Dragon Toolkit: Incorporating Auto-learned Semantic Knowledge into Large-Scale Text Retrieval and Mining

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

Abstract

The majority of text retrieval and mining techniques are still based on exact feature (e.g. words) matching and unable to incorporate text semantics. Many researchers believe that the extension with semantic knowledge could improve the results and various methods (most of them are heuristic) have been proposed to account for concept hierarchy, synonymy, and other semantic relationships. However, the results with such semantic extension have been mixed, ranging from slight improvements to decreases in effectiveness, mostly likely due to the lack of a formal framework. Instead, we propose a novel method to address the semantic extension within the framework of language modeling. Our method extracts explicit topic signatures from documents and then statistically maps them into single-word features. The incorporation of semantic knowledge then reduces to the smoothing of unigram language models using semantic knowledge. The dragon toolkit reflects our method and its effectiveness is demonstrated by three tasks, text retrieval, text classification, and text clustering.

1. Introduction

Text retrieval and mining applications often involve a huge feature space. The comparisons between single documents or document groups using exact feature matching may be unreliable because the decision is based on a very small number of common features. Many researchers believe that the extension with semantic knowledge can improve the quality of the comparison. A straightforward extension is to expand document vectors through ontologies such as WordNet and UMLS. However, this line of approaches often lacks a formal framework. The results have been mixed, ranging from slight improvements to decreases in effectiveness. Liu et al. expanded query terms using WordNet and word sense disambiguation (WSD) and the retrieval accuracy was improved [10]. But Sanderson concluded in [14] that long queries could not take the benefit from WSD and a sense disambiguator contributing to an IR system should have a high accuracy. Zhang et al. compared eight ontology-based similarity metrics for document vector expansions. But, neither of them outperformed the original vector in the setting of k-means document clustering [20].

A more formal approach to the utilization of feature semantics is topic modeling [4] [7], where a document is represented by a fixed number of weighted topics and each topic can be further described as a set of similar or related words. The text retrieval [16] and categorization [5] based on the topic modeling have been empirically proved to be effective. However, this approach has several limitations. First, it is not intuitive; the extraction of "abstract" topics is totally blind to end users. Second, it is difficult to reuse the extracted topics because determining the weights of topics in a new document is a challenging task. Third, topic modeling for a large collection is not efficient.

Berger and Lafferty proposed a statistical translation model for document expansions, which statistically maps document terms into query terms [3]. The incorporation of semantic knowledge then reduces to the smoothing of unigram language models using semantic knowledge. As long as the translation probabilities between words are available, the document expansion is straightforward. Thus, it can solve the first two problems of the topic modeling approach. However, the estimate of translation parameters remains inefficient. Furthermore, it introduces two new problems. First, the "translation" procedure needs a large number of document-query pairs for training, which are difficult to obtain in real world. Second, contextual information is unable to be included and the translation result may be fairly general and contain mixed topics.

We propose in this paper a novel method to address semantic extensions based on the idea of the translation model. Our method extracts explicit topic signatures (e.g. words, multiword phrases, and ontological concepts) from a document and then statistically maps them into single-word features. This semantic mapping component is then used to smooth unigram document models. The semantic mapping from topic signatures to single-word features is based on co-occurrence data which are very easy to collect. The semantic mapping for each topic

signature can be estimated separately, making this approach quite efficient and scalable to large collections. Various topic signatures are compatible with our method. Especially, if topic signatures (e.g. multiword phrases and ontological concept) self-contain context, mappings are highly specific and accurate. The dragon toolkit [25] is an illustration of our approach. We conduct comprehensive experiments on text retrieval, text classification, and text clustering using this toolkit. The results prove that the new approach is not only scalable to large text retrieval and mining applications, but also effective in improving their performance.

2. Overview of the Dragon Toolkit

The dragon toolkit is implemented in Java and consists of six main components as shown in Figure 1. Topic signature extraction and semantic knowledge learning are detailed in section 3. Unlike other tools such as Weka and CLUTO which load all data into memory in the running time, the dragon toolkit is built on sparse matrices which are partially loaded into memory on demand. Thus, the toolkit is highly scalable and capable of handling hundred thousands of documents with limited memory. A collection can generate multiple topic signature representations each of which corresponds to a sparse matrix. The acquisition of co-occurrence data for semantic mapping is then quite straightforward.

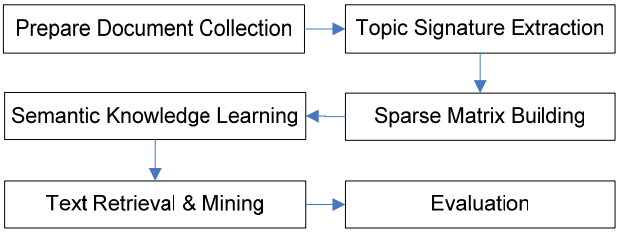

Figure 1: The architecture components of the dragon toolkit

The dragon toolkit provides two types of interfaces. A developer can either directly call well-organized APIs in their programs, or gain access to full functionalities through xml-based configuration files. The configuration files of some experiments in this paper are available at the website of the dragon toolkit [2].

3. Topic Signature Extraction and Mapping

The toolkit is able to extract four types of topic signatures. They are word, multiword phrase, ontological concept, and concept pair. Multiword phrases are extracted by a modified version of Xtract [11]. The detail of the implementation is available in our previous work [24]. The extraction of last two topic signatures needs domain ontologies. The current version of the toolkit has integrated two biomedical ontologies, UMLS and MeSH. UMLS concepts are extracted by MaxMatcher [23], a dictionary-based concept extraction tool. A concept pair is defined as two order-free concepts with semantic and syntactic relationships. The details of the implementation can be found in our previous work [22].

The estimate of semantic mapping from a topic signature to individual words is based on co-occurrence data. We introduce a mixed language model to separate topic information from background information and then use EM algorithm to estimate the parameters. The detail of the algorithm is available in previous work [22] [24]. Examples of semantic mapping are shown in Figure 2.

Space:

space 0.245; shuttle 0.057; launch 0.053; flight 0.042; air 0.035; program 0.031; center 0.030; administration 0.026; develop 0.025; like 0.023; look 0.022; world 0.020; director 0.020; plan 0.018; release 0.017; problem 0.017; work 0.016; place 0.016; mile 0.015; base 0.014;

Program:

program 0.193; washington 0.026; congress 0.026; administration 0.024; need 0.024; billion 0.023; develop 0.023; bush 0.020; plan 0.020;money 0.020; problem 0.020; provide 0.020; writer 0.018; d 0.018; help 0.018; work 0.017; president 0.017; house .017; million 0.016; increase 0.016;

Space Program

space 0.101; program 0.071; NASA 0.048; shuttle 0.043; astronaut 0.041; launch 0.040; mission 0.038; flight 0.037; earth 0.037; moon 0.035; orbit 0.032; satellite 0.031; Mar 0.030; explorer 0.028; station 0.028; rocket 0.027; technology 0.026; project 0.025; science 0.023; budget 0.023;

Figure 2: The demonstration of semantic mapping (only top 20 topical terms are listed). All three examples are trained on the 20-newsgroup corpus.

If topic signatures such as multiword phrases and ontological concepts self-contain contextual information, the mapping is context-sensitive. The corresponding language model smoothing is referred to as context-sensitive semantic smoothing (***CSSS***). Otherwise, the smoothing is referred to as context-insensitive semantic smoothing (***CISS***). From three examples shown in Figure 2, we can see that context-sensitive mapping is more specific and coherent than context-insensitive mapping.

4. Text Retrieval

We compare four retrieval models, Okapi [13], two-stage language model (TSLM) [19], CISS and CSSS. The equation (4.1) describes the retrieval model of CISS and CSSS. It is a mixture of the two-stage language model $p_{ts}(w \mid d)$ and the semantic mapping model controlled by the translation coefficient (λ). If λ is set to zero, it becomes a TSLM; if λ is set to one, it becomes a pure semantic mapping model.

$$p(w \mid d) = (1 - \lambda) p_{ts}(w \mid d) + \lambda \sum_{k} p(t_k \mid d)\, p(w \mid t_k) \qquad (4.1)$$

The aforementioned four models are evaluated on five collections. For CSSS, Genomics 2004 uses UMLS concepts as topic signatures and other four newswire collections take multiword phrases as topic signatures. All models are well tuned. The translation coefficient for CISS and CSSS is set to 0.3 according to our previous work [22]. The results are shown in Table 1. Mean average precision (MAP) and recall at first 1000 documents are two performance metrics for retrieval. TSLM and Okapi achieve similar performance. Both CISS and CSSS outperform significantly than Okapi and TSLM. CSSS achieves slight improvement over CISS on all five collections because CSSS takes the advantage of contextual information and makes the semantic mapping more specific and accurate.

Table 1: The comparison of four retrieval models (Okapi, TSLM, CISS and CSSS) on five testing collections

Collections		Okapi	TSLM	CISS	CSSS
Genomics 2004	MAP	0.369	0.352	0.408	**0.422**
	Recall	6847	6544	7176	**7279**
AP88-89 51-100	MAP	0.239	0.252	0.272	**0.288**
	Recall	3346	3428	3735	**3771**
AP88-89 101-150	MAP	0.220	0.219	0.235	**0.246**
	Recall	3087	3055	3237	**3445**
WSJ90-92 101-150	MAP	0.249	0.239	0.244	**0.256**
	Recall	1488	1510	1568	**1572**
SJMN91 51-100	MAP	0.184	0.190	0.199	**0.208**
	Recall	1348	1350	1427	**1472**

5. Text Classification

We compare six text classifiers. The first four are within the framework of naïve bayesian (NB) [11], but use different class model smoothing techniques. They are Laplacian smoothing [11], background smoothing [8] [18], CISS, and CSSS, respectively. The other two classifiers are an active learning classifier [12] and a SVM classifier [9]. NB has several variants regarding the implementation of class models. We choose multinomial mixture model in this paper because it has proved to be most effective for text classification [11]. The Laplacian smoothing simply adds one count to all features. The background smoothing interpolates a unigram class model with the collection background model, controlled by the parameter β as shown in equation (5.1).

$$p_b(w|c_i) = (1-\beta)p_{ml}(w|c_i) + \beta p(w|D) \qquad (5.1)$$

The equation (5.2) describes the formula for CSSS and CISS. It combines a simple class model $p_b(w|c_i)$ with a semantic mapping model controlled by the translation coefficient (λ).

$$p_s(w|c_i) = (1-\lambda)p_b(w|c_i) + \lambda\sum_k p(w|t_k)p(t_k|c_i) \quad (5.2)$$

The comparative experiments are conducted on three collections, 20-Newsgroups (20NG), Los Angeles Times (LATimes), and OHSUMED. 20NG is collected from twenty different Usenet newsgroups and the data are relatively noise. LATimes contains news articles and we select top 10 categories as described in [24]. OHSUMED consists of scientific abstracts collected from Medline and top 14 categories are selected for the experiment.

All classifiers are well tuned. The parameter β in the background smoothing is set to 0.5 because the classifier achieved the best results around this setting. The best configuration for SVM classifier uses a linear kernel, one-versus-all (OVA) code matrix as well as loss-based multi-class decoder (hinge loss function is used) [1]. For CISS and CSSS, the translation coefficient λ is set to 0.4 and 0.1 for 1% training and 33% training, respectively. Micro-F1 and macro-F1 [17] are used to evaluate the performance of text classifiers. All reported results are the average of ten random runs.

Table 2: Comparisons of all classifiers. 1% of documents are used for training and the remaining 99% for testing.

(a) The result of micro-F1

Collection	SVM	AL	Lap	Bkg	CISS	CSSS
OHSUMED	0.351	0.368	0.352	0.372	0.401	**0.413**
20NG	0.472	0.575	0.427	0.526	**0.623**	0.613
LATimes	0.524	0.566	0.525	0.538	0.577	**0.581**

(b) The result of macro-F1

Collection	SVM	AL	Lap	Bkg	CISS	CSSS
OHSUMED	0.206	0.205	0.205	0.280	0.344	**0.362**
20NG	0.464	0.551	0.421	0.523	**0.616**	0.613
LATimes	0.491	0.536	0.492	0.513	0.549	**0.562**

Table 3: Comparisons of all classifiers. 33% of documents are used for training and the remaining 67% for testing.

(a) The result of micro-F1

Collection	SVM	Lap	Bkg	CISS	CSSS
OHSUMED	**0.680**	0.660	0.667	0.663	0.665
20NG	0.797	0.771	0.802	0.801	**0.820**
LATimes	**0.781**	0.728	0.726	0.724	0.729

(b) The result of macro-F1

Collection	SVM	Lap	Bkg	CISS	CSSS
OHSUMED	0.646	0.626	0.639	0.636	**0.669**
20NG	0.793	0.756	0.787	0.786	**0.820**
LATimes	**0.765**	0.708	0.696	0.693	0.719

In the case of 1% training data, CSSS and CISS significantly outperform Laplacian smoothing and background smoothing. However, in the case of 33% training data, the gap among three smoothing approaches becomes very small. This verifies our hypothesis that semantic smoothing is more effective than Laplacian smoothing and background smoothing for bayesian text classifiers when the number of training documents is small (i.e., the data are sparse).

Taking a closer look at the results, we also find out that Macro-F1 receives more gain than Micro-F1 after

applying semantic smoothing onto highly skewed data such as LATimes and OHSUMED. The result of Micro-F1 is dominated by the performance of some common categories. However, for the metric of Macro-F1, the performance of each category is treated equally regardless the size of the category. This means semantic smoothing is especially effective for small categories. It is reasonable because small categories contain too few training examples and the data sparsity is very serious.

With respect to the effectiveness of CSSS and CISS, the former is slightly better than the latter because the semantic mapping of the former is more specific and accurate. Active learning has been proved to be effective in the case of small training data [12]. Our experiments repeat this finding in the sense that it outperforms NB with Lap. But it is less effective than both CISS and CSSS. Previous empirical studies have shown that SVM using linear kernel outperforms many other text classifiers including NB [17]. In the case of 33 training data, we do find out SVM performs much better than semantic smoothing on OHSUMED and LATimes. But when the training data decrease to 1%, its performance is at the same level of NB and much less effective than semantic smoothing. It is mostly likely due to the fact that a large number of features are blind to SVM when training document set is very small and the power of SVM is compromised while a bayesian classifier can expand meaningful features through semantic smoothing.

6. Text Clustering

We compare six clustering approaches. The first two are the variants of spherical k-means [6]. One uses normalized term frequency score and the other TF.IDF score. Spherical kmeans is considered one of the most effective clustering approaches to text clustering [6]. The remaining four are based on generative model-based k-means [21]. The difference between spherical k-means and model-based k-means lies in the mechanism of document assignment in iteration. The former uses cosine similarity while the latter employs a bayesian classifier. Since bayesian classifiers can take various smoothing methods, model-based k-means also have four variants in this paper. They are Laplacian smoothing, background smoothing, CISS and CSSS.

The testing collections for clustering are the same as for the classification experiment. To mimic the data sparsity problem, we also create five small data sets for each collection. We randomly pick up 100 documents per category and then merge them into a big pool to cluster. Since k-means are sensitive to the initialization, we execute ten runs with random initialization for each dataset and average the results to report. The normalized mutual information (NMI) [2] is used to measure the clustering quality. NMI is a score ranging from 0 to 1. The bigger the score, the better quality the clustering result is.

Table 4: The NMI results of six clustering approaches. The first two are variants of spherical k-means and the last four are model-based k-means with four different smoothing approaches

(a) Small dataset, 100 documents per class

Collection	NTF	TF-IDF	Lap	Bkg	CISS	CSSS
OHSUMED	0.090	0.172	0.080	0.090	**0.227**	0.212
20NG	0.176	0.391	0.240	0.201	**0.476**	0.441
LATimes	0.200	0.185	0.145	0.122	**0.332**	0.322

(b) Large dataset, all documents are used for clustering

Collection	NTF	TF-IDF	Lap	Bkg	CISS	CSSS
OHSUMED	0.085	0.232	0.180	0.165	0.238	**0.239**
20NG	0.192	0.506	0.493	0.489	**0.571**	0.564
LATimes	0.201	0.349	0.382	0.371	0.395	**0.420**

When the dataset to cluster is small, model-based k-means with semantic smoothing (both CSSS and CISS) not only outperform model-based k-means with Laplacian smoothing and background smoothing, but also beats the spherical k-means. This finding is very similar to the one we obtain from the classification experiment. However, clustering with semantic smoothing presents some new features we do not see in the setting of text retrieval and classification. First, no matter the dataset to cluster are small or large, CSSS and CISS always have the best result. Second, semantic smoothing achieves the best result when the translation coefficient is close to one. In the setting of retrieval and classification, the optimal translation coefficient is around 0.3~0.4. Third, CISS performs better than CSSS on small datasets and are comparable to CSSS on large datasets. In the setting of retrieval and classification, CSSS are slightly better than CISS.

A plausible explanation is that semantic smoothing well solves the overfitting problem of k-means. With Lap or Bkg smoothing, documents tend to group into a few large clusters and quickly converge to local maxima. With semantic smoothing, small clusters still have great chance to grow up because small clusters share many significant common words with other documents through semantic mapping. Our experiments do show that model-based k-means with semantic smoothing takes more iterations to converge than the other two smoothing approaches. With such an explanation, it is not difficult to understand three new findings. The optimal translation coefficient is always around one because this maximizes its capability of helping small intermediate clusters jump out of local maxima. Semantic smoothing still works very well on large dataset because large dataset generate small intermediate clusters during first several iterations too. CISS performs more effectively than CSSS on small dataset because the number of words is much larger than

that of context-sensitive topic signatures and hence more powerful to expand features.

7. Conclusions

In this paper, we introduced a new toolkit referred to as dragon toolkit which can utilize auto-learned semantic knowledge for large-scale text retrieval and mining within a formal language modeling framework. The core idea underlying this tool is to identify explicit topic signatures in documents and then statistically map them onto single-word features, i.e. semantic smoothing of unigram language models.

We demonstrated the effectiveness of semantic smoothing on three tasks (text retrieval, classification and clustering). Semantic smoothing performed significantly better than two state-of-the-art retrieval models, Okapi model and two-stage language model. In the setting of classification, a NB classifier with semantic smoothing not only outperformed NB classifiers with Lap and Bkg, but also beat the SVM classifier and the active learning classifier, when the size of training documents is small. On clustering tasks, model-based k-means with semantic smoothing always beat the ones with Lap and Bkg as well as spherical k-means, no matter the dataset to cluster are small or large. CSSS performed slightly better than CISS for retrieval and classification, but slightly worse for clustering where the overfitting problem dominated the results. However, in terms of efficiency, CISS is always worse CSSS because the number of unique words is often much more than the number of unique context sensitive topic signatures such as ontological concepts.

8. 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] Allwein, E.L., Schapire, R.E., and Singer, Y., "Reducing multiclass to binary: A unifying approach for margin classifiers," *Journal of Machine Learning Research*, 1:113–141, 2000.

[2] 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.

[3] Berger, A. and Lafferty J. "Information Retrieval as Statistical Translation," *In ACM SIGIR*, 1999, pp.222-229.

[4] Blei, D., Ng, A. and Jordan, M., "Latent Dirichlet allocation," *Journal of machine Learning Research*, 3, 2003, pp 993-1022.

[5] Cai, L. and Hofmann, T., "Text Categorization by Boosting automatically Extracted Concepts," In *SIGIR 2003*, pp. 182-189

[6] Dhillon, I.S. and Modha, D.S., "Concept Decompositions for Large Sparse Text Data Using Clustering," *Machine Learning*, 42(1):143-175, 2001

[7] Hoffman, T., "Probabilistic latent semantic indexing," *SIGIR'99*, 1999, pp. 50-57

[8] Jelinek, F., "Self-Organized Language Modeling for Speech Recognition," WeiBel A and Lee K-F, Eds., *Readings in Speech Recognition*, Morgan Kaufmann, Los Altos, CA, 1990, pp. 450-505.

[9] Joachims, T., "Text categorization with support vector machines: Learning with many relevant features," In *Proceedings of European Conference on Machine Learning*, pages 137-142, 1998.

[10] Liu, S., Yu, C., and Meng, W., "Word sense disambiguation in queries," *In CIKM 2005*, pp 525 - 532

[11] McCallum, A. and Nigam, K. "A comparison of event models for naive Bayes text classification," *AAAI Workshop on Learning for Text Categorization*, 1998, pp 41-48.

[12] Nigam, K., McCallum, A., Thrun, S., Mitchell, T. "Text Classification from Labeled and Unlabeled Documents using EM," *Machine Learning*, Volume 39, Issue 2-3 (May-June 2000), pp103-134

[13] Robertson, S.E. et al. "Okapi at TREC-4", *In the Fourth Text Retrieval Conference*, 1993.

[14] Sanderson, M., "Word sense disambiguation and information retrieval," *ACM SIGIR*, pp.142-151, 1994

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

[16] Wei, X. and Croft, W.B., "LDA-based document models for ad-hoc retrieval," In *SIGIR 2006*, pp. 178-185

[17] Yang, Y. and Liu, X., "A re-examination of text categorization methods," *ACM SIGIR*, pp 42--49, 1999.

[18] Zhai, C. and Lafferty, J., "A Study of Smoothing Methods for Language Models Applied to Ad hoc Information Retrieval", In *SIGIR*, 2001, pp.334-342.

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

[20] Zhang, X., Jing, L., Hu, X., Ng, M., and Zhou, X., "A Comparative Study of Ontology Based Term Similarity Measures on Document Clustering," *DASFFA2007*

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

[22] Zhou, X., Hu, X., Zhang, X., Lin, X., and Song, I.-Y., "Context-sensitive Semantic Smoothing for Language Modeling Approach to Genomic Information Retrieval," In *ACM SIGIR 2006*, pp. 170-177.

[23] Zhou, X., Zhang, X., and Hu, X., "MaxMatcher: Biological Concept Extraction Using Approximate Dictionary Lookup," In *PRICAI 2006*, Aug 9-11, 2006, Guilin, Guangxi, China, Page 1145-1149

[24] Zhou, X., Zhang, X., and Hu, X., "Semantic Smoothing of Document Models for Agglomerative Clustering," *In IJCAI 2007*, Jan. 6-12, 2007, India, pp. 2922-2927

[25] Zhou, X., Zhang, X., and Hu, X., The Dragon Toolkit, http://www.dragontoolkit.org