Text Categorization Using Unlabeled Data and its Theory

Ko Youngjoong (*yjko@dau.ac.kr*)

Dept. of Computer Engineering
Dong-A University

http://web.dong.ac.kr/yjko/

Warming Up!!

- Pattern classification (Duda & Hart)

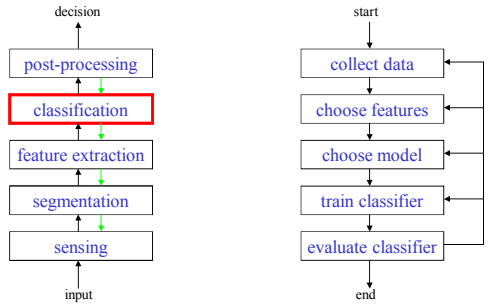


Fig1. The process of the pattern classification system Fig2. The design cycle of the pattern classification system

The Theory of Text Categorization

Ko Youngjoong (*yjko@dau.ac.kr*)

Dept. of Computer Engineering
Dong-A University

Contents

- A definition of the text categorization (TC) task
- The machine learning approach to text categorization
- Indexing and dimensionality reduction
- Methods for constructing classifiers
- Evaluation issues for text categorization

A Definition of the TC Task

- Text Categorization (Sebastiani, 2002)
  - Assign documents to one or more of a predefined set of categories
  - The task of automatically determining an assignment of a value from {0,1} to each entry of the decision matrix.

	d_1	…	…	d_j	…	…	d_n
c_1	a_{11}	…	…	a_{1j}	…	…	a_{1n}
…	…	…	…	…	…	…	…
c_i	a_{i1}	…	…	a_{ij}	…	…	a_{in}
…	…	…	…	…	…	…	…
c_m	a_{m1}	…	…	a_{mj}	…	…	a_{mn}

  - where
    - $C = \{c_1,\ldots,c_m\}$ is a set of pre-defined categories
    - $D = \{d_1,\ldots,d_n\}$ is a set of documents to be categorized
    - A classifier for c_i is a function $f_i' : D \rightarrow \{0,1\}$ that approximates an unknown function $f_i : D \rightarrow \{0,1\}$

A Definition of the TC Task

- Different constraints depending on the application
  - Single-label case : exactly one category must be assigned to each document
  - Multi-label case : general case
- Category and document-pivoted categorization
  - CPC (category-pivoted categorization) : one row at a time
    - a new category may be added to a set of categories after a number of documents have already been categorized under the set of categories
  - DPC (document-pivoted categorization) : one column at a time
    - A user submits one document at a time for categorization
    - The categories may be ranked in decreasing order of estimated appropriateness for the document

The Machine Learning Approaches for TC

- In the 80's, the typical approach is a hand-crafting *expert system* which uses a set of rules of type
  - **If** <*conjunction of terms*> **then** <*category*>
    - bushels & expert → wheat
  - The drawback of this "manual" approach
    - *Knowledge acquisition bottleneck*
- In the 90's, the machine learning approach appears
  - A general inductive process automatically builds a classifier for a category
  - Advantages of this approach
    - construction not of a classifier, but of an automatic builder of classifiers (learner)
    - The effectiveness of these classifiers matches that of hand-crafted classifiers

Training Set and Test Set

- A correct decision matrix

	Training Set				Test Set			
	d_1	…	…	d_g	d_{g+1}	…	…	d_s
c_1	b_{11}	…	…	b_{1g}	$b_{1(g+1)}$	…	…	b_{1s}
…	…	…	…	…	…	…	…	…
c_m	b_{m1}	…	…	b_{mg}	$b_{m(g+1)}$	…	…	b_{ms}

  - A *positive example* of c_i if $b_{ij} = 1$
  - A *negative example* of c_i if $b_{ij} = 0$
- A validation set
  - Use for optimizing its internal parameters
  - A training set may be split into a true training set and a validation set

Indexing and Dimensionality Reduction

- The choice of a text representation
  - ***Lexical semantics***
  - Compositional semantics
- The *bag of words* approach
  - The vector of a document: n weighted terms (or *features*) t_k that occur in d_j.
  - Weight (w_{kj})
    - [0,1]: the most frequent case
    - {0,1}: presence or absence of t_k in d_j
- Lewis have found that more sophisticated representations (linguistic phrases, statistical phrases, etc) yield worse effectiveness

TFIDF Term Weighting Scheme

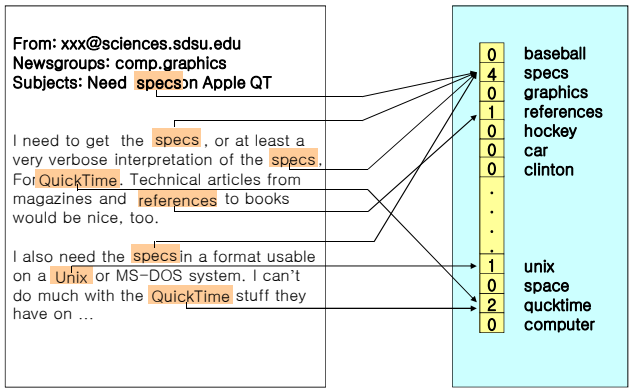
- TFIDF term weight

$$tfidf(t_k, d_j) = \#(t_k, d_j) \cdot \log \frac{|Tr|}{\#(t_k)}$$

- Cosine Normalization
  - The weights resulting from *tfidf*, so as to account for document length

$$w_{kj} = \frac{tfidf(t_k, d_j)}{\sqrt{\sum_{s=1}^{r} (tfidf(t_s, d_j))^2}}$$

The Indexing Process



Dimensionality Reduction (DR)

- Why?
  - Sophisticated learning algorithms for TC do not scale well to high values of r
  - DR reduces *overfitting*
- Two ways of viewing DR
  - **Local DR** : for one category
  - **Global DR** : for all categories
- The second distinction
  - **DR by feature selection** : the chosen r' terms are a subset of the original terms r
  - **DR by feature extraction** : the r terms are not a subset of the original r terms. They are usually obtained by combinations or transformations of the original ones.

Feature Selection Functions

Function	Denoted by	Mathematical Form
Document frequency	$\#(t_k, c_i)$	$P(t_k \mid c_i)$
Mutual information	$MI(t_k, c_i)$	$\log \frac{P(t_k, c_i)}{P(t_k) \cdot P(c_i)}$
Information Gain	$IG(t_k, c_i)$	$P(t_k, c_i) \cdot \log(\frac{P(t_k, c_i)}{P(t_k) \cdot P(c_i)}) + P(\bar{t}_k, c_i) \cdot \log(\frac{P(\bar{t}_k, c_i)}{P(\bar{t}_k) \cdot P(c_i)})$
Chi-square	$\chi^2(t_k, c_i)$	$\frac{g \cdot [P(t_k, c_i) \cdot P(\bar{t}_k, \bar{c}_i) - P(t_k, \bar{c}_i) \cdot P(\bar{t}_k, c_i)]^2}{P(t_k) \cdot P(\bar{t}_k) \cdot P(c_i) \cdot P(\bar{c}_i)}$
Correlation coefficient	$CC(t_k, c_i)$	$\frac{\sqrt{g} \cdot [P(t_k, c_i) \cdot P(\bar{t}_k, \bar{c}_i) - P(t_k, \bar{c}_i) \cdot P(\bar{t}_k, c_i)]}{\sqrt{P(t_k) \cdot P(\bar{t}_k) \cdot P(c_i) \cdot P(\bar{c}_i)}}$
Odds Ratio	$OR(t_k, c_i)$	$\frac{P(t_k \mid c_i) \cdot P(\bar{t}_k \mid \bar{c}_i)}{P(\bar{t}_k \mid c_i) \cdot P(t_k \mid \bar{c}_i)}$

Global DR & Feature Extraction

- The forms for global DR
  - Sum : $f_{sum}(t_k) = \sum_{i=1}^{m} f(t_k, c_i)$
  - Weighted average : $f_{avg}(t_k) = \sum_{i=1}^{m} P(c_i) f(t_k, c_i)$
  - Maximum : $f_{\max}(t_k) = \max_{i=1}^{m} f(t_k, c_i)$
- The best among such measures
  - {OR, CC} > {χ^2, IG} > {#, MI}
- Two approaches of feature extraction
  - *Term clustering*
  - *Latent semantic indexing* : singular value decomposition

Probabilistic Classifiers

- The *Categorization Status Value (CSV)* function
  - CSV_i : D $\rightarrow$ [0,1] (given d_j, for category c_i)
  - The definition of a *threshold* τ_i
    - $CSV_i(d_j) \geq \tau_i$: *a decision to categorize* d_j *under* c_i
- Naïve Bayes classifiers (McCallum & Nigam, 1998)
  - View $CSV_i(d_j)$ in terms of Bayes' theorem

$$P(c_i \mid d_j) = \frac{P(c_i) P(d_j \mid c_i)}{P(d_j)}$$

  - Use of the independence assumption for $P(d_j|c_i)$

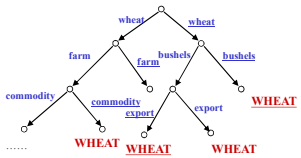
$$P(d_j \mid c_i) = \prod_{k=1}^{r} P(w_{kj} \mid c_i)$$

Neural Networks

- (Wiener, Pedersen, and Weigend, 1995)
- A *neural network (NN)* TC system is a network of units
  - *Input units* : terms appearing in the document
  - *Output units* : categories to be assigned
- NNs are trained by backpropagation

Decision Tree Classifiers

- Build a binary Tree (Lewis and Ringuette, 1994)
  - Internal nodes : labeled by index terms
  - Branches : the value that the index term has in the representation of the test document
  - Leaf nodes : labeled by categories



The Rocchio Classifier

- An adaptation to TC of Rocchio's formula for relevance feedback
  - To compute a profile for c_i by means of the formula

$$w_{ki} = \beta \cdot \sum_{\{\vec{d}_j | b_{ij}=1\}} \frac{w_{kj}}{|\{\vec{d}_j \mid b_{ij}=1\}|} - \gamma \cdot \sum_{\{\vec{d}_j | b_{ij}=0\}} \frac{w_{kj}}{|\{\vec{d}_j \mid b_{ij}=0\}|}$$

  - Typical choices for the control parameters β and γ
    - $\beta = 16$ and $\gamma = 4$, $\beta = 1$ and $\gamma = 0$
  - Advantages
    - the learner is easy to implement
    - Quite efficient
    - Easily interpretable
  - Drawbacks
    - *Seldom very effective, categories are not linearly separable*

Example-based Classifiers

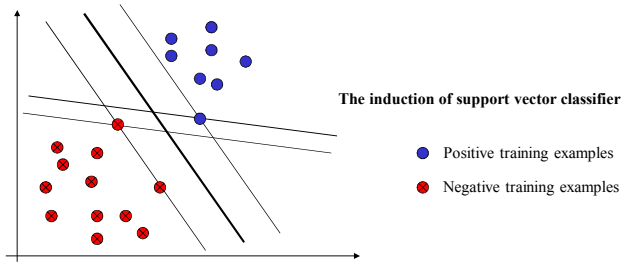
- The distance weighted *k*-NN (Yang, 94)

$$CSV_i(d_j) = \sum_{\vec{d}_z \in Tr_k(d_j)} RSV(d_j, \vec{d}_z) \cdot b_{iz}$$

  - $RSV(d_j, \vec{d}_z)$: a measure or semantic relatedness between d_j and $\vec{d}_z$
    - Ex) vector-based measures : inner-product, cosine similarity
  - The b_{iz} values are from the correct decision matrix of {0,1}
  - $Tr_k(d_j)$ is the set of the k documents $\vec{d}_z$ for which $RSV(d_j, \vec{d}_z)$ is maximum : the k value should be determined on a validation set
- Advantages
  - *High performance*, Not suffer from the "linear separation problem"
- Drawbacks
  - *Too late running time*, lazy learners.

SVM

- The support vector machine (Joachims, 1998)
  - To find *the surface* σ that separate the positive from the negative training examples in the *best* possible way
  - Structural risk minimization principle



Boosting

- The *Boosting* Method for the Classifier Committees
  - By the same learning method (*weak learner*)
  - Trained *sequentially*, one after the other.
    - The training of classifier Φ_t may take into account how classifiers Φ_1, …,Φ_{t-1} perform on the training examples, and concentrate on getting right those examples in which Φ_1, …,Φ_{t-1} have performed worst
  - The ADABOOST algorithm (Schapire & Singer, 2000)
    - Weak learner : decision tree
    - Each $<\vec{d}_j, c_i>$ pair is attributed an importance weight h^t_{ij}
    - Φ_t is then applied to the training documents, and as a result weights h^t_{ij} are updated to yield h^{t+1}_{ij}
      - *Pairs correctly classified by Φ_t will have their weight decreased*
      - *Pairs misclassified by Φ_t will have their weight increased*

Evaluation Issues for TC

- The contingency table for c_i

Category c_i		expert judgments: YES	expert judgments: NO
classifier judgments	YES	TP_i	FP_i
	NO	FN_i	TN_i

  - Precision of c_i (Pr_i) : the *degree of soundness* of the classifier

$$\bar{\Pr}_i = \frac{TP_i}{TP_i + FP_i}$$

  - Recall of c_i (Re_i) : the *degree of completeness* of the classifier

$$\bar{\mathrm{Re}}_i = \frac{TP_i}{TP_i + FN_i}$$

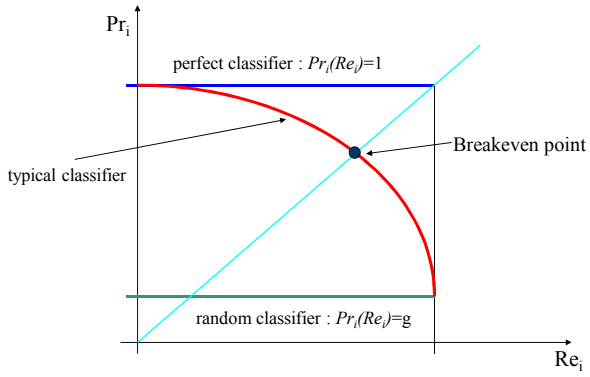
Combined Effectiveness Measures

- The inverse proportion relation between *Pr* and *Re*
  - To obtain 100% Re, one only needs to set every threshold τ_i to 0
- Various combined measures
  - (interpolated) 11-point average precision
    - Each τ_i is set to the values for which *Re* takes up values of 0.0, 0.1, …, 0.9, 1.0
    - *Pr* is computed for the 11 resulting values and averaged
  - F_β function
    - For some $0 \le \beta \le +\infty$

$$F_\beta = \frac{(\beta^2+1)\cdot \mathrm{Pr}\cdot \mathrm{Re}}{\beta^2 \cdot \mathrm{Pr} + \mathrm{Re}}$$

    - *When $\beta = 1$, F_β has equal importance of Pr and Re, called by* ***F_1 measure***
  - Breakeven point
    - The value at which at which *Pr* equals *Re*.
    - Breakeven is always less or equal than F_1 (Yang, 1999)

Combined Effectiveness Measures



Research Issues on Text Categorization

- The state-of-the-art classification systems
- *Unsupervised manner Text Categorization*
- Hypertext classification problems
- Hierarchical classification problems
- Etc.
  - Filtering (Ex. Email)
  - TDT (Topic Detection Tracking)

Text Categorization Using Unlabeled Data

Ko Youngjoong (*yjko@dau.ac.kr*)

Dept. of Computer Engineering
Dong-A University

Contents

Introduction (1)

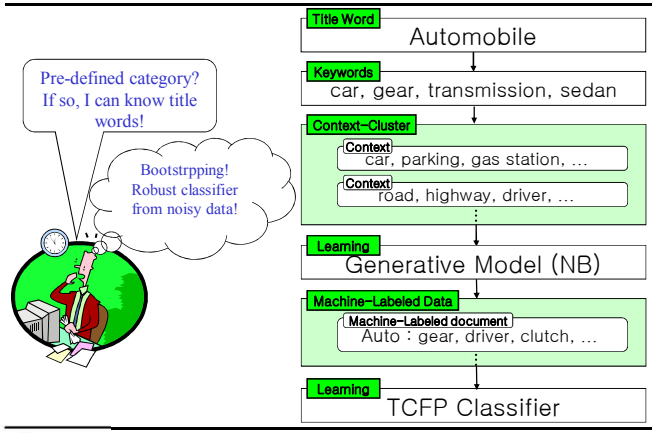
- Text Categorization (TC)
  - Classify documents into one (or several) of a set of *pre-defined* categories (topics of interest)
  - Prominent status in the information system field
    - Explosion of electronic texts from the WWW, E-mail, Digital library etc
  - Until the late ’80s
    - Manual construction of rule sets
    - High accuracy but significant cost
  - In the ’90s, the machine learning paradigm
    - *Supervised learning*
      - Find decision rule from an example set of labeled documents for each category
    - High accuracy and less expensive

Introduction (2)

- **Difficulties of supervised learning in TC**
  - Require large, often prohibitive, number of labeled training data
  - Various application areas: article, web pages, e-mail, and newsgroup, digital library, CRM, biomedical text etc

- ***Our proposal***
  - Automatically constructs labeled training data from unlabeled documents and the title word of each category
    - How can we automatically generate labeled training documents (*machine-labeled data*) from only title words
      - Bootstrapping Framework
    - How can we handle incorrectly labeled documents in the machine-labeled data.
      - TCFP Classifier

Overview



Constructing Context-Clusters for Training(1)

- Context
  - A unit of meaning in our method for bootstrapping
  - Part of a text that surrounds the particular word or a passage
  - Define a context as 60 words

- Creating Keyword list
  - Creating keyword list of each category
    - Keyword : words to be semantically related to a title word
  - Co-occurrence information
    - Cosine similarity

$$sim(T, X) = \frac{\sum_{i=1}^{n} t_i \times x_i}{\sqrt{\sum_{i=1}^{n} t_i^2 \times \sum_{i=1}^{n} x_i^2}}$$

Constructing Context-Clusters for Training(2)

- Extracting and verifying centroid-contexts
  - Centroid-context
    - Contain a keyword or a title word of a category
  - Importance score of centroid-context
    - TF-ICF
      - TF-ICF

$$w_{ij} = TF_{ij} \times ICF = TF_{ij} \times (\log(M) - \log(CF_i))$$

    - Importance Score

$$Score(CC_k, c_j) = \frac{w_{1j} + w_{2j} + \ldots + w_{Nj}}{N}$$

Constructing Context-Clusters for Training(3)

- Creating Contexts-Clusters
  - The goal
    - Assign remaining contexts to each category
  - The Assigning algorithm
    - Measuring similarity based on word & context similarity
      - By Karov & Edelman, 1998
  - Improve this algorithm for our method

Constructing Context-Clusters for Training(4)

- Affinity Formula

$$aff_n(W, C) = \max_{W_i \in C} sim_n(W, W_i)$$
$$aff_n(C, W) = \max_{W \in C_j} sim_n(C, C_j)$$

- Similarity Formulae

$$sim_{n+1}(C_1, C_2) = \sum_{W \in C_1} weight(W, C_1) \cdot aff_n(W, C_2)$$

$$\text{if } W_1 = W_2$$
$$sim_{n+1}(W_1, W_2) = 1$$
$$\text{else}$$
$$sim_{n+1}(W_1, W_2) = \sum_{W_i \in C} weight(C, W_1) \cdot aff_n(C, W_2)$$

Naive Bayes Classifier

- Naive Bayes with minor modification
  - Kullback-Leibler Divergence
  - Produce classification scores with less extreme

$$P(c_j \mid d_i; \hat{\theta}) = \frac{P(c_j \mid \hat{\theta}) P(d_i \mid c_j; \hat{\theta})}{P(d_i \mid \hat{\theta})} \approx P(c_j \mid \hat{\theta}) \prod_{t=1}^{|V|} P(w_t \mid c_j; \hat{\theta})^{N(w_t, d_i)}$$
$$\propto \frac{\log P(c_j; \hat{\theta})}{n} + \sum_{t=1}^{|V|} P(w_t \mid d_i; \hat{\theta}) \log\left(\frac{P(w_t \mid c_j; \hat{\theta})}{P(w_t \mid d_i; \hat{\theta})}\right)$$

- Laplace parameter estimation

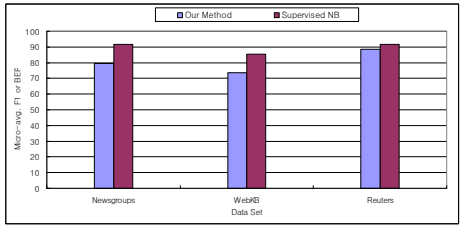
$$\hat{\theta}_{w_t \mid c_j} \equiv P(w_t \mid c_j; \hat{\theta}) = \frac{1 + N(w_t, G_{c_j})}{|V| + \sum_{i=1}^{|V|} N(w_t, G_{c_j})} \qquad \hat{\theta}_{c_j} \equiv P(c_j \mid \hat{\theta}) = \frac{1 + |G_{c_j}|}{|C| + \sum_{c_i} |G_{c_i}|}$$

Empirical Results (1)

- Data sets
  - 3 different types : UseNet newsgroups, web pages, newswire articles
    - Newsgroups data set
    - WebKB data set
    - Reuters-21578 Test Collection
- Experimental setting
  - Five-Fold validation
    - Newsgroups data
    - WebKB
  - Feature Selection : χ^2 statistics
  - Performance Measure
    - Micro-average F1 measure : Newsgroups, WebKB
    - Precision-recall Breakeven Point : Reuters

Empirical Results (2)

- Comparing with supervised NB classifier



Data Set	*Our method*	*Supervised NB*	*Difference*
Newsgroups	79.36	91.72	-12.36
WebKB	73.63	85.29	-11.66
Reuters	88.62	91.64	-3.02

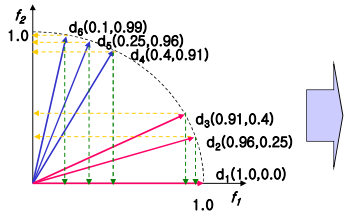
Learning with Machine-labeled Data

- Learning with Machine-labeled Data
  - Obtain finally labeled data of a document unit
  - We can learn supervised classifiers using them
  - A problem
    - Machine-labeled data has a lot of incorrectly labeled documents
    - Need a new robust classifier from noisy data
- ***TCFP*** classifier
  - A new type of text classifier using the feature projection technique
    - With robustness from noisy data
    - Fast execution speed
    - High performance
    - Simple algorithm: easily implement and quickly learn

A New Approach on Feature Projections

- An example of feature projections in Text Categorization

$d=(f_1, f_2)$, $c_1=\{d_1, d_2, d_3\}$, $c_2=\{d_4, d_5, d_6\}$



Document representation in a conventional vector space

w: weight, d: document, c: category

w, d, c
1.0, d_1, c_1
0.96, d_2, c_1
0.91, d_3, c_1
0.4, d_4, c_2
0.25, d_5, c_2
0.1, d_6, c_2

feature projections on feature f_1

w, d, c
0.99, d_6, c_2
0.96, d_5, c_2
0.91, d_4, c_2
0.4, d_3, c_1
0.25, d_2, c_1

feature projections on feature f_2

Feature representation in feature projections

A New Approach on Feature Projections

- A New Text Categorization Algorithm : *TCFP*

test document: $d=<t_1,t_2,...,t_n>$, category set: $C=\{c_1,c_2,...,c_m\}$

begin

for each category c_i
  vote[c_i] =0
for each feature t_i
  $tw(t_i)$ is calculated

for each feature t_i
  for each category c_j
    vote[c_j]=vote[c_j] + vs(c_j,t_m(i))

for each category c_i
  prediction = argmax vote[c_j]

return prediction
end

Empirical Evaluation (1)

- Comparison of TCFP with conventional text classifiers

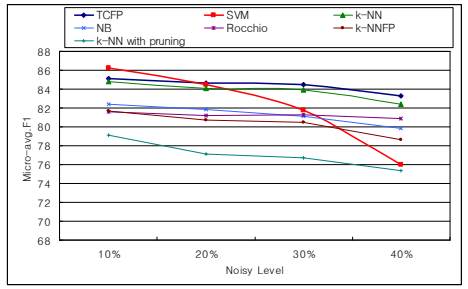
Data Set	TCFP	*k*-NN	SVM	NB	Rocchio
Newsgroups	***86.57***	85.92	***87.97***	82.79	82.37
WebKB	***88.07***	84.82	***91.75***	85.29	86.05
Reuters	***90.01***	88.93	***93.32***	88.62	86.47

- Running Time Observation
  - TCFP is about one hundred times faster classifier than *k*-NN

Data Set	TCFP without context	*k*-NNFP	Rocchio	TCFP	NB	SVM	*k*-NN with pruning	*k*-NN
Newsgroups	***0.68***	0.85	0.8	***1.25***	1.22	14.71	37.97	***142.54***
WebKB	***0.13***	0.23	0.14	***0.55***	0.17	2.72	4.91	***15.25***
Reuters	***2.65***	2.7	3.34	***2.89***	7.01	39.94	15.88	***65.86***

Empirical Evaluation (2)

- Robustness from Noisy Data
  - 4 data sets with from 10% to 40% noisy data in Newsgroups



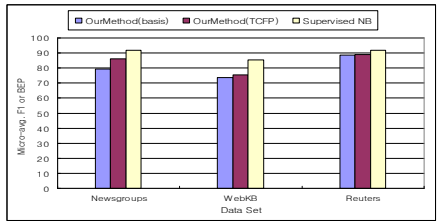
Empirical Evaluation (3)

- Results using machine-labeled documents data

Data Set	*OurMethod (basis)*	*OurMethod (NB)*	*OurMethod (Rocchio)*	*OurMethod (k-NN)*	*OurMethod (SVM)*	*OurMethod (TCFP)*	*Supervised NB*
Newsgroups	***79.36***	83.46	83	79.95	82.49	***86.19***	***91.72***
WebKB	***73.63***	73.22	75.28	68.04	73.74	***75.47***	***85.29***
Reuters	***88.62***	88.23	86.26	85.65	87.41	***89.09***	***91.64***

Empirical Evaluation (4)

- Final Results



Data Set	*OurMethod (basis)*	*OurMethod (TCFP)*	*basis vs. TCFP*	*Supervised NB*	*TCFP vs. Supervised NB*
Newsgroups	79.36	86.19	+6.83	91.72	-5.53
WebKB	73.63	75.49	+1.84	85.29	-9.82
Reuters	88.62	89.09	+0.47	91.64	-2.55

Conclusions

- We propose a new method for learning with only title words and unlabeled documents
- Contributions
  - A text classifier can be built from unlabeled data
  - TCFP classifier with robustness from noisy data, fast execution speed, and high performance
  - Our method is superior to clustering methods
  - Application Area
    - Required low-cost text categorization without labeling task
    - Creating training data
- Future Works
  - Improve the bootstrapping method from title words
  - Need more studies for voting ratio of the TCFP classifier

Publications

- **Y. Ko** and J. Seo, "Using Feature Projection Technique Based on a Normalization Voting Method for Text Classification," *Information Processing & Management, Pergamon-Elsevier Science, Vol.40, No.2, pp.191-208, 2004*.
- **Y. Ko**, J. Park, and J. Seo, "Improving Text Categorization Using the Importance of Sentences," *Information Processing & Management, Pergamon-Elsevier Science, Vol.40, No.1, pp.65-79, 2004*.
- H. Han, **Y. Ko** and J. Seo, "Improving Binary Text Classification Using the EM Algorithm", Proc. of AIRS-04, 2004, pp. 325-328.
- **Y. Ko** and J. Seo, "Learning with unlabeled Data for Text Categorization Using a Bootstrapping and a Feature Projection Technique ," *Proc. of ACL 2004*, pp. 255-262.
- **Y. Ko**, J. Seo, "Text Categorization Using Feature Projections," *Proc. of COLING 2002*, pp.467-473, 2002.
- **Y. Ko**, J. Park and J. Seo, "Automatic Text Categorization Using the Importance of Sentence," *Proc. of COLING 2002*, pp.474-480, 2002.
- **Y. Ko** and J. Seo, "Automatic Text Categorization by Unsupervised Learning," *Proc. of COLING*, pp.453-459, 2000.
- **Y. Ko**, S. Park, and J. Seo, "Web-based Requirements Elicitation Supporting System Using Requirements Categorization," *Proc. Of SEKE*, pp.344-351, 2000.

Cohen, W.W. and Singer Y., 1996, "Context-sensitive learning methods for text categorization," ACM Transactions on Information Systems, 17(2), 141-173.

Charkrabarti, S., Dom, B.E., and Indyk, P., 1998, "Enhanced hypertext categorization using hyperlinks," *Proceedings of SIGMOD-98*, pp.307-318.

Duda, R.O., Hart, P.E., and Stork, D.G., *Pattern Classification*, Second Edition,Wiley-interscience.

Hull, D.A., 1998, "The TREC-7 filtering track: description and analysis," *Proceedings of TREC-7*, 7th Text Retrieval Conference, pp. 33-56, National Institute of Standards and Technology.

Joachims, T., 1998, "Text categorization with support vector machines: learning with many relevant features," *Proceedings of ECML-98*, pp. 137-142.

Klimt, B. and Yang, Y., 2004, "The Enron Corpus: A New Dataset for Email Classification Research," *Proceedings of ECML* 2004.

Koller, D. and Sahami, M., 1997, "Hierarchically classifying documents using very few words," *Proceedings of ICML-97*, pp. 170-178.

Larkey, L.S. and Croft, W.B., 1996, "Combining classifiers in text categorization," *Proceedings of SIGIR-96*, pp. 289-297.

Lewis, D.D., 1992, *Representation and learning in information retrieval*, Ph.D. thesis. Dept. of Computer Science, University of Massachusetts, Amherst, US.

Lewis, D.D., Scharire, R.E., Callan, J.P., and Papka, R., 1996, "Training algorithms for linear text classifiers," *Proceedings of SIGIR-96*, pp, 298-306.

Lewis, D.D., and Ringuette, M., 1994, "A comparison of two learning algorithms for text categorization," *Proceedings of SDAIR-94*, pp. 81-94.

Li, Y.H. and Jain, A.K., 1998, "Classification of text documents," *The Computer Journal,* 41(8), pp. 537-546.

Linquillon, C., 2001, *Enhancing Text Classification to Improve Information Filtering*, Doctoral dissetation, Otto-von-Guericke-Universitat Magdeburg.

McCallum A. et al., 2000, "Automating the Construction of Internet Portals with Machine Learning." *Information Retrieval Journal*, 3, pages 127-163.

McCallum, A. and Nigam, K., 1998, "A Comparison of Event Models for Naïve Bayes Text Classifier, *Proceedings of AAAI-98*.

McCallum, A., Ronald, R., Mitchell, T., and Ng A., 1998, "Improving Text Classification by Shrinkage in a Hierarchy of Classes," *Proceedings of ICML-98*.

Mitchell, T.M., *Machine Learning*, WCB McGraw-Hill.

Nigam, K., 2001, *Using Unlabeled Data to Improve Text Classification*, Doctoral Dissertation.

Schapire, R.E. and Singer, Y., 2000, "BOOSTEXTER: a boosting-based system for text categorization," *Machine Learning*, 39(2/3), pp. 135-168.

Sebastiani, F., 2002, "Machine learning in automated text categorization," *ACM Computing Surveys*, 34(1), pp. 1-47, .

Slonim, N., Fredman, N., and Tishby, N., 2002, "Unsupervised Document Classification Using Sequential Information Maximization," *Proceedings of the SIGIR-02*, pp. 129-136.

Wiener, E., Pedersen, J.O., and Weigend, A.A., 1995, "A neural network approach to topic spotting," *Proceedings of SDAIR-95*, pp. 317-332.

Yang, Y., 1999, "An evaluation of statistical approaches to text categorization," *Information Retrieval 1*, 1-2, pp. 69-90.

Yang, Y., 1994, "Exper network: effective and efficient learning from human decisions in text categorization and retrieval," *Proceedings of SIGIR 94*, pp.13-22.

Yang, Y., 2001, "A study on thresholding strategies for text categorization," *Proceedings of SIGIR-01*, pp 137-145.

Yang, Y. and Pedersen, J.O., 1997, "A Comparative study on feature selection in text categorization," *Proceedings of ICML-97*, pp. 412-420.

Yang, Y. and Xin, L., 1999, "A re-examination of text categorization methods," *Proceedings of SIGIR'99*, pp. 42-49.