OCELOT: A system for summarizing web pages

Adam L. Berger*

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

Vibhu O. Mittal†*

†Just Research
4616 Henry Street
Pittsburgh, PA 15213, USA
mittal@cs.cmu.edu

Abstract We introduce OCELOT, a prototype system for automatically generating the "gist" of a web page by summarizing it. Although most text summarization research to date has focused on the task of news articles, web pages are quite different in both structure and content. Instead of coherent text with a well-defined discourse structure, they are more often likely to be a chaotic jumble of phrases, links, graphics and formatting commands. Such text provides little foothold for extractive summarization techniques, which attempt to generate a summary of a document by excerpting a contiguous, coherent span of text from it. This paper builds upon recent work in *non-extractive* summarization, producing the gist of a web page by "translating" it into a more concise representation rather than attempting to extract a text span verbatim. OCELOT uses probabilistic models to guide it in selecting and ordering words into a gist. This paper describes a technique for learning these models automatically from a collection of human-summarized web pages.

1 Introduction

Condensing a document into a useful and meaningful summary appears to require a level of intelligence not currently available in synthetic form. Therefore, most previous work on summarization has focused on the rather less ambitious goal of *extractive summarization*: selecting text spans—complete sentences or paragraphs—from the original document, and arranging the segments in some order to produce a summary. Unfortunately, this technique is a poor fit for web pages, which may contain only disjointed text.

Our approach to web page summarization is to design a system which can *synthesize* a summary, rather than extract one. The OCELOT system relies on a set of statistical models to guide its choice of words and how to arrange these words in a summary. The models themselves are built using standard machine learning algorithms, the input to which is a large collection of human-summarized web pages. Specifically, we use data from the Open Directory Project [17], a large and ongoing volunteer effort to collect and describe the "best" web sites on the Internet. As of January 2000, the Open Directory Project contained 868,227 web pages, each annotated with a short (roughly 13 word) human-authored summary.

Some important prior work in extractive summarization has explored issues such as cue phrases [13], positional indicators [6], lexical occurrence statistics [15], and the use of implicit discourse structure [14]. Most of this work relies fundamentally on a property of the source text which web pages often lack: a coherent stream of text with a logical discourse structure. Somewhat closer in spirit to OCELOT is work on combining an information extraction phase followed by generation; for instance, the FRUMP system [5] used templates for both information extraction and presentation—but once again on news stories, not web pages.

The very notion that a generic web page summarizer would be useful is predicated, in a sense, on the laziness of web page authors. After all, HTML offers multiple opportunities to web page authors (the title field, for instance, and the `meta description` field) to include a summary of the page's contents. But neither of these fields is required by HTML, and even when present, their content isoften only marginally informative. Moreover, web pages are often automatically generated, which again raises the issue of algorithmic page summarization. Lastly, query-relevant summaries (which are not the focus of this paper) will always need to be generated dynamically anyway.

The OCELOT project bears a close relation to recent work in automatic translation of natural language. The central idea of statistical machine translation is that, starting from bilingual corpus of text, one can apply statistical machine learning algorithms to estimate maximum-likelihood parameter values for a model of translation between the two languages. For instance, the Candide system at IBM [1] used the proceedings of the Canadian parliament—maintained in both French and English—to learn an English-French translation model. In an entirely analogous way, we propose to use Open Directory's "bilingual corpus" of web pages and their summaries to learn a mapping from web pages to summaries. Probably the fundamental difference between OCELOT's task and natural language translation is a degree of difficulty: a satisfactory translation of a sentence must capture its entire meaning, while a satisfactory summary is actually *expected* to leave out most of the source document's content.

Besides its pedigree in statistical machine translation, this work is most similar to the non-extractive summarization system proposed by Witbrock and Mittal [20] in the context of generating headlines automatically from news stories. It also bears some resemblance, in its use of probabilistic models for word relatedness, to some recent work in document retrieval [2].

2 Statistical gisting

Conceptually, we break down the task of building the OCELOT system into three tasks: (a) *content selection*:

determining which words should comprise the summary, (b) *word ordering*: arranging these words into a readable summary, and (c) *search*: for a given web page, find that sequence of words which is optimal in the dual senses of content and readability.

Content Selection

This paper proposes two methods for word selection. The simpler of the strategies is to select words according to the frequency of their appearance in the document $\mathbf{d}$. That is, if word w appears with frequency $\lambda(w \mid \mathbf{d})$ in $\mathbf{d}$, then it should appear in a gist $\mathbf{g}$ of that document with the same frequency:

$$E[\lambda(w \mid \mathbf{g})] = E[\lambda(w \mid \mathbf{d})].$$

Here $E[\cdot]$ is the expectation operator. This technique is essentially identical to the "language modelling approach" to document retrieval proposed recently by Ponte and Croft [18].

A natural extension is to allow words which do not appear in the document to appear in the gist; to do so, we adopt a technique recently introduced in the document retrieval literature [2] for automatically discovering words with similar or related meaning.

Surface Realization

In general, the probability of a word appearing at a specific position in a gist depends on the previous words. If the word `platypus` already appeared in a summary, for instance, it's not likely to appear again. And although `the` might appear multiple times in a summary, it is unlikely to appear in position k if it appeared in position $k-1$. The gisting model which OCELOT uses takes into account the ordering of words in a candidate gist by using an n-gram model of language.

Search

Even though we have described the tasks of content selection and surface realization separately, in practice OCELOT selects and arranges words simultaneously when constructing a summary. That is, the system produces a gist of a document $\mathbf{d}$ by searching over all candidates $\mathbf{g}$ to find that gist which maximizes the product of a content selection term and a surface realization term. We apply generic Viterbi search techniques to efficiently find a near-optimal summary [7].

3 Three models of gisting

This section introduces three increasingly sophisticated statistical models to generate the gist of a given document. We defer until the next section a discussion of how to estimate the parameters of these models.

The idea of viewing document gisting as a problem in probabilistic inference is not prevalent. Intuitively, we justify this perspective as follows. To begin, postulate a probabilistic model $p(\mathbf{g} \mid \mathbf{d})$ which assigns a value (a probability) to the event that the string of words $\mathbf{g} = \{g_1, g_2, \ldots g_n\}$ is the best gist of the document $\mathbf{d} = \{d_1, d_2 \ldots d_m\}$. One way to think about such a model is as the limiting value of a hypothetical process. Give the document $\mathbf{d}$ to a large number of people and ask each to produce a gist of the document. The value $p(\mathbf{g} \mid \mathbf{d})$ is the fraction of participants who produce $\mathbf{g}$ as the number of participants goes to infinity.

Given a document $\mathbf{d}$, the optimal gist for that document is therefore

$$\mathbf{g}^\star \quad = \quad \text{argmax}_{\mathbf{g}} p(\mathbf{g} \mid \mathbf{d}). \tag{1}$$

In this section, we hypothesize a few forms of the model and apply traditional statistical methods—maximum-likelihood estimation and in particular the expectation-maximization (EM) algorithm—to compute the parameters of the hypothesized models.

I. A "bag of words" approach

According to this model, a person gisting a document $\mathbf{d}$ begins by selecting a length n for the summary according to some probability distribution ϕ over possible lengths. Then, for each of the n assigned positions in the gist, he draws a word at random, from the document to be gisted, and fills in the current slot in the gist with that word. In combinatorial terminology, the values of the words in the gist are *i.i.d.* variables: the result of n independently and identically distributed random trials. In imagining a person composes a gist in such a way, this model makes a strong independence assumption among the words in the input document, viewing them as an unordered collection.

Algorithm 1: *Bag of words gisting*

Input: Document $\mathbf{d}$ with word distribution $\lambda(\cdot \mid \mathbf{d})$;
Distribution ϕ over gist lengths;
Output: Gist $\mathbf{g}$ of $\mathbf{d}$

1. Select a length n for the gist: $n \sim \phi$
2. Do for $i = 1$ to n
3. Pick a word from the document: $w \sim \lambda(\cdot \mid \mathbf{d})$
4. Set $g_i \leftarrow w$

Once again denoting the frequency of word w in $\mathbf{d}$ by $\lambda(w \mid \mathbf{d})$, the probability that the person will gist $\mathbf{d}$ into $\mathbf{g} = \{g_1, g_2, \ldots g_n\}$ is

$$p(\mathbf{g} \mid \mathbf{d}) = \phi(n) \prod_{i=1}^{n} \lambda(g_i \mid \mathbf{d}).$$

Though this model is simplistic, it makes one plausible assumption: the more frequently a word appears in a document, the more likely it will be included in a gist of that page. This algorithm is essentially identical (albeit in a different setting) to the language modelling approach to document retrieval introduced by Ponte and Croft [18].

II. Accounting for unseen words

Algorithm 1 is limited in a number of ways, one of which is that the generated summaries can only contain words from the input document. A logical extension is to relax this restriction by allowing the gist to contain words not present in the source document. The idea is to draw (as before) a word according to the word frequencies in the input document, but then replace the drawn word with a related word—a synonym, perhaps, or a word used in similar contexts—before adding it to the gist.

To determine which word to substitute in place of the sampled word, we use a probability distribution $\sigma(\cdot \mid w)$: if u is a very closely related word to v, then we

expect $\sigma(u \mid v)$ to be large. If the system recognizes W words, then the σ model is just a $W \times W$ stochastic matrix. (One could reasonably expect that the diagonal entries of this matrix, corresponding to "self-similarity" probabilities, will typically be large.) We will call this algorithm *expanded-lexicon gisting*, since the lexicon of candidate words for a summary of $\mathbf{d}$ are no longer just those appearing in $\mathbf{d}$.

This "draw then replace with a similar word" model of document gisting is similar to the IBM-style model of language translation [3]. The simplest of this family of statistical models pretends that a person renders a document into a different language by drawing words from it and translating each word—"draw, then translate" rather than "draw, then replace with a related word."

Algorithm 2: *Expanded-lexicon gisting*

Input: Document $\mathbf{d}$ with word distribution $\lambda(\cdot \mid \mathbf{d})$;
Distribution ϕ over gist lengths;
Word-similarity model $\sigma(\cdot \mid u)$ for all words w

Output: Gist $\mathbf{g}$ of $\mathbf{d}$

1. Select a length for the gist: $n \sim \phi$
2. Do for $i = 1$ to n
3. Pick a word from the document: $u \sim \lambda(\cdot \mid \mathbf{d})$
4. Pick a replacement for that word: $v \sim \sigma(\cdot \mid w)$
5. Set $\mathbf{g}_i \leftarrow v$

As before, we can write down an expression for the probability that a person following this procedure will select, for an input document $\mathbf{d}$, a specific gist $\mathbf{g} = \{g_1, g_2, \ldots g_n\}$. Assuming $\mathbf{d}$ contains m words,

$$\begin{aligned} p(\mathbf{g} \mid \mathbf{d}) &= \phi(n) \prod_{i=1}^{n} p(g_i \mid \mathbf{d}) \qquad (2) \\ &= \phi(n) \prod_{i=1}^{n} \left(\frac{1}{m}\right) \sum_{j=1}^{m} \sigma(g \mid d_j) \end{aligned}$$

Algorithm 2 represents a step towards more realistic gisting, but is still lacking in certain capabilities. Perhaps most egregious is the word-independence assumption inherited from Algorithm 1.

III. Generating readable summaries

One can extend Algorithm 2 by enforcing that the sequence of words comprising a candidate gist are coherent. For instance, one could ensure that two prepositions never appeared next to each other in a gist. Algorithm 3 attempts to capture a notion of syntactic regularity by scoring candidate gists not only on how well they capture the essence (the process of content selection) of the original document, but also how coherent they are as a string of English words.

The coherence or readability of an n-word string $\mathbf{g} = \{g_1, g_2, \ldots g_n\}$ comprising a candidate gist is the *a priori* probability of seeing that string of words in text, which we write as $p(\mathbf{g})$.[1] One can factor $p(\mathbf{g})$ into a product of conditional probabilities as

$$p(\mathbf{g}) = \prod_{i=1}^{n} p(g_i \mid g_1, g_2 \ldots g_{i-1})$$

In practice, we use a *trigram* model for $p(\mathbf{g})$, meaning that

$$p(g_i \mid g_1, g_2 \ldots g_{i-1}) \approx p(g_i \mid g_{i-2} g_{i-1}) \qquad (3)$$

Although n-gram models of language make a quite strong (and clearly false) locality assumption about text, they have nonetheless proven successful in many human language technologies, including speech and optical character recognition [11, 16].

To devise a formal model of gisting which accounts for both readability and fidelity to the source document, we apply Bayes' Rule to (1):

$$\begin{aligned} \mathbf{g}^\star &= \text{argmax}_{\mathbf{g}} p(\mathbf{g} \mid \mathbf{d}) \\ &= \text{argmax}_{\mathbf{g}} p(\mathbf{d} \mid \mathbf{g})\ p(\mathbf{g}). \qquad (4) \end{aligned}$$

According to (4), the optimal gist is the product of two terms: first, a fidelity term $p(\mathbf{d} \mid \mathbf{g})$, measuring how closely $\mathbf{d}$ and $\mathbf{g}$ match in content, and a readability term $p(\mathbf{g})$, measuring the *a priori* coherence of the gist $\mathbf{g}$.

For the readability term we can use the language model (3). For the content proximity model $p(\mathbf{d} \mid \mathbf{g})$, we can simply reverse the direction of (2):

$$\begin{aligned} p(\mathbf{d} \mid \mathbf{g}) &= \hat{\phi}(m) \prod_{i=1}^{n} p(d_i \mid \mathbf{g}) \qquad (5) \\ &= \hat{\phi}(m) \prod_{i=1}^{n} \sum_{j=1}^{m} \left(\frac{1}{n}\right) \sigma(d \mid g_j) \end{aligned}$$

Here $\hat{\phi}$ is a length distribution on *documents*, which presumably differs from the length distribution on summaries[2].

Algorithm 3: *Readable gisting*

Input: Document $\mathbf{d}$ with word distribution $\lambda(\cdot \mid \mathbf{d})$;
Distribution ϕ over gist lengths;
Word-similarity model $\sigma(\cdot \mid w)$ for all words w
Trigram language model $p(\mathbf{g})$ for gists

Output: Gist $\mathbf{g}$ of $\mathbf{d}$

1. Select a length n for the gist: $n \sim \phi$
2. Find, by searching, the sequence $\mathbf{g} = \{g_1, g_2, \ldots g_n\}$ which maximizes $p(\mathbf{d} \mid \mathbf{g}) p(\mathbf{g})$

One can think of $p(\mathbf{g})$ as a prior distribution on candidate gists, and $p(\mathbf{d} \mid \mathbf{g})$ as the probability that the document $\mathbf{d}$ would arise from the gist $\mathbf{g}$.

One way to make sense of the seeming reverse order of prediction in equation (4) is with the source-channel framework from information theory. Imagine that the document to be gisted was originally itself a gist: the germ of an idea in the imagination of whoever composed the page. The actual composition of the web page is like a corruption of this original message, and the goal of a gisting algorithm is to recover, from the web page itself, this original, hidden, ideal gist. In doing so, the algorithm makes use of a model $p(\mathbf{d} \mid \mathbf{g})$ of how ideas for web pages get converted ("corrupted") into pages, and a model $p(\mathbf{g})$ for what constitutes an *a priori* likely

[1] We are overloading the term p to refer to multiple probability distributions, as well as values assigned by those distributions, but have endeavored to ensure the proper meaning is clear from context.

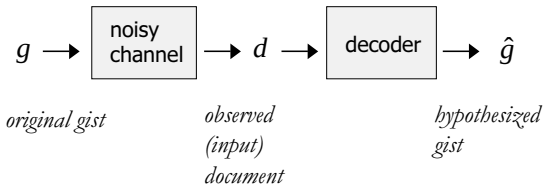


Figure 1: Gisting from a source-channel perspective

and unlikely gist. Figure 1 illustrates this information-theoretic interpretation.

Algorithm 3 leaves unspecified the somewhat involved matter of searching for the optimal $\mathbf{g}$. Speech and handwriting recognition systems face a similar problem in attempting to generate a transcription of a detected signal (an acoustic or written signal) which both accounts for the perceived signal and is a coherent string of words. As mentioned earlier, the most successful technique has been to apply Viterbi-type search procedure, and this is the strategy that OCELOT adopts.

4 A source of summarized web pages

To apply machine learning to this problem, we need a large collection of gisted web pages for training. A good corpus for this task can be obtained from the Open Directory Project (`http://dmoz.org`), a directory of web pages which relies on a large number of volunteers to grow and maintain its contents. What makes Open Directory useful for learning to gist is that each of its entries—individual web sites—is summarized manually, by an Open Directory volunteer.

As of January 2000, Open Directory contained 868,227 descriptions of web pages. We attempted to download these descriptions along with the contents of the associated web page, but since individual web sites often have `robots.txt` files, or otherwise restrict requests for data from automated spider programs, we were unable to obtain many of these pages. Of those we were able to download, we applied the following steps:

- Normalize text: remove punctuation, convert all text to lowercase; replace numbers by the symbol NUM; remove each occurrence of the 100 most common overall words (stopword-filtering).
- Remove all links, images, and meta-information from the web pages
- Remove pages containing adult-oriented content[3];
- Remove HTML markup information from the pages;
- Remove pairs whose pages contained frames;
- Remove pairs whose pages that had been moved since they had been included in the list; in other words, pages which were just "Page not found errors";
- Remove pairs whose Web pages or gists were too short—less than 400 or 60 characters, respectively.
- Remove duplicate web pages;

[2] OCELOT's task is to find the best gist of a document, and the $\hat{\phi}$ term will contribute equally to every candidate gist. We can therefore ignore this term from now on.

[3] Skipping the pages listed in the `Adult` hierarchy goes far, but not the entire way, towards solving this problem.

Filtered: `svenska sidan utsigten antik kuriosa welcome we sell and buy antiques and collectibles of good quality our shop is in central karlskrona sweden at borgmstarekajen close to the county museum and fisktorget see the map you will find swedish porcelain china glass and textiles here we are specialized in porcelain from karlskrona we have been in business since NUM welcome to our shop our opening hours are tuesday wednesday and thursday NUM NUM NUM NUM saturday NUM NUM NUM NUM other times on agreement bookmark this site copyright NUM utsigten antik kuriosa updated NUM NUM NUM contact us with email to utsigtenantikviteter net or phone NUM NUM`

Open Directory gist: `sell and buy antiques and collectibles of good quality our shop is in central karlskrona sweden`

Figure 2: A web page (top), after filtering (middle), and the Open Directory-provided gist of the page (bottom).

- Partition the remaining set of pairs into a training set (99%) and a test set (1%). (Traditionally when evaluating a machine learning algorithm, one reserves more than this fraction of the data for testing. But one percent of the Open Directory dataset comprises over a thousand web pages, which was sufficient for the evaluations we performed.)

At the conclusion of this process, we were left with 103,064 summaries and links in the training set, and 1046 in the test set. Figure 2 shows a "before and after" example of this filtering process on a single web page, along with Open Directory's summary of this page. After processing, the average length of the summaries was 13.6 words, and the average length of the documents was 211.1 words.

5 Training a statistical model for gisting

This section discusses the training of various statistical models for the specific task of gisting web pages.

5.1 Estimating a model of word relatedness

Recall that, in Algorithm 3, the underlying statistical model $p(\mathbf{d} \mid \mathbf{g})$ which measures the "proximity" between a web page $\mathbf{d}$ and a candidate gist $\mathbf{g}$ is a generative model, predicting $\mathbf{d}$ from $\mathbf{g}$. This model factors, as seen in (5), into a product of sums of $\sigma(d \mid g)$ terms: the probability

that a word g in a gist of a web page gives rise to a word d in the page itself. We now describe how one can learn these word-to-word "relatedness" probabilities automatically from a collection of summarized web pages.

If there are W_g different recognized words in gists and W_p different recognized words in web pages, then calculating the parameters of the individual σ models is equivalent to filling in the entries of a $W_g \times W_p$ stochastic matrix. As mentioned above, there exist algorithms, first developed in the context of machine translation [3], for estimating maximum-likelihood values for the entries of this matrix using a collection of bilingual text. In this case, the two "languages" are the verbose language of documents and the succinct language of gists.

For the purposes of estimating the σ parameters, we introduce the notion of an *alignment* $\mathbf{a}$ between sequences of words, which captures how words in gists produce the words in a web page. We'll also make use of an artificial NULL added to position zero of every gist, whose purpose is to generate those words in the web page not strongly correlated with any other word in the gist.

Using $\mathbf{a}$, we can decompose $p(\mathbf{d} \mid \mathbf{g})$ as

$$p(\mathbf{d} \mid \mathbf{g}) = \sum_{\mathbf{a}} p(\mathbf{d}, \mathbf{a} \mid \mathbf{g}) = \sum_{\mathbf{a}} p(\mathbf{d} \mid \mathbf{a}, \mathbf{g}) p(\mathbf{a} \mid \mathbf{g}) \quad (6)$$

Making the simplifying assumption that to each word in $\mathbf{d}$ corresponds exactly one "parent" word in $\mathbf{g}$ (possibly the null word), we can write

$$p(\mathbf{d} \mid \mathbf{a}, \mathbf{g}) = \prod_{i=1}^{m} \sigma(d_i \mid g_{a_i}) \quad (7)$$

Here g_{a_i} is the gist word aligned with the ith web page word. Figure 3 illustrates a sample alignment between a small web page and its summary.

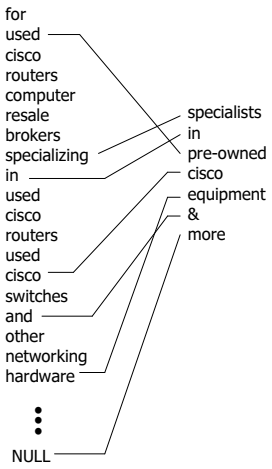


Figure 3: One of the exponentially many alignments between this imaginary document/gist pair. Calculating the score $p(\mathbf{d} \mid \mathbf{g})$ of a document/gist pair involves, implicitly, a sum over *all* possible ways of aligning the words.

If $\mathbf{d}$ contains m words and $\mathbf{g}$ contains $n+1$ words (including the null word), there are $(n+1)^m$ alignments between $\mathbf{g}$ and $\mathbf{d}$. By assuming that all these alignments are equally likely, we can write

$$p(\mathbf{d} \mid \mathbf{g}) = \frac{p(m \mid \mathbf{g})}{(n+1)^m} \sum_{A} \prod_{i=1}^{m} \sigma(d_i \mid g_{a_i}) \quad (8)$$

We view the Open Directory dataset as a collection of web pages and their summaries,

$$\mathcal{C} = \{(\mathbf{d}_1, \mathbf{g}_1), (\mathbf{d}_2, \mathbf{g}_2), (\mathbf{d}_3, \mathbf{g}_3) \ldots$$

The likelihood method suggests that one should adjust the parameters of (8) in such a way that the model assigns as high a probability as possible to $\mathcal{C}$. This maximization must be performed, of course, subject to the constraints $\sum_d \sigma(d \mid g) = 1$ for all words g. Using Lagrange multipliers,

$$\sigma(d \mid g) = Z \sum_{\mathbf{a}} p(\mathbf{d}, \mathbf{a} \mid \mathbf{g}) \sum_{j=1}^{m} \delta(d, d_j) \delta(g, g_{a_j}), \quad (9)$$

where Z is a normalizing factor and δ is the Kronecker delta function.

The parameter $\sigma(d \mid g)$ appears explicitly in the left-hand side of (9), and implicitly in the right. By repeatedly solving this equation for all pairs d, g (in other words, applying the EM algorithm), one eventually reaches a stationary point of the likelihood.

Equation (9) contains a sum over alignments, which is exponential and suggests that the computing the parameters in this way is infeasible. In fact, this is not the case, since

$$\sum_{\mathbf{a}} \prod_{i=1}^{m} \sigma(d_i \mid g_{a_i}) = \prod_{i=1}^{m} \sum_{j=0}^{n} \sigma(d_i \mid g_j) \quad (10)$$

This rearranging means that computing $\sum_{\mathbf{a}} p(\mathbf{d}, \mathbf{a} \mid \mathbf{g})$ requires only $\Theta(mn)$ work, rather than $\Theta(n^m)$.

Figure 4 shows the progress of the perplexity of the Open Directory training data during the six iterations of training. For the EM training, we used all the 103,064 gist/web page pairs in the training set, totaling 24,231,164 words in the web page data and 1,922,393 words in the summaries. The vocabularies were constructed from the top 65535 words appearing at least twice; all other words were mapped to the symbol OOV (for "out of vocabulary").

Table 1 shows the top entries for a few selected words.

5.2 Estimating a language model

OCELOT attempts to ensure that its hypothesized gists are readable with the help of a trigram model of the form (3). For a W-word vocabulary, such a model is characterized by W^3 parameters: $p(w \mid u, v)$ is the probability that the word w follows the bigram u, v.

We constructed such a model by calculating $p(w \mid u, v)$ values from the full set of 868,227 Open Directory gists, comprising 82MB. Building the language model consisted of the following steps:

1. Construct a vocabulary of active words from those words appearing at least twice within the collection of summaries. This amounted to 37,863 unique words.

2. Build a trigram word model from this data using maximum-likelihood estimation.

job	job 0.194	jobs 0.098	career 0.028	employment 0.028
wilderness	wilderness 0.123	the 0.061	national 0.032	forest 0.028
associations	associations 0.083	association 0.063	OOV 0.020	members 0.013
ibm	ibm 0.130	business 0.035	solutions 0.019	support 0.017
camera	camera 0.137	cameras 0.045	photo 0.020	photography 0.014
investments	investments 0.049	investment 0.046	fund 0.033	financial 0.025
contractor	contractor 0.080	contractors 0.030	construction 0.027	our 0.016
quilts	quilts 0.141	quilt 0.074	i 0.036	quilting 0.034
exhibitions	exhibitions 0.059	OOV 0.056	art 0.048	museum 0.041
ranches	ranches 0.089	springs 0.034	colorado 0.032	ranch 0.030

Table 1: Word-relatedness models $\sigma(\cdot \mid w)$ for selected words w, computed in an unsupervised manner from the Open Directory training data.

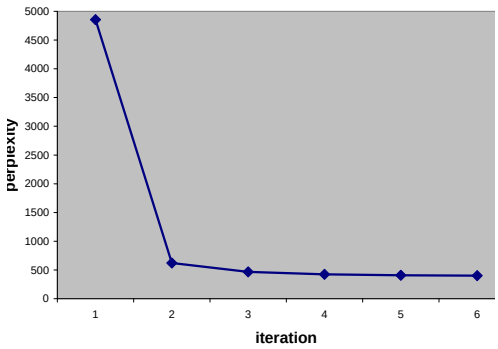


Figure 4: Progress of the EM algorithm, measured in training set perplexity, during six iterations.

Summary length	*Word overlap ratio*
4	0.4116
5	0.3489
6	0.3029
7	0.2745
8	0.2544
9	0.2353

Table 2: A simple evaluation of the bag-of-words gisting algorithm (Algorithm 1).

3. "Smooth" this model (by assigning some probability mass to unseen trigrams) using the Good-Turing estimate [9].

For the final two steps we used the publicly-available CMU-Cambridge Language Modelling Toolkit [4].

6 Evaluation

Summarization research has grappled for years with the issue of how to perform a rigorous evaluation of a summarization system [8, 10, 12]. We have not solved that problem here, but nonetheless present a series of quantitative and qualitative assessments of the functionality of the various components of OCELOT.

6.1 Measuring word overlap

We begin by examining the behavior of the simplest of the proposed gisting algorithms. To this end, we borrow from the speech recognition research literature the notion of *word error rate*. In this context, word error rate corresponds to the lexical overlap between the true and hypothesized gist. Admittedly this is of dubious merit as a criterion for evaluating a summarizer. For one, a hypothesized gist may be as good as the real gist, yet share very few words with it. Conversely, a hypothesized summary may contain exactly the same words as the true summary, but arranged nonsensically.

Despite these caveats, we applied Algorithm 1 to the evaluation set of Open Directory pages, and defined *word overlap* to be the number of words which appeared in both the hypothesized and actual gist, divided by the size of the hypothesized gist. That is, word overlap is meant to measure the fraction of hypothesized words which are likely to be "appropriate." The results appear in Table 2.

6.2 Evaluating the language model

Since OCELOT uses both a language model and a word-relatedness model to calculate a gist of a web page, isolating the contribution of the language model to the performance of OCELOT is a difficult task. Once again the speech recognition literature suggest a strategy: gauge the performance of a language model in isolation from the rest of the summarizer by measuring how well it predicts a previously-unseen collection $\mathcal{G}$ of actual summaries. Specifically, we can calculate the probability which the language model assigns to a set of unseen Open Directory gists; the higher the probability, the better the model.

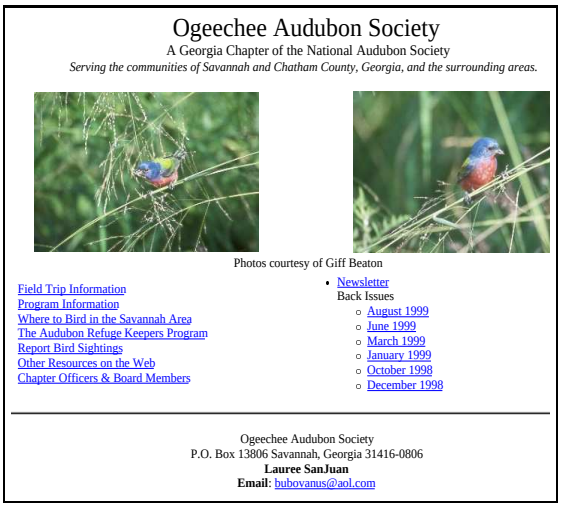
The log-likelihood assigned by λ to an n-word collection $\mathcal{G}$ is

$$\log p(\mathcal{G}) = \sum_{i=1}^{n} \log p(g_i \mid g_{i-2} g_{i-1})$$

The *perplexity* of $\mathcal{G}$ according to the trigram model is related to $\log p(\mathcal{G})$ by

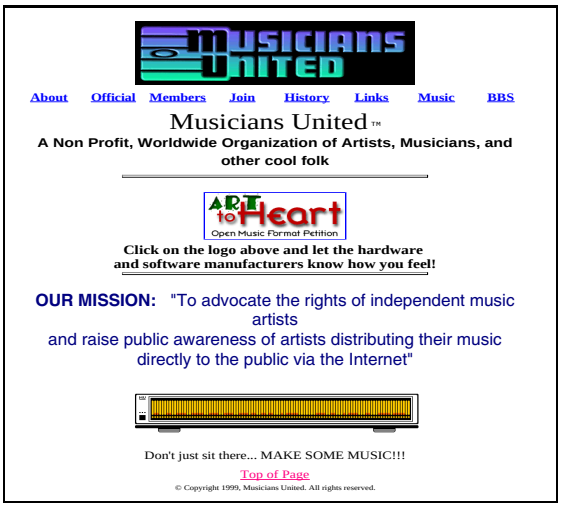
$$\Pi(\mathcal{G}) = \exp\left\{ -\left(\frac{1}{n}\right) \sum_{i=1}^{n} \log p(g_i \mid g_{i-2} g_{i-1}) \right\}$$

Roughly speaking, perplexity can be thought of as the average number of "guesses" the language model must make to identify the next word in a string of text comprising a gist drawn from the test data. An upper bound in this case is $\mid W \mid = 37,863$: the number of different words which could appear in any single position in a gist. To the test collection of 1046 gists consisting of 20,775 words, the language model assigned a perplexity of 362. This is to be compared with the perplexity of the same text as measured by the weaker bigram and unigram models: 536 and 2185, respectively.



Open Directory gist: a chapter of the national audubon society serving the communities of savannah chatham county and the surrounding areas

OCELOT *gist:* audubon society atlanta area savannah georgia chatham and local birding savannah keepers chapter of the audubon georgia and leasing



Open Directory gist: to advocate the rights of independent music artists and raise public awareness of artists distributing their music directly to the public via the internet

OCELOT *gist:* the music business and industry artists raise awareness rock and jazz

Figure 5: Selected output from OCELOT. The original web page is shown above with the actual and hypothesized gists below.

6.3 Gisted web pages

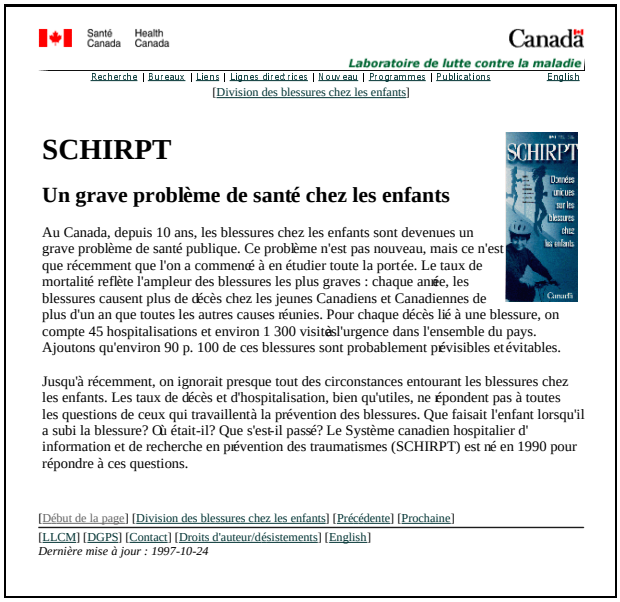
Figure 5 shows two examples of the behavior of OCELOT on web pages selected from the evaluation set of Open Directory pages—web pages, in other words, which OCELOT did not observe during the learning process.

7 Translingual gisting

With essentially no adaptation, OCELOT could serve as a translingual summarization system: a system for producing the gist of a document in another language. The only necessary ingredient is a collection of documents in one language with summaries in another: the word-relatedness matrix would then automatically become a matrix of translation probabilities.

We have conducted some initial proof-of-concept experiments to generate English summaries of French web pages. For this purpose, we used the same language model on (English) summaries as in Section 6. Our attempts to locate a suitably large parallel corpus of French web pages and English summaries from which to estimate σ were not fruitful. Instead, we used the proceedings of the Canadian parliament (known as "Hansards"), transcribed in both French and English, to learn the mapping from French words to English words. The subset of the Hansard corpus we used contained two million parallel English/French sentences, comprising approximately 43 million words. Using a model trained on parliamentary discourse on the domain of web page gisting has its shortcomings: words may sometimes have quite different statistics in the Hansards than in the average web page. In future work we plan to use a web spidering tool to identify and download web pages published in different languages [19].

Figure 6 gives an example of French web page gisted into English.



OCELOT *gist:* health protection branch of the protection of health anti inflation guidelines health of animals in volume may table of contents of our children in central canada review of u.s beginning at page volume final vote day may

Figure 6: Selected output from a French-English version of OCELOT.

8 Conclusion

This paper has described the philosophy, architecture, and performance of OCELOT, a prototype web-page sum-

marization system. OCELOT has the ability to generate coherent summaries that are not excerpts from the original document; in fact, they are likely to contain words not even appearing in the original document. This approach to summarization appears particularly well-suited to web pages, which are often disjointed lists of phrases and links not amenable to traditional extraction-based techniques.

OCELOT represents an initial foray into automating the process of web page gisting. Asked to summarize a web page, a reasonably intelligent person would no doubt make use of information that OCELOT ignores. For instance, text often appears in web pages in the form of images, but this information is lost without a front-end OCR module to extract this text. Also, the system does not exploit structural clues about what's important on the page. For instance, the text within the `<title>` ... `</title>` region is likely to be relatively important, while text within a `<small>` ... `</small>` is probably less important.

The performance of the system could clearly benefit from better (more sophisticated) content selection and surface realization models. For instance, even though Algorithm 3 strives to produce well-formed summaries with the help of a trigram model of language, the model makes no effort to preserve word order between document and summary. OCELOT has no mechanism for distinguishing between the documents `Dog bites man` and `Man bites dog`. In future work, we plan to investigate some of these and other related issues.

Acknowledgments

We thank the Open Directory Project for making available the raw data comprising their ontology. This research was supported in part by an IBM University Partnership Award and by Claritech Corporation.

References

[1] Berger, A., Brown, P., Della Pietra, S., Della Pietra, V., Gillett, J., Lafferty, J., Printz, H., and Ures, L. The Candide system for machine translation. In *Proceedings of the ARPA Human Language Technology Workshop* (1994).

[2] Berger, A., and Lafferty, J. The Weaver system for document retrieval. In *Proceedings of the Eighth Text REtrieval Conference (TREC-8)* (1999).

[3] Brown, P., Della Pietra, S., Della Pietra, V., and Mercer, R. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics 19*, 2 (1993), 263–311.

[4] Clarkson, P., and Rosenfeld, R. Statistical language modeling using the CMU-Cambridge toolkit. In *Proceedings of Eurospeech '97* (1997).

[5] DeJong, G. F. An overview of the FRUMP system. In *Strategies for Natural Language Processing*, W. G. Lehnert and M. H. Ringle, Eds. Lawrence Erlbaum Associates, 1982, pp. 149–176.

[6] Edmundson, H. P. Problems in automatic extracting. *Communications of the ACM 7* (1964), 259–263.

[7] Forney, G. D. The Viterbi Algorithm. *Proceedings of the IEEE* (1973), 268–278.

[8] Goldstein, J., Kantrowitz, M., Mittal, V. O., and Carbonell, J. Summarizing Text Documents: Sentence Selection and Evaluation Metrics. In *Proceedings of the 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-99)* (Berkeley, CA., 1999), pp. 121–128.

[9] Good, I. The population frequencies of species and the estimation of population parameters. *Biometrika 40* (1953).

[10] Hand, T. F. A proposal for task-based evaluation of text summarization systems. In *ACL/EACL-97 Workshop on Intelligent Scalable Text Summarization* (July 1997), pp. 31–36.

[11] Jelinek, F. *Statistical methods for speech recognition.* MIT Press, 1997.

[12] Jing, H., Barzilay, R., McKeown, K., and Elhadad, M. Summarization evaluation methods experiments and analysis. In *AAAI Intelligent Text Summarization Workshop* (Mar. 1998), pp. 60–68.

[13] Luhn, P. H. Automatic creation of literature abstracts. *IBM Journal* (1958), 159–165.

[14] Marcu, D. From discourse structures to text summaries. In *Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization* (1997), pp. 82–88.

[15] Mathis, B. A., Rush, J. E., and Young, C. E. Improvement of automatic abstracts by the use of structural analysis. *Journal of the American Society for Information Science 24* (1973), 101–109.

[16] Nathan, K., Beigi, H., Subrahmonia, J., Clary, G., and Maruyama, H. Real-time on-line unconstrained handwriting recognition using statistical methods. In *Proceedings of the IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP)* (1995).

[17] The Open Directory project: `http://dmoz.org`.

[18] Ponte, J., and Croft, W. A language modeling approach to information retrieval. In *Proceedings of the 21st International Conference on Research and Development in Information Retrieval* (1998), pp. 275–281.

[19] Resnick, P. Mining the Web for bilingual text. In *Proceedings of the ACL'99* (1999).

[20] Witbrock, M., and Mittal, V. O. Headline generation: A framework for generating highly-condensed non-extractive summaries. In *Proceedings of the 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (1999), pp. 315–316.