Online Language Model Adaptation for Spoken Dialog Translation

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

Mauro Cettolo, Nicola Bertoldi, Marcello Federico
FBK - Ricerca Scientifica e Tecnologica, Trento, Italy

Tokyo, Dec 1-2, 2009

Outline

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

Introduction

- Spoken language translation
- 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 IWSLT 2009 CT task, CRR conditions

Model adaptation

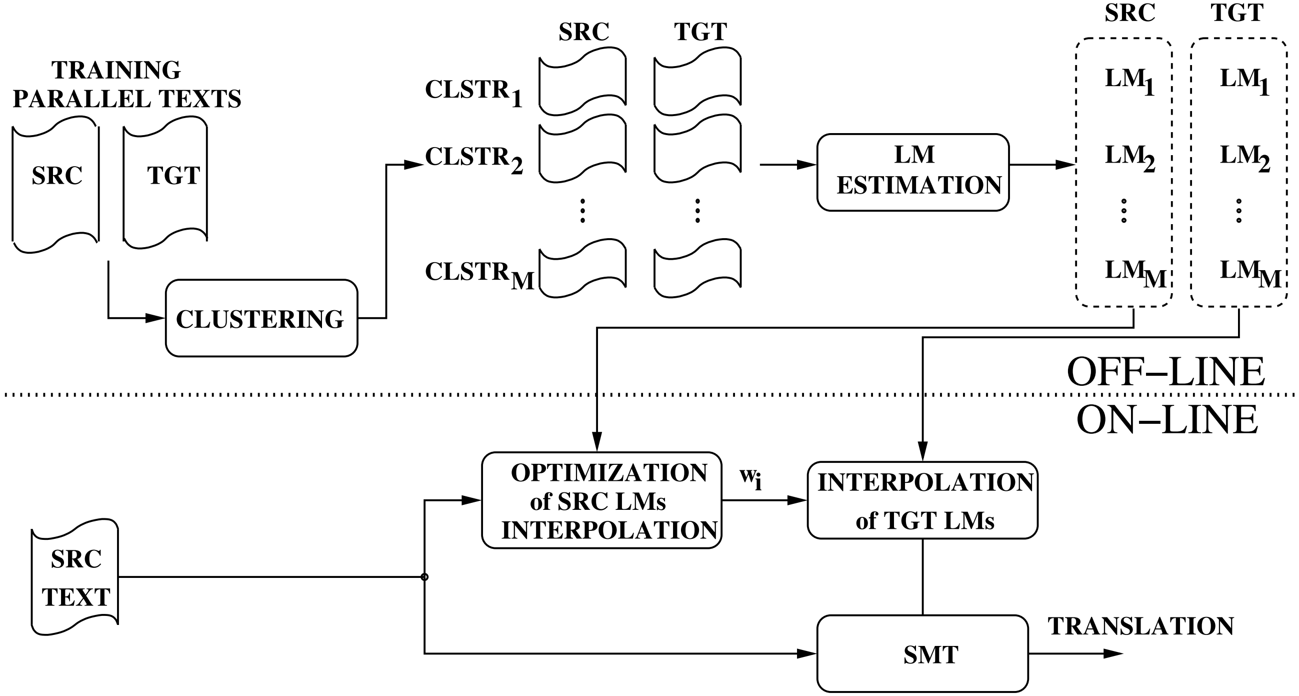
- Most usual translation rule:

$$\mathbf{e}^* = \arg\max_{\mathbf{e}} \max_{\mathbf{a}} \sum_{r=1}^{R} \lambda_r h_r(\mathbf{e}, \mathbf{f}, \mathbf{a})$$

- 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 the source LMs via EM

→ Transfer the resulting weights to the target LM mixture

IWSLT Data

- Experiments carried out on the CT task (both CE and EC)
- We considered the use of Agent, Customer and Interpreter annotations
- We also considered the use of the Dialog tags

Speaker-based statistics of the CT data

		Training			Development		
	speaker	|W|	|V|	$\bar{s}$	|W|	|V|	$\bar{s}$
agent	native	46.7K	2240	14.8	2.5K	427	15.1
	interpreter	26.8K	1626	14.1	0.8K	218	13.2
customer	native	33.3K	2082	13.9	0.5K	152	11.8
	interpreter	33.8K	1878	12.9	1.7K	307	12.3

Nespole! data

- NEgotiating through SPOken Language in E-commerce
- Collected involving Italian speakers, translated into English

Statistics of the Nespole! dialogs.

#turns	|W|	|V|	$\bar{s}$
2522	15335	1344	6.1

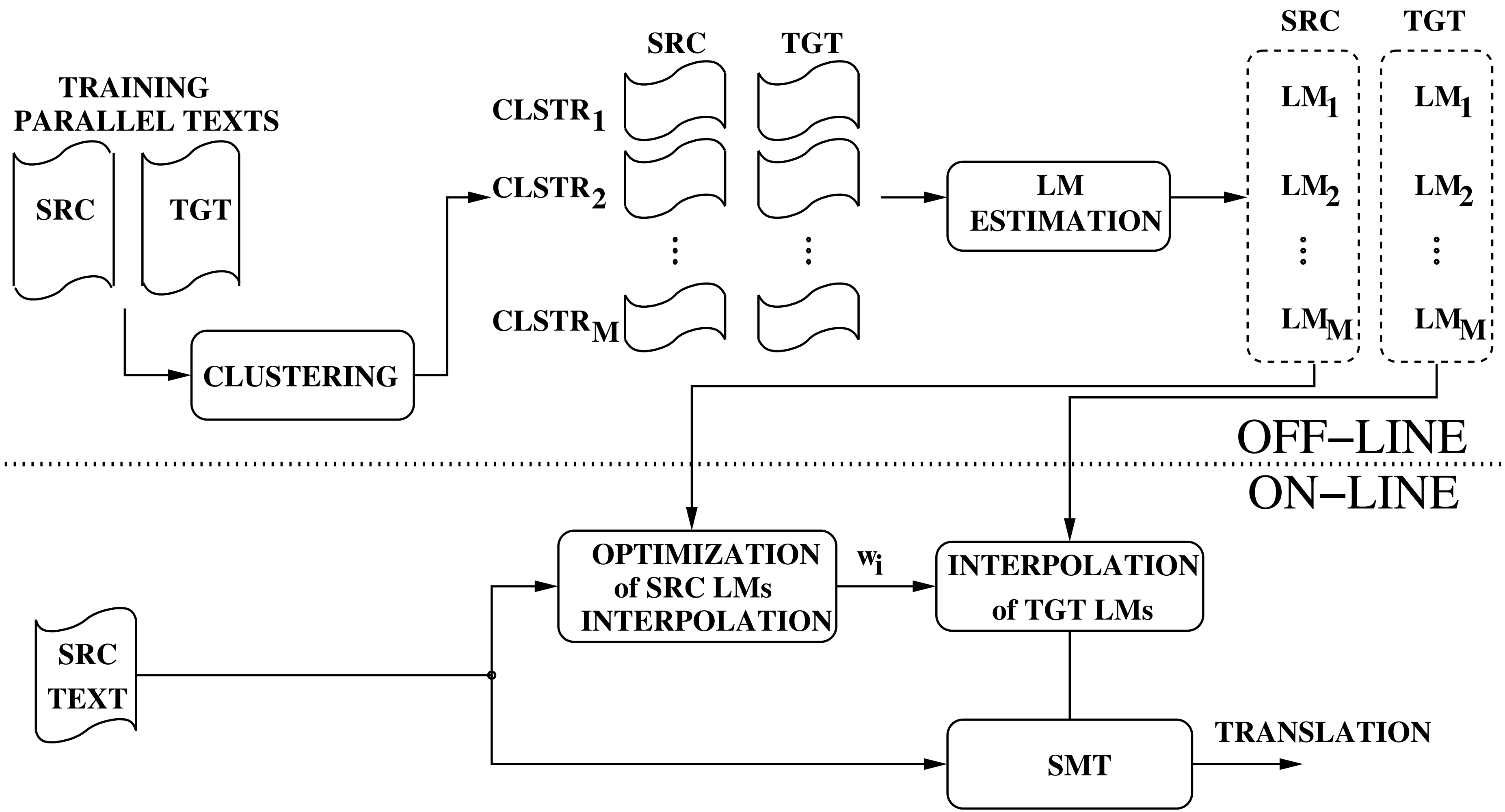
Most frequent Nespole! dialog acts.

label	counter
`give-information`	963
`affirm`	408
`descriptive`	285
`request-information`	199
...	...
total	2522

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`"

Model adaptation



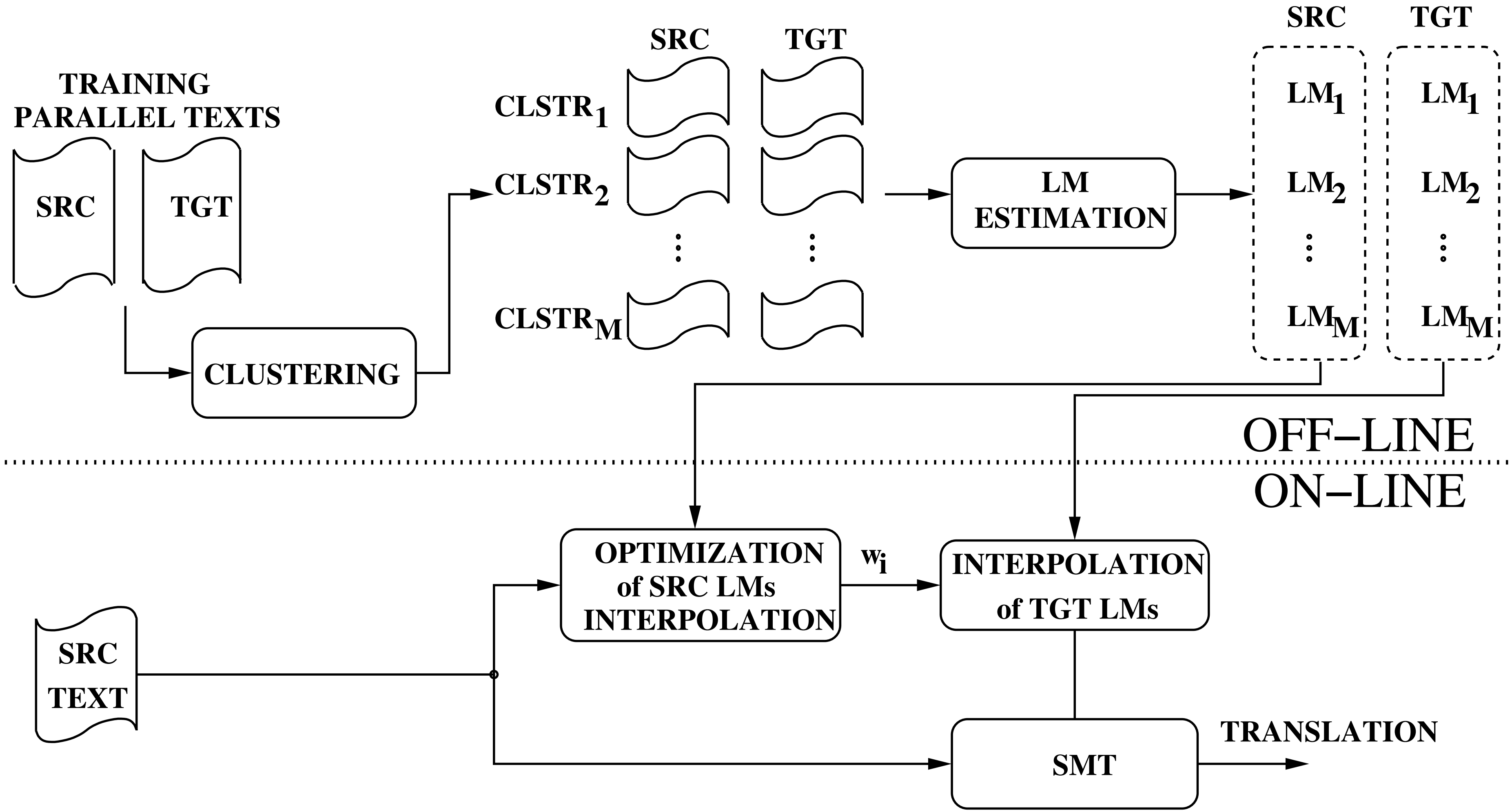
Clustering: IWSLT

- Dialog based
  - Consider each dialog as a bag of source and target words
  - Compute 2, 4, 6 and 8 clusters by means of CLUTO
    - `direct` clustering algorithm
    - cosine distance
  - Additional LM for BTEC+CT data
- Speaker based
  - Specific clusters for native agent/customer, and interpreter agent/customer
  - Additional LMs for BTEC and BTEC+CT data

Clustering: Nespole!

- Three LMs estimated on (English) Nespole! data:
  - `give-information`
  - `request-information`
  - other
- Such LMs are used to partition the IWSLT data on the basis of perplexity
- The clusters are mirrored on the Chinese side
- New LMs were trained on the IWSLT clusters
- Additional LM for all the BTEC+CT data

Model adaptation



On-line weight optimization

Four different approaches:

- 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

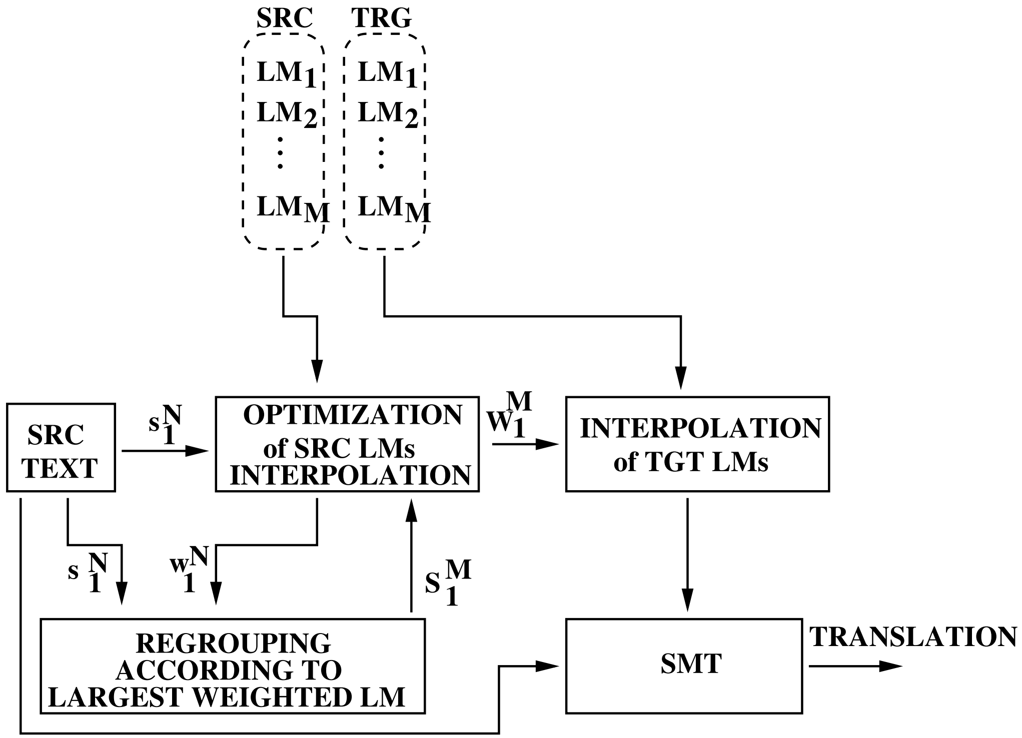
On-line weight optimization

Four different approaches:

- Sentence specific weights:
  - One set of weights for each sentence in the test set
    - + EM procedure allowed complete freedom
    - – Weights estimated on few data
    - ⇒ possibly, less reliable weights

On-line weight optimization

Four different approaches:



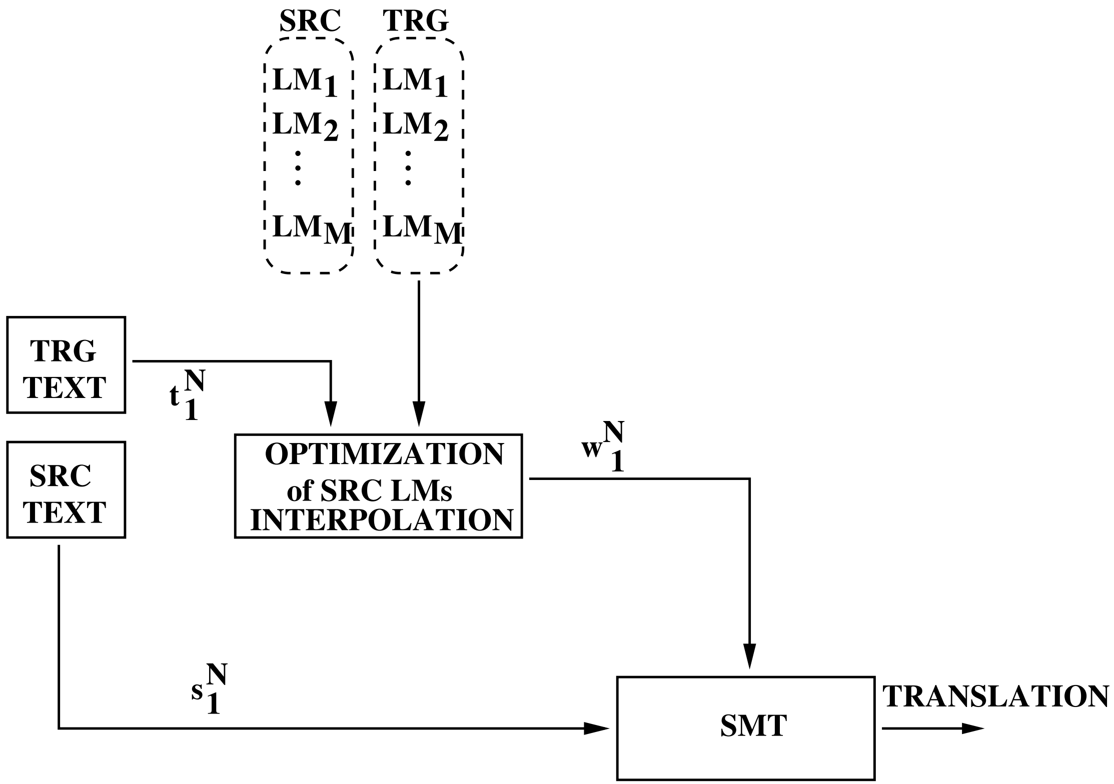
- Two-step weight estimation:
  1. Estimate sentence-specific weights
  2. Assign each source sentence to the cluster with the most weighted LM
  3. Re-estimate one single set of weights for each of such clusters

  \+ Mirror the clustering of the training data into the test set

  \+ Avoid possible data sparseness issues

On-line weight optimization

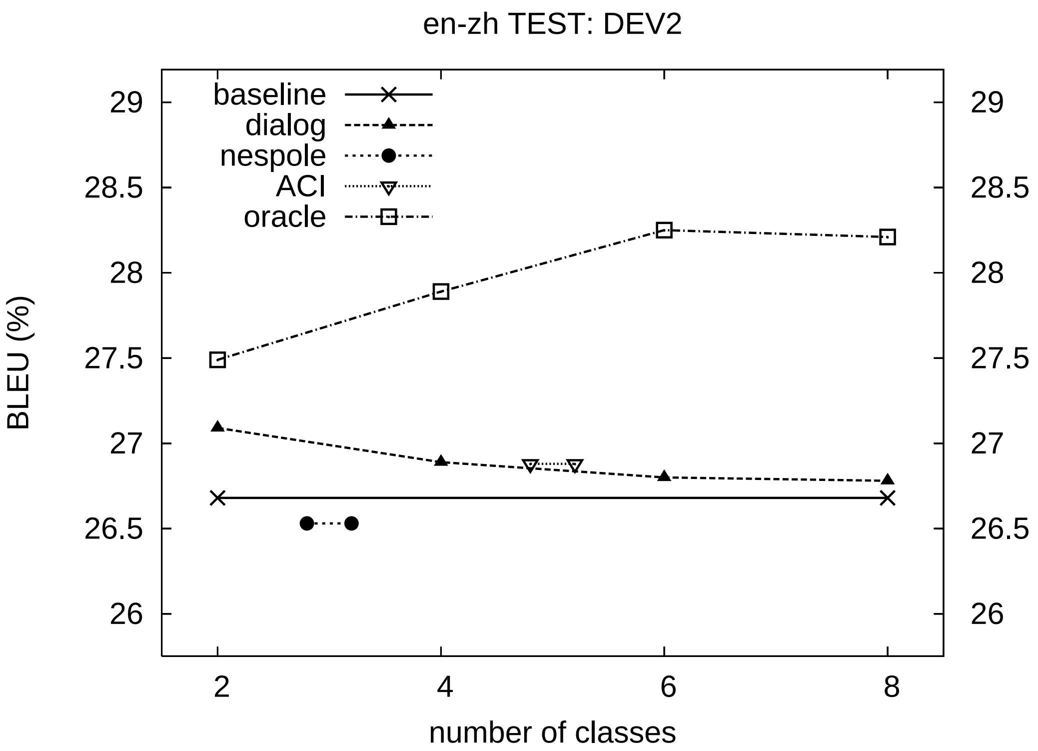
Four different approaches:

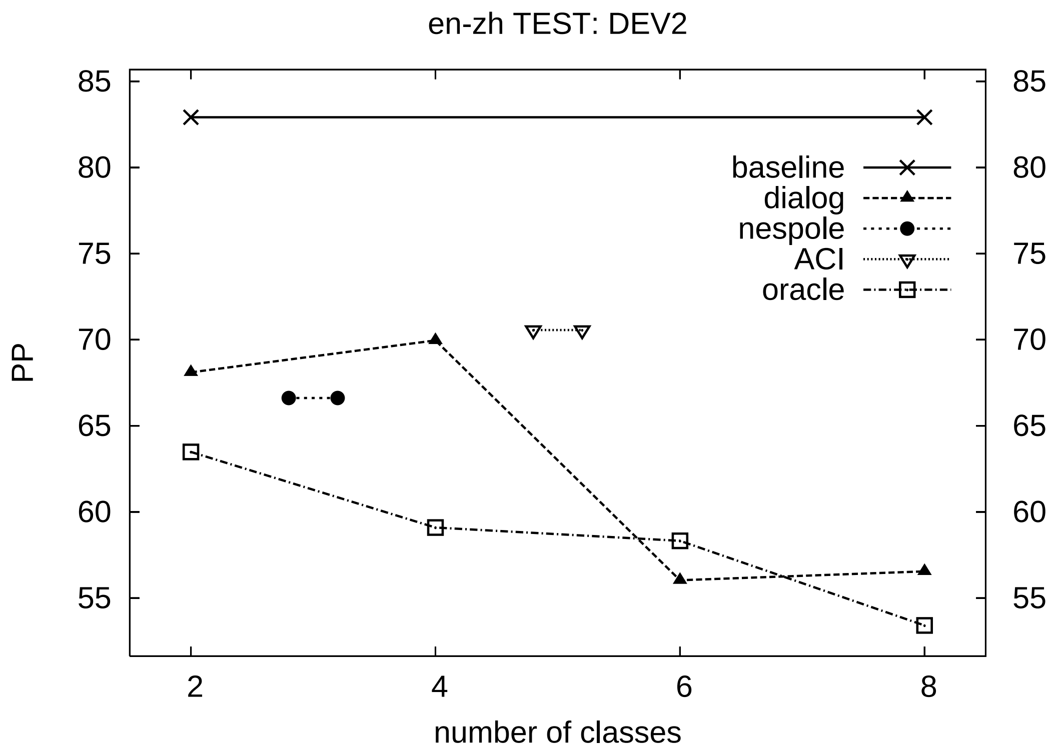


- Oracle weight estimation:
  - Estimate weights at sentence level on the reference texts (i.e. target side)
    - + Provides a sort of upper bound
    - − Not fair

Results

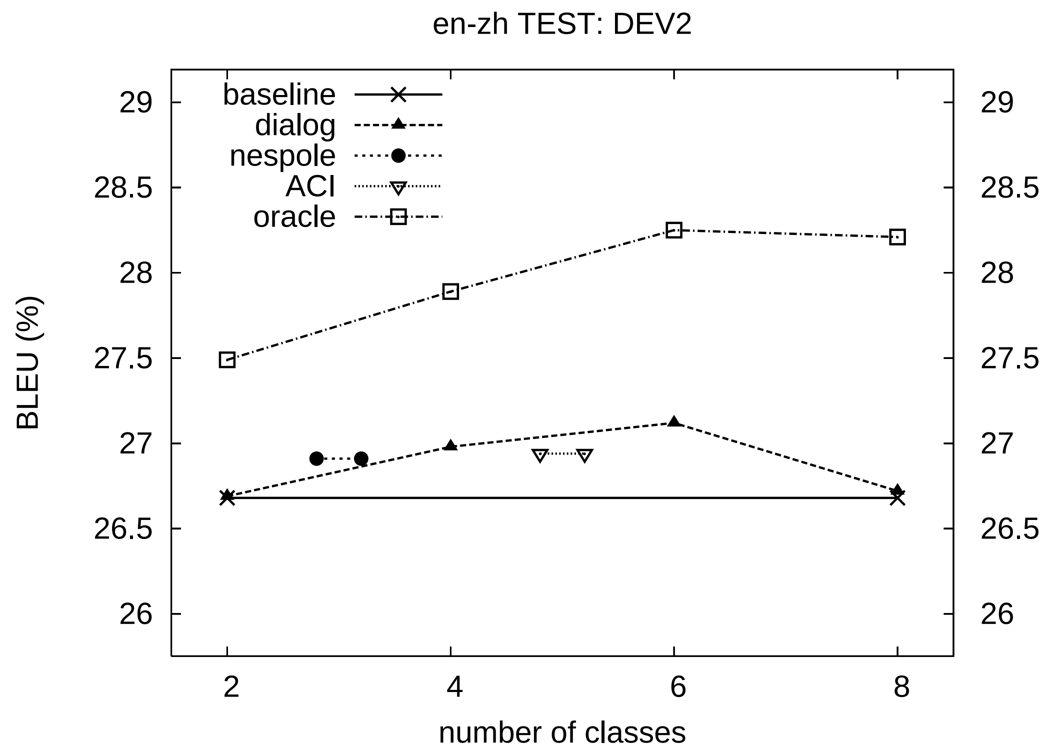
Results for sentence-based weight estimation

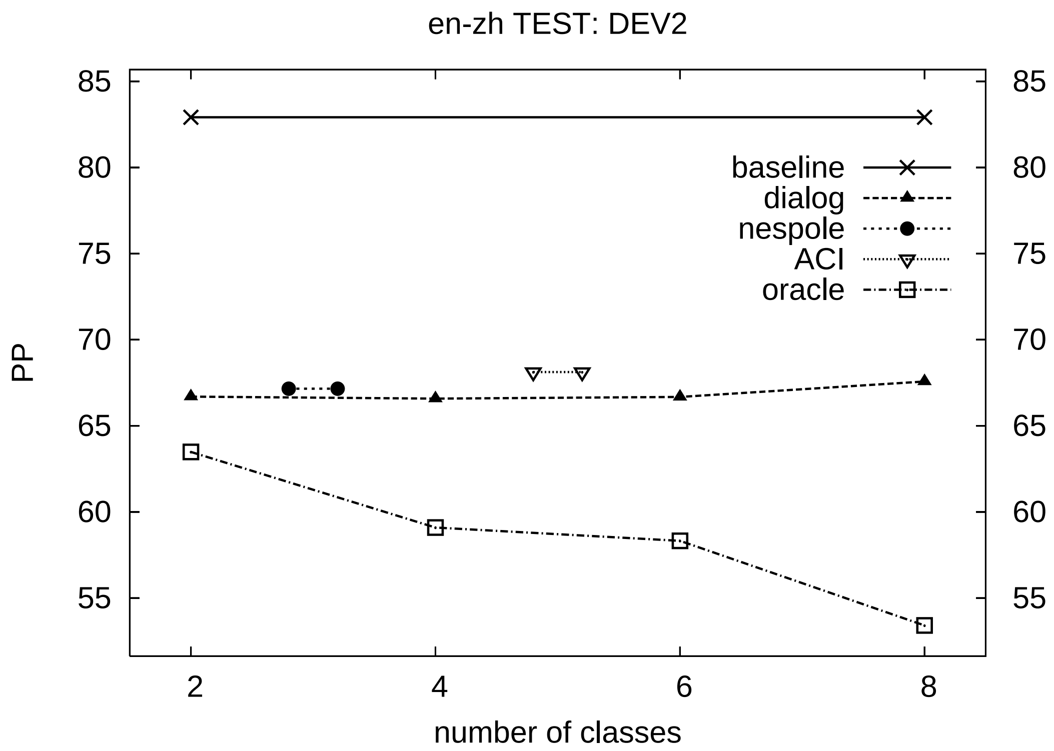




Results

Results for two-step weight estimation





Analysis

- Significant improvements are achieved in terms of perplexity for every setup
- Improvements in perplexity are not always mirrored by BLEU
- Oracle curves are unimodal with peak at six clusters
- Oracle setup confirms that the approach is appealing, room for improvement
- Two-step: does not improve sentence-based, but curves are unimodal
  $\rightarrow$ more predictable
- Dialog clustering improves or is as good as baseline:
  $+$ two-step: seems to guarantee stable improvements
- Nespole! guided clustering does not seem to be effective
- Clustering according to ACI labels works well for EC (not for CE)

Analysis

- Training/development and test conditions are quite different

Table 1: *MERT effect on the BLEU score.*

test on	mert on	Δ BLEU CE	Δ BLEU EC
DEV1	DEV2	-0.19	+3.39
DEV2	DEV1	-0.67	-1.12

- Clustering according to ACI labels produces speaker-specific LMs.

$\longrightarrow$ According to training!

$\longrightarrow$ This is bound to have an important effect

Future work

- Obtain data partitioning in an unsupervised manner
  - Surface form
  - PoS
  - ...
- Perform development/test-driven partitioning of the training data
- Source-to-target weight mapping
- Assess these techniques on larger tasks such as Europarl or NIST

Questions? Comments? Suggestions?