Passage Retrieval by Shrinkage of Language Models

Fei Song, Joe Vasak, and Wei Wang
Dept. of Computing and Information Science
University of Guelph
Guelph, Ontario, Canada N1G 2W1
{fsong, jvasak, wwang01}@uoguelph.ca

1. Introduction

Information retrieval is the process of searching for relevant documents that satisfy a user's information need (usually in the form of queries). Some of its successful applications include library catalogue search, medical record retrieval, and Internet search engines (e.g., Google). As the exponential growth of web pages and online documents continues, there is an increasing need for retrieval systems that are capable of dealing with a large collection of documents and at the same time narrowing the scope of the search results (not only relevant documents but also relevant passages or even direct answers).

A number of conceptual models have been proposed for information retrieval, including the Boolean model [Baeza-Yates and Ribeiro-Neto, 1999], the vector-space model [Salton, 1989], probabilistic models [Robertson and Sparck Jones, 1976], the inference network model [Croft and Turtle, 1992], and the language models [Ponte and Croft, 1998; Hiemstra, 1998; Miller et al., 1999]. Among these models, language models have recently received a lot of attention in the field of information retrieval, since they are based on the solid foundation of statistical natural language processing and are both intuitive and flexible for extensions with more features to handle the retrieval tasks.

In language modeling, we view each document as a language sample and estimate the probabilities of producing individual terms in a document. A query is treated as a generation process. Given a sequence of terms in a query, we compute the probabilities of generating these terms according to each document model. The multiplication of these probabilities is then used to rank the retrieved documents: the higher the generation probabilities, the more relevant the corresponding documents to the given query.

One big obstacle in applying language modeling to information retrieval is the sparse data problem. Unlike a collection of documents where we can control the number of documents in it, a document itself is often small in size and its content is always fixed. Even for a relatively long document, some of the words can still be rare or missing according to the Zipf's law of language usage [Manning and Schütze, 1999]. As a result, the combination of individual probabilities through multiplications will be meaningless if one of the probabilities is zero (for a missing term in a document). Thus, overcoming the sparse data problem is the key for the success of any language modeling system for information retrieval.

For TREC 2006 Genomics Track (see http://ir.ohsu.edu/genomics/ for more information), the data set presents several new challenges for language modeling in specific and information retrieval in general. First of all, the search is targeted to the relevant passages within documents (more or less corresponding to paragraphs), since users of the biomedical domain are likely interested in finding answers along with the context that provides supporting information and links to the original sources. Secondly, there is a need to balance the results across different documents and aspects. An aspect is defined as a group of passages of similar content, which will be judged by human evaluators and identified by a set of MeSH terms for the Genomics data set. By ensuring an adequate coverage of the results across documents and aspects, we can reduce the repeats (or duplicate passages) and maintain a reasonable number of novel/unique passages, which may be particularly useful for

biomedical researchers. Finally, the retrieved passages may need to be trimmed further to highlight the answers, since passages are typically organized as paragraphs and may contain irrelevant wording before and after the relevant answers.

In the rest of the working note, we describe our retrieval method based on the language models and their combinations in section 2. In section 3, we explain the enhancements for balancing the results for documents and aspects and narrowing the spans for the answers in the retrieved passages. In section 4, we discuss our experimental results on the Genomics data set. Finally, we conclude and point future directions of our work in section 5.

2. Language Models and Their Combinations

In Song and Croft [1999], we proposed a general language model for information retrieval. The model is based on a range of data smoothing techniques, including Good-Turing estimates, curve-fitting functions, and model combinations.

For TREC 2006 Genomics Track, we apply our model for information retrieval, but instead of searching for relevant documents, we move deeper to look for relevant passages. Since passages are much smaller than documents, the sparse data problem is even more serious with a large number of terms missing for individual passages. As a result, more attention needs to be directed to smooth the passage models so that the missing terms are allocated with meaningful probability mass and the generation probabilities for each query are non-zero and thus can be used to rank the retrieved passages.

2.1. Smoothing a Passage Model with the Good-Turing Estimate

We can compute the frequencies for each term in a passage. To smooth the probabilities for all the terms, including the missing terms in a vocabulary, the Good-Turing estimate adjusts the raw term frequency (tf) values as follows:

$$tf^* = (tf + 1)\frac{E(N_{tf})}{E(N_{tf+1})}$$

Here, N_{tf} is the number of terms with frequency tf in a passage, and $E(N_{tf})$ is the expected value of N_{tf}. The probability of a term with frequency tf is then defined as tf^*/N_p, where Np is the total number of terms in passage p. Note that when tf = 0, tf* is reduced to $E(N_1)/E(N_0)$ and the probability for a missing term becomes $E(N_1)/E(N_0)N_p$.

However, obtaining $E(N_{tf})$ is almost impossible since a passage is fixed in size and content. In practice, we hope to substitute the observed N_{tf} for $E(N_{tf})$ directly, but this creates two problems. For the terms with the highest frequency tf, their adjusted tf* will be zero, since N_{tf+1} is always zero, which is counter-intuitive. Furthermore, due to the small size of a passage, the number of terms at some middle frequency levels may also be too small or even zero, resulting in an unstable or anomaly distribution.

Table 1. A Typical Term Distribution for a Document Collection

tf	N_{tf}	tf	N_{tf}
0	74,671,100,100	5	68,379
1	2,018,046	6	48,190
2	449,721	7	35,709
3	188,933	8	27,710
4	105,668	9	22,280

One way to get around the above problems is to use a curve-fitting function to smooth the observed N_{tf}'s for the expected values. Table 1 shows a typical term distribution for a document collection[1], taken from Manning and Schütze [1999]. As can be seen, N_{tf} can be approximated by a decreasing curve as tf gets bigger. Such a decreasing curve ensures that $N_{tf} = N_{tf+1}$ and allows us to project a non-zero value for N_{tf+1} with the highest tf.

With a smoothing function $S(N_{tf})$ for N_{tf}, the probability for term t with frequency tf in passage p can be computed as follows:

$$P_{GT}(t \mid p) = \frac{(tf+1)S(N_{tf+1})}{S(N_{tf})N_p}$$

2.2. Curve-Fitting for Good-Turing Estimates

Given a set of data points (x_i, y_i) for i = 1, 2, …, n, linear regression helps us identify a line f(x) = mx + b that fits the data points as tightly as possible. This is done by minimizing the sum of squares of differences:

$$SS(m,b) = \sum_{i=1}^{n}[y_i - f(x_i)]^2 = \sum_{i=1}^{n}(y_i - mx_i - b)^2$$

Using calculus (see Manning and Schütze [1999] for details), we can find the optimal solutions for m and b:

$$m = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \quad \text{and} \quad b = \bar{y} - m\bar{x}$$

where $\bar{x}$ and $\bar{y}$ are the averages of x_i and y_i for i = 1, 2, …, n, respectively.

Clearly, a linear line does not fit the distribution in Table 1 properly, since the values of N_{tf} go down very quickly as tf goes up. This leads us to use a geometric distribution to model a decreasing exponential curve:

$$f(x) = p^x q \quad \Rightarrow \quad \log f(x) = x \log p + \log q$$

By taking the logarithm on both sides, we turn a geometric distribution into a log-linear combination, which can then be solved in a way similar to linear regression.

Unfortunately, a simple geometric curve does not fit the typical term distribution either: although N_{tf} decreases very quickly for smaller tf's, the pace slows down dramatically as tf gets much bigger. To fit the typical term distribution as closely as possible, we replace the variable x by a nested logarithmic function:

Zeroth order: $\log^0 x = x$

First order: $\log^1 x = \log(x+1)$

Second order: $\log^2 x = \log(\log(x+1)+1)$

Third Order: $\log^3 x = \log(\log(\log(x+1)+1)+1)$

…

[1] The table is actually for bigram (word pair) distributions, but a similar pattern is also applied to individual terms due to the Zipf's law for language usage [Manning and Schütze, 1999].

Based on the nested logarithmic function, we develop a greedy algorithm that tries to find an "optimal" geometric distribution:

$$\log f(x) = \log p \cdot \log^m x + \log q$$

Here, the level of nesting m is selected by testing the above formula until no further improvement can be made in terms of sum of squares of the differences for all given data points.

2.3. Combining Language Models by Shrinkage

Good-Turing estimate provides us the first smoothing step towards building a suitable language model for passage retrieval. However, a passage model is not stable and accurate in the sense that there is often a large number of missing terms and there can also be anomaly distributions for certain known terms. In particular, we cannot differentiate the contributions of different missing terms at a passage level. In a biomedical document, for example, a passage may not contain terms "genetic" and "crocodile", but in terms of probability distribution, we would prefer that the probability for "genetic" be higher than that for "crocodile", since the document is about biomedicine. Obviously, we need to borrow information from outside the passage in order to make a proper differentiation for different missing terms.

For TREC 2006 Genomics data set, there are multiple levels of structures: the collection consists of 59 journals[2]; each journal, multiple documents; and each document, multiple passages. Using the Good-Turing estimate, we can also build language models for documents, journals, and even the entire collection. Clearly, the collection model contains the most information. The journal models are more stable and accurate than the document models, and the document models are more stable and accurate than the passage models. For this reason, we want to extend a passage model by adding information from the corresponding document, journal, and collection models. This can be done by the interpolation or shrinkage method:

$$P_{combined}(t \mid p) = l_1 P_{passage}(t \mid p) + l_2 P_{document}(t \mid d) + l_3 P_{journal}(t \mid j) + l_4 P_{collection}(t)$$

where λ_1, λ_2, λ_3, and λ_4 are weighting parameters and $\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1$. Such a linear combination has the advantage that the resulting probabilities are normalized in the range of [0, 1].

The shrinkage combination has two useful effects. One is that we borrow information from outside the passages so that we can further differentiate the contributions of different terms. The other is that we align passage models against the document, journal, and collection models so that the probability distribution for the known terms can be more stable. Intuitively, the document, journal, and collection models provide the average distributions and the passage models add the variations to them. Together, we get more stable and accurate distributions for passage retrieval.

3. Enhancing the Results for Coverage and Specificity

For TREC 2006 Genomics Track, the retrieved results are evaluated at three different levels: passage retrieval, aspect retrieval, and document retrieval so that we can get insight into the overall performance for a user trying to answer a given topic. Since each submitted run can contain only up to 1000 passages per topic, the retrieved passages from the initial search should be further processed to ensure a reasonable coverage for relevant documents and aspects. Such efforts help reduce repeats or duplicate passages and at the same time increase the number of novel/unique passages in the search results.

[2] Actually, there are 49 journals, but one of them is further split into 11 subsets.

3.1 Improving Coverage for Relevant Documents

TREC 2006 Genomics data set provides legal spans for all passages (more or less corresponding to paragraphs). Each passage is identified by three components: PMID (PubMed ID) --- uniquely assigned to each document; passage start --- the bye-offset in the document file where the passage begins; and passage length --- the length of the passage in bytes (8-bit ASCII code). As a result, the corresponding document for a passage can be easily found by analyzing the passage identifier. This leads us to a simple solution to ensure a wider coverage for relevant documents. Given the top ranked passages for a particular topic, we first map them to the corresponding documents. If a document has got a reasonable number of retrieved passages, the remaining passages for this document are thrown away. This frees up more rooms in the top 1000 passages for other documents so that the coverage for relevant documents can be expanded.

3.2 Improving Coverage for Relevant Aspects

Unlike documents that have clear starts and ends, aspects are hidden units corresponding to groups of passages with similar contents, which will be judged by human evaluators and identified by a set of MeSH terms for the Genomics Track evaluation. As stated in the TREC 2006 Genomics Track Protocol, one aspect typically corresponds to multiple passages, but one passage can also be linked to multiple aspects and some passages may overlap and/or belong to multiple aspects.

Because aspects are subjectively determined and currently not available for training purposes, we may have to apply some kind of clustering techniques to discover the natural grouping among the available passages. We limit ourselves to the partition of non-overlapping clusters. In other words, no clusters may contain other clusters and there are no overlaps between clusters. If we model aspects by clusters of passages, this assumption implies that each passage can only belong to one aspect and no two aspects share any passages in common. Clearly, this assumption is too strong and needs to be relaxed further for future experiments.

We use the heuristic clustering method introduced in Salton [1989] for computing aspects, since it allows us to create clusters rapidly with relatively little expense. Each passage is represented as a weighted vector (TF x IDF) and the distance between two vectors is measured by the cosine similarity. As described in [Salton, 1989], heuristic clustering is a one-pass process, which takes the elements to be clustered one at a time in an arbitrary order. The first element is placed into a cluster of its own. Each subsequent element is then compared against all existing clusters and is placed into the cluster that is the most similar to the new element. If the new element is not sufficiently similar to any of the existing clusters, it forms a new cluster of its own. This process is continued until all elements are processed. Each cluster is represented by the centroid vector, which is updated every time a new element is added into the cluster.

Once we obtain the aspects (or clusters) for a set of passages, we can prune the results in a way similar to what we did to documents in section 3.1. In other words, we map the top-ranked passages for a topic to their corresponding aspects. If an aspect gets a reasonable number of retrieved passages, the remaining passages for this aspect are discarded. As a result, we can cover more aspects in the top 1000 retrieved passages in the final results.

3.3 Towards More Specific Answers for Retrieved Passages

The goal of TREC 2006 Genomics Track is to find information that is close to “answers” for a question or information need. Performing search at the passage level is helpful for achieving this goal, since documents are typically too long to be used as “answers”. However, since passages more or less correspond to paragraphs, which are marked for the documents rather than for the answers to users, they may contain irrelevant wording before and after the relevant answers. Thus, to highlight

the answers in the retrieved passages, we may need to trim the irrelevant wording around the answers in the retrieved passages.

We trim irrelevant wording through a two-step process. First, we use a sentence splitter to break a passage into a sequence of sentences. Then, we narrow down the scope of the result by identifying the first and last sentences that match some of the terms in a given topic. Since sentences are natural units for semantic meanings, by keeping the complete sentences in the narrowed result, we can preserve meaning and ensure the readability of the final result for the retrieved passage.

We follow Reynar and Ratnaparkhi [1997] and implement a sentence splitter based on a maximum entropy approach. We use features of the words around an end-of-sentence punctuation mark (usually period, question mark, or exclamation mark) and train the maximum entropy model with labeled documents for end-of-sentence marks. A maximum entropy model allows us to combine features of different kinds, which in our case contain such attributes as person titles, initial capitalization, abbreviations for months and days, etc.

Table 1. University of Guelph Results for TREC 2006 Genomics Track

	UofG0 Run			**UofG1 Run**			**UofG2 Run**		
Topic	**Doc**	**Passage**	**Aspect**	**Doc**	**Passage**	**Aspect**	**Doc**	**Passage**	**Aspect**
160	.201938	.008583	.042549	.209704	.006814	.054574	.198381	.006337	.053360
161	.332930	.062424	.130727	.339145	.044740	.068344	.339295	.044807	.068798
162	.350694	.080812	.271507	.350694	.055556	.200000	.350694	.055556	.200000
163	.610200	.027614	.077983	.631620	.024001	.070498	.605470	.019521	.067078
164	.000934	.000159	.000302	.003026	.000846	.001606	.003058	.000860	.001657
165	.164789	.119675	.475613	.169814	.036491	.373604	.168211	.036674	.374143
166	.221656	.021541	.345639	.221656	.019178	.317630	.221667	.019178	.317641
167	.605977	.162167	.161907	.629965	.039632	.154769	.608553	.037074	.161408
168	.883041	.105436	.383726	.883041	.037130	.298134	.780575	.027088	.210447
169	.401195	.079371	.061101	.410076	.038079	.058015	.417243	.038715	.063333
170	.477868	.011498	.050052	.477868	.000109	.009783	.478161	.000109	.009783
171	.014872	.000000	.000000	.014872	.000000	.000000	.014873	.000000	.000000
172	.169132	.002330	.046692	.185275	.001797	.047749	.179225	.001680	.047434
174	.460182	.081262	.794114	.465273	.068208	.798611	.457086	.067203	.799517
175	.430558	.101289	.221640	.436614	.048859	.167209	.436614	.048861	.167209
176	.318942	.008750	.050105	.318942	.004421	.043447	.319411	.004584	.043447
177	.000000	.000000	.000000	.000000	.000000	.000000	.000000	.000000	.000000
178	.344469	.031602	.068110	.344469	.000000	.000000	.344485	.000000	.000000
179	.032316	.002630	.019841	.033363	.007515	.059524	.034495	.007515	.059524
181	.455263	.077538	.150953	.578879	.067265	.153564	.533192	.059962	.165131
182	.433443	.018683	.027919	.457443	.007514	.018645	.358112	.006618	.020015
183	.307910	.037526	.479292	.309548	.036293	.477777	.310614	.036306	.477993
184	.000992	.000000	.000000	.000992	.000000	.000000	.001004	.000000	.000000
186	.834194	.166325	.713503	.836778	.157531	.610184	.814809	.156807	.610230
186	.391575	.052223	.199145	.496574	.031027	.196430	.494297	.030157	.196986
187	.698413	.030358	.052247	.698413	.000000	.000000	.698413	.000000	.000000
Avg	**.351672**	**.049608**	**.185564**	**.365540**	**.028192**	**.160773**	**.352613**	**.027139**	**.158274**

4. Experimental Results

We submitted three runs of results for TREC 2006 Genomics Track, each of which consists of 1,000 retrieved passages for each topic for a total of 26 topics. The first run "UofG0" is based on the language models with the shrinkage combination (described in section 2); the second run "UofG1" adds the effort for improving coverage of relevant documents (section 3.1); and the third run "UofG2"

tries to improve coverage for aspects (section 3.2). For all three runs, the results are enhanced by narrowing the scopes of the retrieved passages for more specific answers (section 3.3).

For "UofG0" run, we use the greedy curve-fitting algorithm to optimize the Good-Turing estimates for each language model (mostly at the passage, document, and journal levels), and combine all the related models together through the shrinkage method. Since no training data are available from the previous years for TREC 2006 Genomics data set, we set the weighting parameters as follows by intuition: $\lambda_1 = 0.7$, $\lambda_2 = 0.21$, $\lambda_3 = 0.063$, and $\lambda_4 = 0.027$. As can be seen in Table 1, the average MAP (Mean Average Precision) values of "UofG0" run are 0.351672 at the document level, 0.049608 at the passage level, and 0.185564 at the aspect level. These numbers are higher than the corresponding median scores over the 68 automatic runs from Table 2, which are 0.27905 for documents, 0.024008 for passages, and 0.116862 for aspects. Note that a couple of refined conditions are added to further improve the retrieval performance. First, each retrieved passage should have at least one term in common with the given topic in order to avoid over-smoothing with the shrinkage method. Secondly, the first two passages that appear at the beginning or at the end of a document are removed from the search results, since they tend to be titles, authors, and references for the documents in TREC 2006 Genomics data set. In future, we could further improve the retrieval performance when training data become available to fine-tune the weighting parameters.

Table 2. Statistics Over 68 Automatic Runs

	Document AP			**Passage AP**			**Aspect AP**		
Topic	**Best**	**Median**	**Worst**	**Best**	**Median**	**Worst**	**Best**	**Median**	**Worst**
160	.925200	.470600	.000000	.212300	.032200	.000000	.366700	.154900	.000000
161	.933900	.332900	.000000	.185200	.044800	.000000	.886900	.285800	.000000
162	.360700	.160000	.000000	.241700	.003300	.000000	.664300	.020900	.000000
163	.699900	.551300	.040800	.249000	.039700	.000300	.463700	.248500	.002900
164	.619300	.003600	.000000	.399100	.000600	.000000	.744300	.001700	.000000
165	.756500	.212900	.000000	.279300	.036500	.000000	.731500	.409700	.000000
166	.271800	.126100	.000000	.164300	.006200	.000000	.565500	.091500	.000000
167	.750500	.498400	.024400	.218200	.072100	.000000	.348700	.164700	.000000
168	.921600	.751300	.000000	.179300	.089600	.000000	.659800	.212900	.000000
169	.732700	.248500	.000000	.459900	.021300	.000000	.773800	.100400	.000000
170	.916700	.081600	.000000	.334900	.000400	.000000	.989100	.023200	.000000
171	.725800	.002500	.000000	.244700	.000000	.000000	.508500	.000000	.000000
172	.495300	.234700	.019000	.013800	.003400	.000000	.207300	.063200	.001100
174	.674500	.329900	.000000	.327000	.009900	.000000	.941700	.194400	.000000
175	.704200	.379300	.000000	.317300	.030700	.000000	.607400	.186900	.000000
176	.492700	.044300	.000000	.108900	.000400	.000000	.629600	.007100	.000000
177	.750000	.000000	.000000	.143800	.000000	.000000	.764600	.000000	.000000
178	.366700	.011600	.000000	.080500	.000100	.000000	.068100	.001000	.000000
179	.307600	.051800	.000000	.050800	.004200	.000000	.717700	.046600	.000000
181	.830300	.578300	.000000	.333600	.117900	.000000	.431200	.151000	.000000
182	.457400	.235700	.000000	.062500	.009700	.000000	.337500	.059800	.000000
183	.521000	.255800	.000000	.060800	.006600	.000000	.740500	.074700	.000000
184	1.00000	.001000	.000000	.067000	.000000	.000000	.572800	.000000	.000000
185	.836800	.487400	.000000	.252500	.034500	.000000	.713500	.209300	.000000
186	.849800	.614900	.000000	.319500	.056100	.000000	.654300	.302900	.000000
187	1.00000	.590900	.000000	.358400	.004000	.000000	.425600	.027300	.000000
Avg	**.688496**	**.279050**	**.003238**	**.217858**	**.024008**	**.000012**	**.596715**	**.116862**	**.000154**

For "UofG1" run, we set the maximum number of passages per document to be 5 and any more passages from the same document are discarded in the final results. As shown in Table 1, this indeed improves the retrieval performance slightly at the document level with the average MAP of 0.36554, but at the cost of decreasing the performance at the passage and aspect levels: 0.028192 and 0.160773, respectively. Due to the restriction of submitting only three runs, we are unable to test the system

with different threshold values. Clearly, there are conflicting factors in improving the performance across the passage, aspect, and document levels, and some kind of compromise has to be made in order to get a proper balance between the three different levels.

For "UofG2" run, we set the minimum similarity level to be 0.5 in order to merge a passage into one of the existing aspects (clusters). As illustrated in Table 1, the performance does not improve; in fact, all the performance numbers are slightly lower than those for "UofG1" run. This is perhaps not surprising, since the heuristic clustering method is not one of the best clustering methods available and the number of terms within a passage is usually too small for computing the distance between passages. The reason we choose the heuristic clustering method is because it allows us to create clusters rapidly with relatively low cost, which is desirable when we are under the constraints of time and machine resources. Clearly, more work needs to be done in order to model and compute aspects efficiently and accurately.

5. Conclusions and Future Work

We show that language models combined by shrinkage are a promising method for passage retrieval. In particular, extending a passage model with information from the corresponding document, journal, and even collection is desirable since a passage is usually too short to capture enough information for matching and comparison purposes. Our approach for language modeling is intuitive and easy to understand, since models are obtained and optimized individually before they are combined with the shrinkage method.

Although we tried to expand the coverage for documents and aspects, the results did not show much improvement. More study is needed to identify a proper balance for the retrieval performance at the three levels of passages, aspects, and documents. Furthermore, we need to explore different clustering methods to compute the aspects efficiently and accurately. In particular, we need to relax the condition for non-overlapping clusters, since an aspect can correspond to multiple passages, and a passage can also belong to multiple aspects. It will be a big challenge for computing such clustering structures efficiently, since the TREC 2006 Genomics data set contains a total of 12, 641,039 passages, which is quite large to store and compute on a typical personal computer.

Finally, we could continue improve the performance of our system for passage retrieval with training data. For this year, the data and query sets are both new, so no training data are available. For the next year, we could use the evaluated results from this year as training data and investigate whether the weighting parameters for the model combination can be optimized and set automatically so that we can further improve the retrieval performance.

Acknowledgements

The authors would like to thank NSERC (National Sciences and Engineering Council of Canada) and OCE (Ontario Centres of Excellence) for their funding support, which helped us tremendously in completing this research work.

References

Ricardo Baeza-Yates and Berthier Ribeiro-Neto. Modern Information Retrieval. Addison-Wesley, 1999.

Stanley F. Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. In Proceedings of the 34th ACL, pages 310-318, 1996.

W. Bruce Croft and Howard R. Turtle. Text retrieval and inference. In Text-Based Intelligent Systems, edited by Paul S. Jacob, pages 127-155, Lawrence Erlbaum Associates Publisher, 1992.

D. Hiemstra. A linguistically motivated probabilistic model of information retrieval. Second European Conference on Digital Libraries, pages 569-584, 1998.

Christopher D. Manning and Hinrich Schütze. Foundations of Statistical Natural Language Processing. The MIT Press, 1999.

Andrew McCallum, Ronald Resonfeld, Tom Mitchell, and Andrew Y. Ng. Improving text classification by shrinkage in a hierarchy of classes. In Proceedings of the 15th International Conference on Machine Learning, pages 359-367, 1998.

D.R.H. Miller, T. Leek, and R.M. Schwartz. A hidden markov model information retrieval system. In Proceedings of SIGIR'99, pages 214-221, 1999.

Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In Proceedings of SIGIR'98, pages 275-281, 1998.

Jeffrey C. Reynar and Adwait Ratnaparkhi. A maximum entropy approach to identifying sentence boundaries. In Proceedings of the ANLP'97, pages 16-19, 1997.

S.E. Robertson and K. Sparck Jones. Relevance weighting of search terms. Journal of the American Society for Information Sciences, 27(3):129-146, 1976.

Gerard Salton. Automatic Text Processing: the Transformation, Analysis, and Retrieval of Information by Computer. Addison-Wesley, 1989.

Fei Song and W. Bruce Croft. A general language model for information retrieval. In Proceedings of the Eighth ACM International Conference on Information and Knowledge Management (CIKM'99), pages 316-321, 1999.

TREC 2006 Genomics Track. http://ir.ohsu.edu/genomics/.

Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. Proceedings of the ACM-SIGIR'01, pages 334-342, 2001.