Online Language Model Adaptation via N-gram Mixtures for Statistical Machine Translation

Germán Sanchis-Trilles
Instituto Tecnológico de Informática, Universidad Politécnica de Valencia, Spain

Mauro Cettolo
FBK - Ricerca Scientifica e Tecnologica, Trento, Italy

Saint-Raphaël, May 27-28, 2010

Outline

- Introduction
- Model adaptation
- Experiments
- Future work
- Conclusions

Introduction

- Aimed towards introducing more context in the system
- Key idea: enhance target LM by introducing parameters that are adapted to the input text
- LM is implemented as mixture of sub LMs
- Experiments on Europarl v2 task (WMT06)

Model adaptation

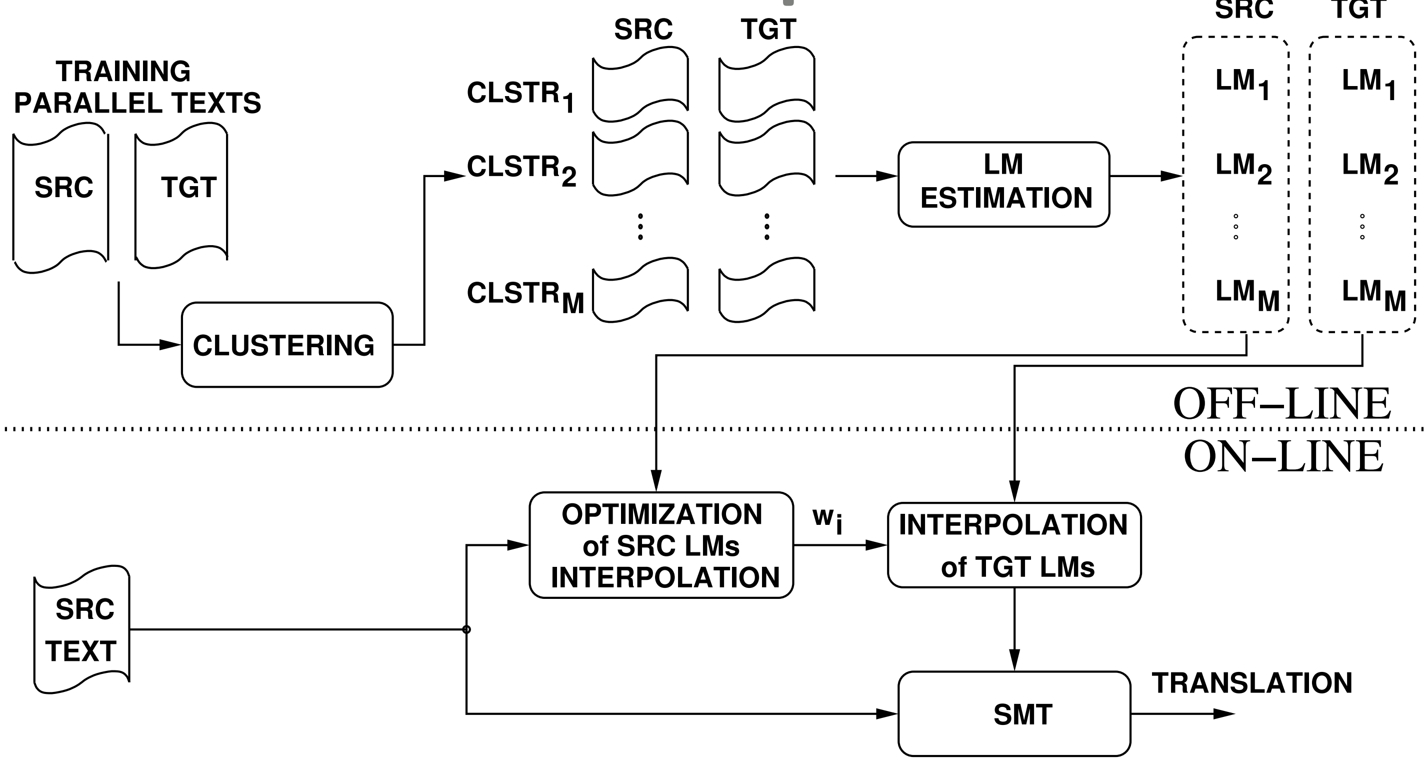
- Most usual translation rule:

$$\mathbf{e}^* = \underset{\mathbf{e}}{\text{argmax}} \sum_{r=1}^{R} \lambda_r h_r(\mathbf{e}, \mathbf{f})$$

- LM can be computed either as a single LM or as a mixture of LMs, i.e.:

$$p(\mathbf{e}) = \sum_{i=1}^{M} w_i p_i(\mathbf{e})$$

Model adaptation



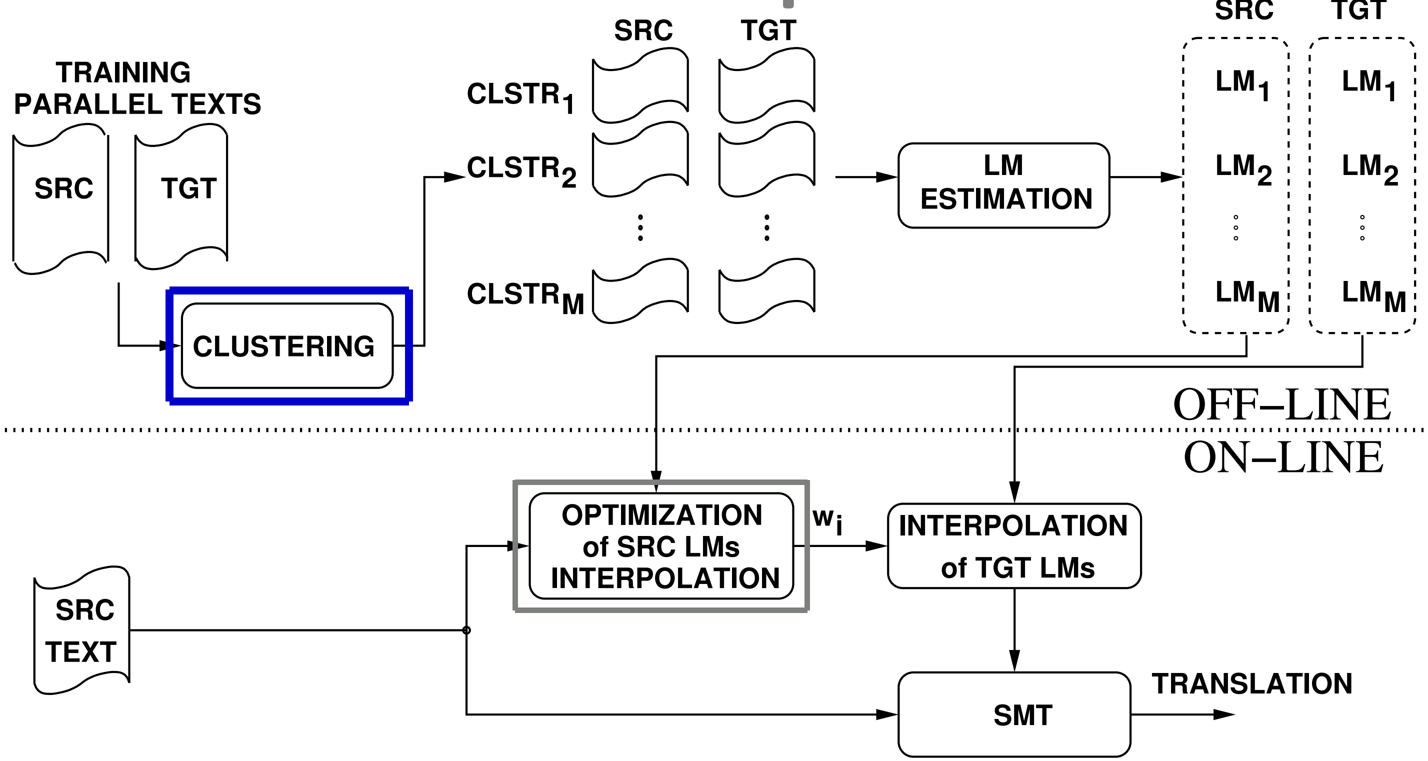
→ Assume a partition of the parallel training data into M bilingual clusters

→ Train specific source/target LMs for each partition

→ Before translation, estimate the optimal weights of source LMs via EM

→ Transfer the resulting weights to the target LM mixture

Model adaptation



→ **Assume a partition of the parallel training data into M bilingual clusters**

→ Train specific source/target LMs for each partition

→ Before translation, estimate the optimal weights of source LMs via EM

→ Transfer the resulting weights to the target LM mixture

Model adaptation: clustering

- Goal: group similar sentences from the lexical point of view
- Sentence pair represented as bag of source and target words
- CLUTO package used, direct k-way partitioning and cosine distance
- Number of clusters set to 4 according to preliminary investigation
- Additional LM built on the whole training data

$\Rightarrow$ First clustering approach: direct clustering of training data

Clustering: Development-induced

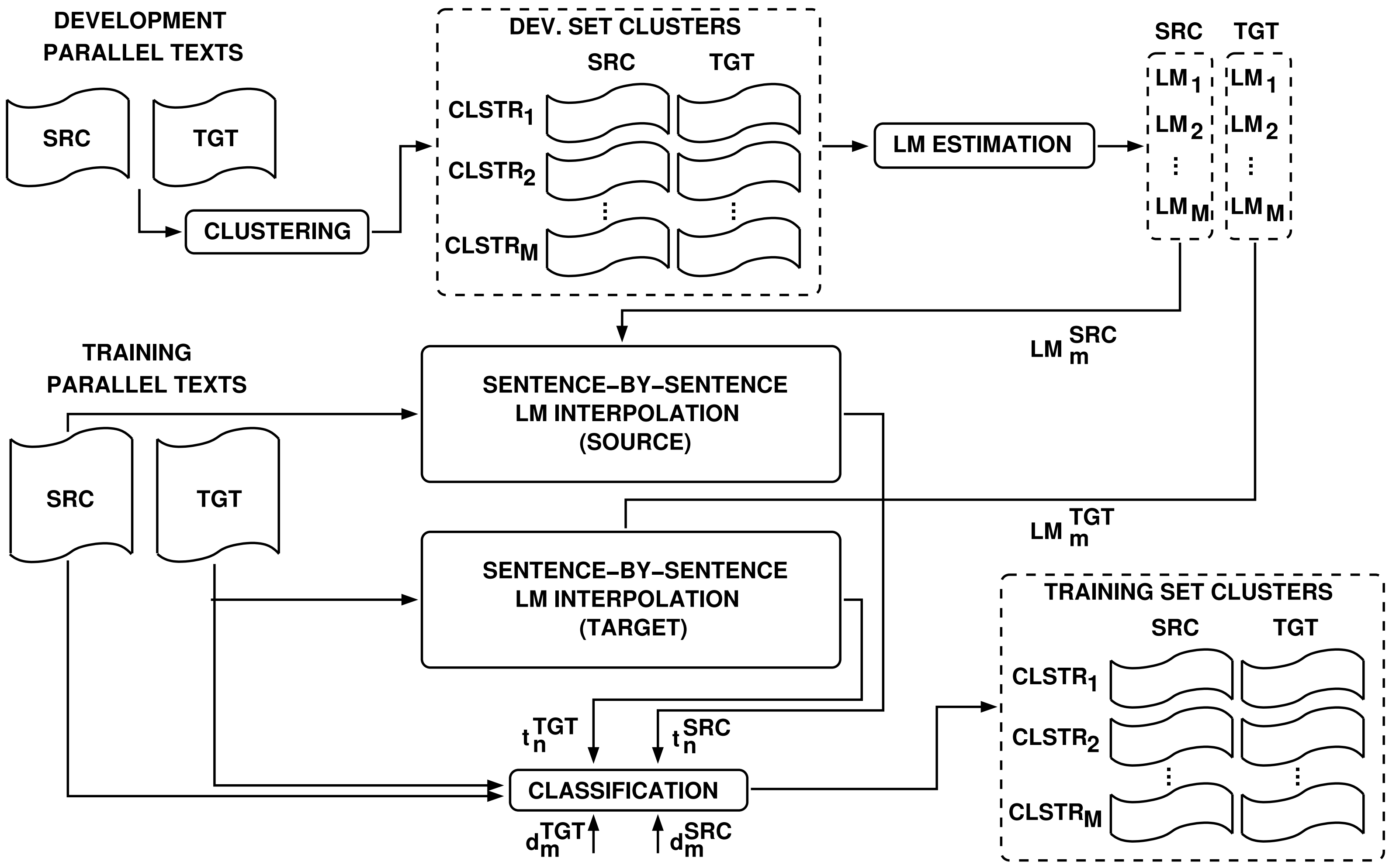
- Adaptation: cover mismatches between training and development/test
  $\rightarrow$ direct clustering may not be the best choice

$\Rightarrow$ Cluster development set and mirror it on training data

1. Cluster bilingual development set
2. Estimate source and target LMs for each cluster
3. For each training sentence:
   - Compute best interpolation of cluster-LMs, in source and target sides
   - Classify it according to most-weighted LMs

- Intuitively:
  - LM is a compact representation of the cluster
  - weights in the optimization provide a measure of similarity

Clustering: Development-induced



Clustering: Test-induced

- Test data can be used to induce the clusterings

⇒ Target side is not available

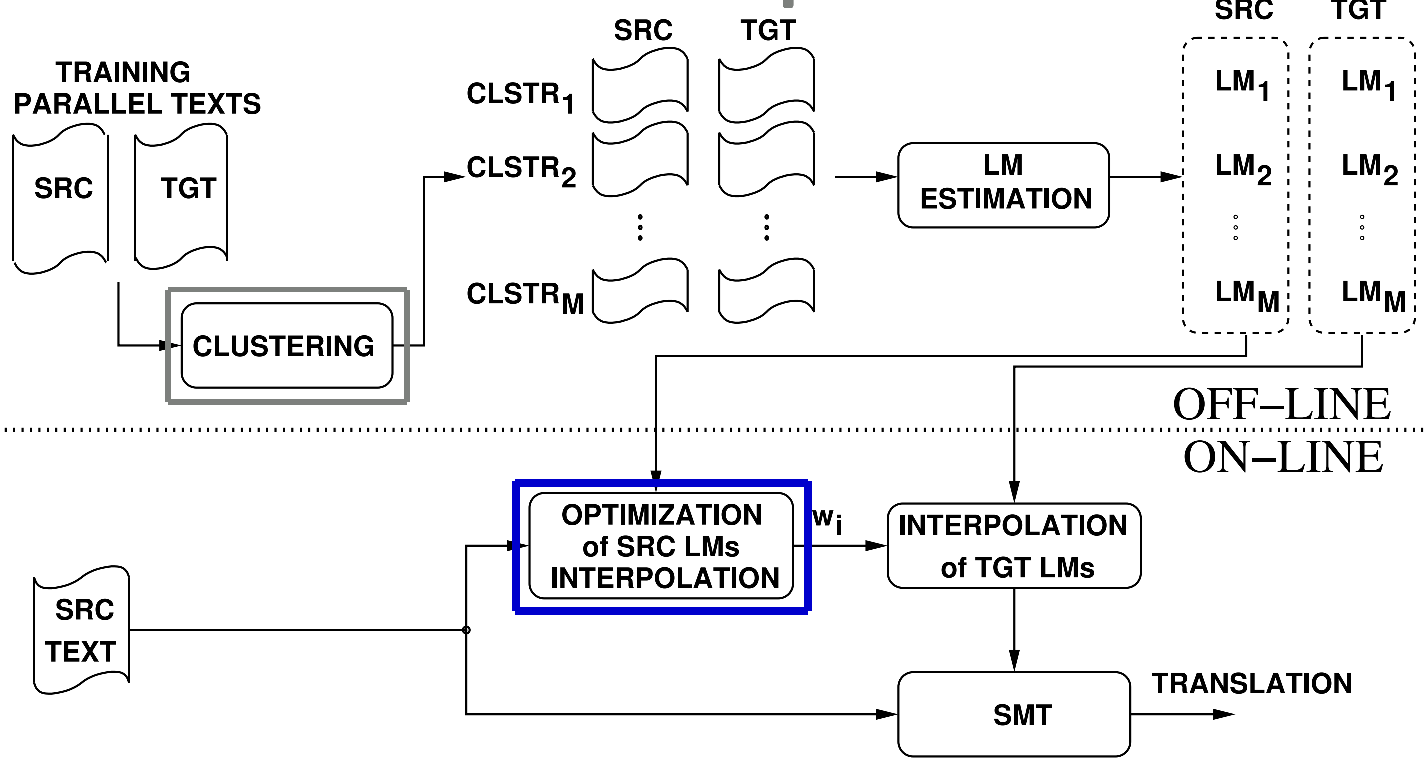
⇒ Only relies on source data, but used to classify both sides!

⇒ May not lead to reliable benefits

⇒ Take advantage of information of the actual test

⇒ Clustering performed only on source data, analogously as for dev-induced

Model adaptation



→ Assume a partition of the parallel training data into M bilingual clusters

→ Train specific source/target LMs for each partition

→ **Before translation, estimate the optimal weights of source LMs via EM**

→ Transfer the resulting weights to the target LM mixture

On-line weight optimization

Three different approaches:

a) Set specific weights

b) Sentence specific weights

c) Two-steps weight estimation

On-line weight optimization

Three different approaches:

a) Set specific weights:

- LM weights estimated on the source side of the complete test set

+ Straightforward
– Does not consider differences between sentences

$\Rightarrow$ benefit of approach may fade

On-line weight optimization

Three different approaches:

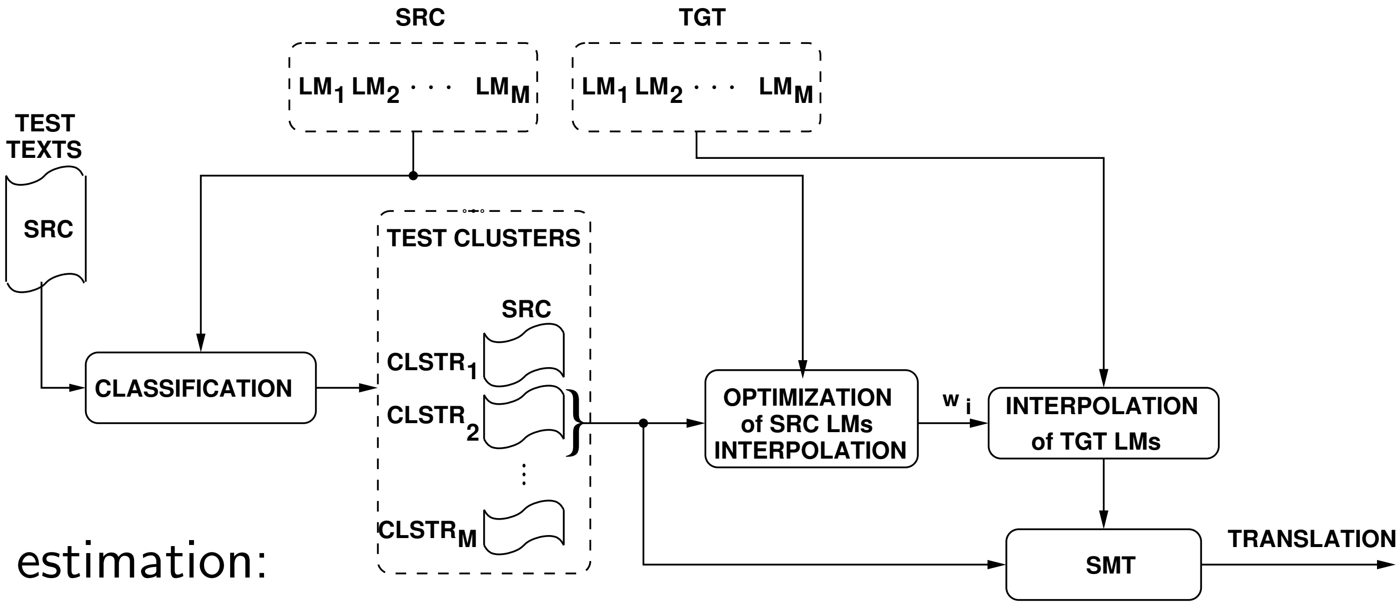
b) Sentence specific weights:

- $*$ One set of weights for each sentence in the test set

$+$ EM procedure allowed complete freedom
$-$ Weights estimated on few data
$\Rightarrow$ possibly, not very reliable weights

On-line weight optimization

Three different approaches:



c) Two-step weight estimation:

1. Estimate sentence-specific weights
2. Assign each source sentence to the cluster with the most weighted LM
3. Re-estimate one single set of weights for each of such clusters

+ Mirror the clustering of the training data into the test set
+ Avoid possible data sparseness issues

Experiments: Corpora

- Experiments conducted on the Europarl corpus (setup of WMT06)
- Consists of transcription of European Parliament speeches
- Experiments conducted on De–En, Es–En and Fr–En, both directions

		De	En
Training	Sentences	751K	
	Run. words	15.3M	16.1M
	Voc.	195K	66K
Dev.	Sentences	2000	
	Run. words	55K	59K
	OoV	432	125
Test	Sentences	2000	
	Run. words	54K	58K
	OoV	377	127

Experiments: Baseline system

- Built upon Moses SMT toolkit. Log-linear model with
  - → Phrase-based translation model
  - → Language model
  - → Word and phrase penalties
  - → Distortion model
- Weights of the log-linear combination optimized with MERT
- Language model: 5-gram with KN smoothing
- Distortion model: "`orientation-bidirectional-fe`"

Experiments

- 10K bootstrap repetitions, 95% confidence level pairwise improvement

Clustering method	Weight optimization	BLEU	TER	Signif BLEU/TER
—	baseline	19.0	67.4	—
direct	sentence	18.2	67.4	yes/no
	two-steps	18.1	67.4	yes/no
	test set	18.0	67.6	yes/no
dev-induced	sentence	**19.2**	**66.7**	yes/yes
	two-steps	**19.2**	**66.7**	yes/yes
	test set	18.7	67.2	yes/no
test-induced	sentence	18.9	67.3	no/no
	two-steps	18.9	67.3	no/no
	test set	18.9	67.1	no/yes

General remarks

- Best results achieved when using:
  - development-induced clustering
  - two-steps (or sentence-based) weight optimization
- Results found to be statistically significant and coherent
- `sentence` and `two-steps` weighting schemes yield similar results
  → For long sentences, `sentence` is best (cheaper)
- Test and development sets are extracted from a narrow time frame
  → development-induced clustering exploits un-even distribution of data better
- Test clustering relies on monolingual data
  → Much less information for clustering (less than half of it!)

Conclusions

- Technique for adapting the LM of SMT systems to actual input
- LM is assumed to be provided as a linear interpolation of sub-LMs
- Weights are estimated dynamically on the text to be translated
- Best results by:
  → Exploiting both source and target of the development set
  → Weight estimation at sentence level or two-steps approach
- Such results yield consistent improvements over the reference baseline

Future work

- Results achieved depend on the clustering technique employed
  $\rightarrow$ Clustering based on n-grams or PoS-tag information
- Supervised clustering
  $\rightarrow$ Detailed supervision is available only for limited amount of data
- Learn source-to-target weight mapping schemes from parallel data

Questions? Comments? Suggestions?