optimal, means that the algorithm consistently assigns the first rank to the correct answer. Table 1 shows the harmonic mean rank on the two collections. The third column of Table 1 shows the result of a QRS system using a uniform fidelity model, the fourth corresponds to a standard tfidf-based ranking method (Ponte, 1998), and the last column reflects the performance of randomly guessing the correct summary from all answers in the document.

	trial	**# trials**	**LM**	**tfidf**	**random**
Usenet	1	554	1.41	2.29	4.20
FAQ	2	549	1.38	2.42	4.25
data	3	535	1.40	2.30	4.19
Call	1	1020	4.8	38.7	1335
center	2	1055	4.0	22.6	1335
data	3	1037	4.2	26.0	1321

Table 1: Performance of query-relevant extractive summarization on the Usenet and call-center datasets. The numbers reported in the three rightmost columns are harmonic mean ranks: lower is better.

4 Extensions

4.1 Question-answering

The reader may by now have realized that our approach to the QRS problem may be portable to the problem of *question-answering*. By question-answering, we mean a system which automatically extracts from a potentially lengthy document (or set of documents) the answer to a user-specified question. Devising a high-quality question-answering system would be of great service to anyone lacking the inclination to read an entire user's manual just to find the answer to a single question. The success of the various automated question-answering services on the Internet (such as AskJeeves) underscores the commercial importance of this task.

One can cast answer-finding as a traditional document retrieval problem by considering each candidate answer as an isolated document and ranking each candidate answer by relevance to the query. Traditional tfidf-based ranking of answers will reward candidate answers with many words in common with the query. Employing traditional vector-space retrieval to find answers seems attractive, since tfidf is a standard, time-tested algorithm in the toolbox of any IR professional.

What this paper has described is a first step towards more sophisticated models of question-answering. First, we have dispensed with the simplifying assumption that the candidate answers are independent of one another by using a model which explicitly accounts for the correlation between text blocks—candidate answers—within a single document. Second, we have put forward a principled statistical model for answer-ranking; $\arg\max_s p(\mathbf{s} \mid \mathbf{d}, \mathbf{q})$ has a probabilistic interpretation as the best answer to $\mathbf{q}$ within $\mathbf{d}$ is $\mathbf{s}$.

Question-answering and query-relevant summarization are of course not one and the same. For one, the criterion of containing an answer to a question is rather stricter than mere relevance. Put another way, only a small number of documents actually contain the answer to a given query, while every document can in principle be summarized with respect to that query. Second, it would seem that the $p(\mathbf{s} \mid \mathbf{d})$ term, which acts as a prior on summaries in (1), is less appropriate in a question-answering setting, where it is less important that a candidate answer to a query bears resemblance to the document containing it.

4.2 Generic summarization

Although this paper focuses on the task of query-relevant summarization, the core ideas—formulating a probabilistic model of the problem and learning the values of this model automatically from FAQ-like data—are equally applicable to generic summarization. In this case, one seeks the summary which best typifies the document. Applying Bayes' rule as in (1),

$$\begin{aligned} \mathbf{s}^\star &\equiv \arg\max_{\mathbf{s}} p(\mathbf{s} \mid \mathbf{d}) \\ &= \arg\max_{\mathbf{s}} \underbrace{p(\mathbf{d} \mid \mathbf{s})}_{\textit{generative}} \underbrace{p(\mathbf{s})}_{\textit{prior}} \qquad (6) \end{aligned}$$

The first term on the right is a generative model of documents from summaries, and the second is a prior distribution over summaries. One can think of this factorization in terms of a dialogue. Alice, a newspaper editor, has an idea $\mathbf{s}$ for a story, which she relates to Bob. Bob researches and writes the story $\mathbf{d}$, which we can view as a "corruption" of Alice's original idea $\mathbf{s}$. The task of generic summarization is to recover $\mathbf{s}$, given only the generated document $\mathbf{d}$, a model $p(\mathbf{d} \mid \mathbf{s})$ of how the Alice generates summaries from documents, and a prior distribution $p(\mathbf{s})$ on ideas $\mathbf{s}$.

The central problem in information theory is reliable communication through an unreliable channel. We can interpret Alice's idea $\mathbf{s}$ as the original signal, and the process by which Bob turns this idea into a document $\mathbf{d}$ as the channel, which corrupts the original message. The summarizer's task is to "decode" the original, condensed message from the document.

We point out this source-channel perspective because of the increasing influence that information theory has exerted on language and information-related applications. For instance, the source-channel model has been used for non-extractive summarization, generating titles automatically from news articles (Witbrock and Mittal, 1999).

The factorization in (6) is superficially similar to (1), but there is an important difference: $p(\mathbf{d} \mid \mathbf{s})$ is a *generative*, from a summary to a larger document, whereas $p(\mathbf{q} \mid \mathbf{s})$ is *compressive*, from a summary to a smaller query. This distinction is likely to translate in practice into quite different statistical models and training procedures in the two cases.

5 Summary

The task of summarization is difficult to define and even more difficult to automate. Historically, a rewarding line of attack for automating language-related problems has been to take a machine learning perspective: let a computer learn how to perform the task by "watching" a human perform it many times. This is the strategy we have pursued here.

There has been some work on learning a probabilistic model of summarization from text; some of the earliest work on this was due to Kupiec *et al.* (1995), who used a collection of manually-summarized text to learn the weights for a set of features used in a generic summarization system. Hovy and Lin (1997) present another system that learned how the position of a sentence affects its suitability for inclusion in a summary of the document. More recently, there has been work on building more complex, structured models—probabilistic syntax trees—to compress single sentences (Knight and Marcu, 2000). Mani and Bloedorn (1998) have recently proposed a method for automatically constructing decision trees to predict whether a sentence should or should not be included in a document's summary. These previous approaches focus mainly on the generic summarization task, not query relevant summarization.

The language modelling approach described here does suffer from a common flaw within text processing systems: the problem of synonymy. A candidate answer containing the term `Constantinople` is likely to be relevant to a question about Istanbul, but recognizing this correspondence requires a step beyond word frequency histograms. Synonymy has received much attention within the document retrieval community recently, and researchers have applied a variety of heuristic and statistical techniques—including pseudo-relevance feedback and local context analysis (Efthimiadis and Biron, 1994; Xu and Croft, 1996). Some recent work in statistical IR has extended the basic language modelling approaches to account for word synonymy (Berger and Lafferty, 1999).

This paper has proposed the use of two novel datasets for summarization: the frequently-asked questions (FAQs) from Usenet archives and question/answer pairs from the call centers of retail companies. Clearly this data isn't a perfect fit for the task of building a QRS system: after all, answers are not summaries. However, we believe that the FAQs represent a reasonable source of query-related document condensations. Furthermore, using FAQs allows us to assess the effectiveness of applying standard statistical learning machinery—maximum-likelihood estimation, the EM algorithm, and so on—to the QRS problem. More importantly, it allows us to evaluate our results in a rigorous, non-heuristic way. Although this work is meant as an opening salvo in the battle to conquer summarization with quantitative, statistical weapons, we expect in the future to enlist linguistic, semantic, and other non-statistical tools which have shown promise in condensing text.

Acknowledgments

This research was supported in part by an IBM University Partnership Award and by Claritech Corporation. The authors thank Right Now Tech for the use of the call-center question database. We also acknowledge thoughtful comments on this paper by Inderjeet Mani.

References

A. Berger and J. Lafferty. 1999. Information retrieval as statistical translation. In *Proc. of ACM SIGIR-99*.

A. Berger, P. Brown, S. Della Pietra, V. Della Pietra, J. Gillett, J. Lafferty, H. Printz, and L. Ures. 1994. The CANDIDE system for machine translation. In *Proc. of the ARPA Human Language Technology Workshop*.

Y. Chali, S. Matwin, and S. Szpakowicz. 1999. Query-biased text summarization as a question-answering technique. In *Proc. of the AAAI Fall Symp. on Question Answering Systems*, pages 52–56.

A. Dempster, N. Laird, and D. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39B:1–38.

E. Efthimiadis and P. Biron. 1994. UCLA-Okapi at TREC-2: Query expansion experiments. In *Proc. of the Text Retrieval Conference (TREC-2)*.

E. Hovy and C. Lin. 1997. Automated text summarization in SUMMARIST. In *Proc. of the ACL Wkshp on Intelligent Text Summarization*, pages 18–24.

F. Jelinek. 1997. *Statistical methods for speech recognition*. MIT Press.

K. Knight and D. Marcu. 2000. Statistics-based summarization—Step one: Sentence compression. In *Proc. of AAAI-00*. AAAI.

J. Kupiec, J. Pedersen, and F. Chen. 1995. A trainable document summarizer. In *Proc. SIGIR-95*, pages 68–73, July.

Chin-Yew Lin. 1999. Training a selection function for extraction. In *Proc. of the Eighth ACM CIKM Conference*, Kansas City, MO.

I. Mani and E. Bloedorn. 1998. Machine learning of generic and user-focused summarization. In *Proc. of AAAI-98*, pages 821–826.

J. Ponte and W. Croft. 1998. A language modeling approach to information retrieval. In *Proc. of SIGIR-98*, pages 275–281.

J. Ponte. 1998. *A language modelling approach to information retrieval*. Ph.D. thesis, University of Massachusetts at Amherst.

S. Sato and M. Sato. 1998. Rewriting saves extracted summaries. In *Proc. of the AAAI Intelligent Text Summarization Workshop*, pages 76–83.

C. Stein. 1955. Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. In *Proc. of the Third Berkeley symposium on mathematical statistics and probability*, pages 197–206.

F. Wancho. 1990. RFC 1153: Digest message format.

M. Witbrock and V. Mittal. 1999. Headline Generation: A framework for generating highly-condensed non-extractive summaries. In *Proc. of ACM SIGIR-99*, pages 315–316.

J. Xu and B. Croft. 1996. Query expansion using local and global document analysis. In *Proc. of ACM SIGIR-96*.