Automatically Building a Corpus for a Minority Language from the Web

Rosie Jones and Rayid Ghani
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213 USA
<firstname>.<lastname>@cs.cmu.edu

Abstract

We present an approach to language-specific query-based sampling which, given a single document in a target language, can find many more examples of documents in that language, by automatically constructing queries to access such documents on the world wide web. We propose a number of methods for building search queries to quickly obtain documents in the target language. They perform accurately and efficiently for building a corpus of documents in Tagalog starting from a single seed document, when these documents are only 2.5% of the documents in a collection. We found that sampling with a query consisting of a word seleccted according to its probability from the minority language corpus constructed so far was very successful. This method built a corpus of documents with word frequencies similar to those in the corpus based on all Tagalog documents in our collection, and required a relatively small number of search queries. It also quickly acquired a good coverage of vocabulary terms. However, adding an element of randomness to the query may give greater coverage, although more queries are required.

1 Introduction

Electronic text corpora are used for modeling language in many language technology applications, including speech recognition (Jelinek, 1999), optical character recognition, handwriting recognition, machine translation (Brown et al., 1993), and spelling correction (Golding and Roth, 1999). They are also useful for linguistic and sociolinguistic studies, as they are readily searchable and statistics can easily be computed.

The Linguistic Data Consortium (LDC) has corpora for twenty languages (Liberman and Cieri, 1998) while web search engines currently perform language identification on about a dozen of the languages they index, allowing language-specific searches in those languages. Documents in many other languages are also indexed, though no explicit labeling of the language they are written in is available. In particular, Tagalog is not one of the languages represented in any readily accessible online corpus. The LDC also has no corpora of web pages, though these are an interesting resource, often written by individuals to describe themselves. More than newswire, they reflect the written language of individuals a corporations, and reflect and broader cross-section of the community.

It would be useful to have an approach which, given a single document in a target language, can find many more examples of documents in that language, by automatically constructing queries to access such documents on the world wide web.

In applications in language technologies, corpora are commonly used to construct language models. The most widely used language models are n-gram language models. Unigram language models assume that all words in a document are independent of each other, and assign a probability to the occurrence of each word in a language.

While this is a simplification, it has been found very useful, for example for database selection (Gravano et al., 1993), and a unigram language model captures frequency information which is useful in a first understanding of the contents of a corpus.

Traditionally, these models are constructed by approximating the word probabilities by maximum likelihood estimation (MLE) from a corpus of a specific language consisting of millions of words and then smoothing the probabilities (Chen and Goodman, 1996). In this paper, we attempt to build a corpus and a unigram language model for a language that is the minority language in a multi-lingual corpus. Our task would be straightforward if (1) we had complete and free access to the multilingual corpus or database, (2) each document was only in one language and (3) each document in the database came with a label specifying its language. The task would then be reduced to the more common problem of building language models from a corpus specific for this purpose.

We treat the World Wide Web as the multilingual corpus and build a corpus and a language model for a language that accounts for less than 3% of the documents in the database. The specific language used in these experiments is Tagalog, the national language of the Phillipines. Since we cannot have complete access to the WWW, and access through a search engine is time-intensive, and not every page on the WWW comes labeled with the language the document was written in, we cannot apply traditional language modeling techniques to our database. Instead, we use the approach introduced by Callan et al. (1999) which uses query-based sampling to acquire monolingual language models from multiple databases. They are motivated by the fact that word occurrences follow a highly skewed distribution, with a few words occurring very often and most words occurring rarely. In the light of evidence suggesting that the important vocabulary words occur frequently in a database (Dumais, 1994; Luhn, 1958; Zipf, 1949) it is probable that these words might be acquired by sampling. They assume that if queries can be run and documents retrieved, then it is possible to sample the contents of each database in a way that will produce an accurate language model for the database.

We extend query-based sampling to handle the case of a multi-lingual database and a monolingual target corpus and language model, and compare several sampling techniques for creating a corpus and a language model of Tagalog by sampling web pages from the WWW. We run experiments with several sampling strategies and evaluate the corpora created by assessing the unigram language models constructed from them.

2 Methodology

We start with an example document from M (the set of documents in the target minority language), and one from O (the set of other documents). We then build a language model for M' (the current sample of M), which we will call $LM_{M'}$, as well as $LM_{O'}$ based on the sample O'. Based on these, we create a one or two-word search query, and retrieve a document to add to the corpus. We filter the document retrieved using the current language models, into either the minority or other language class, then iterate. To evaluate corpus construction, we build a unigram language model over the entire set of possible target minority documents M. We call this the true model, as it represents the knowledge we would have about M if we sampled all possible M documents in the collection. Its language model is written LM_M.

2.1 General Algorithm

1. Select one seed document from each of M and O.

2. Build language models for M' ($LM_{M'}$) and O' ($LM_{O'}$) from the initial documents.

3. Sample a document from the database with replacement. Sampling with replacement simulates duplicates which occur on the web with reasonable frequency, avoids the need for duplicate detection, and simplifies the underlying statistical sampling model.

4. Use the language filter to decide whether to add the new document to the list of documents in M, or those in O.

5. Update the language models for $LM_{M'}$ and $LM_{O'}$ and compare $LM_{M'}$ with the true model LM_M.

6. If the stopping criterion has not been reached, go back to Step 2.

The two important steps for our method are (3) and (4). We discuss the various sampling strategies used in the next section, and the filter we use to decide whether a document is in Tagalog is discussed later. We did not perform extensive experimentation with stopping criteria, but found that there were detectable plateaus in the amount of information acquired with each sample.

2.2 Sampling Methodologies

Our goal is to sample a representative variety of examples of documents in Tagalog with a minimum number of queries. Our approach can rely on the high-dimensionality of the problem, along with the fact that most dimensions (vocabulary) from the two models are not shared.

We will build a corpus of documents in language M by sampling documents from the entire database D. A random selection of documents from D will not suffice since most would not be from M. Applying a language filter would allow us to construct the corpus of those from M, but only very slowly. A more efficient approach is to ensure that most of the documents we examine are from M. Our model suggests that the intersection in vocabulary of M and O is very small. Thus, selecting documents with vocabulary in LM_M and not in LM_O is likely to give us documents in language M. This is the basis for our methods

Table 1: Query construction methodologies. $w_{maxP_{M'}} = argmax_i P(w_i|LM_{M'})$ is the most probable word according to the language model for the current sample M'; similarly for $w_{maxP_{O'}}$. These correspond to the words most frequently seen in the sampled corpora constructed so far. $w_{randP_{M'}}$ is a word chosen randomly, with probability proportional to its frequency in the current sample M'; similarly for $w_{randP_{O'}}$.

Query Method	Include word	Exclude word	Sample Query
`random`			
`most-frequent`	$w_{maxP_{M'}}$		"+sa"
`unigram`	$w_{randP_{M'}}$		"+mga"
`most-frequent-exclude`	$w_{maxP_{M'}}$	$w_{maxP_{O'}}$	"+sa −the"
`unigram-exclude-most-frequent`	$w_{randP_{M'}}$	$w_{maxP_{O'}}$	"+mga −the"
`unigram-exclude-unigram`	$w_{randP_{M'}}$	$w_{randP_{O'}}$	"+mga −antelope"

for sampling. Note that all query-based sampling methods we employ are followed by a language filter described in Section 2.3. Thus it is not imperative that a sampling method choose documents in M at every sampling iteration. However, the more frequently it does, the faster and more efficient corpus creation will be.

Table 1 gives an overview of the query-based sampling methodologies in our experiments. We use uniform random sampling as a baseline for our more "intelligent" sampling techniques. Since only 2.5% of the documents in our experimental corpus belong to M, we expect a document picked at random to probably belong to O. This also serves as motivation for our problem since if a crawler is deployed on the WWW to sample pages at random, it is very unlikely that it is going to find enough web pages in Tagalog to build a corpus for an accurate language model in a reasonable amount of time.

When picking the most-frequent word according to $LM_{M'}$ ($w_{maxP_{M'}}$) or $LM_{O'}$ ($w_{maxP_{O'}}$) we simply take the word seen most frequently in the sample of M' or O' constructed so far, that is, the most probable word according to the maximum likelihood estimated language model for the current sample. For M' this is given by $w_{maxP_{M'}} = argmax_i P(w_i|LM_{M'})$.

In order to pick a random word according to the language model, we generate a number $u \sim Uniform[0,1]$. We then pick the word according to u and the cumulative distribution function (CDF) of the unigram $LM_{M'}$ or $LM_{O'}$. This means that more frequently seen words in our sample so far are more likely to be selected, but that even rarely seen words have a small chance of being selected. Words unseen in the sample cannot be selected in this way.

When a query matches multiple documents, we pick one at random with uniform probability.

2.3 Document Filtering/Language Identification

Since some of our query-based sampling techniques will not find a document in M at every iteration, we construct a filter to detect which distribution (M' or O') is more likely to have generated the document. This determines whether to add it to the growing corpus M' and how use it in building the next language model, which may be used for building the next query. We do not use priors on the classes, but merely the vocabulary from the current estimated model. The filter is updated at every iteration of the experiment.

The algorithm is as follows:

- Count how many occurrences of words in the sampled document d are of words in the vocabulary in $LM_{M'}$.
- Count how many occurrences of words in d are of words in the vocabulary in $LM_{O'}$.
- Assign the document to M' or O' according to which score is higher.

Note that this filtering technique corresponds to a statistical model in which classes M' and O' are given uniform priors, and each word in their vocabularies is equally likely to be generated, i.e. also has a uniform probability. We are finding the maximum likelihood class to have generated the document under these simplifying assumptions.

State-of-the-art techniques in language identification (Dunning, 1994) and document filtering augment such techniques with character-based trigram statistics and class priors. As we will discuss in section 5, our simple language filter was adequate, except when the seed document contained an unusually small number of words.

3 Evaluation Metrics

We evaluate the language models constructed using query-based sampling strategies by comparing them to the true language model as defined by calculating a unigram language model over the entire set of Tagalog documents in our experimental database. The measures we use to evaluate the language models are discussed in this section.

3.1 Kullback Leibler Divergence

We measure the similarity between the true unigram language model LM_M and the unigram language model constructed from the sampled corpus $LM_{M'}$ by Kullback-Leibler (KL) divergence - a measure from information theory that represents the dissimilarity between two probability distributions, also called relative entropy. Let LM_M and $LM_{M'}$ be two distributions over V. $LM_{M'}$ is the sampled unigram language model distribution and LM_M is the actual distribution, while V is the set of terms in the vocabulary. Then the Kullback-Leibler divergence is expressed as:

$$KL(LM_M \| LM_{M'}) = \prod_{i=1}^{|V|} LM_M(w_i) log \frac{LM_M(w_i)}{LM_{M'}(w_i)} \quad (1)$$

where KL($LM_M \parallel LM_{M'}$) $>= 0$, and KL($LM_M \parallel LM_{M'}$) $= 0$ iff LM_M and $LM_{M'}$ are equal. The Kullback-Leibler divergence can be considered as a kind of a distance between the two probability densities, though it is not a distance metric because it is not symmetric.

KL divergence is infinite if any probability used is zero. This can occur frequently since there are often words in the true model which have not yet been seen in our sample. Thus to calculate the KL distance, we assigned any unseen word a uniform prior equal to 1/281379. There are 281,379 Tagalog word occurrences in our experimental collection, so this is smaller than the true probability for any word. Since we did not normalize these terms, the pdf used for calculating the KL distance is improper. This means that our KL divergences are not directly comparable with KL divergence scores calculated from a standard pdf, but direct comparisons with all calculations we performed in this way are meaningful. Note that we used this prior only for completely unseen terms, not as a general Bayesian prior from which we calculated a posterior after sampling. Words seen exactly once in a given sample had the maximum likelihood probability from that sample, generally much higher than unseen words since the number of seen words in any sample is less than the total number of words in the database.

3.2 Percentage of Vocabulary Learned

Percentage of vocabulary learned (Per) measures the proportion of the terms in the actual vocabulary that are found in the sampled vocabulary. As we sample more documents in M, the language model should cover more of the terms found in the true vocabulary. Percentage of vocabulary learned gives equal importance to all the terms in the vocabulary and thus is not a good match for text data because of the skewed distribution of terms in a corpus. According to Callan et al (1999), about 75% of the vocabulary of a text database is words that occur 3 times or less.

3.3 Cumulative Term Frequency (ctf) Ratio

Another measure for the quality of the learned vocabulary is the ctf ratio which gives a weight to each term that is proportional to its frequency in the corpus. Ctf measures the proportion of corpus term occurrences that are covered by terms in the learned language model. For a learned vocabulary V' and actual vocabulary V, and f_i and f_j both frequency counts over the whole minority corpus, the ctf ratio is

$$\frac{\sum_{i \in V'} f_i}{\sum_{j \in V} f_j} \tag{2}$$

4 Data

Our dataset consisted of 16,537 documents collected from the world-wide web. This was broken down into 498 documents in Tagalog and 16,039 documents in other languages. The other language documents were in a mixture of Brazilian Portuguese and English. We removed HTML mark-up and punctuation, converted capitalized letters to lowercase, and considered any string of remaining characters to be a word. The Tagalog sub-collection has a vocabulary of 35,482 words, with a total of 281,379 occurrences. To find out how much of our Tagalog vocabulary intersected with English, we used the file `/usr/dict/words` which contains English words for spelling programs and can be found on all UNIX systems. `/usr/dict/words` has a vocabulary of 45,402 words. Of these, 5393 vocabulary items appeared in our Tagalog documents, accounting for 41,277 word occurrences. The most frequent of these was "at", at 5659 occurrences; which is the Tagalog word for "and". The next most frequent was "the", which at 860 occurred on average twice per document. Tagalog documents contained some "internet English" and so words such as "page", "www" and some English terms occurred in the language model. In total, 28,000 Tagalog vocabulary items were unique to Tagalog in our dataset.

5 Results

A sampling method which samples a large proportion of minority-language documents can be said to be a successful sampling algorithm, as it succeeds in finding a large number of documents in the target language. Figure 1 shows the number of document sampled in Tagalog versus the total number of

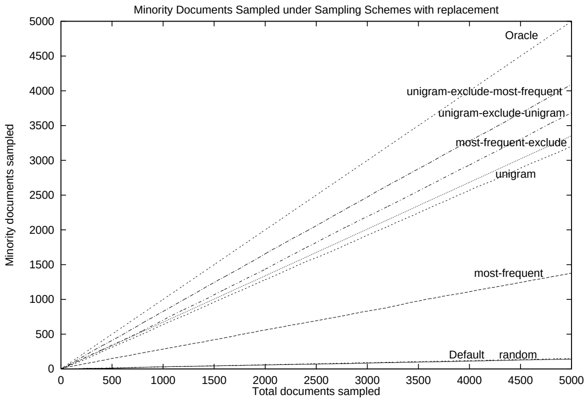

Figure 1: Random selection adds documents to the corpus and language model in the target minority language very slowly, relative to the number of samples taken at any point. Selecting a query term for sampling proportional to its frequency in the corpus constructed so far, and excluding the most-frequent term from the other languages O has a near-optimal rate of sampling documents in the target language.

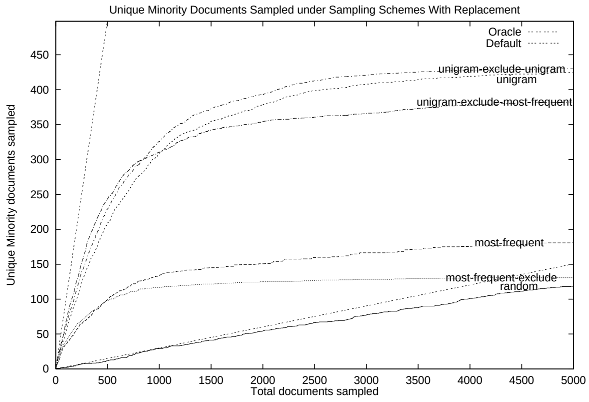

Figure 2: Looking only at the unique documents sampled, we see that excluding the most frequent term from the other languages seen leads the approach to perform worse than unigram exclusion sampling approaches. It is valuable to be able to sample from the entire minority document space. There were a total of 498 unique Tagalog documents in the collection.

documents sampled, averaged over three runs of each algorithm with different seed documents. The "Oracle" sampling rate, in which every sampled document is in the target minority language, is simply the line $y = x$. Since we are sampling with replacement, the number of Tagalog documents sampled can exceed the total number in our collection. Also shown is the default sampling rate, the expected number of minority documents to be sampled, given a particular sample size. This is the line $y = x * 498/16547$. Note that `random` hugs the default line, as is to be expected. Note also that the slopes of all these lines are relatively constant. `most-frequent` sampling samples around 33% of documents in Tagalog, while `unigram` samples significantly greater than the expected number of minority language documents, at around 60%. `most-frequent-exclude` samples at around 66% documents sampled in the target language, and `unigram-exclude-unigram` achieves 70%. The `unigram-exclude-most-frequent` method achieves the highest rate of in-target-group sampling, sampling at 80% accuracy.

Since we are sampling with replacement from a finite set of minority language documents, it is worth examining how the sampling proportion looks when measured by unique documents sampled, as shown in Figure 2.

Note that `most-frequent-exclude` is quick to find unique documents, however it flattens out at just over 130 documents, failing to find the full range of possible minority-language documents. As it finds a local maximum in the space of sample queries, it does not perform any extensive search to find new vocabulary items. By contrast, `most-frequent` is able to find a slightly larger variety of unique documents at just over 150, while `unigram-exclude-most-frequent` covers a variety of documents relatively quickly, and continue to explore the space of possible vocabulary more extensively, levelling off just above 375 unique documents. `unigram` and `unigram-exclude-unigram` acquire new unique documents slightly more slowly, but ultimately lead to a better coverage of the document space at an average of over 420 unique documents of the possible 498.

As we sample more and more documents, the language model for the sample approaches the true one. This is shown by the decreasing KL divergence (Figure 3) and increasing Percentage Learned (Figure 4) and ctf values (Figure 5). It is interesting to note that

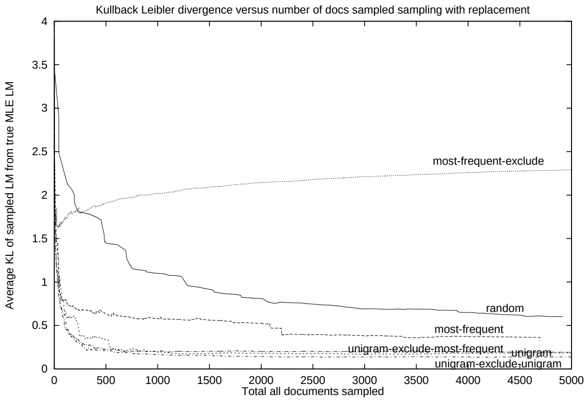

Figure 3: In all schemes, as the total number of documents sampled grows, the Kullback-Leibler divergence of the current language model from the true language model decreases, with the exception of `most-frequent-exclude`, which in some cases diverged. Sampling with `unigram-exclude-unigram` led most quickly and accurately to a corpus with word distributions similar to the complete minority language sub-corpus. However, `unigram` and `unigram-exclude-unigram` were also very accurate.

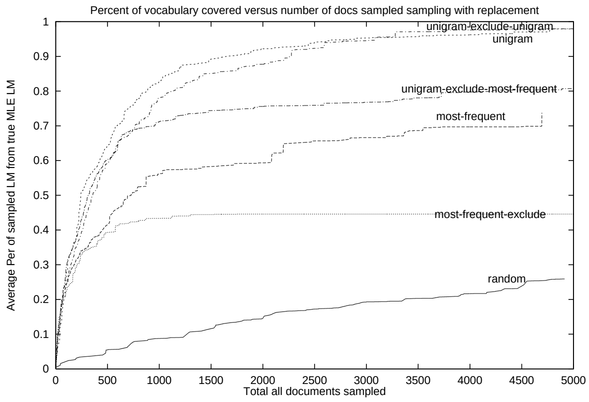

Figure 4: Excluding terms from the Other language model leads to algorithm convergence with a smaller percent of possible vocabulary terms seen. This reflects the mixed and overlapping vocabulary between the minority language and other languages in the corpus.

the `unigram-exclude-unigram` scheme outperforms other schemes throughout the experiments. As the number of documents sampled increases, two other schemes (`unigram` and `unigram-exclude-most-frequent`) approach it in accuracy.

An algorithm which performs very well as a function of all documents sampled may be very good at sampling from the minority document sub-collection M, but may do a poorer job of modeling the unigram due to concentrating on a sub-part of the space. Our preliminary results suggested that excluding the most-frequent term indeed displayed this behavior, and that unigram exclusion, and non-excluding methods out-performed them when reckoned as a function of in-target-group documents sampled, as they selected a greater variety of documents, enabling them to better estimate the unigram.

While our language filter was fast to implement and run, it broke down when the seed document in Tagalog was an unusual one containing very few words. In experiments not shown here, we started with a seed document containing only two distinct words. In the case of `random` sampling, `unigram` sampling and `unigram-exclude-unigram` sampling our

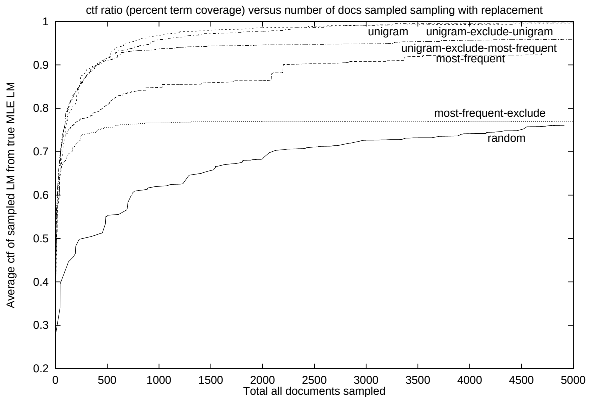

Figure 5: In term coverage ratio, `unigram-exclude-unigram` performs best, with `unigram` a very close second. `most-frequent-exclude` has reached a plateau covering under 75% of terms in the target minority language.

filter erroneously put the first Tagalog documents found into the O class, and then never recovered. Thus in some unusual cases it may be important select the initial document to be a more typical one, or to supervise the language filter on the first few iterations.

6 Conclusions and Future Work

We have shown that a simple query construction technique combined with a simple language filter can be used to automatically build a corpus of a minority language from the world-wide web. The methods which lead to fastest acquisition of documents (those using most frequent vocabulary items) may not explore the entire vocabulary space, and more randomized approaches may prove more effective if sufficient time is available. The choice of initial document may also prove important when a simple language filter is used.

We would like to extend this model to provide a theoretical account of the degree of vocabulary intersection which can exist between the minority language and the other documents in the collection, as well as how the results scale to larger corpora, and languages as varying proportions of the corpus. Another interesting question is whether these techniques give us a random selection of documents from language M. Another application would be to incorporate these sampling strategies in a spider that uses reinforcement learning to crawl the web and collect documents in the desired language efficiently (Rennie and McCallum, 1999). We also would like to investigate the effect of more sophisticated document filtering techniques.

7 Acknowledgements

This research has been supported in part by DARPA under research contract F30602-97-1-0215, and by the NSF under LIS Grant REC-9720374. We would like to thank John Lafferty, Steve Fienberg, Greg Aist and Kenji Sagae for helpful comments.

References

P. Brown, S. Della Pietra, V. Della Pietra, and R. Mercer. 1993. The mathematics of statistical machine translation. *Computational Linguistics*, 19(2).

J. Callan, M. Connell, and A. Du. 1999. Automatic discovery of language models for text databases. In *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, pages 479–490, Philadelphia. ACM.

Stanley F. Chen and Joshua T. Goodman. 1996. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 310–318.

S. Dumais. 1994. Latent semantic indexing (LSI) and TREC-2. In D. K. Harma, editor, *The Second Text REtrieval Conference (TREC-2)*, pages 105–115. Gaithesburg, MD.

Ted Dunning. 1994. Statistical identification of language. Technical Report MCCS 94-273, New Mexico State University.

Andrew R. Golding and Dan Roth. 1999. A winnow-based approach to context-sensitive spelling correction. *Machine Learning*, 34(1-3):107–130.

Luis Gravano, Hector Garcia-Molina, and Anthony Tomasic. 1993. The efficiency of GLOSS for the text database discovery problem. Technical Report CS-TN-93-2, Stanford University.

Frederick Jelinek. 1999. *Statistical Methods for Speech Recognition.* MIT Press.

Mark Liberman and Christopher Cieri. 1998. The creation, distribution and use of linguistic data. In *Proceedings of the First International Conference on Language Resources and Evaluation.*

H. P. Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2).

Jason Rennie and Andrew McCallum. 1999. Using reinforcement learning to spider the web efficiently. In *Proceedings of the 16th International Conference on Machine Learning.*

George K. Zipf. 1949. *Human behavior and the principle of least effort: An introduction to human ecology.* Addison-Wesley, Cambridge MA.