Coarse-To-Fine Learning for Neural Machine Translation

[1]Zhirui Zhang, [2]Shujie Liu, [3]Mu Li, [2]Ming Zhou, [1]Enhong Chen

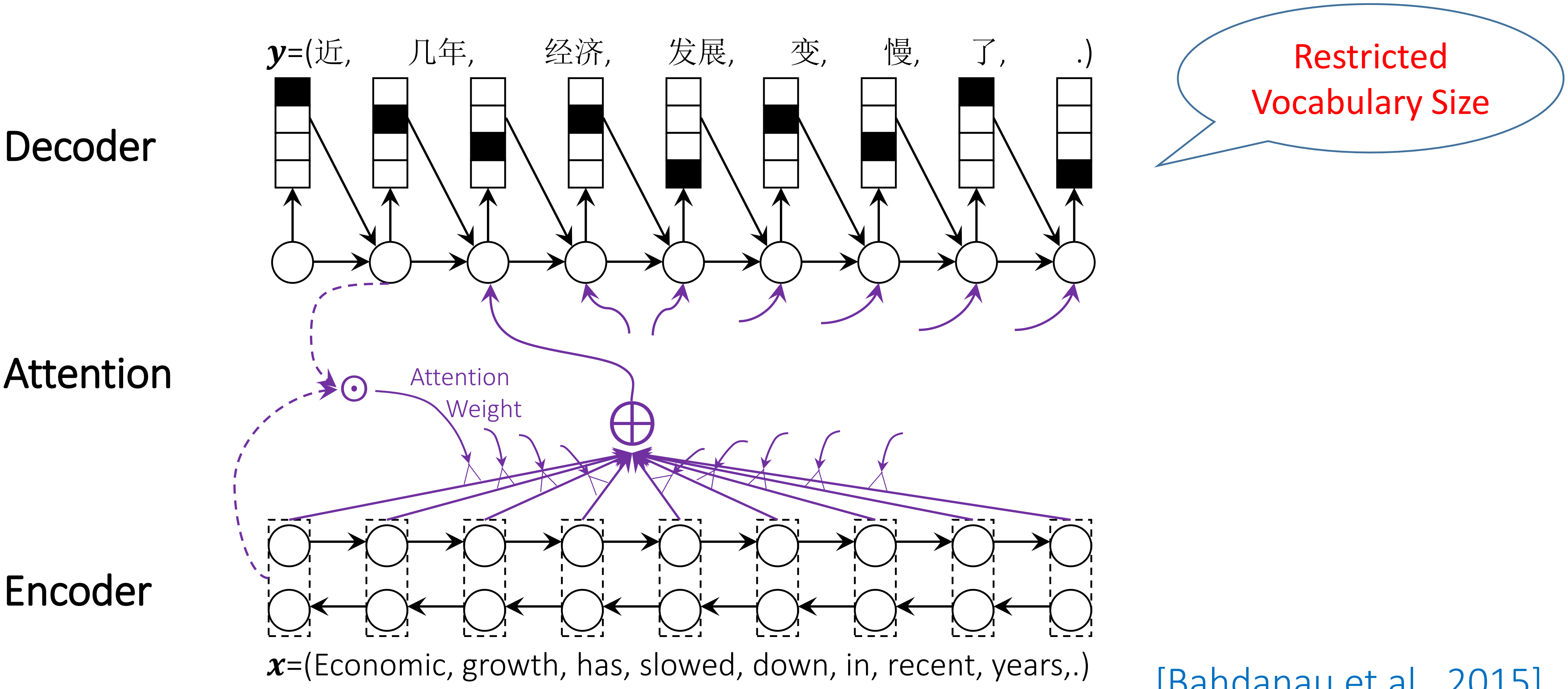
[1]University of Science and Technology of China, Hefei, China

[2]Microsoft Research Asia

[1]zrustc11@gmail.com cheneh@ustc.edu.cn

[2]{shujliu, mingzhou}@microsoft.com [3]limugx@outlook.com

Neural Machine Translation (NMT)



[Bahdanau et al., 2015]

Related Work

- UNK Replacement (Luong et al. 2015)

Related Work

- UNK Replacement (Luong et al. 2015)
- Using Large Vocabulary (Jean et al. 2015)

Related Work

- UNK Replacement (Luong et al. 2015)

- Using Large Vocabulary (Jean et al. 2015)

Input:	他 篡改 老师 与 学生 对话 的 录音 .
Output:	He teacher the recording of teacher and student conversation .
Reference:	He tempered with the recording of conversation between the teacher and the student .

Figure 1: Example of incorrect translation of less-frequent word.

Related Work

- UNK Replacement (Luong et al. 2015)
- Using Large Vocabulary (Jean et al. 2015)
- Byte Pair Encoding (BPE) (Sennrich et al. 2016)
- ...

Our Approach

- Leveraging coarse-to-fine paradigm to learn better NMT model parameters for less-frequent words

- Inspired by a common linguistic observation
  - A group of words belonging to the same syntactic/semantic class tend to share certain properties such as collocations and translations
  - They are expected to be close to each other in embedding space
  - For instance, *large, enormous, gigantic, mammoth*

Coarse-To-Fine Learning Framework

- Conceptually there are two major steps in our coarse-to-fine learning method:
    - Constructing a hierarchical cluster tree
    - Learning a sequence of gradually refined NMT models

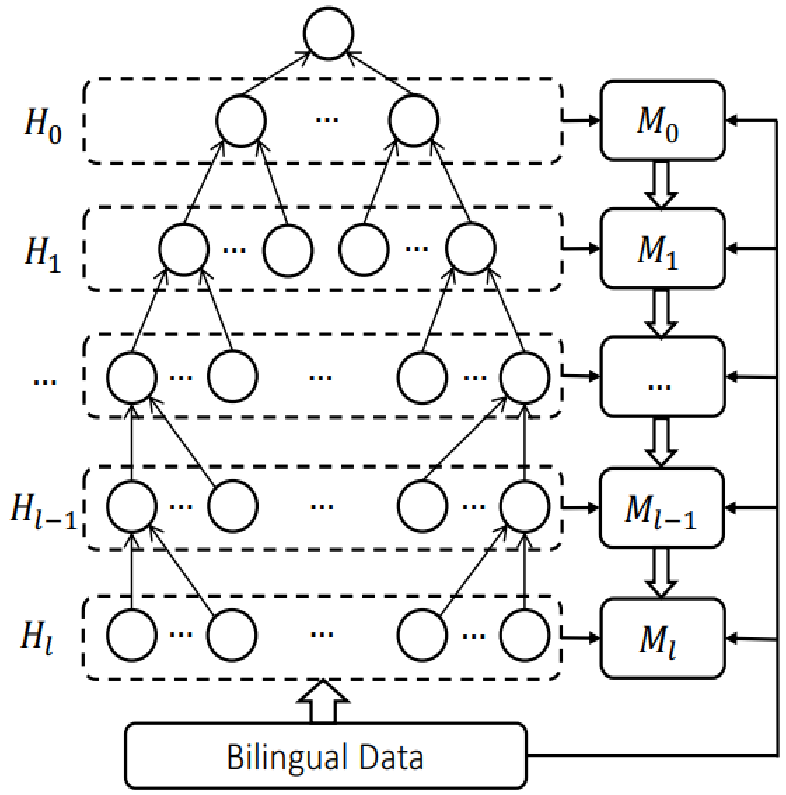


Fig. 2: The coarse-to-fine learning framework for nerual machine translation.

Hierarchical Clustering

- Start with every word as a singleton cluster

$$C_0 = \{a_0 = \{w_0\}, a_1 = \{w_1\}, \ldots, a_n = \{w_n\}\}$$

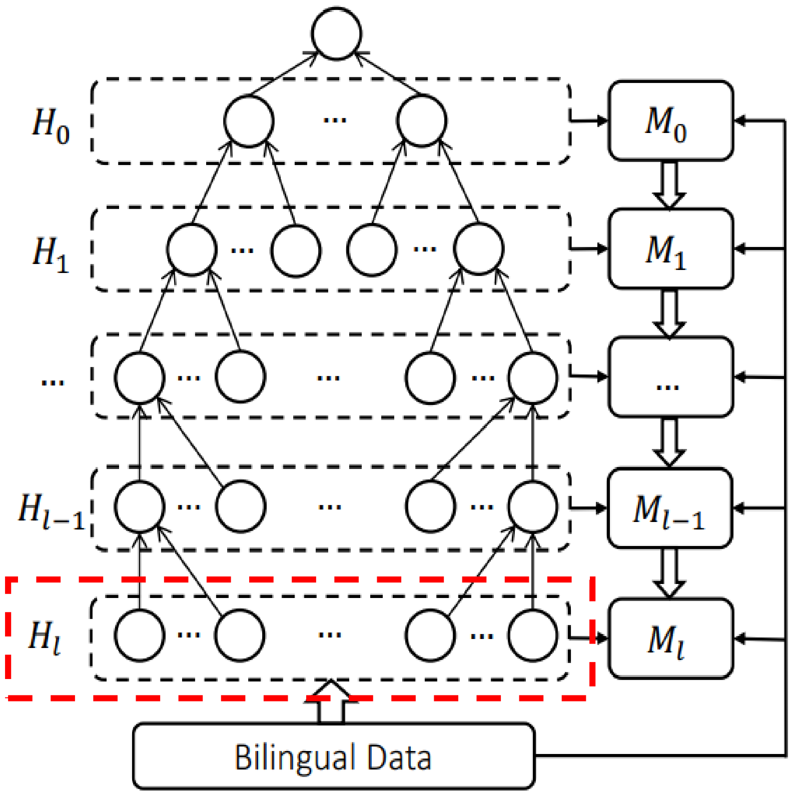


Fig. 2: The coarse-to-fine learning framework for nerual machine translation.

Hierarchical Clustering

- Start with every word as a singleton cluster
- At each step, we calculate the similarity for each pair of clusters and combine two closest clusters to form a new cluster

$$C_{k+1} = (C_k \setminus \{a_u, a_v\}) \cup \{a'\}$$

$$C_0 \to C_1 \to \ldots \to C_V$$

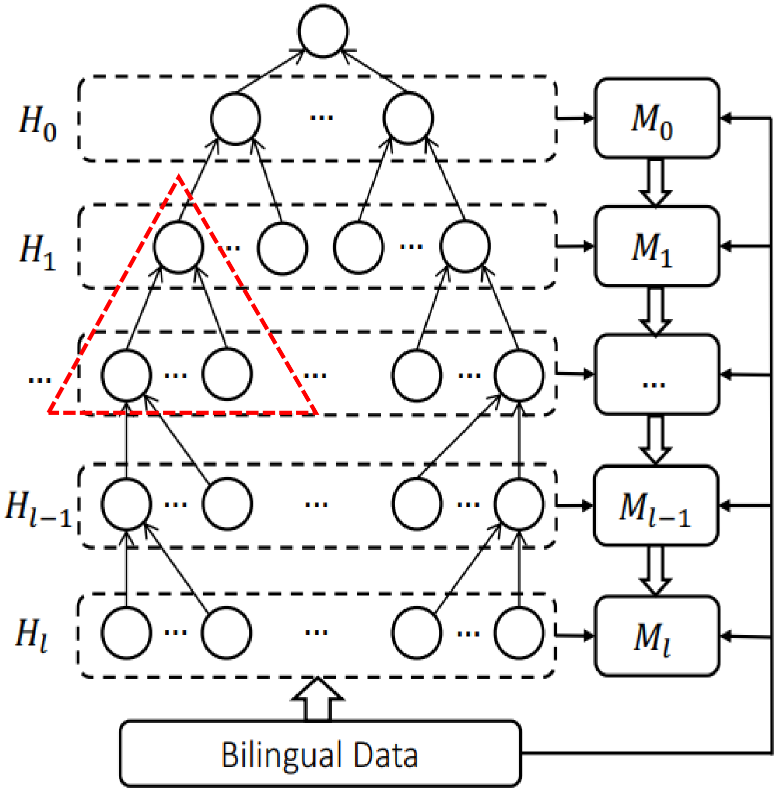


Fig. 2: The coarse-to-fine learning framework for nerual machine translation.

Hierarchical Clustering

- Start with every word as a singleton cluster
- At each step, we calculate the similarity for each pair of clusters and combine two closest clusters to form a new cluster
- $H_0, \ldots, H_l$ are selected in a way that the number of clusters will grow at a geometric rate γ

$$H_i = C_k, \quad n_0\gamma^i = |C_k|$$

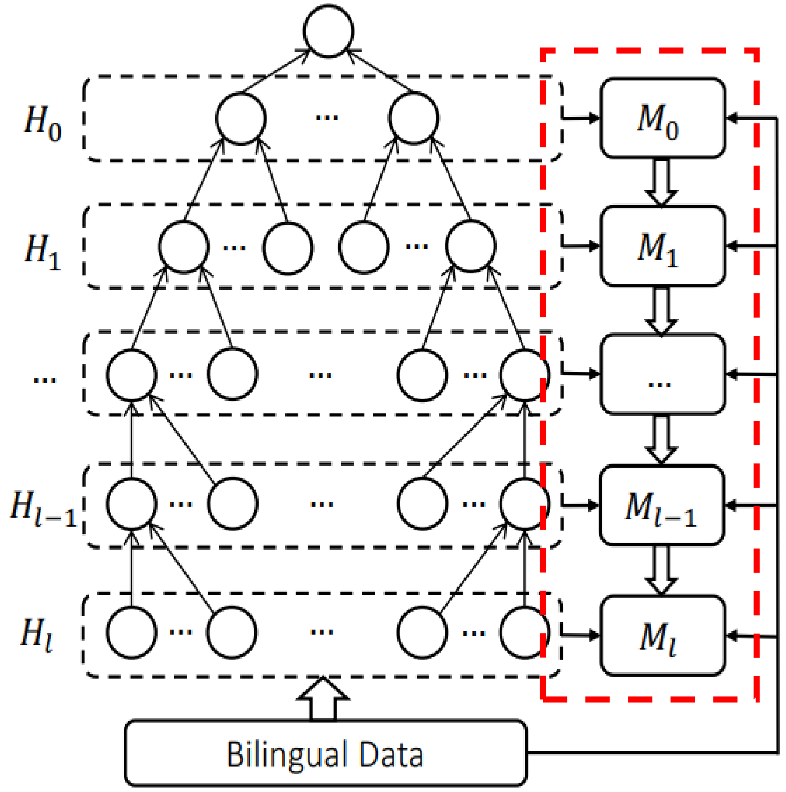


Fig. 2: The coarse-to-fine learning framework for nerual machine translation.

Hierarchical Clustering

- Start with every word as a singleton cluster
- At each step, we calculate the similarity for each pair of clusters and combine two closest clusters to form a new cluster
- $H_0, \dots, H_l$ are selected in a way that the number of clusters will grow at a geometric rate γ

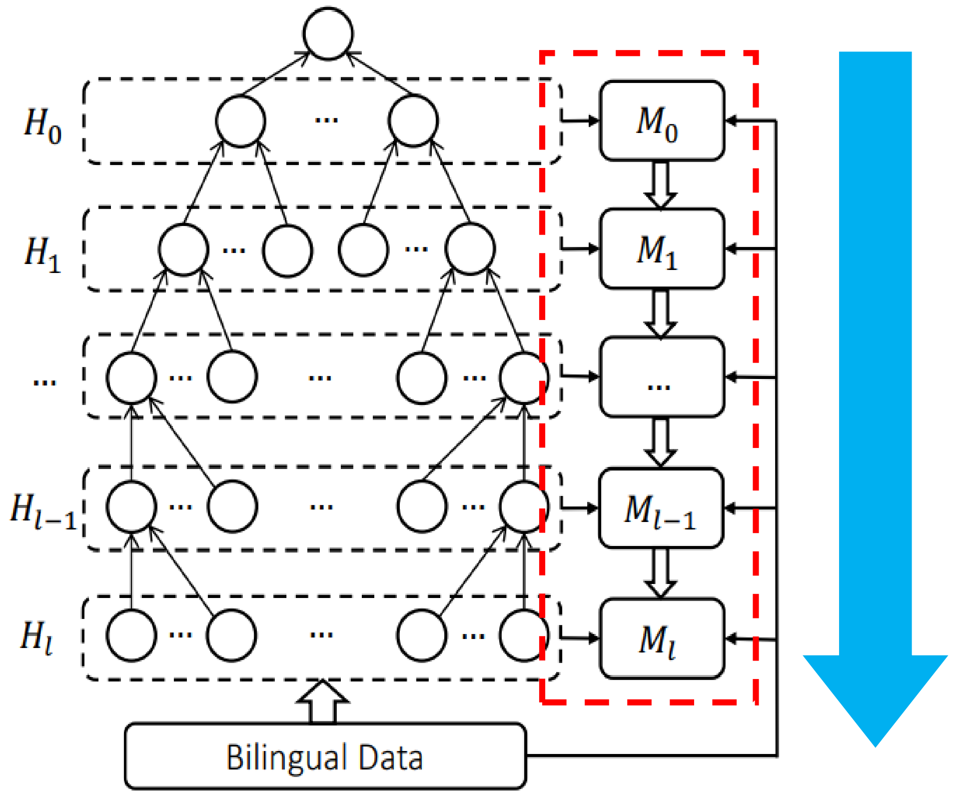


Fig. 2: The coarse-to-fine learning framework for nerual machine translation.

NMT Model Refinement

- Vocabulary Mapping
  - Train M_i on the vocabulary defined by H_i instead of the original vocabulary V

Algorithm 1: Coarse-To-Fine Training Algorithm for NMT

Input : Bilingual data $T = \{(x^n, y^n)\}$;
Validation set D;
Cluster hierarchies $H_0, \ldots, H_l$;
Output: A sequence of NMT models $M_0, \ldots, M_l$;

for $i \leftarrow 0$ **to** l **do**
    if $i == 0$ **then**
        Initialize θ_0 in M_0 ;
    else
        $\theta_i = \Gamma(\theta_{i-1}, H_{i-1}, H_i)$;
    end
    $\{(cx^n, cy^n)\} = \mathrm{Map}(\{(x^n, y^n)\},\ H_i)$;
    for $e \leftarrow 0$ **to** max_epoch **do**
        $\theta_j^e = \arg\max_{\theta_j} \sum_T \log p(cy^n | cx^n)$;
        $ppl^e = \mathrm{CalcPerpelxity}(D, \theta_j^e)$;
        $\Delta PPL = \frac{ppl^{e-1} - ppl^e}{ppl^{e-1}}$;
        if $\Delta PPL < \alpha$ **then**
            break ;
        end
    end
end

NMT Model Refinement

- Vocabulary Mapping
    - Train M_i on the vocabulary defined by H_i instead of the original vocabulary V
- Parameters Inheriting
    - All parameters in model M_{i+1} is inherited from M_i using hierarchical cluster tree

Algorithm 1: Coarse-To-Fine Training Algorithm for NMT

Input : Bilingual data $T = \{(x^n, y^n)\}$;
Validation set D;
Cluster hierarchies $H_0, \ldots, H_l$;
Output: A sequence of NMT models $M_0, \ldots, M_l$;
for $i \leftarrow 0$ **to** l **do**
    if $i == 0$ **then**
        Initialize θ_0 in M_0 ;
    else
        $\theta_i = \Gamma(\theta_{i-1}, H_{i-1}, H_i)$;
    end
    $\{(cx^n, cy^n)\} = \text{Map}(\{(x^n, y^n)\},\ H_i)$;
    for $e \leftarrow 0$ **to** max_epoch **do**
        $\theta_j^e = \arg\max_{\theta_j} \sum_T \log p(cy^n | cx^n)$;
        $ppl^e = \text{CalcPerpelxity}(D, \theta_j^e)$;
        $\Delta PPL = \frac{ppl^{e-1} - ppl^e}{ppl^{e-1}}$;
        if $\Delta PPL < \alpha$ **then**
            break ;
        end
    end
end

NMT Model Refinement

- Vocabulary Mapping
  - Train M_i on the vocabulary defined by H_i instead of the original vocabulary V
- Parameters Inheriting
  - All parameters in model M_{i+1} is inherited from M_i using hierarchical cluster tree
- Switch Condition
  - The perplexity change ratio $\Delta PPL < \alpha$ on validation set D

Algorithm 1: Coarse-To-Fine Training Algorithm for NMT

Input : Bilingual data $T = \{(x^n, y^n)\}$;
Validation set D;
Cluster hierarchies $H_0, \ldots, H_l$;

Output: A sequence of NMT models $M_0, \ldots, M_l$;

```
for i ← 0 to l do
    if i == 0 then
        Initialize θ_0 in M_0 ;
    else
        θ_i = Γ(θ_{i−1}, H_{i−1}, H_i) ;
    end
    {(cx^n, cy^n)} = Map({(x^n, y^n)}, H_i);
    for e ← 0 to max_epoch do
        θ_j^e = arg max_{θ_j} Σ_T log p(cy^n | cx^n) ;
        ppl^e = CalcPerpelxity(D, θ_j^e) ;
        ΔPPL = (ppl^{e−1} − ppl^e) / ppl^{e−1} ;
        if ΔPPL < α then
            break ;
        end
    end
end
```

Experiment

- Evaluate our approach on Chinese-English and English-French translation tasks

- NMT systems
  - **RNNSearch**: attention-based NMT system (Bahdanau et al., 2015)
  - **RNNSearch+BPE**: attention-based NMT system with BPE method (Sennrich et al., 2016)
  - **CTF-NMT**: our coarse-to-fine learning method
  - **CTF-NMT+BPE**: combine our training process with BPE method

Experiment: Chinese-English Translation

- Dataset
  - Bilingual data: 5.2M sentence pairs from LDC corpus
  - Development data: NIST2006
  - Test data: NIST2003, NIST2005, NIST2008
- Result

System	NIST2006	NIST2003	NIST2005	NIST2008	Average
RNNSearch	36.97	39.17	38.97	29.35	36.11
RNNSearch + BPE	37.58	39.73	39.87	30.48	36.92
CTF-NMT	39.14	41.69	41.02	32.66	38.63
CTF-NMT + BPE	**39.72**	**42.20**	**42.24**	**32.90**	**39.26**

Table 1: Case-insensitive BLEU scores (%) on Chinese-English translation. The "Average" denotes the average results of all datasets.

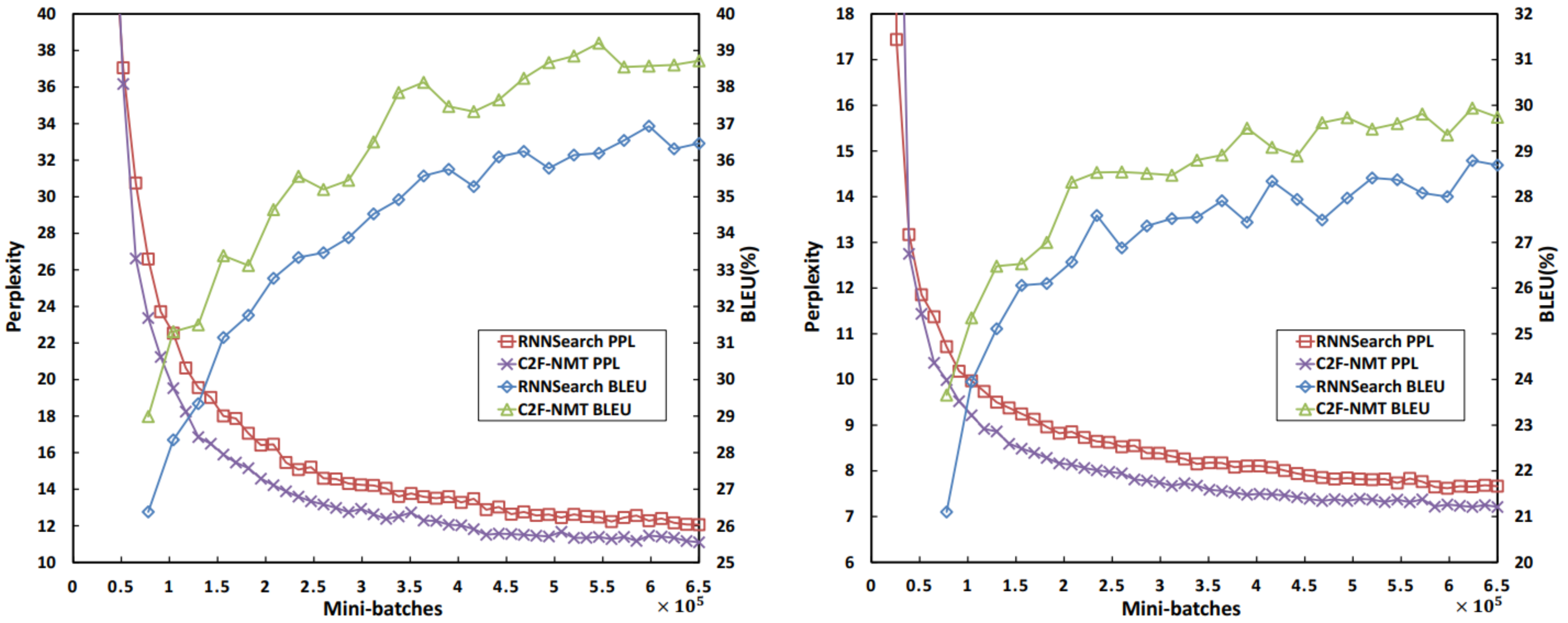
Experiment: English-French Translation

- Dataset
  - Bilingual data: 12M sentence pairs from WMT 2014 training corpus used in Jean et al. 2015
  - Development data: the concatenation of news-test 2012 and news-test 2013
  - Test data: news-test2014
- Result

System	Architecture	Vocab Size	Test
Sutskever et al. (2014)	LSTM with 4 layers	80K	30.59
Luong et al. (2015c)	LSTM with 6 layers + PosUnk	40K	32.70
Shen et al. (2016)	Gated RNN with search + PosUnk + MRT	30K	34.23
Jean et al. (2015)	Gated RNN with search + PosUnk + LV	500K	34.60
Wang et al. (2017)	LAU with 4 layers	30k	35.10
Zhou et al. (2016)	LSTM with 16 layers + F-F connections	30k	35.90
RNNSearch	Gated RNN with search + PosUnk	80K	34.33
RNNSearch + BPE	Gated RNN with search + BPE	80K	35.15
CTF-NMT	Gated RNN with search + PosUnk	80K	35.67
CTF-NMT + BPE	Gated RNN with search + BPE	80K	**36.12**

Table 2: Case-sensitive BLEU scores (%) on English-French translation. The "PosUnk" denotes Luong et al. (2015c)'s technique of handling rare words. The "MRT" denotes minimum risk training proposed in Shen et al. (2016). The "LAU" represents Linear Associative Unit proposed in Wang et al. (2017).

Learning Curve



(a) Chinese-English Translation

(b) English-French Translation

Figure 3: The perplexity (PPL) and BLEU scores on Chinese-English and English-French validation sets for RNNSearch and CTF-NMT as training progresses.

Analysis

- Impact of α

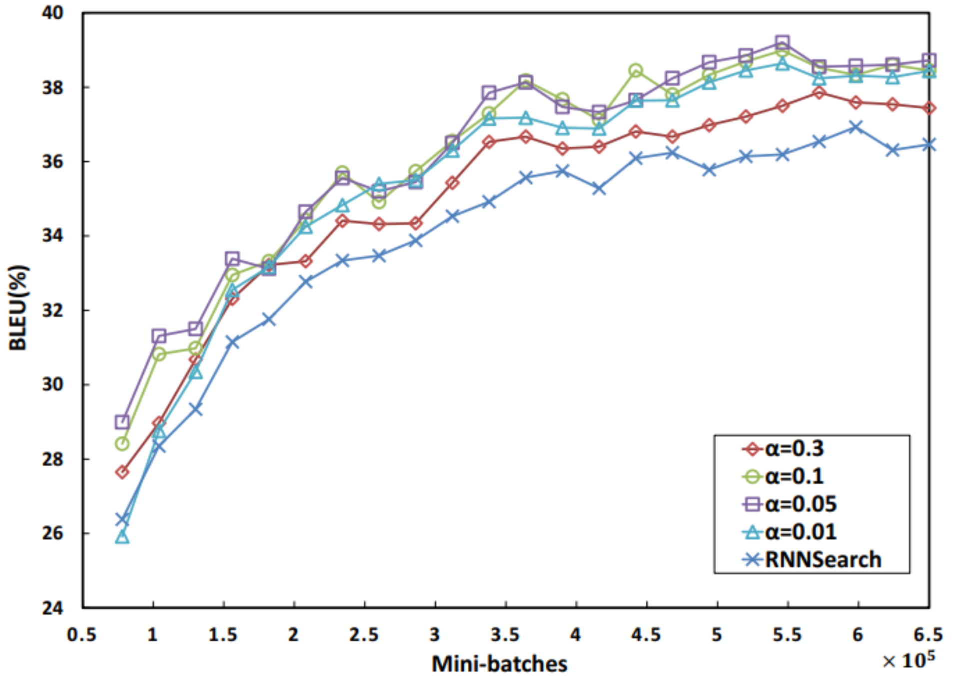


Figure 4: Impact of α on the Chinese-English validation set.

- Impact of γ

	NIST 2006	Average
RNNSearch	36.97	36.11
$l = 1/\gamma = 1000$	38.35	37.60 (+1.49)
$l = 2/\gamma = 100$	38.80	38.26 (+0.66)
$l = 3/\gamma = 10$	39.14	38.63 (+0.37)
$l = 4/\gamma = 6$	39.19	38.66 (+0.03)

Table 3: Impact of γ on Chinese-English translation task.

Conclusion

- Propose a coarse-to-fine learning framework for NMT
  - Constructing a hierarchical cluster tree
  - Building a sequence of NMT models where each model refines its previous one
- Significant improvements on Chinese-English/English-French tasks
- Future work
  - Extend this method to other NLP tasks and seq2seq models (such as Transformer)
  - Explore the possibility to leverage this method to speed-up the training process

Thanks!