Phrase Table Smoothing with Word Classes

Yunsu Kim
kimyunsu@i6.informatik.rwth-aachen.de

Master Thesis Final Talk - August 10, 2015

Human Language Technology and Pattern Recognition
Lehrstuhl für Informatik 6
Computer Science Department
RWTH Aachen University, Germany

Outline

1. Introduction

2. Word Class Models
   - Refinements with the Class Membership Probability
   - Optimizing Word Classes for Word Class Models

3. Translation using Word Classes

4. Translation Examples

5. Conclusion

Motivation: Smoothing with Word Classes

Word class: group of words with similar syntactic/semantic roles

- **Automatically learned from training data (w/o linguistic knowledge)**
- **Examples of words in the same class [Brown & deSouza$^+$ 92]**
  - **Class 1:** had hadn't hath would've could've should've must've might've
  - **Class 2:** head body hands eyes voice arm seat eye hair mouth

Phrase consisting of word classes

- **Simple word-level generalization**

$$\text{word} \longmapsto \text{word class}$$

- **Local context preserved**
- **More robust counts**

State of the Art

1. Word class models

[Wuebker & Peitz$^+$ 13] Improving Statistical Machine Translation with Word Class Models, *EMNLP 2013*.

- **Train existing translation/reordering models with word classes**

2. Translation using word-level labels

[Yang & Kirchhoff 06] Phrase-based Backoff Models for Machine Translation of Highly Inflected Languages, *EACL 2006*.

- **Extract hierarchical paraphrases using a morphological analysis**

[Koehn & Hoang 07] Factored Translation Models, *EMNLP 2007*.

- **Integrate word-level labels as factors in the translation**
- **Word-to-label mapping $\Rightarrow$ Label-to-label translation $\Rightarrow$ Label-to-word generation**

Goals

1. Word clustering ✔
   - Implement bilingual clustering [Och 99] ✔
   - Compare between monolingual/bilingual approach in SMT ✔

   ⇒ No difference in the performance of the word class models

2. Train word alignments from word classes ✔
   - Merge with the original word alignment ✔
   - Extract phrase pairs and train word class models ✔

   ⇒ Degradation of the translation quality

Goals

3. Word class models

- Implement the state-of-the-art [Wuebker & Peitz$^+$ 13] in Jane ✔
- Develop a novel smoothing formulation of the phrase translation model ✔
- Refine word class models with class membership probability ✔
- Optimize word classes for word class models in SMT

4. Translation using word classes

- Paraphrase the extracted phrase pairs using word classes
- Modify the standard phrase-based decoder to use the paraphrases

Phrase-based System: Notations

Phrase segmentation of sentence pair $(f_1^J, e_1^I) = (\tilde{f}_1^K, \tilde{e}_1^K)$

$$k \rightarrow s_k := (i_k; b_k, j_k) \quad \text{for} \quad k = 1, ..., K$$

- $i_k =$ end position of k-th target phrase
- $b_k =$ begin position of k-th source phrase
- $j_k =$ end position of k-th source phrase

Phrase pairs $(\tilde{f}_k, \tilde{e}_k)$

$$\tilde{f}_k := f_{b_k}, \ ..., \ f_{j_k}$$
$$\tilde{e}_k := e_{i_{k-1}+1}, \ ..., \ e_{i_k}$$

Decision rule of log-linear modeling

$$f_1^J \longmapsto \hat{e}_1^{\hat{I}}(f_1^J) = \underset{I, e_1^I}{\operatorname{argmax}} \ \max_{K, s_1^K} \left\{ \sum_{m=1}^{M} \lambda_m h_m(e_1^I, s_1^K; f_1^J) \right\}$$

Word Classes: Notations

Word vocabulary $\mathbb{W}_e$

Class labels $\mathbb{C}$

Word-class mapping $\mathcal{C}$

$$\begin{aligned} \mathcal{C} : \mathbb{W}_e &\rightarrow \mathbb{C} \\ e &\longmapsto \mathcal{C}(e) \\ e_1^I &\longmapsto \mathcal{C}(e_1^I) = \mathcal{C}(e_1),\ ...,\ \mathcal{C}(e_I) \end{aligned}$$

Assumption: each word belongs to one class
(word classes = partitions of words)

Bilingual word-class mapping $(\mathcal{C}_f, \mathcal{C}_e)$

$$\begin{aligned} f &\longmapsto \mathcal{C}_f(f) \\ \tilde{f}_k &\longmapsto \mathcal{C}_f(\tilde{f}_k) = \mathcal{C}_f(f_{b_k}),\ ...,\ \mathcal{C}_f(f_{j_k}) \\ e &\longmapsto \mathcal{C}_e(e) \\ \tilde{e}_k &\longmapsto \mathcal{C}_e(\tilde{e}_k) = \mathcal{C}_e(e_{i_{k-1}+1}),\ ...,\ \mathcal{C}_e(e_{i_k}) \end{aligned}$$

Word Class Translation Models (wcTM)

[Wuebker & Peitz+ 13]

Word class phrase translation model

$$p(\tilde{f}_k|\tilde{e}_k) = p(\mathcal{C}_f(\tilde{f}_k)|\mathcal{C}_e(\tilde{e}_k))$$

Word class IBM1 lexicon model

$$h_{Lex}(e_1^I, s_1^K; f_1^J) = \sum_{k=1}^{K} \log \prod_{j=b_k}^{j_k} \frac{1}{|\tilde{e}_k|} \sum_{i=i_{k-1}+1}^{i_k} p(\mathcal{C}_f(f_j)|\mathcal{C}_e(e_i))$$

- Train existing models with word class corpus
- Similarly applied to:
  - Hierarchical reordering models (HRM) $\Rightarrow$ wcHRM
  - Language model $\Rightarrow$ wcLM
- Generalize every word of the phrase without distinction

Class Smoothing Models (CSM)

Notation: selective class mapping

$$\mathcal{C}^{(\{i\})}(e_1^I) := e_1, \ \ldots, \ \mathcal{C}(e_i), \ \ldots, \ e_I$$

Model CSM_{src}

$$p(\tilde{f}_k|\tilde{e}_k) = \sum_{j=b_k}^{j_k} \frac{w_j}{\sum_{j'} w_{j'}} \cdot p(\mathcal{C}_f^{(\{j\})}(\tilde{f}_k)|\tilde{e}_k)$$

- w_j = averaging weight (e.g. equal weight: $w_j = 1$)

Example:

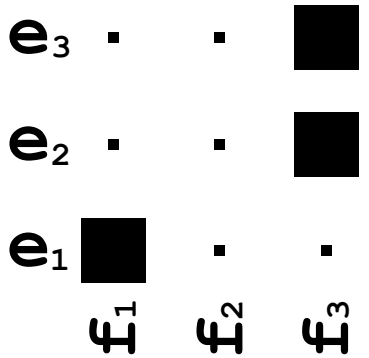
$$\begin{aligned} p(f_1\ f_2\ f_3|e_1\ e_2\ e_3) = \frac{1}{3} \cdot \Big[& p(\mathcal{C}_f(f_1)\ f_2\ f_3 \mid e_1\ e_2\ e_3) \\ & + p(f_1\ \mathcal{C}_f(f_2)\ f_3 \mid e_1\ e_2\ e_3) \\ & + p(f_1\ f_2\ \mathcal{C}_f(f_3) \mid e_1\ e_2\ e_3) \Big] \end{aligned}$$

- More fine-grained generalization than wcTM

Class Smoothing Models (CSM)

Model $\text{CSM}_{\text{src+tgt}}$

$$p(\tilde{f}_k|\tilde{e}_k) = \sum_{j=b_k}^{j_k} \frac{w_j}{\sum_j w_j} \cdot p(\mathcal{C}_f^{(\{j\})}(\tilde{f}_k)|\mathcal{C}_f^{(a_j)}(\tilde{e}_k))$$



Example:

$$p(f_1\ f_2\ f_3 \mid e_1\ e_2\ e_3) = \frac{1}{3} \cdot \Big[\ p(\mathcal{C}_f(f_1)\ f_2\ f_3 \mid \mathcal{C}_e(e_1)\ e_2\ e_3)$$
$$+\ p(f_1\ \mathcal{C}_f(f_2)\ f_3 \mid e_1\ e_2\ e_3)$$
$$+\ p(f_1\ f_2\ \mathcal{C}_f(f_3) \mid e_1\ \mathcal{C}_e(e_2)\ \mathcal{C}_e(e_3))\Big]$$

- Generalize also on the target side
- Alignment information is encapsulated

Class Smoothing Models (CSM)

Weighting over source positions

- (inverse) unigram of the replaced word

$$\frac{1}{w_j} = \frac{N(f_j)}{\sum_{f'} N(f')}$$

- (inverse) source phrase replacement probability

$$\frac{1}{w_j} = \frac{N(f_{b_k} \ldots f_j \ldots f_{j_k})}{\sum_{f'} N(f_{b_k} \ldots f' \ldots f_{j_k})}$$

- factorizing likelihood

$$w_j = N(\mathcal{C}_f^{(\{j\})}(\tilde{f}_k))$$

Refinements with the Class Membership Probability

Class membership probability

$$p(e|\mathcal{C}(e)) = \frac{N(e)}{\sum_{e':\mathcal{C}(e')=\mathcal{C}(e)} N(e')}$$

- **in wcTM**

$$p(\tilde{f}_k|\tilde{e}_k) = \left[\prod_{j=b_k}^{j_k} p(f_j|\mathcal{C}_f(f_j))\right] \cdot p(\mathcal{C}_f(\tilde{f}_k)|\mathcal{C}_e(\tilde{e}_k))$$

- **CSM**

$$p(\tilde{f}_k|\tilde{e}_k) = \sum_{j=b_k}^{j_k} \frac{w_j}{\sum_j w_j} \cdot p(f_j|\mathcal{C}_f(f_j)) \cdot p(\mathcal{C}_f^{(\{j\})}(\tilde{f}_k)|\tilde{e}_k)$$

- **wcLM**

$$p(e_i|e_1^{i-1};\mathcal{C}_e) = p(e_i|\mathcal{C}_e(e_i)) \cdot p(\mathcal{C}_e(e_i)|\mathcal{C}_e(e_{i-n+1}^{i-1}))$$

CSM: Model Comparison

IWSLT 2012 De→En		dev		test	
	Integration	BLEU [%]	TER [%]	BLEU [%]	TER [%]
PBT + wcLM		30.3	50.0	28.3	52.2
+ CSM_{src}	linear	30.4	49.5	28.8	51.5
	log-linear	30.5	49.7	29.0	51.6
+ $CSM_{src+tgt}$	linear	30.9	48.8	29.1	50.9
	log-linear	30.8	48.6	28.9	50.6
+ CSM_{tgt}	linear	30.3	49.6	28.7	51.8
	log-linear	30.4	49.4	29.0	51.4
+ $CSM_{tgt+src}$	linear	29.9	50.1	28.2	52.1
	log-linear	30.7	48.9	29.1	50.9

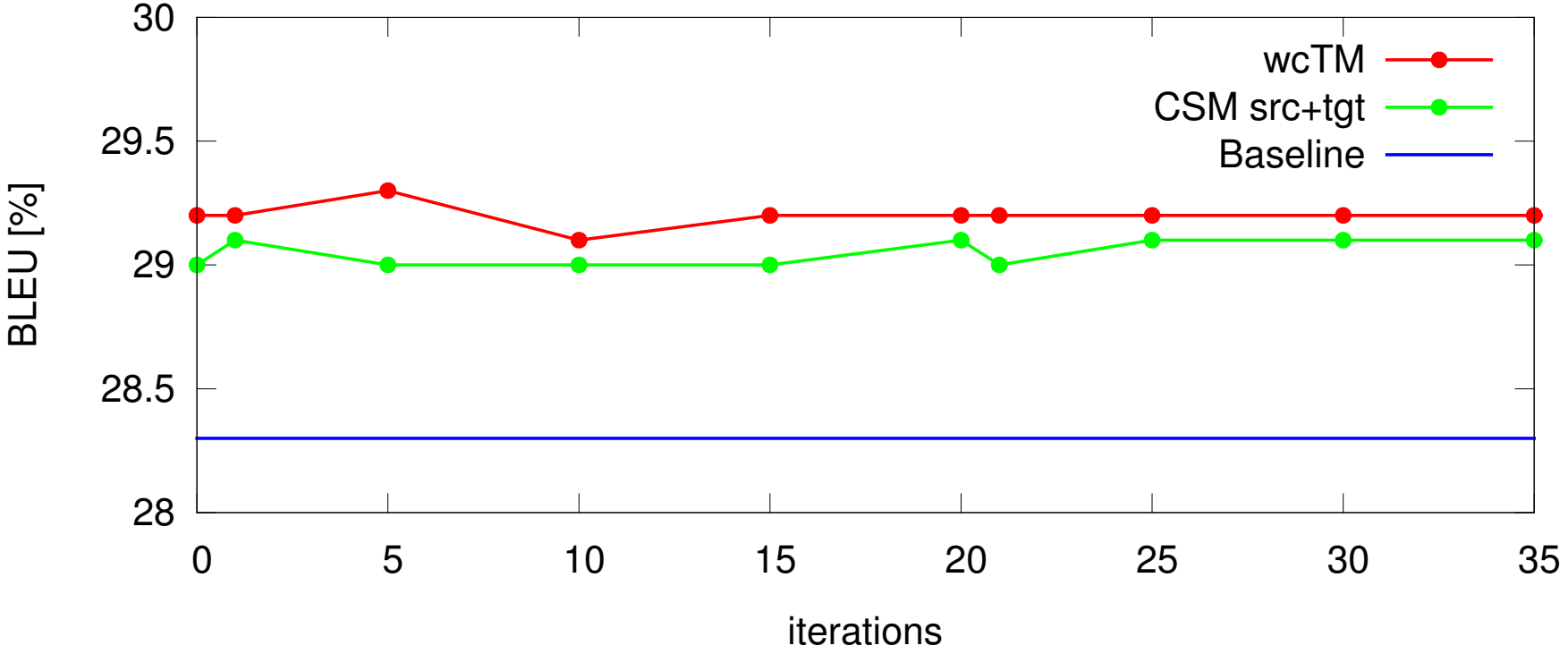
Refinements with the Class Membership Probability

		IWSLT 2012 De→En test		WMT 2014 En→De newstest13		WMT 2015 En→Cs newstest14	
	Refinements	BLEU [%]	TER [%]	BLEU [%]	TER [%]	BLEU [%]	TER [%]
PBT		28.1	51.4	14.3	70.2	20.0	65.3
+ wcLM	none	28.3	52.2	14.6	69.8	20.0	65.5
	membership	28.5	51.5	14.6	69.7	20.4	64.4
+ wcTM	none	29.2	51.2	14.8	69.4	20.4	64.9
	membership	29.2	51.2	15.2	67.8	20.7	64.2
+ $CSM_{src+tgt}$	none	29.1	50.9	15.2	68.5	20.4	65.1
	membership	29.2	50.8	15.2	68.4	20.5	64.9

- Class membership probability enhances the word class models

Optimizing Word Classes for wcTM/CSM

- **Clustering iterations do not affect the translation quality significantly**



- **Similar results for varying:**
  - **Initial clustering**
  - **The number of classes**
  - **Clustering algorithm (monolingual/bilingual)**

Optimizing Word Classes for wcLM

IWSLT 2012 De→En		dev†		test		
	#classes	BLEU [%]	TER [%]	BLEU [%]	TER [%]	Perplexity
PBT		29.8	49.2	28.1	51.4	
+ wcLM	100	30.2	49.2	28.5	51.5	263.3
	200	30.4	49.6	28.4	51.8	231.7
	500	31.5	48.1	29.4	50.6	193.6
	1000	31.5	48.1	29.2	50.4	178.5
	2000	31.3	48.8	29.1	51.3	166.6
	5000	31.0	49.4	29.1	51.3	156.3

- The number of classes must be tuned for the optimal performance of wcLM

Motivation: Translation using Word Classes

Phrase pairs are extracted from a limited amount of bilingual data

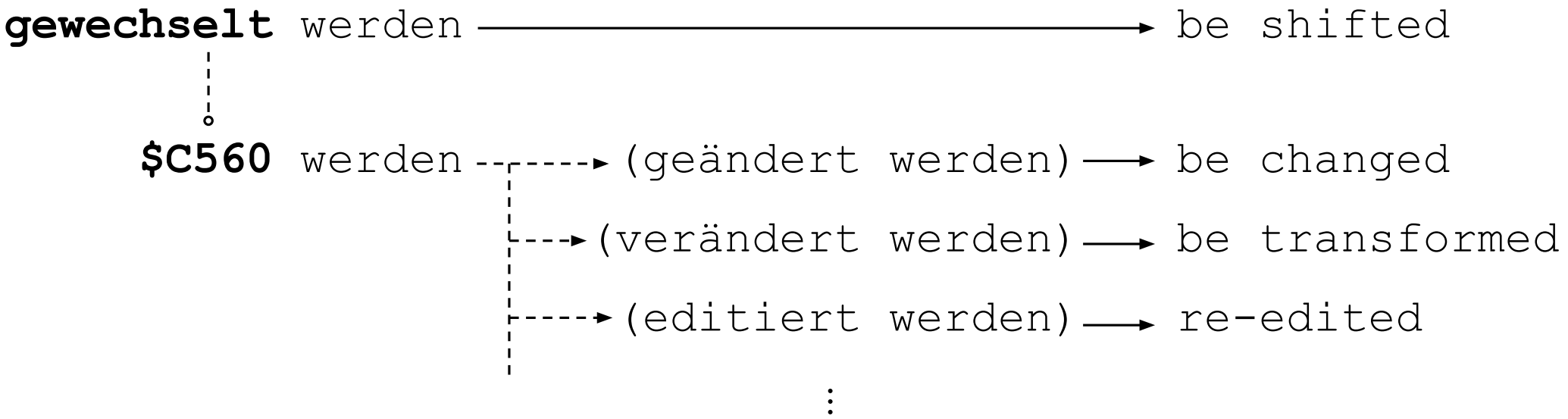
- Might not have all necessary phrases to correctly translate a given test set
- Use more bilingual data: not feasible for low-resource language pairs

Manipulate the extracted phrases with word classes

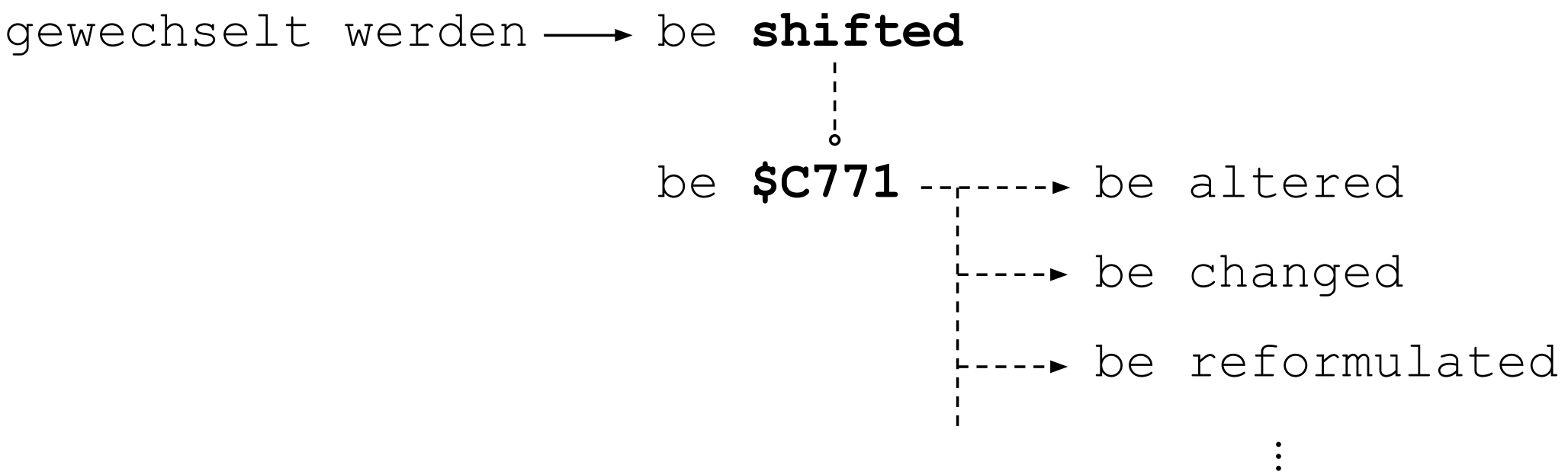
1. Replace a word in a phrase with the corresponding word class
2. Replace the class with other member words within the same class
3. Create new phrase pairs which could not be extracted from the training corpus

- No need for additional bilingual data
- We can reuse CSM phrase pairs

Translation using Word Classes: Source Side



Translation using Word Classes: Target Side



Translation using Word Classes: Details

Parameter τ_u: determine the candidate words for the back-off

- **Back-off only if $N(f_j) \leq \tau_u$ (or $N(e_i) \leq \tau_u$)**
- **Rare words are backed off first**

OOV translation ($N(f_j) = 0$)

- **OOV words of the bilingual training data have no translation options**
- **Even if bilingual data is scarce, monolingual data is easier to obtain**

Estimate word classes from a large monolingual data

- **Might have a class mapping of the OOV words**

  `versprechendes --> $C626`

- **An OOV word can be replaced with other words in the same class**

  `versprechendes --> $C626 --> versprechende`

- **Obtain phrase pairs including OOV words**

  `versprechendes --> versprechende` → `promising`

Translation using Word Classes: Results

IWSLT 2012 De→En		dev			test		
	τ_u	BLEU [%]	TER [%]	#OOV	BLEU [%]	TER [%]	#OOV
PBT + wcLM		30.3	50.0	398	28.3	52.2	484
+ wcTrans	0	29.8	51.2	117	28.6	52.6	89
	100	30.1	50.0	117	28.4	51.8	89
	1000	30.1	49.6	117	28.3	51.3	89
	∞	29.2	50.5	117	27.9	52.1	89
+ wcTrans (w/o OOV)	100	30.1	50.0	398	28.4	51.8	484
	1000	30.4	49.4	398	28.7	51.7	484
	∞	30.6	49.2	398	29.2	51.0	484

wcTrans: including OOV translation

- 2000 classes from a large monolingual corpus (∼0.7B words)

wcTrans (w/o OOV): excluding OOV translation

- 100 classes from the bilingual training data (∼2.5M words)

CSM: Translation Examples

Source	unsere Zeit Schriften werden von Millionen gelesen .
Reference	our magazines are read by millions .
PBT + wcLM	our magazines are killed by millions .
+ $CSM_{src+tgt}$	our magazines will read from millions .

Source	viele von ihnen sind von Gesichtern verdeckt usw.
Reference	so many of them are occluded by faces , and so on .
PBT + wcLM	many of them are of faces , and so on .
+ $CSM_{src+tgt}$	many of them are covered by faces , and so on .

Translation using Word Classes: Examples

Source	… ein neues und viel versprechendes Erlebnis …
Reference	… a new and promising experience …
PBT + wcLM	… a new and much UNKNOWN experience …
wcTrans	… a new and promising experience …

viel $C626 → promising

Source	… mit der wir diese kleinen Stücke zusammensetzen und die Fehler korrigieren konnten .
Reference	… for putting these little pieces together and correct all the errors .
PBT + wcLM	… with which we have these little pieces , and the correct mistakes .
wcTrans	… with which we have these little pieces together and to correct the mistakes .

$C74 Fehler → the mistakes

Conclusion

1. CSM has a competitive performance to wcTM with a smaller number of models
   - improves PBT systems by up to +0.9 BLEU and -1.4 TER
2. Performance of word class models are enhanced by integrating the class membership probability
   - up to +0.4 BLEU and -1.6 TER
3. Performance of wcTM and CSM is not significantly affected by:
   - clustering iterations
   - initial clustering
   - the number of classes
   - clustering algorithm

Conclusion

4. **Performance of wcLM is considerably affected by increasing the number of classes**
   - **up to +0.9 BLEU and -0.9 TER**
5. **Using word classes, we can paraphrase the extracted phrase pairs to enlarge the search space**
   - **enables OOV translation**
   - **improves the PBT systems by up to +0.9 BLEU and -1.2 TER**

Future Work

1. Word class models beyond the phrase context
   - Train joint translation and reordering models (JTR) with word classes
2. More variants of wcLM
   - LM with other word-level labels (e.g. morphological stems, part-of-speech tags)
   - Word classes as inputs to neural language models
   - Combination of various wcLMs
3. More study on translation using word classes
   - Find the optimal usage of OOV translation
   - Test on large-scale SMT tasks

Thank you for your attention

Yunsu Kim

kimyunsu@i6.informatik.rwth-aachen.de

http://www-i6.informatik.rwth-aachen.de/

References

[Brown & deSouza+ 92] P.F. Brown, P.V. deSouza, R.L. Mercer, V.J.D. Pietra, J.C. Lai: Class-based n-gram Models of Natural Language. *Computational Linguistics*, Vol. 18, No. 4, pp. 467–479, December 1992.

[Koehn & Hoang 07] P. Koehn, H. Hoang: Factored Translation Models. In *2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)*, pp. 868–876, Prague, Czech Republic, June 2007.

[Och 99] F.J. Och: An Efficient Method for Determining Bilingual Word Classes. In *9th Conference on European Chapter of the Association for Computational Linguistics (EACL 1999)*, pp. 71–76, Bergen, Norway, June 1999.

[Wuebker & Peitz+ 13] J. Wuebker, S. Peitz, F. Rietig, H. Ney: Improving Statistical Machine Translation with Word Class Models. In *2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pp. 1377–1381, Seattle, USA, October 2013.

[Yang & Kirchhoff 06] M. Yang, K. Kirchhoff: Phrase-based Backoff Models for Machine Translation of Highly Inflected Languages. In *11th Conference on*

***European Chapter of Association for Computational Linguistics (EACL 2006)*, pp. 3–7, Trento, Italy, April 2006.**

Appendix: Corpus Statistics (IWSLT 2012 De→En)

		German	English
train*	Sentences	130K	
	Running Words	2.5M	2.5M
	Vocabulary	71K	49K
dev	Sentences	883	
	Running Words	20K	21K
	Vocabulary	4K	3K
	OOVs (Rate)	776 (4%)	639 (3%)
test	Sentences	1565	
	Running Words	32K	27K
	Vocabulary	5K	5K
	OOVs (Rate)	1068 (3%)	753 (2%)

* Web Inventory of Transcribed and Translation Talks (WIT3) 2012-03

Appendix: Corpus Statistics (WMT 2014 En→De)

		English	German
train*	Sentences	4M	
	Running Words	104M	105M
	Vocabulary	648K	659K
newstest11	Sentences	3003	
	Running Words	66K	81K
	Vocabulary	14K	13K
	OOVs (Rate)	2128 (3%)	1736 (2%)
newstest12	Sentences	3003	
	Running Words	73K	81K
	Vocabulary	10K	13K
	OOVs (Rate)	1827 (2%)	1688 (2%)
newstest13	Sentences	3000	
	Running Words	56K	70K
	Vocabulary	13K	12K
	OOVs (Rate)	1426 (2%)	1310 (2%)

* Europarl v7 + Common Crawl + News Commentary v9 + newstest08/09/10

Appendix: Corpus Statistics (WMT 2015 En→Cs)

		English	Czech
train*	Sentences	930K	
	Running Words	2.4M	2.1M
	Vocabulary	161K	345K
newstest12	Sentences	3003	
	Running Words	73K	65K
	Vocabulary	10K	17K
	OOVs (Rate)	1336 (2%)	2393 (4%)
newstest13	Sentences	3000	
	Running Words	65K	57K
	Vocabulary	9K	15K
	OOVs (Rate)	1170 (2%)	2023 (4%)
newstest14	Sentences	3003	
	Running Words	69K	60K
	Vocabulary	9K	16K
	OOVs (Rate)	1298 (2%)	2190 (4%)

* Europarl v7 + Common Crawl + News Commentary v10

Appendix: Optimizing Word Classes for wcLM

wcLM + `membership` factorizes over words (not word classes)

$\Rightarrow$ possible to compare the perplexity with LM

Corpus	Perplexity		
	LM	wcLM + `membership`	LM + wcLM + `membership`
IWSLT 2012 De$\rightarrow$En	105.43	263.27	104.08
WMT 2014 En$\rightarrow$De	636.67	1206.71	510.83
WMT 2015 En$\rightarrow$Cs	624.88	1851.37	500.72

Appendix: Translation using Word Classes: Technical Details (1)

In phrase extraction:

1. **Create CSM phrase pairs with its standard model scores**

```
2.80336 0.81093 5.02395 4.13453 ...  # $C560 werden # be changed # ...
```

2. **Append the CSM phrase entries to the original phrase table**

In phrase matching:

1. **For each source phrase, back off each source word f_j if $N(f_j) \leq \tau_u$**

$$\texttt{gewechselt werden} \dashrightarrow \texttt{\$C560 werden}$$

2. **Query the phrase table with the backoff phrase**

$$\texttt{\$C560 werden} \rightarrow \texttt{be changed}$$

3. **Store the query result in the separate matching list**

$$\texttt{gewechselt werden} \rightarrow \texttt{be changed}$$

Appendix: Translation using Word Classes: Technical Details (2)

4. For each target phrase, back off each target word e_i if $N(e_i) \leq \tau_u$

   `be shifted --> be $771`

5. Replace the replaced class with its member words

   `be $771` $\rightarrow$ `be altered`

6. Store the expanded phrase in the separate matching list

   `gewechselt werden` $\rightarrow$ `be altered`

In translation: expand hypotheses with the original phrase matching list and CSM phrase matching list

- Same pruning parameters are applied to both lists

Appendix: Translation using Word Classes: Technical Details (3)

CSM phrase pairs aggregate the counts with respect to the classes

- **Different scale of scores from the original phrase pairs**
- **Balanced by introducing the membership probabilities of the class replacements**

Additional model for the class membership probabilities

$$h_{mbs}(e_1^I, s_1^K; f_1^J) = \sum_{k=1}^{K} \left[\sum_{j=b_k}^{j_k} 1_{B_f}(j) \cdot \log p(f_j|\mathcal{C}_f(f_j)) + \sum_{i=i_{k-1}+1}^{i_k} 1_{B_e}(i) \cdot \log p(e_i|\mathcal{C}_e(e_i)) \right]$$

- B = **a set of back-off positions**
- **For non-back-off positions, the score is zero**
- **The weight λ_{csm} is automatically tuned with MERT**