Semantic Smoothing for Model-based Document Clustering

Xiaodan Zhang, Xiaohua Zhou, Xiaohua Hu
College of Information Science & Technology, Drexel University
xzhang@ischool.drexel.edu, xiaohua.zhou@drexel.edu, thu@ischool.drexel.edu

Abstract

*A document is often full of class-independent "general" words and short of class-specific "core" words, which leads to the difficulty of document clustering. We argue that both problems will be relieved after suitable smoothing of document models in agglomerative approaches and of cluster models in partitional approaches, and hence improve clustering quality. To the best of our knowledge, most model-based clustering approaches use Laplacian smoothing to prevent zero probability while most similarity-based approaches employ the heuristic TF*IDF scheme to discount the effect of "general" words. Inspired by a series of statistical translation language model for text retrieval, we propose in this paper a novel smoothing method referred to as context-sensitive semantic smoothing for document clustering purpose. The comparative experiment on three datasets shows that model-based clustering approaches with semantic smoothing is effective in improving cluster quality.*

1. Introduction

Agglomerative and partitional approaches are two common strategies for document clustering. A volume of previous work has been focused on this area. In this paper, we aim at improving the cluster quality of these two approaches from the point of view of model estimation and smoothing.

Steinbach et al. [7] argue that the agglomerative hierarchical clustering perform poorly because the nearest neighbors of a document belong to different classes in many cases. According to their examination on the data, each class has a "core" vocabulary of words and remaining "general" words may have similar distributions on different classes. Thus, two documents from different classes may share many general words (e.g. stop words) and will be viewed similar in terms of vector cosine similarity. To solve this problem, we should "discount" general words and "emphasize" more importance on core words in a vector. Besides, we think the poor performance of the agglomerative clustering can also be attributed to the sparsity of core words in a document. A document is often short and contains very few number of core words. Thus, two documents from the same class may share few core words and be falsely grouped into different clusters when using vector cosine similarity metric. To solve this problem, we should assign reasonable positive counts to "unseen" core words if its related topical words occur in the document.

Partitional approaches estimate cluster models instead of document models. A cluster often contains much more than one document. Thus, the data sparsity problem is not as serious as in pairwise document similarity calculation. But if the size of the dataset for clustering is small or the dataset is extremely skewed upon different classes, the sparsity of core words will still be a serious problem. Besides, no matter how many documents a cluster have, general words always dominate the cluster; thus, discounting the effect of general words is always helpful to improve cluster quality.

Discounting seen words and assigning reasonable counts to unseen words are two exact goals of the probabilistic language model smoothing. To the best of our knowledge, the effect of model smoothing has not been extensively studied in the context of document clustering. Most model-based clustering approaches simply use Laplacian smoothing to prevent zero probability [5] [11] while most similarity-based clustering approaches employ the heuristic TF*IDF scheme to discount the effect of "general" words [7]. In contrast, the study of language model smoothing has been a hot topic in the community of information retrieval (IR) with the increasing popularity of the language modeling approach to IR in recent years [2] [8] [9] [12]. In this paper, we will adapt the smoothing techniques used in IR to the context of document clustering and hypothesize that the document or cluster model smoothing can significantly improve the quality of model-based document clustering.

Berger and Lafferty [2] proposed a kind of semantic smoothing approach referred to as the statistical translation language model which statistically mapped document terms onto query terms.

$$p(q \mid d) = \sum_{w} t(q \mid w) p(w \mid d) \tag{1}$$

where $t(q|w)$ is the probability of translating the document term w to the query term q and $p(w|d)$ is the maximum likelihood estimator of the document model. With term translations, a document containing "star" may be returned for the query "movie" because two words have strong association on the topic of entertainment. However, this approach suffers from the context-insensitivity problem, i.e., it is unable to incorporate contextual and sense information into the model. Thus, the resulting translation may be fairly general and contain mixed topics.

To overcome the context-insensitivity problem, we propose a novel context-sensitive semantic smoothing method suitable for document clustering, inspired by the topic signature language model [12]. The basic idea of the new smoothing method is to identify multiword phrases and then statistically map multiword phrases into individual document terms. Here, a multiword phrase can be viewed as a sub-topic or a latent theme. The semantics of a phrase is clear and explicit since a multiword phrase is unambiguous in most cases. Thus, the translation of phrases to individual terms will be very specific. In addition, unlike the distribution of individual terms, the distribution of most multiword phrases depends on the topic. Therefore, phrases are very helpful to cluster documents. For example, a document containing word "Israel" may be merged into the same cluster with a document containing phrase "Arab country" because "Arab country" is highly associated with "Israel" in the corpus.

We evaluate our semantic smoothing method in conjunction with a model-based agglomerative algorithm and a model-based K-Means algorithm on three datasets: 20-newsgroups, TDT2, and LA Times using the Dragon Toolkit [13] developed by ourselves.

2. Phrase extraction and translation

In our previous work [12], we proposed a topic signature language model for IR use. We implemented topic signatures as concept pairs in that model and developed an ontology-based approach to extract concepts and concept pairs from documents. However, for many domains, ontology is not available. For this reason, we propose the use of multiword phrases as topic signatures and employ *Xtract* [6] to identify phrases in documents. *Xtract* is a kind of statistical extraction tool with some syntactic constraints. It is able to extract noun phrases frequently occurring in the corpus without any external knowledge. *Xtract* uses four parameters, strength (k_0), peak z-score (k_1), spread (U_0), and percentage frequency (T), to control the quantity and quality of the extracted phrases. In the experiment, the four parameters are set to 1, 1, 4, and 0.75, respectively.

We index all documents in a given collection C with terms (individual words) and topic signatures (phrases). For each phrase t_k, we have a set of documents (D_k) containing that phrase. Intuitively, we can use the document set D_k to estimate the translation model for t_k, i.e., determining the probability of translating the given phrase to terms in the vocabulary. If all terms appearing in the document set center on the topic signature t_k, we can simply use maximum likelihood estimator and the problem is as simple as term frequency counting. However, one document also contains terms from other topic signatures and background collection model, both referred as noise. We then use a mixture language model to remove noise. Assuming the set of documents containing t_k is generated by a mixture language model (i.e., all terms in the document set are either translated by the given topic signature model $p(w|\theta_{t_k})$ or generated by the background collection model $p(w|C)$), we have:

$$p(w|\theta_{t_k}, C) = (1-\beta)p(w|\theta_{t_k}) + \beta p(w|C) \quad (2)$$

where β is a coefficient accounting for the background noise and θ_{t_k} denotes the translation model parameters. Using the EM algorithm [4], we obtain the following update formulas for model parameters:

$$\hat{p}^{(n)}(w) = \frac{(1-\beta)p^{(n)}(w|\theta_{t_k})}{(1-\beta)p^{(n)}(w|\theta_{t_k}) + \beta p(w|C)} \quad (3)$$

$$p^{(n+1)}(w|\theta_{t_k}) = \frac{c(w, D_k)\hat{p}^{(n)}(w)}{\sum_i c(w_i, D_k)\hat{p}^{(n)}(w_i)} \quad (4)$$

where $c(w, D_k)$ is the document frequency of term w in D_k., i.e., the cooccurrence count of w and t_k in the whole collection.

In the experiment, we set the background coefficient β=0.5. We also truncate terms with extreme small translation probabilities for two purposes. First, with smaller number of translation space, the document smoothing will be much more efficient. Second, we assume most terms with extreme small probability are noise (i.e. not semantically related to the given topic signature). In detail, we disregard all terms with translation probability less than 0.001 and renormalize the translation probabilities of the remaining terms.

3. Clustering methods

In this section, we briefly describe the model-based clustering algorithms and then discuss the method of the document model smoothing for agglomerative approaches and the cluster model smoothing for partitional approaches.

3.1. Model-based agglomerative clustering

The key of agglomerative clustering is to measure the distance of two clusters, which is further reduced to the calculation of pairwise document distance. We take the complete linkage criterion for agglomerative clustering. The Kullback-Leibler divergence [4] between documents models with context-sensitive semantic smoothing is used as the document distance metric. Given two probabilistic document models $p(w|d_1)$ and $p(w|d_2)$, the KL-divergence distance is defined as:

$$\Delta(d_1, d_2) \equiv \sum_{w \in V} p(w \mid d_1) \log \frac{p(w \mid d_1)}{p(w \mid d_2)} \quad (5)$$

where V is the vocabulary of the corpus. The KL-divergence distance will be a non-negative score. It gets the zero value if and only if two document models are exactly same. However, KL-divergence is not a symmetric metric. Thus, we define the distance of two documents as the minimum of two KL-divergence distances:

$$dist(d_1, d_2) \equiv \min\{\Delta(d_1, d_2),\ \ \Delta(d_2, d_1)\} \quad (6)$$

$$p_{bt}(w|d) = (1\text{-}\lambda) p_b(w|d) + \lambda p_t(w|d) \quad (7)$$

The estimation of the document model is described in equation (7). It is a mixture model with two components: *a simple language model* and *a topic signature translation model*. The translation coefficient (λ) is to control the influence of two components in the mixture model. With training data, the translation coefficient can be optimized by maximizing certain clustering quality metrics such as NMI [1].

The simple language model can be easily obtained using the maximum likelihood estimator (MLE) together with background smoothing such as *Jelinek-Mercer*, *dirichlet*, *absolute discount* [8] and *two-stage language model* [9]. In this paper, we take the *Jelinek-Mercer* smoothing. That is,

$$p_b(w|d) = (1-\alpha) p_{ml}(w|d) + \alpha p(w|C) \quad (8)$$

where α is a coefficient accounting for the background collection model $p(w|C)$ and $p_{ml}(w|d)$ is the MLE document model. In our experiment, α is set to 0.5.

The translation model smoothes document models by statistically mapping context-sensitive topic signatures onto individual terms. That is,

$$p_t(w|d) = \sum_k p(w|t_k) p_{ml}(t_k|d) \quad (9)$$

Here t_k denotes topic signatures identified in a document. The probability of translating t_k to individual term w is estimated with the approach described in Section 2. The likelihood of a given document generating the topic signature t_k can be estimated with

$$p_{ml}(t_k \mid d) = \frac{c(t_k, d)}{\sum_i c(t_i, d)} \quad (10)$$

where $c(t_i, d)$ is the frequency of the topic signature t_i in a given document d.

3.2. Model-based partitional clustering

The model-based partitional clustering assumes that there are k parameterized models, one for each cluster. Basically, the algorithm iterates between a model re-estimation step and a sample re-assignment step [11].

Some previous studies [11] showed that the multinomial model consistently outperformed the multivariate Bernoulli model. For this reason, we choose multinomial model for evaluation. Based on the naive Bayes assumption, the log likelihood of document d generated by the j-th multinomial cluster model will be:

$$\log p(d \mid c_j) = \sum_{w \in V} c(w,d) \log p(w \mid c_j) \quad (11)$$

where $c(w,d)$ denotes the frequency count of word w in document d and V denotes the vocabulary. Thus, the problem remains to estimate parameters $p(w \mid c_j)$ for the cluster model. The parameter estimation of multinomial models is as simple as counting word frequency in the cluster. However, one has to smooth the model in order to prevent zero probability caused by data sparsity. Laplacian smoothing approach assigns all unseen words of a given cluster a fixed probability. It is very simple and frequently used, but not very effective. We approach cluster model estimation with semantic smoothing. The cluster model smoothing is very similar to the document model. That is,

$$p(w|c_j) = (1\text{-}\lambda) p_b(w|c_j) + \lambda p_t(w|c_j) \quad (12)$$

The translation coefficient λ is to control the influence of two components in the mixture model: a simple language model $p_b(w|c_j)$ and a statistical translation model $p_t(w|c_j)$. The simple language model is again the mixture of the maximum likelihood model and the corpus model.

$$p_b(w|c_j) = (1-\alpha) p_{ml}(w|c_j) + \alpha p(w|C) \quad (13)$$

Here, we set α to 0.5. The topic signature translation model is estimated with the formula below

$$p_t(w \mid c_j) = \sum_k p(w \mid t_k) p(t_k \mid c_j) \qquad (14)$$

Here, t_k denotes the topic signature and $p(w|t_k)$ is the probability of translating t_k to w.

4. Experiment setting and result analysis

Cluster quality is evaluated by three extrinsic measures, *purity* [10], *entropy* [12], and *normalized mutual information (NMI)* [1]. The results in terms of three measures are consistent with each other on all runs. Here we will analyze the results of NMI, an increasing popular measure for evaluating cluster quality. Results of purity are also listed to support our conclusion, but not explained as it's consistent with NMI. Because of space restriction, result of entropy would not be listed.

We run agglomerative hierarchical clustering with complete linkage criterion. The two document distance metrics used for the comparative experiment are the traditional vector cosine and the Kullback-Leibler divergence proposed in this paper. For cosine similarity, we try two different representation schemes: term frequency and TF-IDF. For KL-divergence metric, we test eleven translation coefficients (λ) ranging from 0 to 1. When λ=0, it is actually the simple background smoothing.

We take the model-based k-means and the standard k-means for partitional clustering. For model-based k-means, we compare the effectiveness of three different smoothing methods: Laplacian smoothing, background smoothing, and semantic smoothing. Similar to the run of agglomerative clustering, we test eleven translation coefficients (λ) ranging from 0 to 1 for semantic smoothing. For standard k-means, we use traditional vector cosine as similarity measure on three different representation schemes: TF, normalized TF, and TF-IDF. Since the result of K-Means clustering varies with the initialization, we run ten times with random initialization and take the average as the result. During the comparative experiment, each run has the same initialization.

We do clustering experiments on three datasets: TDT2, LA Times (from TREC), and 20-newsgroups (20NG). We selected 7,094 documents in TDT2 that have a unique class label, 18,547 documents from top ten sections of LA Times, and all 19,997 documents in 20-newsgroups. The ten classes selected from TDT2 are *20001, 20015, 20002, 20013, 20070, 20044, 20076, 20071, 20012, and 20023*. The ten sections selected from LA Times are *Entertainment, Financial, Foreign, Late Final, Letters, Metro, National, Sports, Calendar, and View*. All 20 classes of 20NG are selected for testing.

We hypothesize that the effect of semantic smoothing on small dataset is stronger than on large dataset because small dataset has serious data sparsity problem when using model-based K-Means. To test this hypothesis, we run partitional clustering on both large and small datasets. A large dataset contains all the documents from selected classes. To build small datasets, we randomly pick 100 random documents from each selected class of a given dataset and then merge them into a big pool for clustering. For each dataset, we create five small datasets and average the experiment results. We test agglomerative clustering on only small datasets because it suffers from the $O(n^2)$ clustering time. For all experiments, we cluster data into k (=the number of selected classes) clusters.

4.1. Agglomerative clustering results

Table 1. NMI results of the agglomerative hierarchical clustering with complete linkage criterion. "Bkg" and "Semantic" denote simple background smoothing and semantic smoothing, respectively.

Dataset	Vector Cosine		KL-Divergence	
	TF	TF-IDF	Bkg	Semantic
TDT2	0.369	0.484	0.550	**0.743(λ=0.8)**
20NG	0.135	0.135	0.155	**0.227(λ=0.6)**
LA Times	0.059	0.054	0.104	**0.202(λ=0.8)**

Table 2. Purity results of the agglomerative clustering with complete linkage criterion.

Dataset	Vector Cosine		KL-Divergence	
	TF	TF-IDF	Bkg	Semantic
TDT2	0.446	0.54	0.606	**0.826(λ=0.8)**
20NG	0.095	0.103	0.158	**0.236(λ=0.5)**
LA Times	0.128	0.127	0.227	**0.333(λ=0.8)**

As shown in table 1, when using the vector cosine for pairwise document similarity measure, the TF-IDF scheme performs slightly better than the TF scheme. As we discussed before, the heuristic TF-IDF weighting scheme can discount "general" words and strengthen "specific" words in a document vector. Thus, it can improve the agglomerative clustering quality. The KL-divergence similarity measure combined with background smoothing of document models (i.e., λ=0) consistently outperforms the cosine measure on both Normalized TF and TF-IDF schemes. As expected, the KL-divergence measure with context-sensitive semantic smoothing significantly improves the quality of the agglomerative clustering on all three datasets. The semantic smoothing not only weakens the effect of class-independent general words, but also assigns reasonable probability to unseen words by phrase-word translations. Since the direct comparison of two documents suffers from the severe data sparsity problem, the semantic smoothing can dramatically improve the quality of agglomerative clustering.

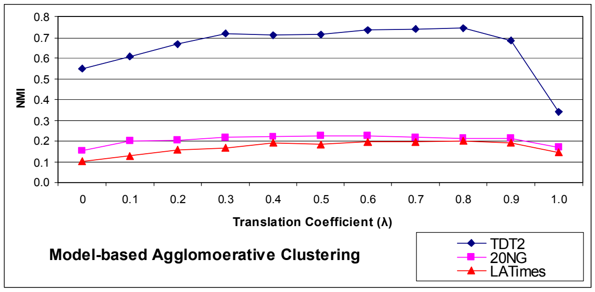

Figure 1. The variance of the cluster quality with the translation coefficient (λ)

To see the robustness of the semantic smoothing effect, we show the performance curve in Figure 1. Except for the point of λ=1, the semantic smoothing always improve the cluster quality over the simple background smoothing (λ=0). In general, NMI will increase with the increase of translation coefficient till the peak point (around 0.7 in our case) and then go downward. The phrase-based semantic smoothing will cause information loss because not all topics in a document can be expressed by multiword phrases. Thus, we interpolate the translation model with a simple language model to make up the loss. Now it is easy to understand why the NMI goes downward when the influence of the semantic smoothing is too high.

4.2. Partitional clustering results

The clustering NMI results of partitional clustering on small datasets and large datasets are listed in Table 3&4, respectively. Of three smoothing schemes for model-based K-Means, Laplacian smoothing always performs the worst on both large datasets and small datasets. This indicates that assigning an equal probability to unseen words is a bad smoothing scheme. It just technically prevents zero probabilities, but can not weaken the effect of general words as the background smoothing does.

The comparisons of the semantic smoothing and the background smoothing show different patterns on small datasets and large datasets. As shown in Table 3 & 4, the semantic smoothing performs consistently better than the background smoothing on small datasets whereas the results of two smoothing methods are almost same on large datasets. This is because when the dataset is small, the cluster models still suffers from the data sparsity problem more or less and when the size of the dataset increases, the data sparsity problem disappears and the semantic smoothing can not take the advantage over the background smoothing any more. This pattern can be observed apparently in Figure 2 & 3.

Table 3. NMI results of partitional clustering on small data sets. “Lap”, “Bkg”, and “Semantic” denote Laplacian smoothing, background smoothing, and semantic smoothing, respectively.

Dataset	Standard K-Means			Model-based K-Means		
	TF	NTF	TF-IDF	Lap	Bkg	Semantic
TDT2	0.792	**0.805**	0.790	0.246	0.607	0.672(λ=0.6)
20NG	0.197	0.161	**0.374**	0.107	0.187	0.318(λ=0.9)
LATimes	0.194	0.197	0.166	0.097	0.120	**0.215(λ=0.7)**

Table 4. NMI results of partitional clustering on large data sets.

Dataset	Standard K-Means			Model-based K-Means		
	TF	NTF	TFIDF	Lap	Bkg	Semantic
TDT2	0.685	0.699	**0.710**	0.468	0.661	0.665(λ=0.1)
20NG	0.188	0.176	**0.521**	0.093	0.443	0.472(λ=0.6)
LATimes	0.193	0.203	**0.352**	0.243	0.320	0.327(λ=0.5)

Table 5. Purity results of partitional clustering on small data sets.

Dataset	Standard K-Means			Model-based K-Means		
	TF	NTF	TFIDF	Lap	Bkg	Semantic
TDT2	0.872	**0.886**	0.848	0.105	0.708	0.758(λ=0.7)
20NG	0.234	0.210	**0.396**	0.055	0.245	0.319(λ=0.5)
LATimes	0.318	0.320	0.311	0.103	0.270	**0.342(λ=0.6)**

Table 6. Purity results of partitional clustering on large data sets.

Dataset	Standard K-Means			Model-based K-Means		
	TF	NTF	TFIDF	Lap	Bkg	Semantic
TDT2	0.873	0.870	**0.877**	0.582	0.841	0.850(λ=0.1)
20NG	0.236	0.220	**0.508**	0.092	0.452	0.463(λ=0.4)
LATimes	0.340	0.350	**0.467**	0.334	0.453	0.457(λ=0.5)

It’s worth noting that the agglomerative clustering with TF*IDF scheme performs significantly worse than with semantic smoothing whereas the partitional clustering with TF*IDF scheme is consistently as good as or slightly better than with semantic smoothing. We believe this difference is caused by the following two things. First, as mentioned before, the major problem with agglomerative clustering is data sparsity and semantic smoothing is very effective in solving data sparsity problem. Second, the data sparsity issue in partitional clustering becomes minor because a cluster often contains much more than one document; instead, the density of general words becomes the major issue.

Although semantic smoothing also has the effect of "discounting" general words, TF*IDF scheme does much more aggressively than semantic smoothing.

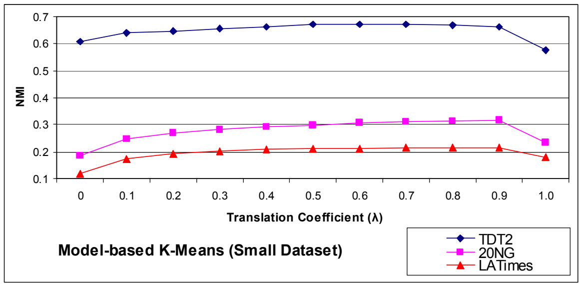

Figure 2. The variance of the cluster quality on small datasets

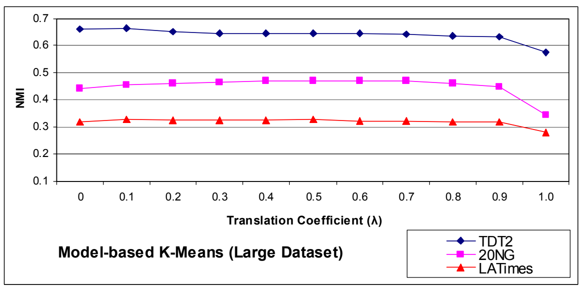

Figure 3. The variance of the cluster quality on large datasets

5. Conclusions

In this paper, we evaluated the effect of semantic smoothing with a model-based agglomerative clustering and a model-based partitional clustering on three different datasets. The experiment shows that our semantic smoothing is very promising for model-based text clustering. In detail, we obtained following interesting findings from the experiment by comparing semantic smoothing to other five schemes: Laplacian smoothing, background smoothing, TF-cosine, NTF-cosine, and TF*IDF-cosine: (1) Semantic smoothing is much more effective than other schemes on agglomerative clustering where data sparsity is the major problem. (2) The effectiveness of semantic smoothing with partitional clustering depends on the size of the dataset. When dataset is small and data sparsity is the major problem, semantic smoothing is very effective; otherwise, it equals to background smoothing. (3) Although both semantic smoothing and background smoothing can weaken the effect of general words, they are less effective than TF*IDF which is more aggressive on discounting general words. (4) Laplacian smoothing is the worst among all tested schemes.

6. Acknowledgement

This work is supported in part by NSF Career grant (NSF IIS 0448023), NSF CCF 0514679, PA Dept of Health Tobacco Settlement Formula Grant (No. 240205 and No. 240196), and PA Dept of Health Grant (No. 239667).

References

[1] Banerjee, A. and Ghosh, J. Frequency sensitive competitive learning for clustering on high-dimensional hperspheres. *Proc. IEEE Int. Joint Conference on Neural Networks*, pp. 1590-1595.

[2] Berger, A. and Lafferty J. Information Retrieval as Statistical Translation. In *Proceedings of the 22nd ACM SIGIR Conference*, 1999, pp.222-229.

[3] Kaufman, L. and Rousseuw, P.J. *Finding Groups in Data:* an Introduction to Cluster Analysis, John Wiley and Sons, 1990.

[4] Kullback, S. and Leibler, R.A. On information and sufficiency. Annals of Mathematical Statistics, 22(1):79–86, March 1951.

[5] Nigam, K., McCallum, et. al, Text Classification from Labeled and Unlabeled Documents using EM, *Machine Learning*, Volume 39 , Issue 2-3, pp103-134

[6] Smadja, F. Retrieving collocations from text: Xtract. *Computational Linguistics*, 1993, 19(1), pp. 143--177.

[7] Steinbach, M., Karypis, G., and Kumar, V. *A Comparison of document clustering techniques*. Technical Report #00-034, Dept. of Computer Science and Engineering, University of Minnesota, 2000.

[8] Zhai, C. and Lafferty, J. A Study of Smoothing Methods for Language Models Applied to Ad hoc Information Retrieval, SIGIR'01, pp.334-342.

[9] Zhai, C. and Lafferty, J. Two-Stage Language Models for Information Retrieval., *SIGIR'02*.

[10] Zhao, Y. and Karypis, G. *Criterion functions for document clustering: experiments and analysis*, Technical Report, CS Dept., Univ.of Minnesota, 2001.

[11] Zhong, S. and Ghosh, J. Generative model-based document clustering: a comparative study. *Knowledge and Information Systems*, 8(3): 374-384, 2005.

[12] Zhou, X., Hu, X., Zhang, X., Lin, X., and Song, I.-Y. Context-Sensitive Semantic Smoothing for the Language Modeling Approach to Genomic IR, SIGIR'06

[13] Zhou, X., Zhang, X., and Hu, X., The Dragon Toolkit, Data Mining & Bioinformatics Lab, iSchool at Drexel University, http://www.ischool.drexel.edu/dmbio/dragontool