[Day-2] Unsupervised Learning for Natural Language Processing

(Part V: Suggested Strategies for Better Performance)

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

Part V: Suggested Strategies for Better Performance

- Lessons Learned from POS Tagging
- Phenomena Frequently Observed in the Past
- Basic Principles
- Essential Elements
  - Adopting Appropriate Language Model
  - Educated Initial Guess
  - Enhancing Discrimination Power
  - Enhancing Robustness
- Suggested Unsupervised Learning Steps

Lesson Learned from POS Tagging (1)

- Unsupervised learning for tagging part-of-speech delivered satisfactory results (although not as good as supervised learning)
- However, very poor performance was reported from the attempt for selecting desired parse-tree via unsupervised learning
- Why unsupervised training works for POS tagging?
  - The Task of tagging part-of-speech is relatively simple (you can get about 90% accuracy rate by always selecting the most frequent tag)
    - Tagging part-of-speech delivers better result than selecting parse tree If both are conducted under supervised-learning mode (which is the upper bound for the unsupervised-learning under the same model)

Lesson Learned from POS Tagging (2)

- Why unsupervised training works for POS tagging (Cont.)?
  - 62% of the words in Brown Corpus have only one tag
    - 62% of the words are also directly observable (human preference is unveiled) in the training set; virtually only 38% ambiguous
    - Not much difference between the complete data space and the incomplete data space.
    - Enough hints for human preference

Lesson Learned from POS Tagging (3)

- During unsupervised-learning iterations:
  - Errors are randomly distributed (uniformly and independently distributed in the first iteration)
  - Those 62% uni-tag words make the correct sub-patterns dominate (appear more times)
    - Many word bi-grams (or even tri-grams, 4-grams, …) can be tagged unambiguously; thus, they enhance the dominance of those correct patterns
    - For example, [det n] are unambiguous in some cases (e.g., The (det) dog (n)), although they would be randomly selected in other cases (e.g., The (det) design (n/v) of (prep) ..)

Lesson Learned from POS Tagging (4)

- During unsupervised-learning iterations (Cont.):
  - Once those correct bi-grams dominate the tagged corpus in terms of their numbers, the dominance will be enhanced iteratively.
  - Those anchor points also impose constrains on their adjacent words; thus, significantly help reducing the task complexity

Lesson Learned from POS Tagging (5)

- What lesson can we learn ?
  - Naive unsupervised learning will not work when the perplexity of the model (task difficulty) is high
    - If task complexity is high, then it usually implies that it will have many local maximums. It will have little chance to converge to the desired point without additional guidance
    - Constraints should be added to the model to make the problem to be solved easier in the adopted feature space
  - Human preference should be told explicitly or implicitly

Phenomena Frequently Observed in the Past

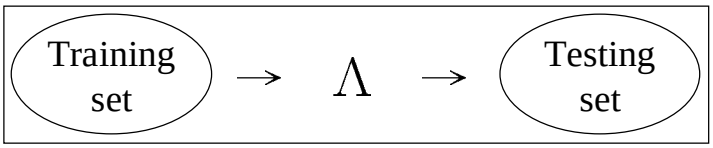
- Linear Expectation versus Non-linear Frustration
  - First Part is most juicy: (20-80 rule)
  - Examples:
    - N-gram (Scope)
    - Corpus-Size (1M,2M,100M,1B)
    - Iteration
    - K-NN
  - After that, you only get little juice even you suck very hard

Phenomena Frequently Observed in the Past (Cont.)

- Changing Domain/Style means a lot comparing to changing techniques:
  - Various Classifiers:
    - Viterbi, EM
    - ME, SVM, AdaBoost, …
    - The reports frequently only cover a specific problem, and the results are compared with the un-refined (baseline or non-optimized) forms of other techniques
  - Smoothing Techniques:
    - Good-Turing,
    - Back-off,
    - Log-Linear,….
- Adopt Lexicon Resource (e.g., WordNet) generates better performance (e.g., Q&A task)

Main (old) Problems:

- Discrimination
  - ML (Measure for Model Fitness) v.s. Error-Rate (Measure for Human Preference)
  - They must be made highly correlated



- Robustness
  - What Characteristic is most likely to be Preserved (invariant) in the testing set ?

OLCR Example:

- Robustness (Cont.)

64

刀 64 $(256)^{64 \times 64}$

刀 (Knife) , 力 (Strength) , 刃 (Blade)

- Only Relative Structure, not the image or crossing-count, will be preserved in the second writing.

Basic Principle for Performance Improvement

- Discrimination: correlating likelihood value with error rate as much as possible
  - Coupling the Stochastic Language Model with the Human Preference Model
- Robustness: making model robust in the testing set
  - Only those features that human really care tend to be preserved in the next sentence

Essential Elements

- Adopting Appropriate Language Model
  - Adopting Appropriate Feature Set
  - Adopting Appropriate Parametric Form
- Educated Initial Guess
- Enhancing Discrimination Power
- Enhancing Robustness

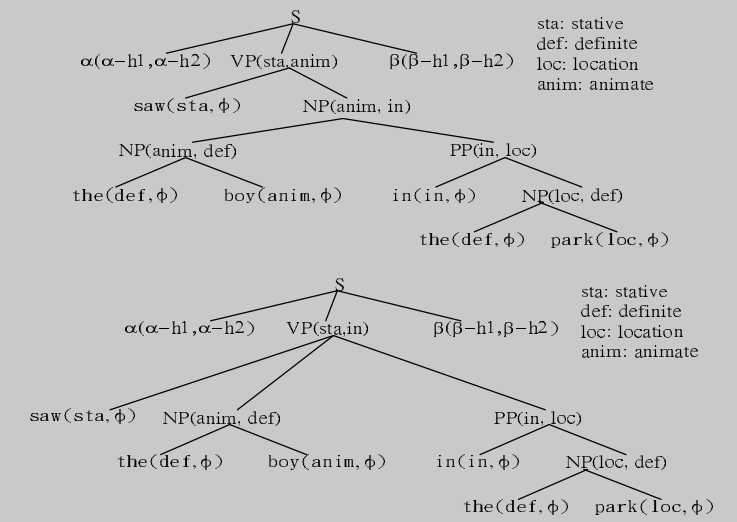
Adopting Appropriate Feature Set (1)

- Heuristically collecting the initial feature set
  - Adopt those existing Explicit Linguistic Features (which echo human preference)
  - Add all features that can impose linguistic constrains on your task
    - Example: the corresponding target sentences in aligned bilingual sentence pairs
  - Add other helpful Implicit Features:
    - They can be probed by using the statistical measures such as Mutual Information (or the correlation dependency test) to probe

Adopting Appropriate Feature Set (2)

- Heuristically collecting the initial feature set (cont.)
  - Add feature percolation mechanism to dynamically provide the required context-sensitive features
    - Build stochastic language model on top of those non-terminal symbols
    - Non-terminal symbols can be used to provide the percolation mechanism required for projecting those head-features
    - Example: Subject-Verb agreement is a long-distance dependency problem in the surface level; however, it is a local dependency problem in the level of NP and VP
      - As NP and VP are adjacent to each other, bi-gram model is enough to handle the agreement problem in this level
  - You can also add the information provided from other competitive classes as features to enhance the discrimination power
    - Probability measures from each class can be used as features [Su 94]

Example for Head-Feature Percolation [Chang 90, 92]



Adopting Appropriate Feature Set (3):

- Select Robust Features from the Initial Feature Set
  - Features possessing Large Discrimination Power are more robust
    - Which are features that have large inter-classes distance & small intra-class variance
  - Discarding Non-discriminative Features to enhance robustness
    - Some features are vulnerable (can easily be contaminated) and are considered harmful in the testing set ; therefore, they should be discarded
  - Use Cross-Validation set for Feature Selection
  - Feature selection is actually executed through the adopted model; therefore, it is an iterative design process (features decide the model; however, the model can be also used to select features)

Adopting Appropriate Parametric Form (1):

- Closely Echo Human Preference Network
  - Follow Human Preference Network and introduce required intermediate form
    - Example:

$$P(T_i|S_j) = \sum_{SubTree} P(T_i, SubTree|S_j)$$

  - Adopting Non-terminal Symbols to Handle Long-distance Dependency
    - Class-based Modeling: e.g., (NP, VP) versus N-gram Markov Chain
  - Drop Terms according to their Relevant Ranking
    - e.g., using Pearson's Chi-square Test for testing (and ranking) degree of independence, drop features that are most independent w.r.t. the outcome (i.e. the numerator term)

Adopting Appropriate Parametric Form (2):

- Closely Echo Human Preference Network (cont.)
  - Example of adopting different feature dependencies:

$$P(x_1^n) = \prod_i P(x_i|x_1^{i-1})$$

$$= \prod P(x_i|x_{i-(n-1)}^{i-1}) \qquad (n-gram)$$

$$= \prod P(x_i|y_{i-m+1}^{i-1}) \quad y_{i-m+1} \xrightarrow{predict} y_{i-m} \cdots \longrightarrow x_i \quad (causal\ chain)$$

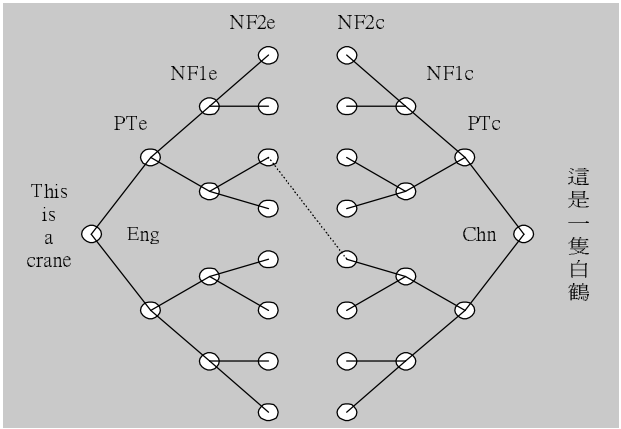
$$y_j = H_j(x_{i-(m-1)}^{i-1}); \quad (non-terminal, or\ head\ features)$$

Adopting Appropriate Parametric Form (3)

- Embedding All Possible Constraints in the Model (to reduce task complexity)
  - Adopting Correlated Resources as Training Corpus (e.g., bilingual corpus)
  - Modeling with Implicit Constraints Embedded
  - Example: Adopting Bilingual Corpus
    - Build aligned bilingual sentence pairs
    - Sentence in each language side would help to impose the constrains on its corresponding sentence in other side
    - For example, in the task of Sense Disambiguation (in Source Side), the possible lexicon senses of each language is restricted by the possible sense set (listed in the dictionary) of the corresponding lexicon in another language

Embed Constraints through two-way learning

- Example: the possible target words (such as, 白鶴) place constraints on possible senses of the source words “crane” through the given bi-lingual sentence pair



Embed Constraints through two-way learning (Cont.)

- Original Optimizing Function (one way unsupervised learning)

$$\hat{\Lambda} = \arg\max_{\Lambda} \{ \max_{I} P(S_1^n, I | \Lambda) \}$$

- Modified Optimizing Function (two way unsupervised learning)

$$\overline{\Lambda} = \arg\max_{\tilde{\Lambda}} \{ \max_{\tilde{I}} P\left([S_i]_1^n, \tilde{I} \,\middle|\, [T_i]_1^n, \tilde{\Lambda}\right),$$

$\hat{\Lambda}$ is in the subspace of that of $\overline{\Lambda}$.

  - The dimensionality of new parameter space would be larger

Adopting Appropriate Form (4):

- Select appropriate modeling unit according to training data size
  - Adopt a larger context-sensitive unit (e.g., Lexicalized-Production-Rule) when data is enough
  - It is the issue of trading in the discrimination power for enhancing the robustness capability
- Eliminate the influence from those Non-discriminative parts by ignoring them or sharing the same sub-model
  - For example, eliminate stop words (or function words) in documents classification
  - Different non-discriminative sub-models can be tied together to form one sub-model by pooling their corresponding training data together (e.g., E-Set)

Adopting Appropriate Form (5):

- Weight different knowledge sources (or parts) according to their contribution to the discrimination power
  - Different knowledge sources (such as lexical, syntactic, and semantic, etc.) have different dynamic ranges. They should be weighted according to their contribution to the discrimination power [Chiang et al. 96]
  - Log-linear weighting model can be adopted
    - e.g., W_{lex} x Log P(Lex|Words) + W_{syn} x Log P(Syn|Lex, Words) ….
  - The weights could also be learned from the seed corpus as a set of parameters

Adopting Appropriate Form (6):

- Adopt Class-Based Modeling, if necessary, to avoid over-fitting
  - Class-based modeling also trade the discrimination power for the robustness
  - Better to be non-uniformly adopted
    - Use the detailed model (e.g., lexicon-based) for the case when its corresponding training data is sufficient, and adopt the class-based model for the case when its corresponding training data is not enough
- Using Cross-Validation Set to Select Appropriate Model Complexity
  - Compromise between the Size of the Training Corpus and the Model Complexity
  - For example, in the case of deciding the number of word-classes we should divide: the larger number of word-classes you choose, the bigger the maximum likelihood value you can obtain from the training set; however, it might even deteriorate the performance in the testing set

Guess Initial Parameters Heuristically

- Heuristically Guessing Initial Parameters
  - EM/Viterbi can be easily trapped at a local maximum that is not matching human preference
  - Try to provide a good starting point for leading the searching process to converge to the desired local maximum point
    - Using problem (or domain) knowledge to make heuristically guessing for the initial parameter set
    - Example (1): PP-Attachment prefer minimum attachment and right association (=> assign heuristic guessing on left/right association, say as [0.3, 0.7], according to prior knowledge).
    - Example (2): non-uniform initial state segmentation in E-Set Speech Recognition (assigning more states to the consonant part)
    - Example (3): adopt unigram priori probabilities as initial values for trigram part-of-speech tagging model

Bootstrapping With a Seed Corpus (1)

- Adopting a seed corpus (annotated) to guide the learning process (unveil the human preference)
- Compromise between Corpus Annotation Cost & Performance
- Obtain hints of human preference, thus has a better chance to converge toward desired local maximum
- Bootstrap in incremental stages
  - Avoid the effect of seed corpus to be over-ridden by the fluctuation resulted from the guessing of the large corpus to be mixed

Bootstrapping With a Seed Corpus (2)

- Procedure
  - Use a small annotated corpus as a seed to estimate the initial parameter values.
  - Smoothed the parameter before going to the next stage of bootstrapping (for handling the unseen cases in previous stages)
  - Enlarge the training set by adding more un-annotated data
  - Use EM or Viterbi training algorithms to re-estimate the parameters, from the enlarged corpora (however, the annotation associated with the seed corpus must be remained untouched).
  - Incremental bootstrapping: repeat the above process stage by stage by adding more unannotated data each time
    - Avoid overriding the hints, brought in from previous iterations, by the perturbation resulted from un-supervised guessing; thus we will have a better initial starting parameter set in each stage.

Enhancing Discrimination Power

- Perform Adaptive Learning if some kinds of related mapping are available (Observable)
  - For example, although the associated state-sequence is unobservable in speech recognition, its corresponding text string is known
  - Another example, although the associated intermediate forms (e.g., parse-tree, word-sense, etc.) are not known, its corresponding target-sentence is known in two-way training
  - Multiple Modules Learning: [Chiang *et. al*, 96]
    - Incremental Learning: modules are trained on-by-one; parameters of current module are trained to optimize the performance criteria of the system
    - Joint Learning: parameters of all modules are trained simultaneously to optimize the system performance criteria
  - Avoid over-tuning by checking the validation set

Enhancing Discrimination Power (cont.)

- Enhancing Learning Efficiency: Parameter Tying [Lin 95, Chiang 95]
  - Tie those rare and highly correlated events into a new class
  - Enlarge the training procedure coverage scope:
    - Such rare parameters will be trained (instead of being ignored from training) when their correlated events are trained
  - Training Efficiency will be higher, as more percentages of the parameters will be better trained

Enhancing Robustness

- Adopt Robust Features and avoid the perturbation from non-discriminating parts
  - Usually those features that human really cares will be preserved in the testing set (otherwise, the sentence cannot convey his/her intention)
  - Check with the Cross-Validation set
  - Train those non-discriminating parts with the same model by pooling their associated data
- Smoothing
  - Managing those unseen/un-reliable/under-trained parameters
  - Back-off smoothing, interpolation (from multiple sources), and tying

Enhancing Robustness (Cont.)

- Enlarging Tolerance Margins between Correct Label and its corresponding Most-Competitive Candidate during the Adaptive Learning Procedure
  - Attempt to achieve the maximum separation between different classes
  - Provide the safety zone for tolerating possible statistical variation and data scatting over in the testing set

Suggested Unsupervised Learning Steps (1)

0. Start with a labeled seed corpus
   - It is strongly suggested not to directly jump into unsupervised learning, unless you are already familiar with the problem
   - Since supervised learning is easier to trace and debug, and the associated behavior is more predictable, it is highly recommended to start with supervised learning
   - First try your model under supervised learning on your seed corpus, until you have got the sense about the model behavior, ranges of parameter values, etc.
   - Use the supervised model as the baseline performance, and then proceed to the unsupervised learning

Suggested Unsupervised Learning Steps (2)

1. Develop models that reflect Human Inference Behavior most, and embed Constrains as much as possible to the model
   - Select Discriminative Features based on which human make preference
     - Should be jointly considered with the adopted form
   - Select Appropriate Form
     - Determine appropriate Feature Dependency
     - Decide suitable Model Complexity with a Cross-Validation Set
   - Determining desired Feature Space and Form is an Iterative Design Process

Suggested Unsupervised Learning Steps (3)

2. Initial Guess
   - Adopting Annotated Seed Corpus for Initial Model Parameters
   - Smoothing Parameters for Unseen Events (with respective to seed corpus) in Training Set before processing those un-annotated corpus
3. Conduct Discriminative/Robust Learning in Seed Corpus (Tying Parameters)
   - to compensate for criteria mismatch
4. Re-generating Prediction According to New Model Parameters in un-labeled corpus (however, don't re-label seed corpus)
   - EM: Re-calculating the Expectation of Sufficient Statistic
   - Viterbi: Re-labeling Corpus according to new Model Parameters

Suggested Unsupervised Learning Steps (4)

5. Re-Estimation of Model Parameters via MLE
   - EM: Using the expectation (which implies that every possibility is considered)
   - Viterbi: Using the guessed labels (only one possibility is considered)
6. Repeat the above Prediction and Estimation Steps until overtuning effect starts to appear
   - Check the overtuning effect with a cross-validation set
7. Conduct Discriminative/Robust Learning in training Corpus (Tying Parameters), if it is possible. (However, don't re-label seed corpus.)
   - To compensate for criteria mismatch

Suggested Unsupervised Learning Steps (5)

8. Bootstrap Seed Corpus Incrementally Stage by Stage
   - Training data is increased incrementally to avoid overriding the Seed
9. Using the Cross-Validation Set to Check the Effectiveness of Each Step
   - Check whether the performance is starting to degrade
10. Iterate the above design procedures until you are satisfied
   - You should end up with a model that better reflects human preference with well-trained parameters

Future Directions:

- The degree of complexity implied by the model under unsupervised learning must be controlled
  - It has been found that baby cannot learn language by just watching Video Tape (Steven Pinker, The Language Instinct, 1994)
  - Children learn languages with some pre-specified patterns
- Perform Unsupervised Learning (with seed) on Deep Structure: Transform into clean space.
  - Semantic Normal Form
  - Discourse Structure
  - Contextual-Knowledge
  - Domain (Project/Product) Knowledge

Future Directions (Cont.):

- Use existing resource (e.g. WordNet, HowNet, etc), and linguistic models (theories)
  - Don't start from scratch, which implies adopting Uniform priori distribution (and thus gives you a handicap from the start)