

Abstract

Problem: It is often expensive to manually tag the class label of a document, which directly causes the lack of training documents for text classification. Without sufficient training data, the Bayesian text classifier has serious data sparsity problem. The frequently used Laplacian smoothing and background smoothing approaches to data sparsity are not very effective, however.

Solution: Why does human fully understand the meaning of a short document or passage? Human knows the semantics of each word and phrase in the document and thus the size of the document is not a matter. Inspired by this general idea and the statistical translation language model [1] used in the setting of information retrieval, we proposed the semantic smoothing (i.e., based on the semantics of words, phrases, or concepts in the document) for Bayesian text classification. Different from parallel corpora training in [1], we simply use co-occurrence data to learn translation probabilities. The cost of collecting co-occurrence data is much cheaper than the parallel corpora.

Result: The new classifier with semantic smoothing was tested on four collections, 20NG, TDT2, LA Times, and OHSUMED (see Table 1). In the case of 5% training data, the semantic smoothing significantly outperforms two other approaches on all four collections (see Table 2). The experiment on 20NG (see Figure 3) showed that the smaller the training dataset, the more advantage the semantic smoothing takes. The F1 score was improved from 12% to 32% in the case of 1 training doc per class. The classification result based on phrase-word translation is slightly better than the word-word translation (see Table 3) though the latter is unable to incorporate contextual information, but the former is much more efficient. The learned semantic knowledge is also reusable (see the last row of Table 2 and 3)

Background: Bayesian Classifier & Smoothing

$$C(d) = \underset{c_i}{\operatorname{argmax}} p(c_i) p(d | c_i)$$

The first term is the class prior which can be computed by the formula below:

$$p(c_i) = \frac{1 + N(c_i, D)}{|C| + |D|}$$

With independence assumption, the second term could be simplified to:

$$p(d | c_i) = \prod_{k=1}^{|d|} p(w_{d_i, k} | c_i)$$

Smoothing Techniques for Bayesian Classifiers

Now the problem is reduced to the computation of $p(w|c_i)$. Due to the data sparsity, the terms in the testing document may not appear in the training documents and technically cause the zero-probability problem.

Laplacian Smoothing: add one count to all terms

$$p(w | c_j) = \frac{1 + c(w, c_j)}{|V| + \sum_w c(w, c_j)}$$

Background Smoothing: linearly interpolate the unigram language model with a background collection model

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

Semantic Smoothing

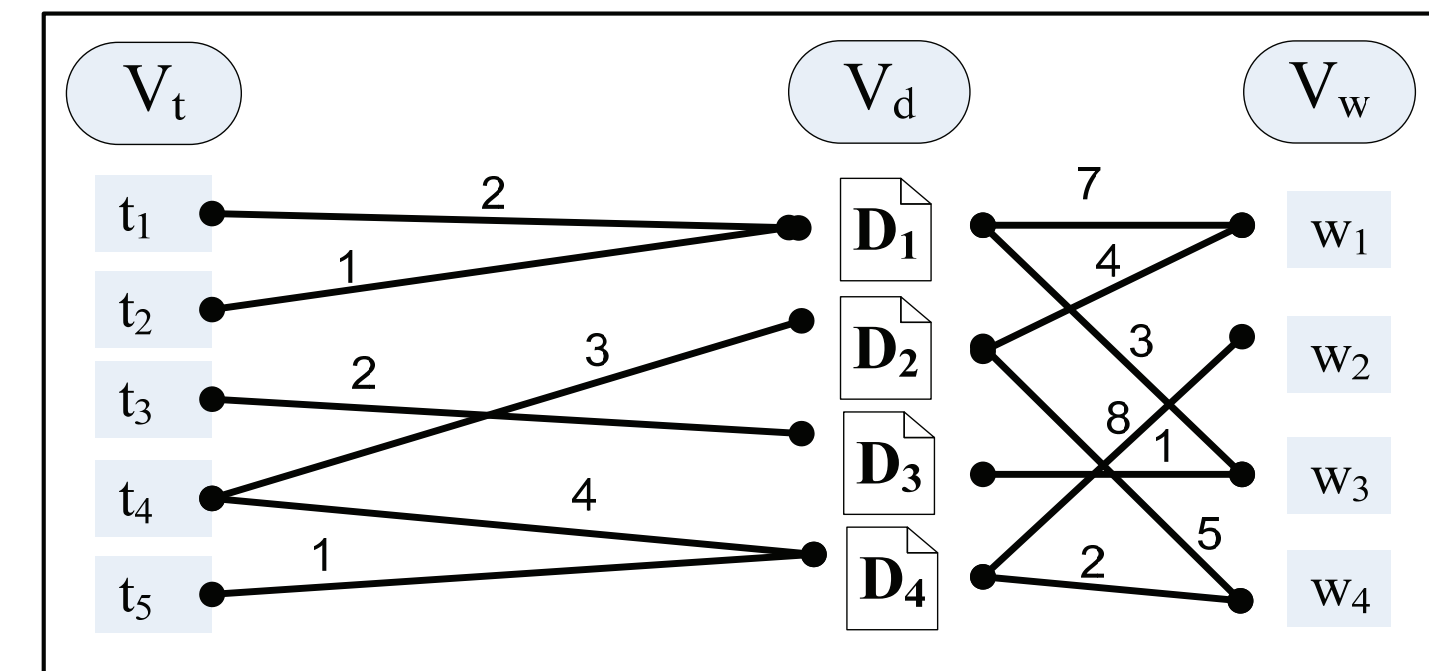


Fig. 1 The illustration of document indexing. V_t , V_d , and V_w are topic signature set (multiword phrases, concepts, etc.), document set and word set, respectively. The number on each line denotes the frequency of topic signatures or terms the doc.

Translation Probability Estimates

All terms in the document set are either translated by the given topic signature model or generated by the background collection model, we have:

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

The probability of translating a topic signature t_k to a concept w can be estimated by the Expectation Maximum (EM) algorithm with the following update formulas:

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

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

Notes: D_k is the set of documents containing phrase t_k ; $c(w, D_k)$ is the frequency count of word w in D_k ; A is the background noise coefficient; C denotes the background model.

Translation probability of phrase "space program" to words

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;

Fig. 2 The demonstration of translations (only top 20 topical terms are listed). All three translation models are trained on the 20-newsgroup corpus. The result of the phrase looks more specific than single words probably because the phrase itself contains contextual information.

Bayesian Classifier with Semantic Smoothing

Context-Sensitive Semantic Smoothing (CSSS): based on phrase (concept)-word Translation

$$p(w | c_j) = (1 - \lambda) p_b(w | c_j) + \lambda p_i(w | c_j)$$

Context-Insensitive Semantic Smoothing (CISS): based on word-word Translation

$$p_i(w | c_j) = \sum_k p(w | t_k) p(t_k | c_j)$$

Notes: λ is called translation coefficient which controls the influence of the translation component in the mixture model and P_b is the simple language model with background smoothing.

Evaluation

Table 1: The descriptions and basic statistics of the four collections for evaluation. The first three collections use automated multiword phrases as topic signatures while the fourth uses UMLS concepts.

Dataset Name	20NG	TDT2	LATimes	OHSUMED
Domain	Online Com.	News	News	Biomedicine
# of classes	20	10	10	14
# of indexed docs	19,997	7,094	21,623	7,400
# of phrases in corpus	10,902	8,256	10,414	28,857
# of phrases per doc	9	21	8	61
# of unique phrases per doc	7	17	7	33
# of words in corpus	140,277	37,994	63,510	27,676
# of words per doc	157	221	99	116
# of unique words per doc	91	138	75	69

Table 2: The comparisons among three smoothing approaches on all four collections. Lap, Bkg, and CSSS stand for Laplacian smoothing, background smoothing, and context sensitive semantic smoothing. * denotes that the translation probabilities are learned from the full TDT2 corpus. 5% dataset is used for training.

Collection	Micro-F1					Macro-F1				
	Lap	Bkg	CSSS	Vs. Lap	Vs. Bkg	Lap	Bkg	CSSS	Vs. Lap	Vs. Bkg
OHSUMED	0.352	0.372	0.413	17.3%	10.9%	0.205	0.280	0.362	76.2%	29.1%
20NG	0.427	0.526	0.613	43.7%	16.6%	0.421	0.523	0.613	45.5%	17.2%
TDT2	0.926	0.934	0.944	2.0%	1.0%	0.851	0.903	0.937	10.2%	3.8%
LATimes	0.525	0.538	0.581	10.7%	7.9%	0.492	0.513	0.562	14.3%	9.5%
LATimes*	0.525	0.538	0.559	6.5%	3.8%	0.492	0.513	0.541	10.0%	5.4%

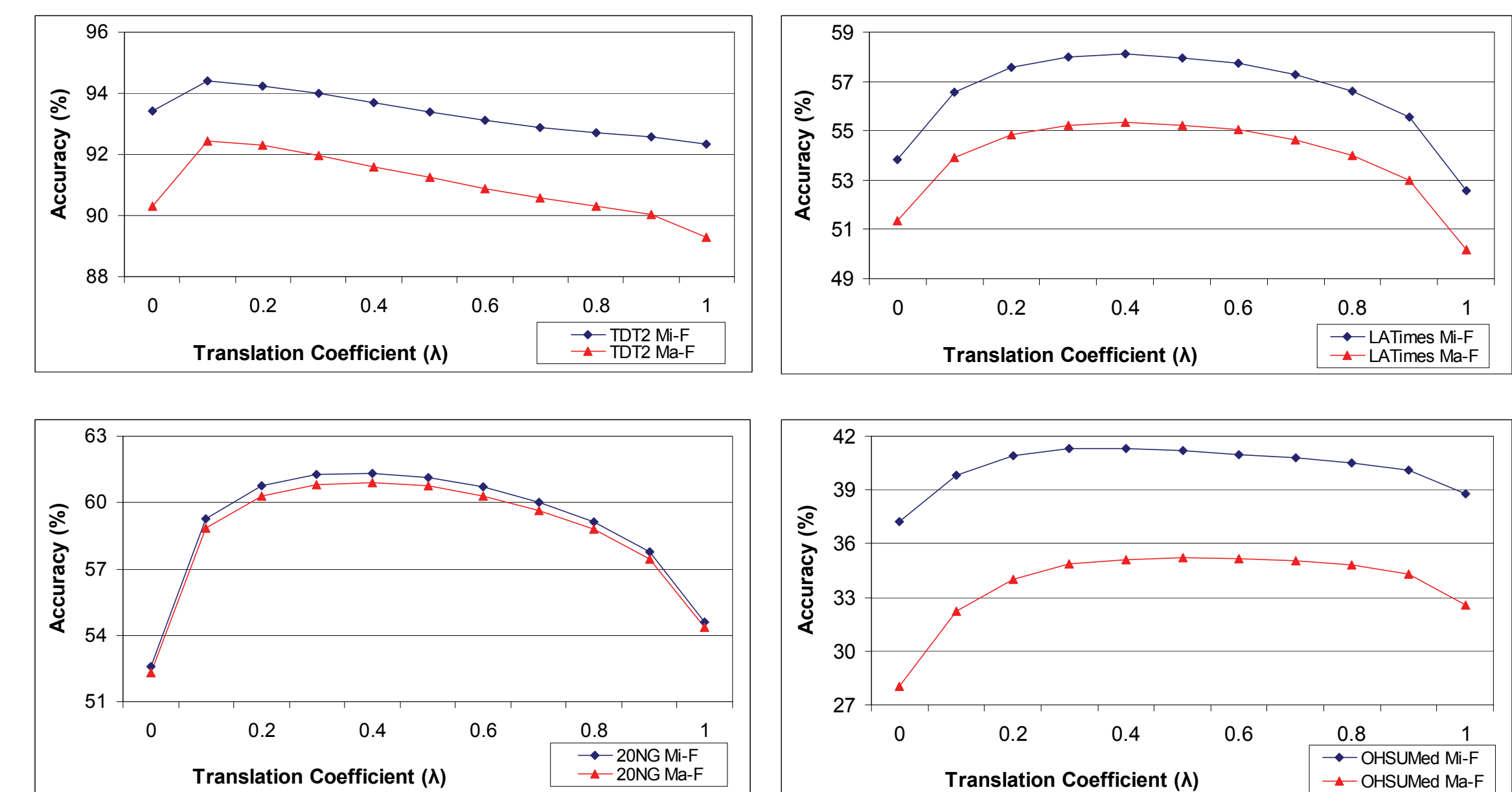


Fig. 4 Tuning of the translation coefficient. Except for the TDT2 corpus, the best results are achieved around the point of 0.4 and also the semantic smoothing significantly outperforms the background smoothing (i.e. $\lambda=0$) in a wide range (0.1 to 0.9 or even 1.0). This shows the robustness of the semantic smoothing approach.

Reuse Learned Semantic Knowledge

We applied translation probabilities learned from the full TDT2 corpus to the classification of LATimes news articles. Although phrase coverage is less than 50%, the classification on LA Times is still got improved over the Lap. And Bkg. See the last row of Table 2 and 3.

Context Sensitive vs. Context Insensitive

Task	Effectiveness	Efficiency
Extraction	N/A	CISS <i>much faster than</i> CSSS
Translation	CSSS <i>much better than</i> CISS	CSSS <i>much faster than</i> CISS
Classification	CSSS <i>slightly better than</i> CISS	CSSS <i>much faster than</i> CISS

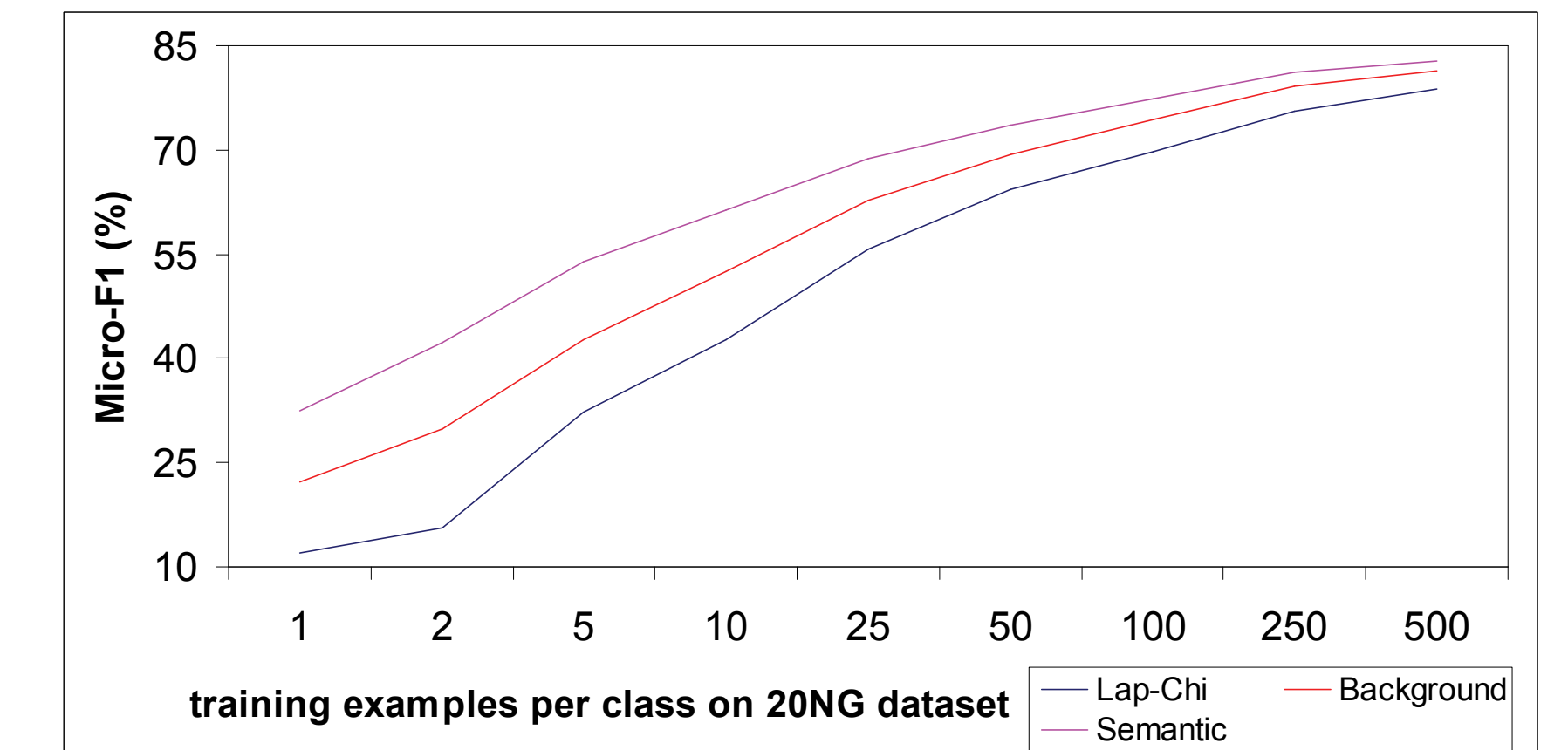


Fig. 3: The comparisons among three smoothing approaches on 20NG corpus when the training size ranges from 1 doc to 500 docs. The smaller the training dataset, the more advantages the semantic smoothing takes over the other two.

Table 3: The comparisons of context-sensitive semantic smoothing (CSSS) to context insensitive semantic smoothing. CSSS looks slightly better than CISS in terms of effectiveness.

Collection	Micro-F1			Macro-F1		
	CISS	CSSS	Change	CISS	CSSS	Change
OHSUMED	0.401	0.413	2.8%	0.344	0.362	5.3%
20NG	0.623	0.613	-1.6%	0.616	0.613	-0.5%
TDT2	0.942	0.944	0.2%	0.922	0.937	1.7%
LATimes	0.577	0.581	0.8%	0.549	0.562	2.5%
LATimes*	0.558	0.559	0.2%	0.529	0.541	2.4%

Findings

- ◆ Semantic smoothing significantly outperforms other smoothing approaches when the training dataset becomes smaller. This finding is of practical value because manual labeling is expensive.
- ◆ The learned semantic knowledge is reusable. It can be applied to the classification of documents in similar domains or collections.
- ◆ The translation results of topic signatures make more sense than those of single words because the former contains contextual information.
- ◆ The Bayesian classifier based on context sensitive semantic smoothing is slightly better than context insensitive semantic smoothing probably because too few number of topic signatures are extracted (see Table 1)

References

- [1] Berger, A. and Lafferty J., "Information Retrieval as Statistical Translation," *SIGIR '99*, 222-229.
- [2] Smadja, F., "Retrieving collocations from text: Xtract," *Computational Linguistics*, 1993, 19(1), 143-177.
- [3] Zhou X., Hu X., Zhang X., Lin X., Song I-Y., "Context-Sensitive Semantic Smoothing for the Language Modeling Approach to Genomic IR," *SIGIR 2006*, 70-77
- [4] Zhou, X., Zhang, X., and Xiaohua Hu, "Semantic Smoothing of Document Models for Agglomerative Clustering," *IJCAI 2007*, 2928-2933