Unsupervised Training for Large Vocabulary Translation Using Sparse Lexicon and Word Classes

Yunsu Kim, Julian Schamper, Hermann Ney
Human Language Technology and Pattern Recognition Group, RWTH Aachen University, Aachen, Germany

Lehrstuhl Informatik 6
Human Language Technology and Pattern Recognition

Unsupervised Machine Translation

- Given: two **monolingual corpora** on source/target (not sentence-aligned)
  - ▶ **No parallel corpora, no seed lexicon**
  - ▶ Target language model (LM): trained beforehand
- To train: word lexicon model $p(f|e)$
- Task assumption: **1-1 monotone** word alignment

Source words f_1^N:	f_1	f_2	...	f_n	...	f_N
	|	|	|	|	|	|
Target words e_1^N:	e_1	e_2	...	e_n	...	e_N

  - ▶ Computationally infeasible to consider phrases and reorderings
- Data preparation / Task setup
  1. Learn word alignments of a parallel corpus
  2. Reorder/Drop source words to make the alignment 1-1 monotonic
  3. Divide the corpus into two parts:

	Source	Target
1st part	Training data	Reference (only for evaluation)
2nd part	-	LM training data

Baseline Framework

- Hidden Markov model (HMM)

$$p(e_1^N, f_1^N) = \prod_{n=1}^{N} \underbrace{p(e_n|e_{n-1})}_{\text{fixed}} \underbrace{p(f_n|e_n;\theta)}_{=\theta_{f|e}}$$

- Training: expectation-maximization (EM) algorithm

$$L(\theta) = \sum_{e_1^N} p(e_1^N, f_1^N)$$

  - ▶ Latent variable: target sentence e_1^N
  - ▶ E-step: compute posteriors $p_n(e|f_1^N)$ (forward-backward algorithm)
  - ▶ M-step: update lexicon table $\theta_{f|e}$
- This work: first attempt at **100k-vocabulary** scenarios

Sparse Lexicon

- Problem 1: full table $\theta_{f|e}$ is **too large to fit in memory**
  - ▶ How can we represent the lexicon efficiently?
- Solution: filter out unlikely entries for each iteration
  1. Select the lexicon entries with a high probability (**threshold** τ)

$$\mathcal{F}(e) = \{f \mid \hat{\theta}_{f|e} \geq \tau\}$$

  2. Renormalize over the selected entries, setting other entries to zero

$$p_{\text{sp}}(f|e) = \begin{cases} \dfrac{\hat{\theta}_{f|e}}{\sum\limits_{f' \in \mathcal{F}(e)} \hat{\theta}_{f'|e}} & \text{if } f \in \mathcal{F}(e) \\ 0 & \text{otherwise} \end{cases}$$

  3. Smooth with a uniform back-off model $p_{\text{bo}}(f)$

$$p(f|e) = \lambda \cdot p_{\text{sp}}(f|e) + (1-\lambda) \cdot p_{\text{bo}}(f)$$

  - ▶ Enforces multinomial sparsity throughout the training
  - ▶ Reduces the model size on the fly
- Results on EuTrans es-en (no pruning)

Lexicon	τ	Accuracy [%]	Memory [%]
Full	-	70.2	100
Sparse	0.005	69.0	2.7
	0.002	**72.3**	**5.1**
	0.0001	70.1	9.1

  - ▶ Outperforms full table by setting τ properly
  - ▶ Greatly reduces the memory usage

Initialization Using Word Classes

- Problem 2: **harsh pruning is inevitable** for large hypothesis lattices
  - ▶ EM algorithm does not converge properly
  - ▶ How can we stabilize the training?
- Solution: learn an initial lexicon on **word class vocabulary**
  1. Estimate word-class mappings on both sides ($\mathcal{C}_{\text{src}}, \mathcal{C}_{\text{tgt}}$)
     - Exchange algorithm, e.g. `mkcls` tool
  2. Map each word in the corpus to its class

$$f \mapsto \mathcal{C}_{\text{src}}(f) \qquad e \mapsto \mathcal{C}_{\text{tgt}}(e)$$

  3. Train a class-to-class full lexicon p_c (using a target class LM)
  4. Convert p_c to a word lexicon score by mapping each class back to its member words (not normalized yet)

$$\forall (e,f) \quad \theta_{f|e} := p_c(\mathcal{C}_{\text{src}}(f)|\,\mathcal{C}_{\text{tgt}}(e))$$

  5. Apply the thresholding and renormalization to 4 (sparse lexicon)
  - ▶ Class vocabulary $\ll$ word vocabulary: marginal increase in memory/time
- Results on EuTrans es-en (pruning with beam size 10)

Initialization			Accuracy [%]
Uniform			63.7
	#Classes	Class LM	
Word Classes	25	2-gram	67.4
	50	2-gram	69.1
	100	2-gram	72.1
	50	3-gram	76.0
	50	4-gram	**76.2**

  - ▶ More performance gain with:
    - larger number of classes
    - better class LMs

Large Vocabulary Experiments

- Corpus statistics

Task		Source (Input)	Target (LM)
Europarl es-en	Running Words	2.7M	42.9M
	Vocabulary	32k	**96k**
IWSLT 2014 ro-en	Running Words	2.8M	13.7M
	Vocabulary	**99k**	**114k**

- Results

Task	Accuracy [%] Supervised	Accuracy [%] Unsupervised	Memory [%]
es-en	77.5	**54.2**	**0.06**
ro-en	72.3	**32.2**	**0.03**

  - ▶ Significantly high accuracy with $< 0.1\%$ memory
  - ▶ Conventional decipherment methods are not applicable

Conclusion and Outlook

- First promising results in 100k-vocabulary unsupervised machine translation
  - ▶ Sparse lexicon = no memory bottleneck + effective model structure
  - ▶ Initialization using word classes = robust training + performance boost
- Outlook
  - ▶ Incorporating local reorderings
  - ▶ Neural network lexicon models
  - ▶ Using more training data and more powerful LMs