[Day-2] Unsupervised Learning for Natural Language Processing

(Part VI: Advanced Topic: Co-Training)

Keh-Yih Su and Jing-Shin Chang
kysu@bdc.com.tw, shin@nlp.csie.ncnu.edu.tw

(2002/8/18)

Behavior Design Corporation (BDC)
No. 5, 2F, Industrial East Road IV, Science-Based Industrial Park
Hsinchu, Taiwan 30077, R.O.C.

Department of Computer Science and Information Engineering
National Chi-Nan University
Puli, Nantou, Taiwan 545, R.O.C.

Day-2: Unsupervised Learning for Natural Language Processing

Co-Training
Combining Labeled and Unlabeled Data

- Why semi-unsupervised learning?
- Brief History
- What is Co-Training ?
- When will Co-Training Work ?
  - High agreement rate (redundancy among features)
  - Conditional independency of feature given class (too strong)
- Weak Hypothesis Dependence
- General Observation

Why semi-unsupervised learning?

- Both supervised and un-supervised learning have drawbacks
  - Supervised, with fully hand-labeled training set
    - Labeled data: expensive and rare
  - Unsupervised, with unlabeled training data
    - not always competitive with supervised training
- We need a compromise solution: Semi-unsupervised Learning
  - Use as much unlabeled data as possible without sacrificing performance

How can Unlabeled Data be Useful ?

- Since we don't know the associated class label of an unlabeled example, we cannot use it to directly train classifiers
- But, we can be almost sure of its class label, if …
  - Some classifiers consistently assign the same class labels to an example with high confidence
- Co-Training: take advantages of the consistent classification results of two classifiers to incrementally assign class labels to unlabeled data, and update classifier parameters, to
  - incrementally improve performance of classifiers
  - incrementally enlarge labeled data

Co-Training Procedure

- Basic Procedure
  - Start with a small labeled data set and a large unlabeled data set
  - Train separated models of two classifiers (with different views to the data) from labeled data set
  - Two classifiers independently classify those unlabeled data set
  - For those tokens that are assigned the same class-label by each classifier with high confidence, put them into the labeled data set
  - Retrain the models with enlarged labeled data set, and repeat the above procedure
  - Stop when it has converged (nothing has changed)

Brief History

- [Yarowsky 95] Word Sense Disambiguation
  - Using “One sense per collocation” (view of context word) and “One sense per discourse” (view of document consistency)
  - Obtain 96.5% performance, rivals supervised learning (96.1%)
  - First experiment of Co-Training. No supporting theory is given
- [Blum & Mitchell 98] PAC (Probably Approximately Correct) model
  - Propose PAC supporting model under “Conditional Independent” assumption, and first coin out the name of “Co-Training”
  - Obtain 5.0% error rate in web-page classification with 12 samples, which would have 11.1% error rate in supervised learning (12 data)
- [Collins & Singer, 99] Propose to make two classifiers agree
  - Discuss the limitations of [Blum & Mitchell, 98]
  - Propose CoBoost, which is based on AdaBoost [Freund & Schapire, 99], to make two classifiers agree on those unlabeled data as much as possible. Test on Named Entity Classification

Brief History (Cont.)

- [Nigam & Ghani, 2000] Test effectiveness of feature split
  - Test effectiveness of feature split under various situations
  - Integrate with EM: Co-EM, self-training
- [Dasgupta, Littman and McAllester, 01] PAC-style model formally justify the Collins and Singer suggestion
  - Give a bound on the generalization error of each classifier in terms of the empirical agreement rate between two classifiers
- [Abney, 02] Weak Hypothesis Dependence
  - Release the conditional independence assumption
  - Proposing more realistic *Weak Hypothesis Dependence* instead
- Co-Training applied to NLP
  - Good: Sense Tagging [Yarowsky 95], Statistical Parsing [Sarkar 01]
  - So So: Tagging [Abney et al, 99], Base Noun Identification [Pierce & Cardie, 01], Reference Resolution [Muller, Rapp and Strube, 02]

[Blum and Mitchell, 98] Assumptions

- Feature Redundancy
  - Two views used by different classifiers are not completely correlated
  - Each view is sufficient for classification
- View Independency of Features, given Class
  - Each view provide consistent classification results

Definition 1 *A pair of views x_1, x_2 satisfy view independence just in case:*

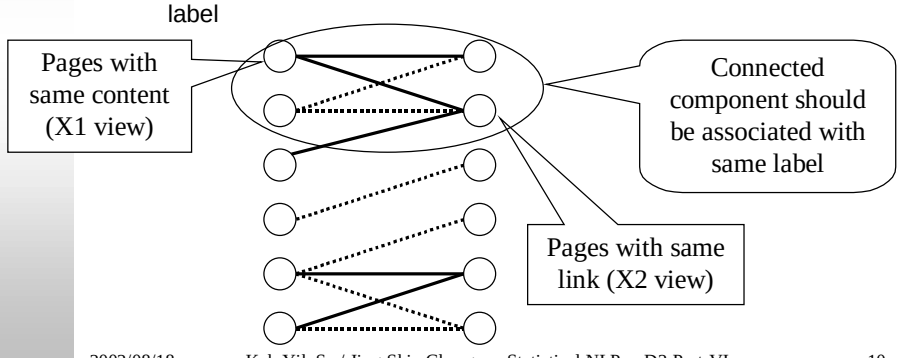
$$\Pr\left[X_1 = x_1 \mid X_2 = x_2, Y = y\right] = \Pr\left[X_1 = x_1 \mid Y = y\right]$$

$$\Pr\left[X_2 = x_2 \mid X_1 = x_1, Y = y\right] = \Pr\left[X_2 = x_2 \mid Y = y\right]$$

A classification problem instance satisfies view independence just in case all pairs X_1, X_2 satisfy view independence.

[Blum & Mitchell, 98] Basic Idea

- Bipartite graph representation:
  - Edges: examples with non-zero probability under D (distribution)
  - Solid edges: examples observed in some finite labeled sample S
  - Under co-training assumptions, even without seeing any labels, the learning algorithm can deduce that any two examples belonging to the same connected component in Gs should have the same class-label



[Blum and Mitchell, 98] approach

- [Blum & Mitchell, 1998]
  - Use labeled data (L) to initialize two classifiers with two independent views of the training data
  - Classify unlabeled data (U') from an unlabeled data pool (U) with the two different classifiers for most confidently identified positive and negative examples
  - Augment the labeled set with examples of high confidence (not necessarily agreed by both classifiers)
  - Re-train the two classifiers with the augmented set, and re-classify data from unlabeled pool
  - Use a combined classifier:

$$\text{wrong: } P(C_j \mid x_1, x_2) \leftrightarrow P(C_j \mid x_1) \times P(C_j \mid x_2) \qquad \text{[B\&M98]}$$

$$\text{correct: } \hat{C} = \arg\max_{C_j} P(x_1 \mid C_j) \times P(x_2 \mid C_j) \times P(C_j)$$

[Blum & Mitchell, 98] Web-Page Classification (1)

- Target: find "course home page"
  - Corpus: 1051 web pages from CS departments of 4 University (all labeled)
  - Positive : Negative = 22% : 78% ~= 1:3.
  - Test Set: 25% (263 pages)
  - Training Set: 75%, 3 positive + 9 negative as labeled, the others as unlabeled (through a 5-fold heldout evaluation)
  - Classifiers: two naïve Bayes classifiers, each using a view of the web pages.
- Two views (independent features) of a web page:
  - "bag of words" on a web page
  - "bag of words" of hyperlinks pointing to this web page (e.g., "my advisor")

[Blum & Mitchell, 98] Co-Training Algorithm

Given:

- a set L of labeled training examples
- a set U of unlabeled examples

Create a pool U' of examples by choosing u examples at random from U

Loop for k iterations:

Use L to train a classifier h_1 that considers only the x_1 portion of x

Use L to train a classifier h_2 that considers only the x_2 portion of x

Allow h_1 to label p positive and n negative examples from U'

Allow h_2 to label p positive and n negative examples from U'

Add these self-labeled examples to L

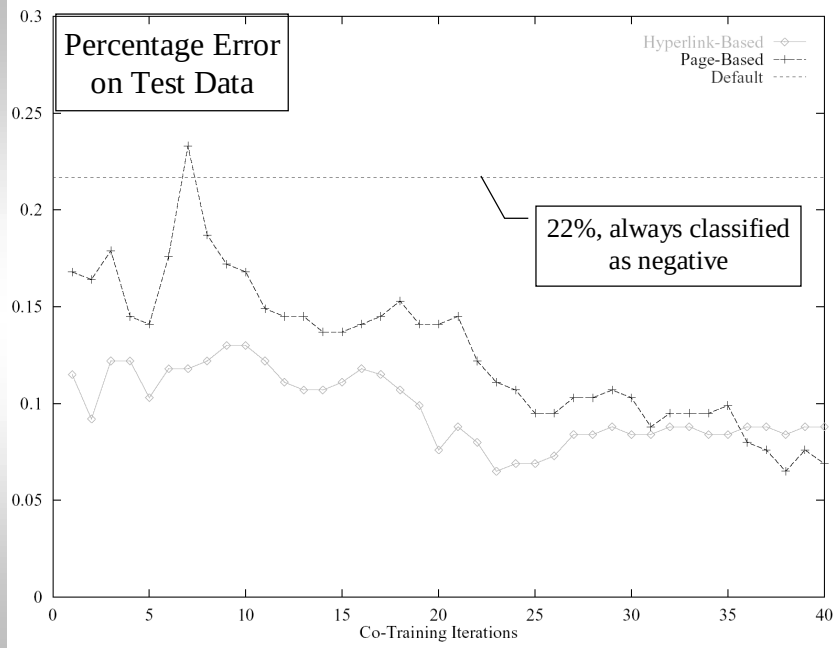
Randomly choose $2p + 2n$ examples from U to replenish U'

[Blum & Mitchell 98] Web-Page Classification (2)

- Test Set Performance:

	Page-based classifier	Hyperlink-based classifier	Combined classifier
Supervised training	12.9	12.4	11.1
Co-training	6.2	11.6	5.0

Table 2: Error rate in percent for classifying web pages as course home pages. The top row shows errors when training on only the labeled examples. Bottom row shows errors when co-training, using both labeled and unlabeled examples.



15

[Collings & Singer, 99] CoBoost Test

- Name-Entity Classification
  - Corpus: 88,962 (*spelling, context*)
  - Test Set: 1,000 out randomly draw from the above corpus
  - One view is “Spelling”, another view is “Context”
- Performance

Learning Algorithm	Accuracy (Clean)	Accuracy (Noise)
Baseline	45.8%	41.8%
EM	83.1%	75.8%
(Yarowsky 95)	81.3%	74.1%
Yarowsky-cautious	91.2%	83.2%
DL-Cotrain	91.3%	83.3%
CoBoost	91.1%	83.1%

Table 2 : Accuracy for different learning methods. The baseline method tags all entities as the most frequent class type (organization).

[Nigam & Ghani 2000] Feature Split Test (1)

- Web-Page Classification with Naïve Bayes Classifier

Table 2 : Classification error rates for co-training, EM and naive Bayes on the WebKB-Course dataset. This dataset does not demonstrate that co-training algorithms are better than other algorithms even when the features naturally divide.

Algorithm	#Labeled	#Unlabeled	Error
Naive Bayes	788	-0-	3.3%
Co-training	12	766	5.4%
EM	12	766	4.3%
Naive Bayes	12	-0-	13.3%

[Nigam & Ghani 2000] Feature Split Test (2)

- Text Classification with Naïve Bayes Classifier
  - Make two feature sets conditionally independent

Table 4 : Classification error rates on the News 2x2 dataset. On a dataset with true class-conditional independence between the two feature sets, co-training outperforms EM, which does not explicitly use the feature split.

Algorithm	#Labeled	#Unlabeled	Error
Naive Bayes	1006	-0-	3.9%
Co-training	6	1000	3.7%
EM	6	1000	8.9%
Naive Bayes	6	-0-	34.0%

[Nigam & Ghani 2000] Feature Split Test (3)

- Text Classification with Naïve Bayes Classifier
  - Make two feature sets conditionally independent

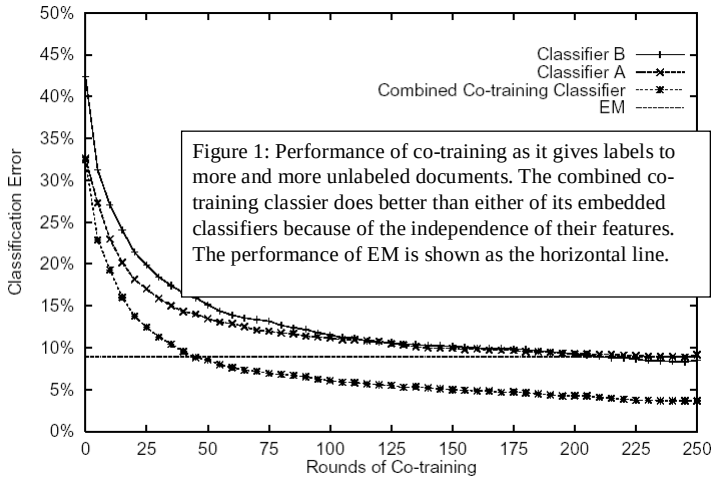


Figure 1: Performance of co-training as it gives labels to more and more unlabeled documents. The combined co-training classier does better than either of its embedded classifiers because of the independence of their features. The performance of EM is shown as the horizontal line.

[Nigam & Ghani 2000] Feature Split Test (4)

- Hybrid algorithms for conditionally independent two feature sets
  - Co-training: incrementally uses the unlabeled data
  - EM: iteratively uses the unlabeled data
  - Combine Feature-Split and Training-Mode: Co-EM and Self-Training

Table 5: The space of algorithms using labeled and unlabeled data.

Method	**Uses Feature Split?**	
	Yes	**No**
Incremental	co-training	self-training
Iterative	co-EM	EM

  - Co-EM: Converge is not guaranteed

Table 6: Classification error rates for four algorithms using labeled and unlabeled data on the News 2x2 dataset.

Method	**Uses Feature Split?**	
	Yes	**No**
Incremental	3.7%	5.8%
Iterative	3.3%	8.9%

[Nigam & Ghani 2000] Feature Split Test (5)

- Randomly Split Feature-Sets

Table 7: Classification error rates for four algorithms using labeled and unlabeled data on the News 2x2 dataset.

Method	Uses Random Feature Split?	
	Yes	No
Incremental	5.5%	5.8%
Iterative	5.1%	8.9%

Table 8: Classification error rates for four algorithms using labeled and unlabeled data on the News5 dataset.

Method	Uses Random Feature Split?	
	Yes	No
Incremental	28.0%	27.0%
Iterative	29.9%	31.2%

[Nigam & Ghani 2000] Feature Split Test (6)

- How to select conditionally independent feature sets
  - Conditionally independent => conditional mutual information between two feature set is zero
  - Calculate the conditional mutual information between every pair of possible features
  - Create a V-regular undirected graph with the weights on each edge being the associated conditional mutual information
  - Make a 2-way balanced cut in the graph with minimum sum of weights of the edges to be cut

[Nigam & Ghani 2000] Feature Split Test (7)

- Observation and Conjecture [Nigam & Ghani, 2000]
  - EM is a likelihood-based approach, and is more sensitive to the violation of underlying model
  - EM is expected to do well when its underlying assumption about the data is correct
  - Co-Training ranks the data by confidence, not directly uses the actual posterior probabilities, is thus a more discriminative approach
  - Self-learning might be more resistant to local maximum (than EM), as new data is added to the training set at each iteration
- Adding only a few most confident data at each iteration seems a good strategy
  - Also observed in the following Chinese New Lexicons Extraction Experiment

[Dasgupta, Littman and McAllester, 01] Relation to Empirical Agreement Rate (1)

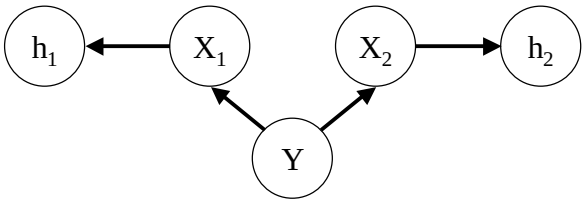
- Intuitive motivation
  - EM is often subject to local minima and will over-fit the data when there is a large number of parameters with the model
  - Avoiding training on low-confidence filled-in labels one might avoid the self-justifying local optima encountered by EM
- Main theorem
  - If the sample size is large, and h_1 and h_2 largely agree on the unlabeled data, then $\hat{P}(\cdot)$ is a good estimate of the error rate P(..)

Theorem 1 : With probability at least $1-\delta$ over the choice of the sample S, we have that for all h_1 and h_2, if $\gamma_i(h_1, h_2, \delta) > 0$ for $1 \le i \le k$ then (a) f is a permutation and (b) for all $1 \le i \le k$,

$$P(h_1 \neq i \mid f(y) = i, h_1 \neq \perp) \le \frac{\hat{P}(h_1 \neq i \mid h_2 = i, h_1 \neq \perp) + \varepsilon_i(h_1, h_2, \delta)}{\gamma_i(h_1, h_2, \delta)}$$

[Dasgupta, Littman and McAllester, 01] Relation to Empirical Agreement Rate (2)

- PAC-style model formally justify the Collins and Singer suggestion
  - Conditionally independence implies that the mutual information between x_1 and x_2 given y is zero
  - $I(h_1 ; y) >= I(h_1 ; h_2)$: any mutual information between h_1 and h_2 must be mediated through y. If h_1 and h_2 agree to a large extent, then they must reveal a lot about y. And yet finding such a pair (h_1, h_2) requires no labeled data at all



The co-training scenario with rules h_1 and h_2

[Abney, 02] Hypothesis Independent (1)

- Hypothesis Independent:
  - P(F = u | G = v, Y = y) = P(F = u | Y = y)
  - P(G = v | F = u, Y = y) = P(G = v | Y = y)
- Theorem 1:
  - View Independence implies hypothesis independence

[Abney, 02] Hypothesis Independent (2)

- Theorem 2:

Theorem 2: For all $F \in H_1, G \in H_2$ that satisfy hypothesis independence and are nontrivial predictors in the sense that $\min_u \Pr[F = u] > \Pr[F \neq G]$, one of the following inequalities holds:

$$\Pr[F \neq Y] \leq \Pr[F \neq G]$$
$$\Pr[\overline{F} \neq Y] \leq \Pr[F \neq G]$$

- If F agrees with G on all but epsilon unlabelled examples, then either F or F bar predicts Y with error no greater then epsilon
- A small amount of labeled data suffices to choose between F and F bar

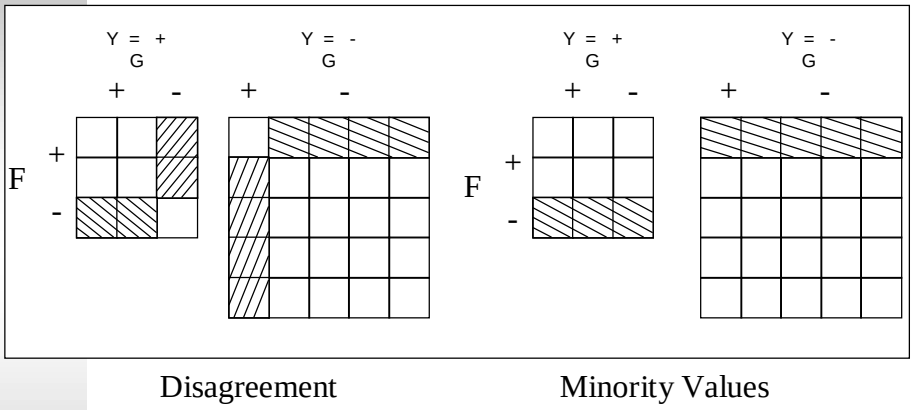


Figure 1: Disagreement upper-bounds minority Probabilities.

[Abney, 02] Weak Hypothesis Dependence (1)

- Hypothesis Independence is too strong:
  - If hypothesis independence holds, knowing the precision of any one hypothesis allows one to exactly compute the precision of every other hypothesis given only unlabeled data and knowledge of the size of the target concept
- Weak Hypothesis Dependence
  - Conditional dependence

Conditional dependence of F and G given $Y = y$:

$$d_y = \frac{1}{2}\sum_{u,v}\left|\Pr(G = v \mid Y = y, F = u) - \Pr(G = v \mid Y = y)\right|$$

if F, G are conditioanlly independent then $d_y = 0$.

[Abney, 02] Weak Hypothesis Dependence (2)

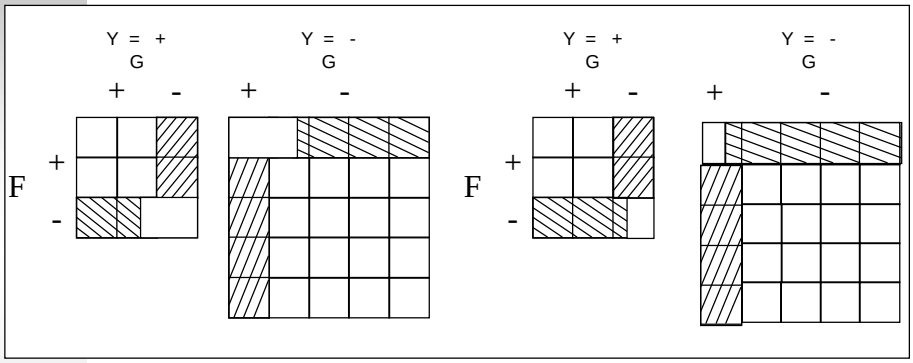
- Weak Hypothesis Dependence (Cont.)
  - Weak Hypothesis Dependence

Definition 4 *Hypotheses F and G satisfy weak hypothesis dependence just in case, for* $y \in \{+,-\}$:

$$d_y \le p_2 \frac{q_1 - p_1}{2 p_1 q_1}$$

where $p_1 = \min_u \Pr\left[F = u \mid Y = y\right]$,

$p_2 = \min_u \Pr\left[G = u \mid Y = y\right]$, *and* $q_1 = 1 - p_1$.



(a) Positive correlation (b) Negative correlation

Figure 2 : Deviation from conditional independence.

[Abney, 02] Weak Hypothesis Dependence (3)

Theorem 3 *For all* $F \in H_1$, $G \in H_2$ *that satisfy weak hypothesis dependence and are nontrivial predictors in the sense that* $\min_u \Pr[F = u] > \Pr[F \neq G]$, *exactly one of the following inequalities holds:*

$$\Pr[F \neq Y] \leq \Pr[F \neq G]$$
$$\Pr[\overline{F} \neq Y] \leq \Pr[F \neq G]$$

- The area of disagreement (B union C) upper bounds the area of minority value of F (A union B)

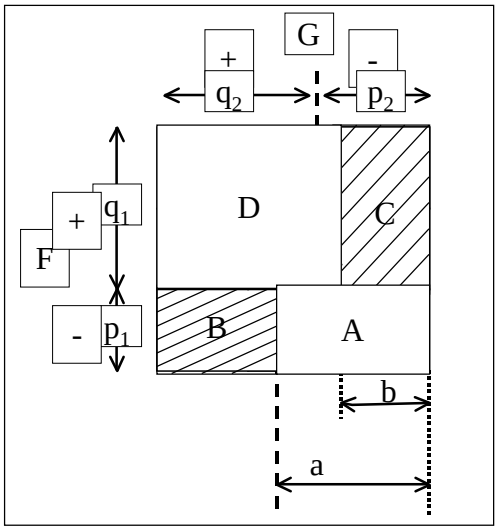


Figure 3: Positive correlation, Y = +.

[Abney, 02] Weak Hypothesis Dependence (4)

- Proposed Greedy Agreement Algorithm:
  - At each iteration, each possible extension to one of the hypothesis is considered and scored. The best one is kept, and attention shifts to the other hypothesis

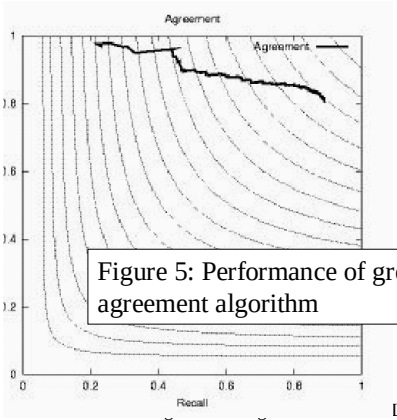


Figure 5: Performance of greedy agreement algorithm

[Abney, 02] Precision Independence

- Yarowsky's algorithm is actually based on the assumption of Precision Independence

Definition 5 *Feature F and labeled set G satisfy* precision independence, *just in case, for all l,*

$$P(Y_l \mid F, G) = P(Y_l \mid F)$$

A bootstrapping problem instance satisfies precision independence just in case all labeled sets G and all hypotheses F that nontrivially overlap with G (both $F \cap G$ *and* $F - G$ *are nonempty) satisfy precision independence.*

[Abney, 02] Precision Independence (Cont.)

- Theorem 5

Theorem 5 *If the assumptions of precision independence and balanced errors are satisfied, then the Yarowsky algorithm with threshold* θ *obtains a final hypothesis whose precision is at least* θ*. Moreover, recall is bounded below by* $N_t\theta/N_l$*, a quantity which increases at each round.*

- Intuitively, the Yarowsky's algorithm increases recall while holding precision above a threshold that represents the desired precision of the final hypothesis
- Yarowsky's algorithm is not a special case of co-training. Precision independence and view independence are distinct assumptions; neither implies the other

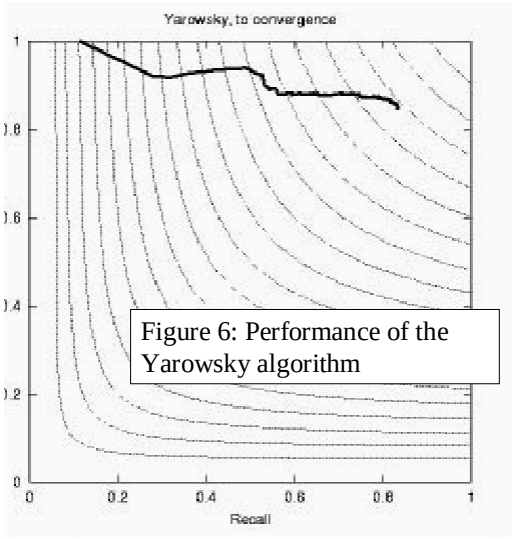


Figure 6: Performance of the Yarowsky algorithm

General Observation

- The improvement usually shrinks when the size of initial labeled data increases
- Benefit cannot be consistently observed across different NLP applications
- Learning curve is not smoothly converged
  - Bias from data added in (only confidence ones are added)
  - Quality degradation of labeled data set lately added in
- Classifiers which obey the assumption and have high degree of agreement might not be easy to find
- Although it is a theoretically sound approach, the theory only provide an upper bound, which is frequently useless when compared with other approaches

Example: Extracting Chinese New Words [Chang 97a, 97b]

- Task Definition
  - Generate potential new word list from the given corpus
  - Optimization Criteria: improve precision and recall simultaneously
- System Architecture
  - Word segmentation: with contextual constraints (one view)
  - Cohesion judge: ranking module according to the likelihood values (two classes model with association features, another view)
  - Two-stage iterative approaches to improve recall, in addition to improving precision
- Consideration in designing Unsupervised Learning
- Prospective Improvement for Unsupervised Learning

Task Definition: Extracting Chinese New Words

- Task: extract new words from the given un-segmented text corpus
  - **Input:** An un-segmented Chinese Text Corpus, and a system dictionary of known words
  - **Output:** Potential New Words in the Text Corpus (that were not in the system dictionary)
  - **Criteria:** Improve joint precision-recall performance

Example of New Word Extraction

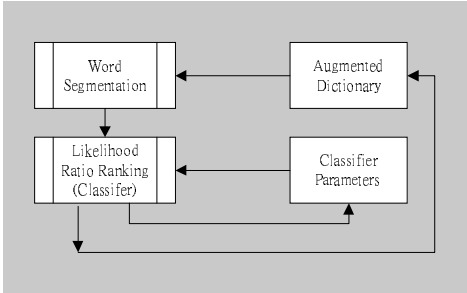
- China Times 1997/7/26:
  - **[台經院]**指出，隨著股市**[活絡]**與景氣**[回溫]**，第一季車輛及零件營業額成長十六.八一%，顯示民間需求**[回升]**。再加上爲加入ＷＴＯ，開放進口已是時勢所趨，也將帶動消費成長。 **[台經院]**預測今年民間消費全年成長率可提昇至六.七四%。
  - 在投資方面，第一季國內投資出現**[回升]**走勢， **[固定資本]**形成實質增加六.五六%，其中民間投資實質增加八.九五%。在持續有民間大型投資計畫進行、國內**[房市]** **[回溫]** 、與政府開放投資、加速執行公共工程等多項因素下，預測今年全年民間投資將成長十一.八%。
  - **[台經院]**表示， **[口蹄疫]** **[連鎖效應]**在第二季顯現，使第二季出口貿易成長率比預期低，出口**[年增率]**二.一%，比去年低。而進口**[年增率]**爲七.三八%，因此第二季貿易出超僅十七.一四億美元，比去年第二季減少四十三.六五%。不過，由於第三、四季爲出口旺季，加上國際組織均預測今年世界**[貿易量]**擴大， **[台經院]**認爲我商品出口應可轉趨順暢。
- New words: proper names, jargons, lexicalized compounds, ...

Basic Language Models and System Architecture

- Two Modules
  - Word Segmentation Module (One Classifier):
    - A Viterbi Training module: to get best word segments
    - According to an augmented dictionary, i.e., the union of system dictionary plus high frequency character n-grams
  - Likelihood Ratio Test Module (Another Classifier):
    - A two-class classification module: used to rank word candidates (in best segments) by likelihood ratio
    - Can also be used to determine whether an n-gram is a word, but was not used in this manner

Basic Language Models and System Architecture

- Integration of the Modules
  - Iteratively apply word segmentation and use the relative rank information of the segments to improve the augmented dictionary for segmentation
    - improve the segmentation parameters and classifier parameters as well



Language Model for Word Segmentation (Viterbi Training)

- Segmentation Stage: Find the best segmentation pattern S*

$$S^*(V(t)) = \arg\max_{S_j} P(S_j = w_{j,1}^{j,m(j)} | c_1^n, V(t))$$

- which maximizes the following likelihood function of the input corpus

$$P(S_j = w_{j,1}^{j,m(j)} | c_1^n, V(t)) \approx \prod_{i=1,m(j)} P(w_{j,i} | V(t))$$

- c_1^n : input characters c_1, c_1, ..., c_n
- S_j : j-th segmentation pattern, consisting of { $w_{j,1}$, $w_{j,2}$, ..., $w_{j,m(j)}$ }
  - V(t): vocabulary (n-grams in the augmented dictionary) used for segmentation at the t^{-th} iteration
  - S*(V): the best segmentation (is a function of V)

Language Model for Word Segmentation (cont.) (Viterbi Training)

- Re-estimation Stage: Estimate the word probabilities which maximize the likelihood of the input text:
  - Initial Estimation:

$$P(w_{j,i}|V) = \frac{Number(w_{j,i})\ in\ corpus}{Number\ of\ all\ w_{j,i}\ in\ corpus}$$

  - Re-estimation:

$$P(w_{j,i}|V) = \frac{Number(w_{j,i})\ in\ best\ segmentation}{Number\ of\ all\ w_{j,i}\ in\ best\ segmentation}$$

Language Model for Two-Class Classifier (Log-Likelihood Ratio Ranking Module)

- **Input:** n-grams in the given un-segmented text corpus
- **Output:** assign a class label ("word" or "non-word") to each n-gram
- **Classifier:** a log-likelihood ratio (LLR) tester (minimum error classifier)

$$g(\mathbf{x}) = LLR(\mathbf{x}) = \log \frac{f(\mathbf{x}|\mathbf{W})}{f(\mathbf{x}|\overline{\mathbf{W}})}$$

- **Decision Rules:**

$$class(w(\mathbf{x})) = \begin{cases} +word \quad (word) & if\ LLR(\mathbf{x}) \geq \lambda_0 \\ -word \quad (non-word) & if\ LLR(\mathbf{x}) < \lambda_0 \end{cases}$$

- **Advantage:** ensure minimum classification error (with λ_0 =0) if the distributions are known.
- **NOTE:** We don't really use it for assigning class label when joining in unsupervised learning. Instead, the associated LLR's are used for sorting to identify **relative ranking order** of character n-grams, and hence it works as a ranking module.

Integration of Knowledge Sources

- Conventional System Schemes:
  - Segmentation (with known words) + Merge adjacent characters + Qualification with a filter
- Characteristics:
  - Independent knowledge sources, one-pass, non-iterative
    - Word Segmentation: Use contextual constraints (or contextual probabilities) to find the best segmentation
    - Filter: Use word association features (e.g., mutual information, dice) to filter out unlikely compound words
      - many filtering approaches filter out unlikely candidates in a feature-by-feature filtering manner, one feature one filtering step
    - No information sharing between the two modules

Problems with Segment-Merge-Filtering Schemes

- Merge-type errors cannot be recovered:
  - Types of errors: over-segmentation, under-segmentation (mis-merging)
  - **New words** may be merged with neighbors into **known words** in a system dictionary, and thus will not be extracted
    - Example: known word:土地公 & new word:公有
    - [土地公有政策] => [土地公][有][政策]
- **Simple filtering will *never* improve recall**
  - Successful filtering ⇒ precision improved, recall **unchanged**
  - Unsuccessful filtering ⇒ both precision and recall **degraded**

Problems with Segment-Merge-Filtering Schemes

- Association features not used jointly; instead, used independently
  - Worse than jointly considering all association features
- Information cannot be shared between word segmentation and filtering
  - Inherent contextual constraints cannot be used by filter
  - Word association features do not help select candidate word for segmentation module
- Model parameters are not improved iteratively
  - Performance of segmentation and filtering is unlikely to be perfect in only one pass with unsupervised mode

Strategies for Extracting Chinese New Words

- Strategies
  - Use augmented dictionary (system dictionary+high frequency n-grams)
    - to prevent from pre-mature rejection of new words by using only known words for segmentation
    - new words have the chance to compete with known words during segmentation
  - Iterative Approach to provide a chance for improving **recall**:
    - Word Segmentation → Qualification (→ Re-estimate Parameters) → Segmentation → Qualification (→ Re-estimate Parameters) …
    - Why: (See Next Slide)
  - Use a two-class classifier which jointly considering all features: likelihood ratio test
  - Use ranks of likelihood ratio to identify very likely or very unlikely candidates, instead of using the value for filtering out candidates with non-positive values
  - ☆ Filter => Likelihood Ratio Ranking Module (aka LRRM)

Extracting Chinese New Words

- Why Iterative ?
  - **Recall Improvement:** Truncated candidates could be replaced by other more likely segments (judged by contextual probability) at later segmentation iterations, thus extracting likely new words
    - ⇨ **Recall** could be improved, in addition to improving precision (by filtering)
    - ⇨ Joint improvement of precision-recall becomes possible
  - **Information Sharing:** Contextual probability used by Word Segmentation and association features used by filter help each other in improving the model parameters
    - WS: producing better segments iteration by iteration, highly probable new words are moved to the word-class, thus refine two-class classifier model
    - Filter: provide correct candidate ranking for truncating unlikely n-grams, thus improve the dictionary used by the word segmentation module
    - ⇨ Contextual information and Association features are iteratively integrated

Unsupervised Training for New Word Extraction

- **Initialization:**
  - Initial augmented dictionary = {system dictionary + high frequency n-grams in text (frequency count >=5)}
  - Initial word segmentation probability = relative frequency in text corpus
  - Initial two-class classifier parameters: divide n-grams into word & non-word according to system dictionary & estimate feature distribution for the two classes
- **Jointly train & improve two modules:**
  - Word Segmentation+Ranking Module
    - LRRM: a two-class classifier, using likelihood ratio between word-class and non-word class to rank possibility of an n-gram being a word

Unsupervised Training for New Word Extraction (cont.)

- **Jointly train & improve two modules (cont.)**
  - **Viterbi Training:** for Training Word Segmentation Module:
    - Use initial probabilities for finding the best word segments
    - Re-estimate word probabilities from best segments
    - **Repeat:** until converge or running a specified iterations
  - Sort word list in Word Segmentation results by Likelihood Ratio
  - Delete unlikely words (not in system dictionary) from augmented dictionary
  - **Update word/non-word class parameters of LRRM:** with highly likely new words (change the estimates to the word-class)
- **Repeat:** Joint Training to Iteratively improve the Viterbi-Training and LRRM modules

Viterbi Training for Extracting New Words

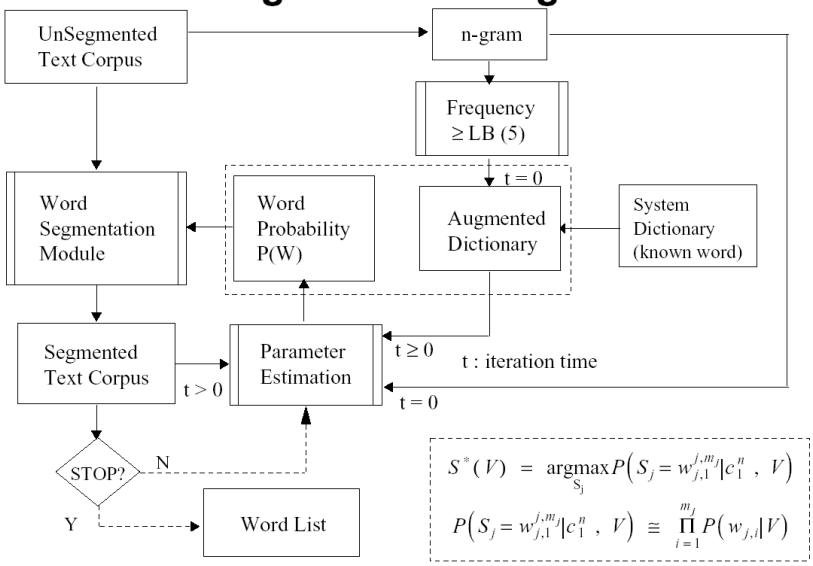


Figure 1 The Viterbi training model for unsupervised new word identification

Integrated System for New Word Identification

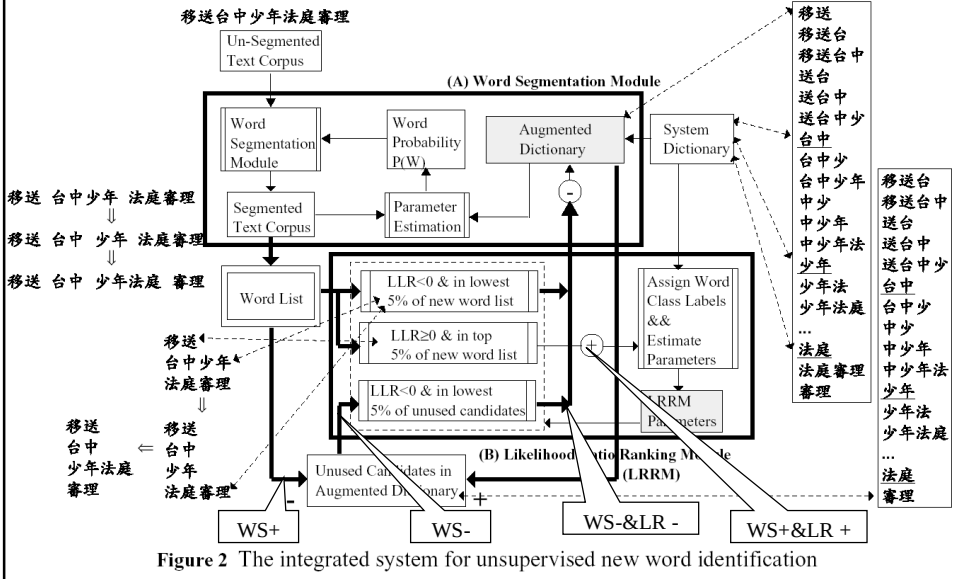
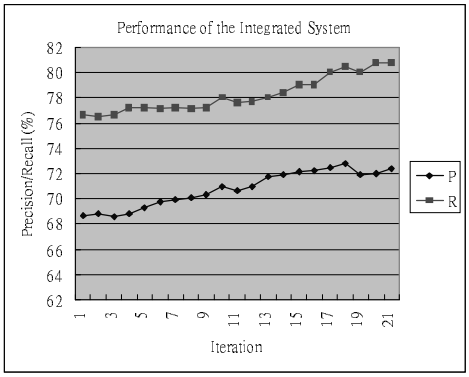


Figure 2 The integrated system for unsupervised new word identification

Extracting Chinese New Words

- Results: precision and recall both increase almost monotonically without sacrificing one for another
  - bigram new word precision (P) & recall (R):



Comparison between Example & Suggested Steps (I)

1. Develop Models that Reflect Human Inference, Embed Constrains and Fit Training Data

- Select Discriminative Features based on which human make preference
  - Segmentation: uses character N-grams (could be integrated with POS tags)
  - Classifier: uses (mutual information, entropy) vector jointly, instead of using them as individual association measures for filtering candidates

Comparison between Example & Suggested Steps (II)

- 1. Develop Models (cont.)
  - Select Appropriate Form
    - Determine appropriate Feature Dependency:
      - segmentation model is known to be conditional on potential candidate list, thus it motivates us to design procedures for refining augmented dictionary, which is used for segmentation, iteratively
      - classifier was based on likelihood ratio test for minimum error rate
    - Decide suitable Model Complexity with Cross-Validation Set: Not applied to the two-class ranking module (since there are only two features)
      - Feature selection could be conducted as feature number increases
    - Integrate the two different knowledge source in an iteratively improved manner

Comparison between Example & Suggested Steps (III)

2. Initial Guess

- Adopting Annotated Seed Corpus for Initial Model Parameters: segmented seed corpus is currently not available; will be a plus if available (was applied in another task [Chang 95])
- Using a System Dictionary and high frequency n-grams as possible anchor points for word segmentation, and estimating segmentation parameters as relative frequency in un-segmented input
- Using the System Dictionary for dividing n-grams into two classes (word/non-word) for estimating initial classifier parameters
- Smoothing Parameters for Unseen Events (with respective to seed corpus) in Training Set: N/A

Comparison between Example & Suggested Steps (IV)

3. Re-generating Prediction According to New Model Parameters
   - Viterbi-type labeling for word segmentation
4. Re-Estimation of Model Parameters via MLE
   - Viterbi Training within word segmentation module
5. Repeat the Prediction and Estimation Steps until joint likelihood value of the training corpus converge
   - within each joint training iteration of two module, the likelihood associated with the word segmentation module is maximized

Comparison between Example & Suggested Steps (V)

6. Conduct Discriminative/Robust Learning in Seed Corpus (Tying Parameters)
   - currently not applied; using a segmented seed for adjusting segmentation parameters and the two-class parameters would be helpful
7. Bootstrap Incrementally Stage by Stage
   - not applied in this task; may better utilize the seed if applied

Comparison between Example & Suggested Steps (VI)

8. Using the Cross-Validation Set to Check the Effectiveness of Each Step
   - not applied
9. Iterate the above design procedures until you are satisfied
   - using a two-stage iterative approach to integrate the two modules did provide a system that meets our expectation: improving precision and recall simultaneously without trading one for another

Refinement to the Unsupervised Learning Procedure

- Possible Future Refinement
  - Feature set: joint more useful association measures in the classifier (ranking module), including feature selection mechanism for the best subset; using tagging information to help segmentation, etc.
  - Initial Guess: will be a plus with segmented seed corpus; applying smoothing to the initial parameters could be helpful as well
  - Discriminative/Adaptive Learning on Seed: could be applied to adjust the parameters to get maximize precision and recall (in terms of weighting sum or F-measure) as we did for English compound extraction [Chang 97]
  - Bootstrapping: Incrementally enlarge the training size with the seed fixed would possibly leads us to a better set of parameters