

Robust Pseudo Feedback Estimation and HMM Passage Extraction: UIUC at TREC 2006 Genomics Track

Jing Jiang, Xin He, ChengXiang Zhai
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801

1 Introduction

The University of Illinois at Urbana-Champaign (UIUC) participated in TREC 2006 Genomics Track. Our focus this year was to apply two language modeling techniques for information retrieval that have been proposed recently by our group [4, 1]. These two techniques have been shown to be effective for general English text. It is not clear, though, how they perform on text in special domains such as the biomedical domain. We therefore tested their effectiveness for this year's genomics task.

First, we tried to improve the pseudo relevance feedback mechanism in the retrieval model by applying a recently proposed regularized estimation method [4]. In the KL-divergence retrieval framework, pseudo relevance feedback documents can be used to better estimate the query model [5]. While in the original proposed method [5], this estimation involved two parameters that need to be empirically set, recent work showed that a more robust, regularized estimation method that involves less parameter tuning can be effective [4]. We therefore applied this estimation method to this year's genomics task to see whether it can improve the pseudo relevance feedback mechanism in biomedical information retrieval as well. Second, since this year's task is defined as passage retrieval rather than document retrieval, a challenge is how to extract coherent and relevant passages from whole documents. Previously, we proposed a hidden Markov model (HMM)-based passage extraction method that was shown to be effective in the general English domain [1]. We applied this method to this year's genomics task to see whether this method is also effective for biomedical text.

Besides the two language modeling techniques, we also tested the use of user relevance feedback for retrieval, to see how much human interaction can help improve the performance. We obtained some manual judgments from two domain experts, and used them in the two interactive runs.

Our experiment results showed that the regularized estimation method for pseudo relevance feedback performed

similarly to the original estimation method when both methods were under the optimal parameter setting, and outperformed the original estimation method when both methods were under the default parameter setting. Because in reality we do not know the optimal parameter setting, the regularized estimation method is thus more robust than the original estimation method. Our experiment results also showed that the HMM-based passage extraction method outperformed a baseline method that returns whole paragraphs as passages. However, our HMM-based passage extraction method tends to return relatively long and coherent passages, which may not be optimal for the genomics task this year, because in this task the information need is more specific. Finally, our experiment results showed that user relevance feedback was very effective, as we expected.

2 Robust Feedback Estimation

This year's task was defined as a passage retrieval task, where legal passages must be within single paragraphs from the full-text articles. Since the relevant passages are query-dependent, they can be of variable length and their boundaries cannot be determined without considering the specific query. We thus divided the task into two steps. For each query, we first retrieved the top 1000 paragraphs using the KL-divergence retrieval method. Then from each of the 1000 paragraphs, we extracted a relevant passage using our HMM-based passage extraction method. We assumed that there was only a single relevant passage within a relevant paragraph for a given query. Although this assumption may not hold in all cases, it is a relatively reasonable assumption, and can greatly simplify the task.

In the rest of this section, we will discuss the paragraph retrieval subtask, and focus on the pseudo relevance feedback mechanism. Then in Section 3, we will discuss the passage extraction subtask.

2.1 The KL-Divergence Retrieval Model

The first subtask is paragraph retrieval, for which we could use any existing document retrieval method. In our experiments, we used the KL-divergence retrieval model, which was first proposed in [2]. In this model, queries and documents are all represented by unigram language models, which are essentially word probability distributions. Assuming that these language models can be appropriately estimated, KL-divergence retrieval model scores a document D with respect to a query Q by computing the Kullback-Leibler divergence between the query language model θ_Q and the document language model θ_D as follows:

$$D(\theta_Q||\theta_D) = \sum_{w \in V} p(w|\theta_Q) \log \frac{p(w|\theta_Q)}{p(w|\theta_D)},$$

where V is the set of all words in the vocabulary. In practice, for the sake of efficiency, we often truncate the query language model θ_Q and use only a certain number of words with the highest probabilities to score documents. In our experiments, we set this number to 50.

What remains to be solved is how to appropriately estimate the query language model θ_Q and the document language model θ_D . θ_D is estimated from the document D , and is usually smoothed with a background language model θ_B , which is estimated from the whole document collection as follows:

$$p(w|\theta_B) = \frac{\sum_{D \in \mathcal{C}} c(w, D)}{\sum_{D \in \mathcal{C}} |D|},$$

where \mathcal{C} is the document collection, $c(w, D)$ is the number of occurrences of word w in document D , and $|D|$ is the length of document D . One of the most effective smoothing methods is the Dirichlet prior smoothing method, as shown below:

$$p(w|\theta_D) = \frac{c(w, D) + \mu p(w|\theta_B)}{|D| + \mu},$$

where μ is a parameter that controls the degree of smoothing, and is usually set empirically. In our experiments, we used the Dirichlet prior smoothing method, and set μ to 1000 based on our preliminary experiments with the queries and documents from TREC 2005 Genomics Track.

Estimation of the query language model is more complicated. The simplest way to estimate the query language model is to use only the query. θ_Q can thus be estimated as follows:

$$p(w|\theta_Q) = p(w|Q) = \frac{c(w, Q)}{|Q|},$$

where $c(w, Q)$ is the number of occurrences of word w in query Q , and $|Q|$ is the length of query Q . However, since

queries are usually very short, they can hardly capture the user's information need completely. Several methods have been proposed to improve the estimation of the query language model [2, 3, 5].

In [5], the query language model is estimated as follows. First, we assume there is a set \mathcal{F} of feedback documents for the query. These documents can be obtained either from the user explicitly or from the top-ranked documents after the first round of retrieval, in which case they are considered pseudo relevance feedback. We then assume that these feedback documents are generated from a two-component mixture model, where one component is the background language model θ_B , and the other component is a topic language model θ_T that is related to the query Q . The log likelihood of generating the feedback documents is thus

$$\begin{aligned} & \log p(\mathcal{F}|\theta_T) \\ &= \sum_{D \in \mathcal{F}} \sum_{w \in V} c(w, D)((1 - \alpha)p(w|\theta_B) + \alpha p(w|\theta_T)), \end{aligned}$$

where α is an interpolation parameter that controls the degree to which a feedback document is generated from the topic language model, and is usually empirically set. As proposed in [5], the topic language model θ_T can be estimated by fixing θ_B and α , and maximizing the log likelihood $\log p(\mathcal{F}|\theta_T)$ using the E-M algorithm. Once θ_T is estimated, the query language model θ_Q is computed as a linear interpolation of the original query language model and the topic language model, as shown below:

$$p(w|\theta_Q) = (1 - \lambda)p(w|Q) + \lambda p(w|\theta_T),$$

where λ is another parameter to control the degree to which we trust the estimated topic language model, and is again usually set empirically.

2.2 Regularized Estimation of the Mixture Models for Feedback Documents

The above method to estimate the query language model from the feedback documents involves two parameters, α and λ , that need to be empirically set. It has been shown that although the method can perform well, it is sensitive to the setting of both α and λ [5]. To address this problem, in [4], the authors proposed a regularized estimation method.

First, the authors pointed out that the parameter α should be document-specific because different feedback documents may contain different amount of relevant information. The log likelihood of the feedback documents thus becomes

$$\begin{aligned} & \log p(\mathcal{F}|\Lambda) \\ &= \sum_{D \in \mathcal{F}} \sum_{w \in V} c(w, D)((1 - \alpha_D)p(w|\theta_B) + \alpha_D p(w|\theta_T)), \end{aligned}$$

where $\Lambda = \{\theta_T, \{\alpha_D\}_{D \in \mathcal{F}}\}$ is the set of parameters. θ_B can still be fixed as before, but both θ_T and α_D 's are now estimated from the data.

Second, the authors proposed to use a prior for the topic language model. Previously, there was no constraint imposed on the topic language model. However, the topic language model is expected to be close to the original query language model. The authors therefore proposed to construct a conjugate Dirichlet prior for the topic language model from the original query language model. Formally, the prior states that

$$p(\theta_T) \propto \prod_{w \in V} p(w|\theta_T)^{\mu p(w|Q)},$$

where μ is a parameter that indicates the confidence on the prior. The authors did not impose any prior on the parameter α_D 's. As a result, the prior for all the parameters is

$$p(\Lambda) \propto p(\theta_T) \propto \prod_{w \in V} p(w|\theta_T)^{\mu p(w|Q)}.$$

Using maximum a posterior (MAP) estimation, the parameters $\Lambda = \{\theta_T, \{\alpha_D\}_{D \in \mathcal{F}}\}$ can be estimated as follows:

$$\hat{\Lambda} = \arg \max_{\Lambda} p(\mathcal{F}|\Lambda)p(\Lambda).$$

It can be shown that this MAP estimation can be implemented by modifying the M-step in the E-M algorithm. The updating formulas for the E-M algorithm thus become the following:

E-step:

$$p(Z_{w,D} = 1) = \frac{\alpha_D^{(n)} p^{(n)}(w|\theta_T)}{\alpha_D^{(n)} p^{(n)}(w|\theta_T) + (1 - \alpha_D^{(n)}) p(w|\theta_B)}.$$

M-step:

$$\alpha_D^{(n+1)} = \frac{\sum_{w \in V} p(Z_{w,D} = 1) c(w, D)}{\sum_{w \in V} c(w, D)},$$

$$p^{(n+1)}(w|\theta_T) = \frac{\mu p(w|Q) + \sum_{D \in \mathcal{F}} p(Z_{w,D} = 1) c(w, D)}{\mu + \sum_{w' \in V} \sum_{D \in \mathcal{F}} p(Z_{w',D} = 1) c(w', D)}.$$

In this model, the parameter μ can be manually set, but the authors observed two problems with a fixed value of μ . (1) μ should be dependent on the amount of relevant information accumulated from the feedback documents, which is not known beforehand. (2) On the one hand, μ needs to be set to a large value in order for the topic language model θ_T to stay close to the original query language model; otherwise, θ_T could be easily drifted away to fit the popular topic(s)

in the feedback documents. On the other hand, setting μ to a large value would prevent θ_T from picking up from the feedback documents those query-related words that do not occur in the query. To solve this dilemma, the authors then proposed to use a regularized estimation method in which the parameter μ is initially set to a large value, and gradually decreased during the E-M iterations. This allows the topic language model θ_T to start from a model very close to the original query model, and then gradually deviate from the original query model to fit the data, i.e. the feedback documents. The authors introduced a parameter η to control when to stop the E-M iteration. Define

$$r = \sum_{w' \in V} \sum_{D \in \mathcal{F}} c(w', D) p(Z_{w',D} = 1),$$

the E-M iterations will stop when $\mu\eta \leq r$.

We applied this new method in our experiments to incorporate either pseudo relevance feedback or user relevance feedback.

3 HMM Passage Extraction

As we have discussed in Section 2, the second subtask is passage extraction. Since the relevant passages are query-specific, the passage boundaries should be determined dynamically based on the specific query. In our experiments, we applied a hidden Markov model (HMM)-based passage extraction method, which we proposed in [1]. This method has been shown to be effective on some general English corpora.

The intuition behind this method is that a relevant document containing a relevant passage and some non-relevant parts can be seen as a sequence of words generated from a linear hidden Markov model, where the relevant passage is generated from a relevance state that emits words according to a relevance language model (i.e. a relevance word probability distribution), and the non-relevant parts are generated from a number of background states that emits words according to a background language model. If the emission probabilities and the transition probabilities of the hidden Markov model are set appropriately, given a document, we can decode this sequence of words and find the sequence of states that have generated this word sequence with the highest probability. This state sequence tells us which words are generated from the relevance state and thus which part of the document belongs to the relevant passage.

Figure 1 shows the actual hidden Markov model we used. R is the relevance state, which has an emission probability distribution (i.e. the relevance language model) highly related to the query we consider. For example, this language model can be set to the original query language model as defined in Section 2. B_1 and B_3 are background states

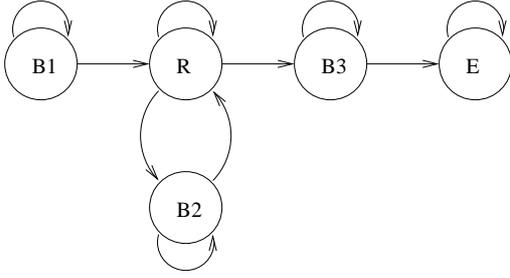


Figure 1. A 5-state HMM for passage extraction.

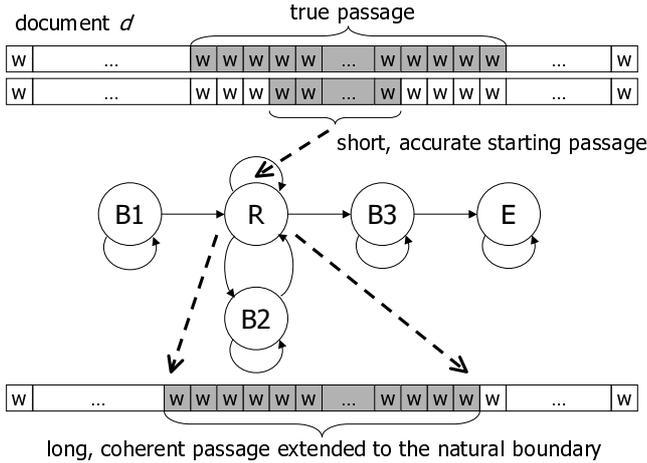


Figure 2. Incorporating pseudo relevance feedback into the HMM-based passage extraction method.

that generate the non-relevant text before and after the relevant passage in the document. B_2 is a background state for smoothing because a relevant passage also contains words that are not in the query. Words generated by state R and state B_2 are all considered inside the relevant passage. The emission probability distribution at states B_1 , B_2 and B_3 is estimated by the whole document collection, as in Section 2.

The transition probabilities in the hidden Markov model can affect the length of the relevant passage. For example, large transition probabilities from R to R , from B_2 to B_2 , and between R and B_2 would result in a relatively long relevant passage. Since passage length is query-dependent, these transition probabilities should not be manually fixed. What we proposed to do is to estimate these transition probabilities for each document by unsupervised learning. In another word, we try to find the transition probabilities that can maximize the likelihood of the document (word sequence) using the Baum-Welch algorithm, which is essentially an EM algorithm.

This HMM-based passage extraction also allows using relevance feedback to improve the estimation of the relevance language model at state R . Instead of using the original query language model, we can set the relevance

language model at state R to a feedback language model θ_F , where θ_F is estimated from the relevant passage extracted from the same document after a first round of passage extraction. Figure 2 illustrates this idea. Starting from a conservative relevance language model at state R , we can obtain a short and presumably accurate relevant passage. There may be relevant text surrounding this short, accurate passage. Let this starting short passage be P . We can then set the relevance language model at state R to

$$p(w|\theta_R) = p(w|P) = \frac{c(w, P)}{|P|},$$

where $c(w, P)$ is the number of occurrences of word w in P , and $|P|$ is the length of P . In this way, state R can now attract not only query words but also query-related words that are seen in the starting passage. The passage boundary can thus be extended to the natural topical boundary.

In our experiments, we applied our HMM-based passage extraction method, where we used paragraphs as documents. However, since learning the transition probabilities for each paragraph individually would be too time-consuming, we first trained the HMM on a sample of paragraphs, and used the average transition probabilities learned from these paragraphs for all paragraphs. Again, to save time, for state R , we simply used the original query language model. But we extended the extracted passage to the nearest sentence boundary at both ends.

4 User Relevance Feedback

Besides the two language modeling techniques shown above, we also experimented with user relevance feedback, in order to see how much true relevance feedback can help improve the performance compared with pseudo relevance feedback.

To obtain user relevance feedback, we first used KL-divergence retrieval method to retrieve top 10 paragraphs for each query. We then asked two domain experts to judge the relevance of these top 10 paragraphs for each query. The paragraphs that were judged to be relevant were then used for feedback.

5 Experiment Results

5.1 Official Runs

We submitted three official runs, including an automatic run, *UIUCauto*, and two interactive runs, *UIUCinter* and *UIUCinter2*. For all three runs, we first retrieved 1000 paragraphs using the KL-divergence retrieval model. Then from each retrieved paragraph, we used the HMM-based passage extraction method to extract the most relevant passage.

	Doc MAP	Psg MAP	Asp MAP
UIUCauto	0.3842	0.04864	0.2407
UIUCinter	0.4176	0.05906	0.2976
UIUCinter2	0.4243	0.06038	0.2900

Table 1. Results of the three official runs.

In *UIUCauto*, we used pseudo relevance feedback with the regularized estimation of the feedback mixture models, as discussed in Section 2. The top 10 documents for each query were used for pseudo relevance feedback. We used the default parameter $\eta = 1$.

In *UIUCinter* and *UIUCinter2*, we used user relevance feedback. In *UIUCinter*, the feedback mechanism was the same as in *UIUCauto*, that is, we also used the regularized feedback estimation method with $\eta = 1$. In *UIUCinter2*, we used the original feedback estimation method as proposed in [5]. We manually set the two parameters $\lambda = 0.75$ and $\alpha = 0.5$. λ was set to a value greater than the default value 0.5 because we believe we should trust these true relevance feedback documents more.

Table 1 shows the performance of the three official runs. As we can see, using user relevance feedback performed better than using pseudo relevance feedback, which is not surprising. However, using the original feedback estimation method (*UIUCinter2*) performed better than using the regularized estimation method (*UIUCinter*). Because we used the default parameter value for the regularized estimation method but non-default parameter values for the original estimation method, we could not directly draw any conclusion from the comparison between *UIUCinter* and *UIUCinter2*.

In order to study compare the regularized feedback estimation method with the original estimation method under the same setting, and to evaluate the HMM-based passage extraction method, we ran a number of diagnostic experiments.

5.2 Pseudo Relevance Feedback

First, we compared the performance of using pseudo relevance feedback with that of using no feedback. We show this comparison in Table 2. In the baseline method, we used the KL-divergence retrieval method and a Dirichlet prior for smoothing with $\mu = 1000$. No feedback was used in the baseline method. We then used the original feedback estimation method to incorporate pseudo relevance feedback. The top 10 documents for each query were used. Similarly, we used the regularized feedback estimation method with the top 10 documents for each query. For both feedback methods, we show the performance of two settings in the table: a default parameter setting, and an optimal parameter setting obtained by parameter tuning. Note that the regularized estimation method with the default parameter setting

Method		Doc MAP
Baseline		0.3484
Original Estimation	Def ($\lambda = 0.5, \alpha = 0.5$)	0.3606
	Opt ($\lambda = 0.8, \alpha = 0.8$)	0.3943
Regularized Estimation	Def ($\eta = 1.0$)	0.3842
	Opt ($\eta = 3.0$)	0.3942

Table 2. Pseudo relevance feedback using two feedback estimation methods (#Doc = 10).

#Doc	Doc MAP	
	Original Estimation	Regularized Estimation
10	0.3943	0.3942
20	0.3879	0.3934
30	0.3811	0.3801
50	0.3709	0.3725
100	0.3563	0.3473
150	0.3547	0.3523
200	0.3521	0.3465

Table 3. Pseudo relevance feedback using two feedback estimation methods with different numbers of feedback documents. For the original estimation method, $\lambda = 0.8$ and $\alpha = 0.8$. For the regularized estimation method, $\eta = 3.0$.

corresponds to the official run *UIUCauto*.

We can see from Table 2 that using pseudo relevance feedback with both estimation methods outperformed the baseline method. When the parameters were set to the optimal values, the two estimation methods performed similarly. But when the parameters were set to the default values, the regularized estimation method outperformed the original estimation method. Since in reality we cannot tune the parameters, the regularized estimation method is more robust than the original estimation method in practice.

To see how the number of pseudo relevance feedback documents affects the performance and how sensitive the two estimation methods are to this number, we tried a set of different numbers of feedback documents. The optimal parameter setting for each estimation method was used. Table 3 shows the results. We can see from the table that using more feedback documents decreased the performance. When the number of feedback documents was relatively small (under 50), the performance of the regularized estimation method did not decrease as much as the original estimation method. But for larger number of feedback documents (100 and above), the regularized estimation method performed worse than the original estimation method. Overall, it is not clear whether the original estimation method is more sensitive to the number of feedback documents than the regularized estimation method. This is different from what was found in [4].

Method		Psg MAP
UIUCauto	Paragraph	0.03753
	HMM Passage	0.04864
	Rel. Impr.	29.6%
UIUCinter	Paragraph	0.04481
	HMM Passage	0.05906
	Rel. Impr.	31.8%
UIUCinter2	Paragraph	0.04580
	HMM Passage	0.06038
	Rel. Impr.	31.8%

Table 4. Comparison between whole paragraphs as passages and passages extracted by the HMM-based method.

5.3 HMM-based Passage Extraction

To see the effectiveness of the HMM-based passage extraction method, we compared with a baseline method which returns whole paragraphs as passages. Recall that we treated paragraphs as documents and used KL-divergence method to retrieve paragraphs before we extracted the relevant passages within each retrieved paragraph. Therefore, this baseline method simply excluded the passage extraction step. For each of *UIUCauto*, *UIUCinter* and *UIUCinter2*, we applied this baseline method. Table 4 shows the comparison between the two methods. We can see that the HMM-based passage extraction method outperformed the baseline method in all three cases.

However, compared with the true relevant passages, the passages extracted by the HMM-based method were still far from optimal. In Table 5, we list the maximum length (MAX), minimum length (MIN), average length (AVG) and the standard deviation of the length for three kinds of passages. The first kind of passages are the true relevant passages. The second are the passages that are returned by the HMM-based method and that overlap with some true relevant passages. We only consider these overlapping passages because we do not want the retrieval performance to affect this analysis. The third kind of passages are whole paragraphs that overlap with some true relevant passages. As we can see from the table, the passages extracted by the HMM-based method are much longer than the true relevant passages on average. To see why our passages are in general longer, we examined some passages. Table 6 shows such an example. We can see that in this example, our passage has two more sentences than the true passage. This is because the last sentence in our passage contains the word "diseases", which is a query word. Our passage indeed represents a coherent piece of text, which is exactly what the HMM-based method tries to do. However, such kind of passages may be at a coarser granularity than what the genomics task desires. The passages extracted by the HMM-based method can be seen as upper bounds for the

	MAX	MIN	AVG	STD
True Psg	6928	27	399.8	489.4
HMM Psg	6955	34	1525.8	949.7
Paragraph	8670	60	2105.4	1136.8

Table 5. Some statistics of the length of three kinds of passages.

Method		Doc MAP
Baseline		0.3484
Original Estimation	Def ($\lambda = 0.5, \alpha = 0.5$)	0.3986
	Opt ($\lambda = 0.9, \alpha = 0.9$)	0.4511
Regularized Estimation	Def ($\eta = 1.0$)	0.4261
	Opt ($\eta = 6.0$)	0.4509

Table 7. User relevance feedback using two feedback estimation methods.

true passages.

One way to modify the HMM method to fit this task is maybe to use the document language model as the background model in the HMM structure. This change would make it easier for the HMM to distinguish a query-related region and the background regions in the document. We experimented with this idea, and found it helpful in a few cases. However, further investigation is needed to understand the problem.

5.4 User Relevance Feedback

Finally, we show the effect of using user relevance feedback. Similar to pseudo relevance feedback, we also used the two estimation method to estimate a topical language model from the feedback documents. The only different is that here we used true relevance feedback documents from domain experts. Some topics had no true relevant documents among the top-10 retrieved documents, so for these topics, we did not use any feedback. Table 7 shows the results. We can see that true relevance feedback is indeed more effective than the pseudo relevance feedback. The optimal parameter values reflect the fact that we should trust the true relevance feedback documents more than the pseudo relevance feedback documents. However, in reality, we still do not know the optimal parameter values. If we choose to use the default parameter setting, the regularized estimation method can still outperform the original estimation method, as shown in the table.

The regularized estimation method with default parameter setting in Table 7 should correspond to *UIUCinter*. However, in *UIUCinter*, we made a mistake by using the results from *UIUCauto* for those topics that did not get any judged relevant documents. As a result, *UIUCinter* performed worse than it should be.

Prion diseases, which include Creutzfeldt-Jacob disease in humans, mad cow disease in cattle, and scrapie in sheep, involve the misfolding of the benign cellular prion protein (PrP C) 1 to the infectious disease-causing scrapie isoform PrP Sc. The prion protein (PrP C) is a copper-binding cell surface glycoprotein. The role of copper in the normal function of PrP, as well as in prion diseases, has been the subject of a number of excellent reviews. The mature cellular form of PrP consists of residues 23 to 231 and is tethered to the cell surface via a glycosylphosphatidylinositol anchor at the C terminus. There are now a number of NMR solution structures of copper-free mammalian PrPs. A crystal structure of PrP C has also been published; this structure is dimeric involving domain swapping of the monomeric form.

Table 6. A passage example. The whole paragraph is a single paragraph from the original article. The words in italic type are inside the passage extracted by the HMM-based method. Words in bold type are inside the true passage.

6 Summary

In summary, in this year's Genomics Track, we focused on testing the effectiveness of two language modeling techniques for information retrieval on biomedical text. The general observation is that the two techniques are still effective to some degree on biomedical text. The regularized feedback estimation method is more robust than the original feedback estimation method because it needs less parameter tuning. The HMM-based passage extraction method can outperform paragraph-based passages. However, since the HMM-based method is not designed to extract short passages with very specific information, it needs some modification in order to fit this task. Finally, user relevance feedback is very effective.

7 Acknowledgments

We thank Xu Ling for judging the results to provide user relevance feedback. We thank Matt J. Wood and Fabien Campagne for providing a sentence locator tool. We also thank Martijn Schuemie for the parsed HTML documents.

References

- [1] J. Jiang and C. Zhai. Extraction of coherent relevant passages using hidden markov models. *ACM Transactions on Information Systems (TOIS)*, 24(3):295–319, 2006.
- [2] J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'01)*, 2001.
- [3] V. Lavrenko and W. B. Croft. Relevance-based language models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'01)*, 2001.
- [4] T. Tao and C. Zhai. Regularized estimation of mixture models for robust pseudo-relevance feedback. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'06)*, 2006.
- [5] C. Zhai and J. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of the 10th ACM International Conference on Information and Knowledge Management (CIKM'01)*, 2001.