Semi-supervised learning for statistical machine translation

Anoop Sarkar
(joint work with Gholamreza Haffari)
{anoop,gholamreza_haffari}@cs.sfu.ca

School of Computing Science
Simon Fraser University
Vancouver, Canada

http://natlang.cs.sfu.ca/

December 9, 2006

1 The task: statistical machine translation
- The baseline SMT system
- The hypothesis

2 Previous work in semi-supervised learning for SMT

3 Our approach: Yarowsky algorithm applied to SMT

4 Experiments
- Inductive vs. Transductive
- Experimental Setup
- Experiments

Phrase-based SMT: Train

- Input to training: a set of aligned sentences, $\bigcup_i \{\mathbf{f}_i, \mathbf{e}_i\}$.
- First step in training: train a generative alignment model using EM (unsupervised learning) in both directions: $\mathbf{f} \to \mathbf{e}$ and $\mathbf{e} \to \mathbf{f}$,
- Second step: produce Viterbi alignments for $\mathbf{f} \to \mathbf{e}$ and $\mathbf{e} \to \mathbf{f}$,
- Third step: Extract all phrase pairs upto a fixed length and estimate models for phrasal alignment,
- Fourth step: Discriminative training of $\Pr_{\lambda_1^M}(\mathbf{e} \mid \mathbf{f})$, a log linear combination of M models including various phrasal alignment models, a target language model feature $\Pr(\mathbf{e})$ and others.

Phrase-based SMT: Decode and Test

- Training provides a log-linear model $\Pr_{\lambda_1^M}(\mathbf{e} \mid \mathbf{f})$.
- Decode the test data $\mathbf{f}$: $\mathbf{e}^* = \operatorname{argmax}_{\mathbf{e}} \left\{ \Pr_{\lambda_1^M}(\mathbf{e} \mid \mathbf{f}) \right\}$
- For each test data sentence, evaluate against $4 - 10$ human translations for that sentence.
- Bleu-4 score: weighted combination of upto 4-gram precision scores and a brevity penalty, $\mathrm{Bleu} = bp \cdot \exp\left(\sum_{n=1}^{N} \frac{\log p_n}{N} \right)$
- Baseline system
  - Implementation = $\textsc{Giza}^{++}$, $\textsc{Sri-lm}$ and $\textsc{Moses}$;
  - Dataset = EuroParl corpus from SMT shared task 2006.
  - With 25000 sent pairs in training, $\text{Bleu\%} = 20.9$;
  - With 50000 sent pairs, $\text{Bleu\%} = 22.6$

Improving quality of output translations

- The SMT system:

$$\mathbf{e}^* = \underset{\mathbf{e}}{\operatorname{argmax}} \left\{ Pr_{\lambda_1^M}(\mathbf{e} \mid \mathbf{f}) \right\}$$

- Estimates for the target language model $\Pr(\mathbf{e})$ can be improved by adding large amounts of target **e** text.
- In practice, adding more target **e** text has been shown to improve translation quality considerably.
- Our hypothesis: adding more source **f** text can also provide improvements.
    - Unlike adding target **e** text, this hypothesis is a natural semi-supervised learning (SSL) problem.
    - We need translations for the additional source **f** text before they can be useful in SMT.

Improving quality with additional source text

- French input:
  j'en viens maintenant l'autre point faible: le soutien de l'opinion publique, l'intrieur et l'extrieur de l'union europenne .
- With 2000 English-French parallel text we get English output:
  i have just said to be another point: the support of the public opinion to the internal and medicines completely dependent on the outside the european union. faible now in
- Using only additional monolingual French text we get:
  i come now to another weak point: the support of the public, inside and outside the european union.

SSL for word alignment (Callison-Burch et al, 2004)

- Model IBM-M4: generative model for word alignment extracted using unsupervised learning on parallel text.
- Model SUP: model trained on small amount of hand annotated word alignment data.
- Mixture model provides a probability for word alignment using: $\lambda \ \mathrm{SUP} + (1 - \lambda) \ \mathrm{IBM\text{-}M4}$
- Experiments show $\lambda = 0.9$ performed best (large weight on labeled data).
- However, word alignment does not equal translation quality.

SSL for word alignment (Fraser and Marcu, 2006)

- EM is used to train a generative model of word alignment from a large parallel text. The generative model is decomposed into several sub-models using independence assumptions.
- Each sub-model can be used in a log linear model for word alignment. The weights for the log linear model are trained on a small set of hand aligned sentences.
- Iteratively alternate between approximate EM (Neal and Hinton, 1998) and gradient descent for log linear model until error rate on a held out set is minimized.
- Predicted Viterbi word alignments are used to train a phrase-based SMT system.
- Arabic-English, Bleu%: 49.16 $\Rightarrow$ 50.84;
  French-English, Bleu%: 30.63 $\Rightarrow$ 31.56.

SSL for multiple language pairs (Callison-Burch, 2002)

- Consider source languages **a**, **b**, **c**, **d** which all translate into target language **e**.
- In addition, **a**, **b**, **c**, **d** are sentence aligned with each other.
- If a sentence in **c** is found to be accurately translated into sentence in **e**, then the corresponding aligned sentences in **a**, **b** and **d** now have new labeled parallel text, e.g. $d \rightarrow c \rightarrow e$.
- One language pair creates data for another language pair and can be naturally used in a (Blum and Mitchell, 1998) style co-training algorithm.
- Experiments on the EuroParl corpus show word error rate improvement of 2.5% for German-English (other pairs had lower WER).
- When run long enough, large amounts of co-trained data injected too much noise and performance degraded.

Self-training for SMT (Ueffing, 2006)

- In this workshop!
- Run a log linear phrase-based SMT decoder on source $\mathbf{f}$ text.
- Use word alignments in newly labeled parallel text to extract new phrase pairs,
- Augment the log linear model with new feature functions based on phrasal alignments from newly labeled source $\mathbf{f}$ text.
- This results in a new SMT system that exploits phrase pairs from unlabeled data.

The Yarowsky algorithm: classifier version

- Input: each example x is either labeled $L(x)$ in some annotated data, or unlabeled as $U^0(x) := \perp$.
- Input: function **train** that provides θ for classifier $\pi = \Pr(j \mid x, \theta)$ from labeled training data
- For $t \in \{0, 1, \ldots\}$:
  - Training step: **train** $\pi^{(t+1)}$ using L and U^t
  - For each example x:
    - Labeling step: $\hat{y} = \text{argmax}_{j \in \mathcal{L}} \pi_x^{(t+1)}(j)$
    - Selection step:

$$U^{(t+1)}(x) = \begin{cases} \hat{y} & \text{if } U^{(t)}(x) \neq \perp \text{ or } \pi_x^{(t+1)}(\hat{y}) > \text{threshold } \zeta \\ \perp & \text{otherwise} \end{cases}$$

  - For all x: if $U^{(t+1)}(x) = U^{(t)}(x)$ then **stop**

Analysis of the Yarowsky algorithm (Abney 2004)

Definition

Prediction distribution: $\pi_x(j)$

$$\pi_x(j) = \Pr(j \mid x, \theta)$$

with model parameters θ

Definition

Empirical labeling distribution: $\phi_x(j)$

- For labeled example x and label $j \in \mathcal{L}$:

$$\phi_x(j) = \begin{cases} 1 & \text{if } j \text{ the label of } x \\ 0 & \text{otherwise} \end{cases}$$

- For unlabeled example x: $\phi_x(j) = \frac{1}{|\mathcal{L}|}$ (ϕ_x is uniform)

Analysis of the Yarowsky algorithm (Abney 2004)

- Minimum threshold $\zeta = \frac{1}{|\mathcal{L}|}$.
- Each example x in U once labeled remains labeled but label can change.
- The algorithm produces a sequence of labelings: $\phi^{(0)}, \phi^{(1)}, \ldots$
- And it produces a sequence of classifiers (model parameters): $\pi^{(1)}, \pi^{(2)}, \ldots$
- Classifier $\pi^{(t+1)}$ is trained on the labeling $\phi^{(t)}$.
- Labeling $\phi^{(t+1)}$ is created using $\pi^{(t+1)}$.
- Assuming that

$$\sum_x D(\phi_x^{(t)} || \pi_x^{(t+1)}) - \sum_x D(\phi_x^{(t)} || \pi_x^{(t)}) \leq 0$$

- (Abney, 2004) shows that H is the objective function:

$$H = \sum_x H(\phi_x) + D(\phi_x || \pi_x)$$

MT-Yarowsky: SSL for machine translation

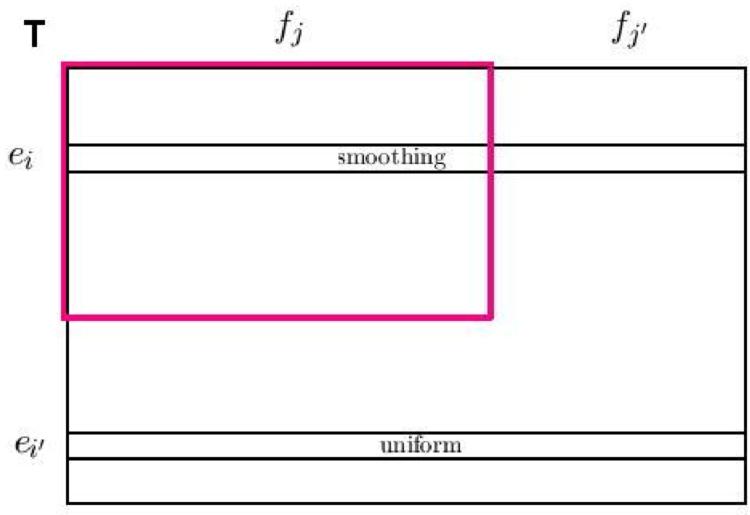
- Machine translation is very different from classification
- Consider an unlabeled instance f: there are many candidate e sentences that could lead to the same Bleu score.
- We want to use the labeling distribution ϕ_f to separate a large number of good translations from a large number of bad translations.

  $\Rightarrow$ Intuition from the splitting and uneven margin ideas from (Shen, Sarkar, Och, 2003) and (Shen and Joshi, 2005)
- We modify the classifier-based Yarowsky algorithm to use a SMT system.
- We use importance sampling to collect all useful translations (possibly sampling multiple translations even for the same source f sentence).

MT-Yarowsky: SSL for machine translation

- *Input*: training set L of parallel sentence pairs.
- *Input*: unlabeled set U of source $\mathbf{f}$ text.
- Set the pool of training data T to L; $t := 0$.
- **repeat**
  - Training step: estimate $\pi^{(t)} = Pr_{\lambda_1^M}(\mathbf{e} \mid \mathbf{f})$ from T.
  - Reset training data: $T = L$; Set $X = \{\}$.
    X will be the set of *confident* translations for this iteration.
  - Labeling step: **for each** sentence $f \in U$:
    Decode f using $\pi^{(t)}$ to obtain n-best sentence pairs:
    $X = X \cup \{(\mathbf{e}, \mathbf{f})\}^n$ with scores $\{\pi_{\mathbf{f}}^{(t)}(\mathbf{e})\}^n$.
  - For $(\mathbf{e}, \mathbf{f}) \in X$, $\pi'(\mathbf{e}) = \left(\pi_{\mathbf{f}}^{(t)}(\mathbf{e})\right)^{\frac{1}{|\mathbf{e}|}}$ (length normalized)
  - Importance sampling to get k sentence pairs: $\{(\mathbf{e}, \mathbf{f})\}^k \sim \pi'(\mathbf{e})$
  - Add $\{(\mathbf{e}, \mathbf{f})\}^k$ to T; $t := t + 1$.
- **until** labeling distribution $\phi_{\mathbf{f}}(\cdot)$ converges

MT-Yarowsky: SSL for machine translation



Inductive vs. Transductive

- Transductive: produce a label only for the available unlabeled data.
  - The output is not a classifier that can be applied to new data.
  - Typically, semi-supervised learning is performed on the test data.
- Inductive: Not only produce label for unlabeled data, but also produce a classifier.
- Analogy from (Zhu, 2005):
  - Transductive learning: take-home exam.
  - Inductive learning: in-class exam.

Inductive vs. Transductive

- However a transductive SVM is an inductive learner! A TSVM can be naturally used on unseen data.
- However, the name TSVM originates from the following argument from (Vapnik, 1998):
  - Learning on the entire data space is solving a more difficult problem.
  - If the task is to annotate the test data, the only work on the observed data (L+T): solve a simpler problem first!
- TSVM can be seen as an alternative way to do supervised learning:

Inductive vs. Transductive

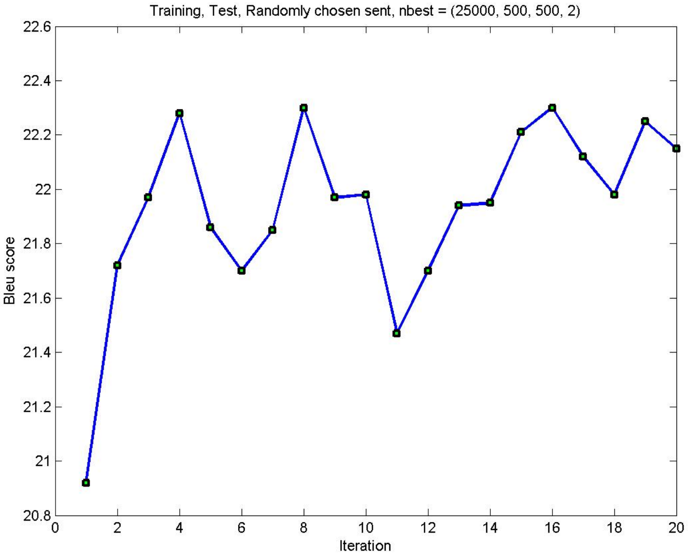
- TSVM can be seen as an alternative way to do supervised learning:
  - Advantages: getting around the i.i.d. assumption by learning a classifier geared towards each test case (or all test cases considered together)
  - For example, in digit recognition, transduction can leverage information in the test data in cases where the test data is all written by the same person.
  - Generative model approach in (Hinton and Nair, 2005).
- In the case of machine translation, transductive learning would be able to adapt to test data from a different domain.

Experimental settings

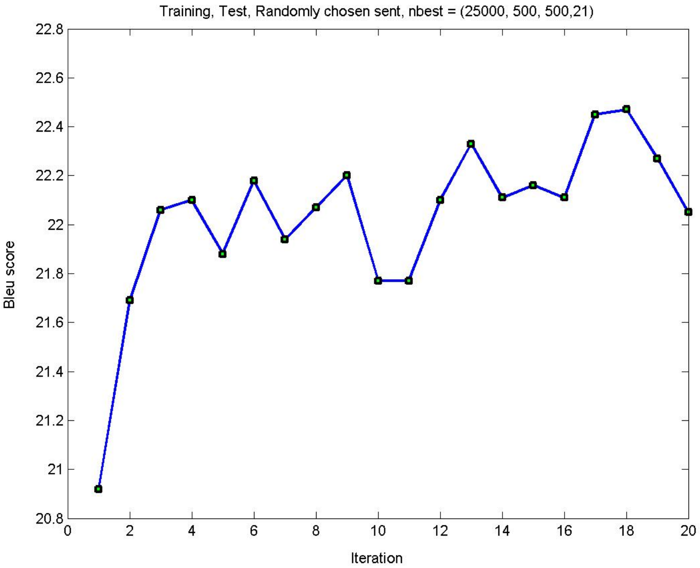
- Dataset = EuroParl corpus from SMT shared task 2006.
- With 25000 sent pairs in training, Bleu% = 20.9;
- With 50000 sent pairs in training, Bleu% = 22.6
- Labeled data set L: 25000 sent pairs.
- Unlabeled data set U = Test set = 500 sentences (transductive learning)
- Expensive decoding of different test and unlabeled data in each bootstrapping iteration is avoided in the transductive setting.
- No reference translations for test set were used for SSL.
- n-best translations: $n = 21$ and $n = 2$.
- Sample size per iteration $k = 500$.

  Note that the same source $\mathbf{f}$ sentence could contribute multiple target $\mathbf{e}$ sentences in each iteration.

MT-Yarowsky: Experiment 1



MT-Yarowsky: Experiment 2



Summary

- Error rate is more stable when sampling from *n*-best list.
- Transductive learning with MT-Yarowsky provides an improvement in the Bleu score is almost equivalent to doubling the training data from 25000 to 50000.

  double training data: 20.9 $\Rightarrow$ 22.6
  MT-Yarowsky SSL: 20.9 $\Rightarrow$ 22.3
- Moving from transductive to inductive learning: avoid re-training full model in the Training step.
- Instead, create a mixture model of phrase pair probabilities from unlabeled data with static phrase probabilities from training data.
- Extension to large data track SMT.