DISCRIMINATIVE INSTANCE WEIGHTING FOR DOMAIN ADAPTATION IN STATISTICAL MACHINE TRANSLATION

Authors: George Foster, Cyril Goutte, and Roland Kuhn (NRC Canada)

Presenter: Avneesh Saluja

The Claim

- Domain adaptation in SMT, and in NLP in general, a popular topic
- By incorporating several ideas:
  - Instance-weighting approach, at the level of phrase pairs
  - Overlapping features, designed to elicit “general language” and “similarity” characteristics
  - ML, instead of ME, training/learning criterion

the authors come up with an (improved?) domain adaptation scheme for MT

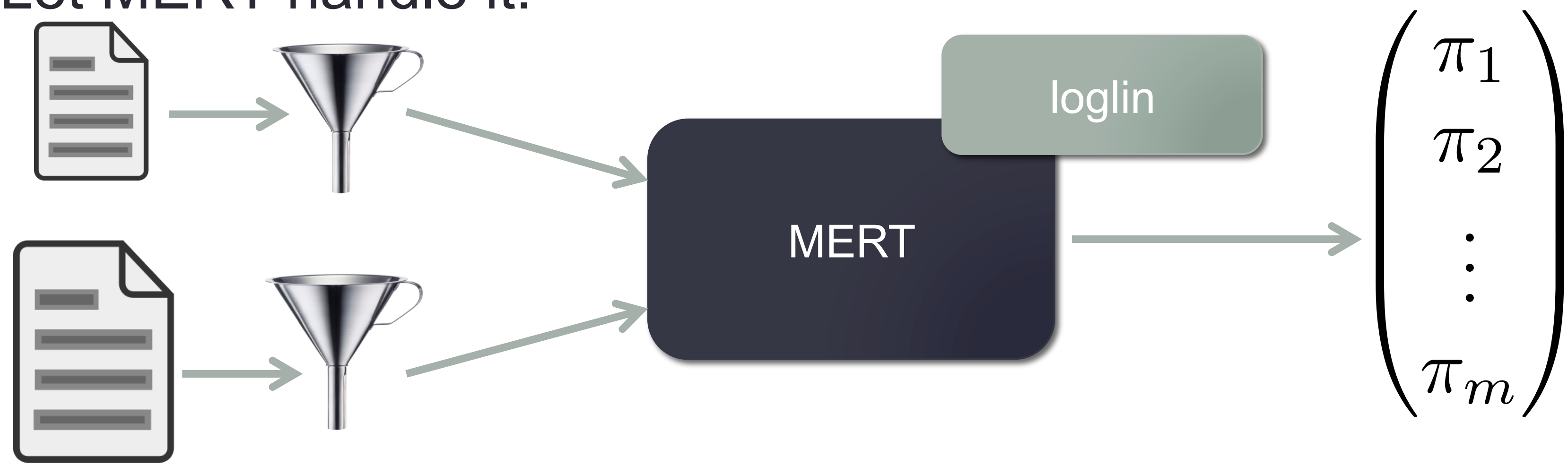
Why Domain Adaptation?

- Workshops, theses, papers, etc.
  - The brittleness of our models…
- In action: LMs for MT: Original vs. Translated Texts
- Theoretical background:
  - A theory of learning from different domains (Ben-David et al., Machine Learning, 2010)
  - Domain Adaptation of NLP Systems (J. Blitzer’s Thesis, 2008)
  - Domain Adaptation in Regression (Cortes & Mohri, ALT 2011)
- In MT: the pipeline approach prevents end-to-end adaptation scheme
- Assumption: all OOD data is homogeneous

Baseline Setups: Simplest Methods

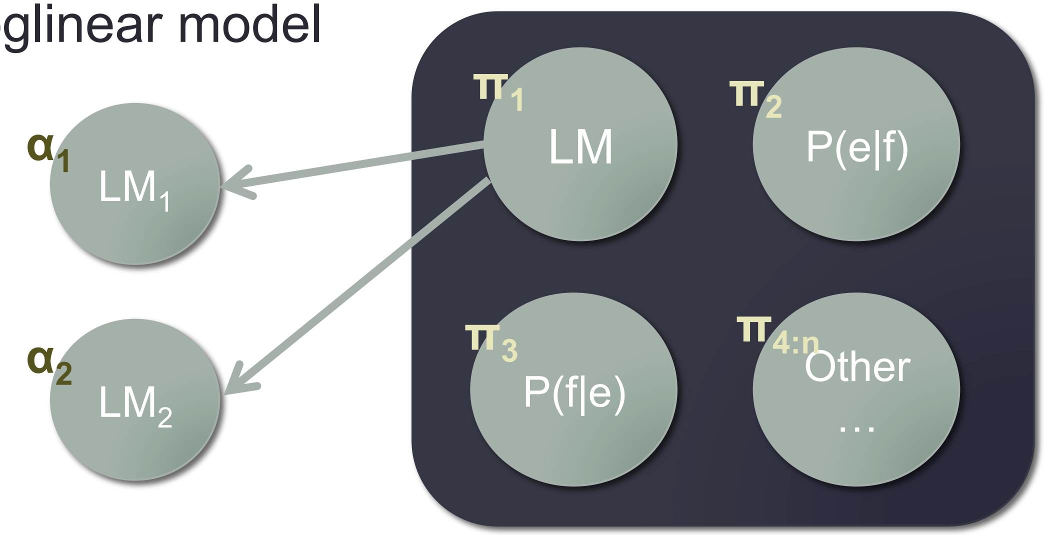
- Throw everything into a big bucket:

- Let MERT handle it:



Baseline Setups: Linear Combination

- Linear models and MERT for adaptation problematic:
  - MERT assumes a flat loglinear model
- Optimize corpus log-likelihood instead of minimizing error



LM Weights: $$\hat{\alpha} = \arg\max_{\alpha} \sum_{w,h} \tilde{p}(w,h) \log \sum_{i} \alpha_i p_i(w|h)$$

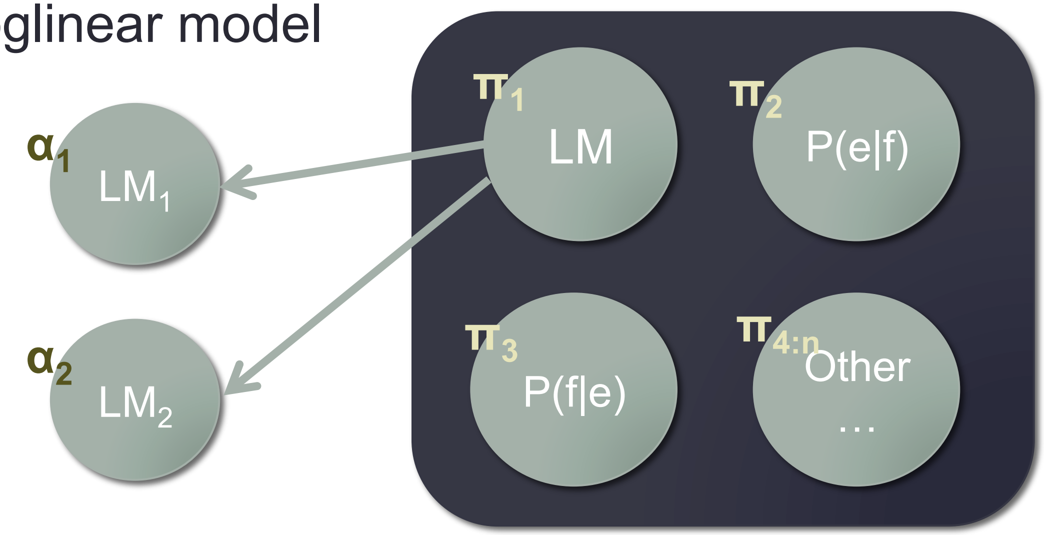
TM Weights: $$\hat{\alpha} = \arg\max_{\alpha} \sum_{s,t} \tilde{p}(s,t) \log \sum_{i} \alpha_i p_i(s|t)$$

lin lm

lin tm

Baseline Setups: Linear Combination

- Linear models and MERT for adaptation problematic:
  - MERT assumes a flat loglinear model
- Optimize corpus log-likelihood instead of minimizing error



LM Weights: $$\hat{\alpha} = \arg\max_{\alpha} \sum_{w,h} \tilde{p}(w,h) \log \sum_{i} \alpha_i p_i(w|h)$$

TM Weights: $$p(s|t) = \frac{c_I(s,t) + \beta p_0(s|t)}{c_I(t) + \beta}$$

lin lm

map tm

Baseline Setups: IR style

ir

- Select “similar” sentence pairs from from OOD that match sentences from ID
- Trained LM with in-domain data, evaluated on target side of OOD data
  - Select lowest perplexity sentences
  - Number of sentences to select tuned (optimize dev-set BLEU)

Instance Weighting: Model & Training

- Instance = Phrase Pair
- Potentially overlapping features defined for phrase pairs
- LM adaptation as in baseline
- TM adaptation: $p(s|t) = \alpha_t p_I(s|t) + (1-\alpha_t) p_o(s|t)$

$$\underbrace{c_o(s,t)\left[1+\exp\left(-\sum_i \lambda_i f_i(s,t)\right)\right]^{-1}}_{w_\lambda(s,t)} \leftarrow \frac{c_\lambda(s,t)+\gamma u(s|t)}{\sum_{s'} c_\lambda(s',t)+\gamma}$$

- Jointly optimize feature and mixture weights via L-BFGS

$$(\hat{\alpha}, \hat{\lambda}) = \arg\max_{\alpha,\lambda} \sum_{s,t} \tilde{p}(s,t) \log p(s|t;\alpha,\lambda)$$

$\gamma = 0$: iw all
$\gamma \neq 0$: iw all map

Interpretation of the Model

- Why does downweighting original joint OOD counts work?
- Ideally, we want to maximize (log) likelihood w.r.t. (i.e., weighted by) "true" joint distribution of in-domain data:

$$\hat{\theta} = \arg\max_{\theta} \sum_{s,t} p_{\hat{I}}(s,t) \log p_{\theta}(s|t)$$

s,t → Over all OOD phrase pairs

$$\approx \arg\max_{\theta} \sum_{s,t} \frac{p_{\hat{I}}(s,t)}{p_{\hat{O}}(s,t)} c_o(s,t) \log p_{\theta}(s|t) \Rightarrow$$

$$p_{\hat{\theta}}(s|t) = \frac{\frac{p_{\hat{I}}(s,t)}{p_{\hat{O}}(s,t)} c_o(s,t)}{\sum_{s'} \frac{p_{\hat{I}}(s,t)}{p_{\hat{O}}(s,t)} c_o(s,t)}$$

compare with

$$\frac{c_o(s,t) w_{\lambda}(s,t) + \gamma u(s|t)}{\sum_{s'} c_o(s',t) w_{\lambda}(s',t) + \gamma}$$

Uniform prior in experiments

$$\frac{1}{1 + e^{-\sum_i \lambda_i f_i(s,t)}} \approx \frac{p_{\hat{I}}(s,t)}{p_{\hat{O}}(s,t)} \Rightarrow$$

Ranges between 0 and 1
Does it make sense to "upweight"?

Features Used

General Language

- Phrase pair length
- Frequency of pair
- Rare source/target phrase frequencies (2x)
- IBM1 (OOD) ppl (2x)
- Mean & Min "document" or block frequencies (4x)
- Burstiness features (4x)

Similarity

- ID LM ppl over 1 & 2-grams (4x)
- OOV counts w.r.t. ID LM (2x)
- ID IBM1 model (2x)

SVM Feature:

- SVM classifier to classify ID and OOD phrase pairs
- Classifier result used as additional feature

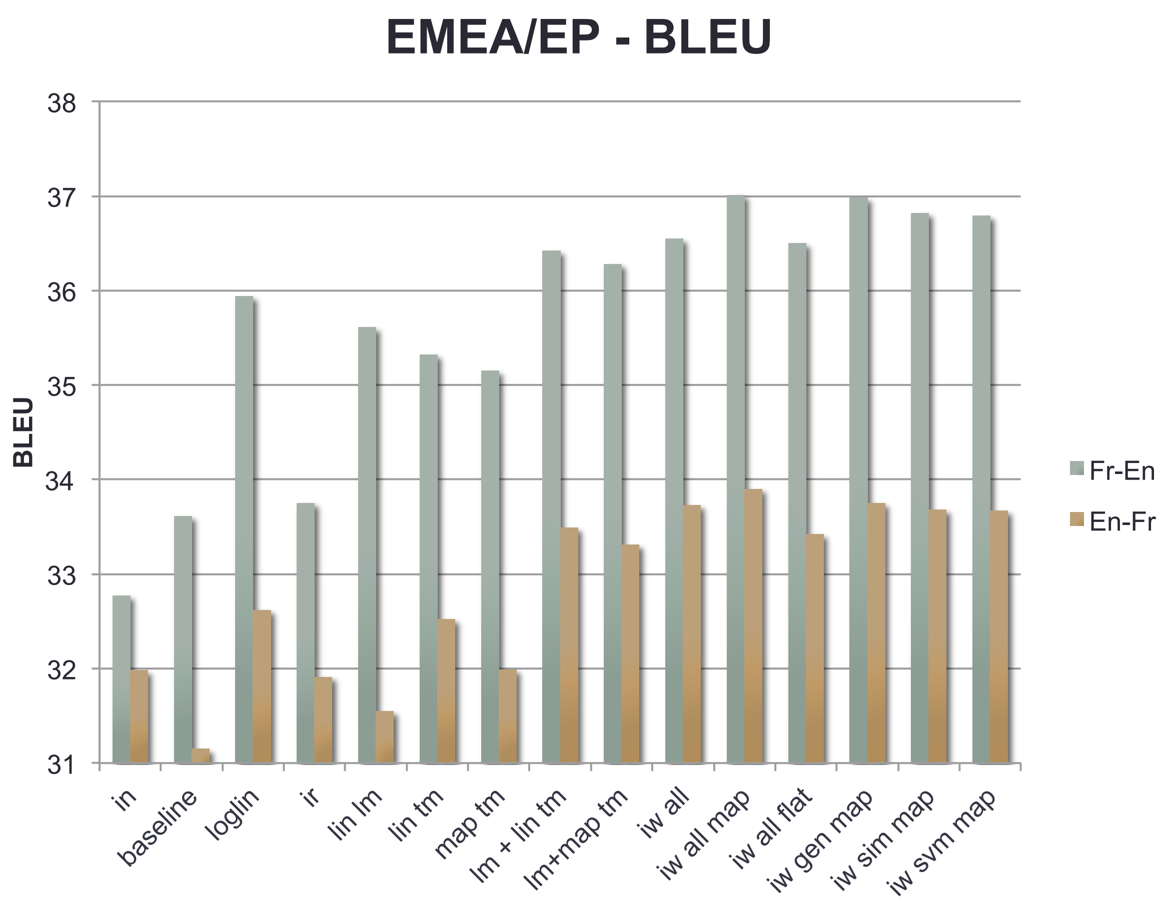
Corpora & Setup

- English <-> French
  - ID: EMEA Medical corpus
  - OOD: Europarl
  - Dev/test: from EMEA corpus
- Chinese -> English
  - ID: NIST09 news-related corpora
  - OOD: Rest of NIST09
  - Dev: NIST05 evaluation + random training set sentences
  - Test: NIST06 & NIST08
- Standard phrase-based setup; 4-gram LM
- HMM + IBM2 WA union

corpus	sentence pairs
Europarl	1,328,360
EMEA train	11,770
EMEA dev	1,533
EMEA test	1,522
NIST OUT	6,677,729
NIST IN train	2,103,827
NIST IN dev	1,894
NIST06 test	1,664
NIST08 test	1,357

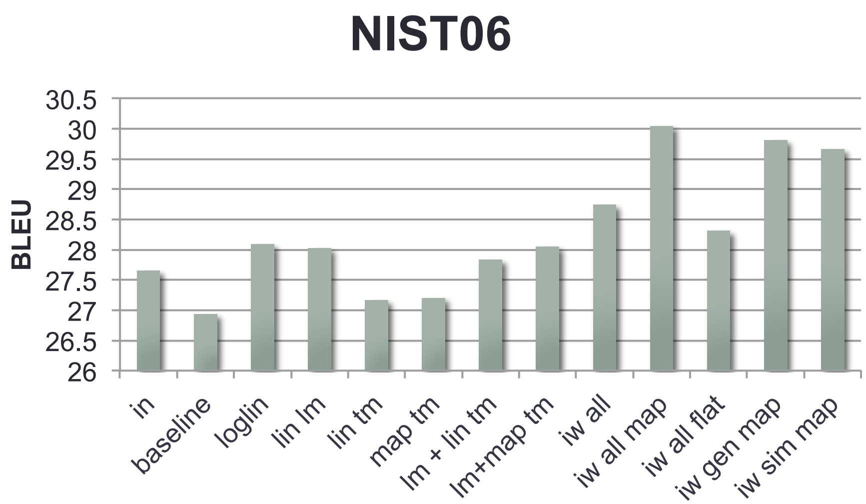
Table 1: Corpora

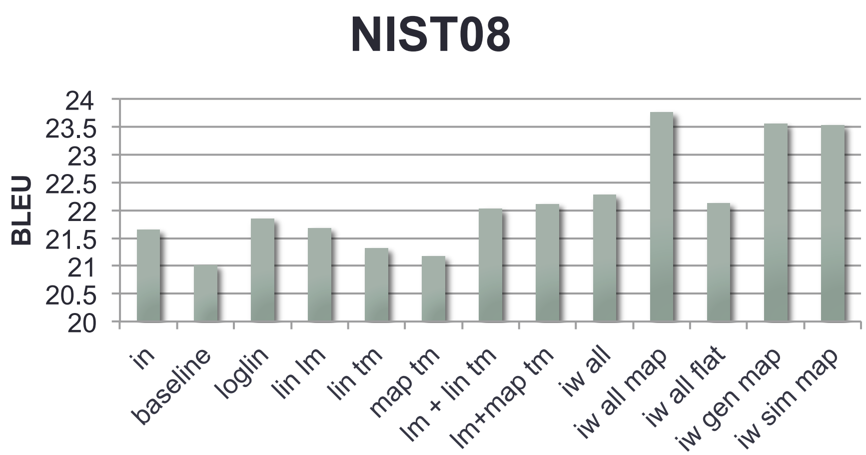
Results – EMEA/EP



method	EMEA/EP fren	EMEA/EP enfr
in	32.77	31.98
out	20.42	17.41
baseline	33.61	31.15
loglin	35.94	32.62
ir	33.75	31.91
lin lm	35.61	31.55
lin tm	35.32	32.52
map tm	35.15	31.99
lm+lin tm	36.42	33.49
lm+map tm	36.28	33.31
iw all	36.55	33.73
iw all map	**37.01**	**33.90**
iw all flat	36.50	33.42
iw gen map	36.98	33.75
iw sim map	36.82	33.68
iw svm map	36.79	33.67

Results - NIST





method	NIST	
	nst06	nst08
in	27.65	21.65
out	19.85	15.71
baseline	26.93	21.01
loglin	28.09	21.85
ir	——	——
lin lm	28.02	21.68
lin tm	27.16	21.32
map tm	27.20	21.17
lm+lin tm	27.83	22.03
lm+map tm	28.05	22.11
iw all	28.74	22.28
iw all map	**30.04**	**23.76**
iw all flat	28.31	22.13
iw gen map	29.81	23.56
iw sim map	29.66	23.53
iw svm map	——	——

Related Work

- Linear combination framework: Foster & Kuhn (ACL WMT, 2007)
  - Mixture weights are a function of several distance metrics
  - Downhill simplex to maximize BLEU on development set
- Motivation for instance weighting in NLP: Jiang & Zhai (ACL 2007)
  - Maximize expected log likelihood w.r.t. ID development set
  - This work applies the general concepts to MT
- Instance weighting through feature-based discriminative model: Matsoukas et al. (EMNLP 2009)
  - Sentence-level features, instead of phrase pair-level
  - Perceptron, instead of logistic regression
  - Optimize expected TER (over N-best) instead of log-likelihood
  - L-BFGS also
- General language & similarity features: Daumé (ACL 2007)

Conclusion

- Linear combination + instance weighting method for SMT domain adaptation
- Two-stage weighting:
  - Combine multinomial models: linearly
  - OOD phrase pair count weights: feature-based discriminative model
- Joint training of both sets of weights
- EMEA/EP (vs. strongest baseline):
  - Fr->En: +0.60 BLEU
  - En->Fr: +0.41 BLEU
- NIST (vs. strongest baseline)
  - NIST06: +0.99 BLEU
  - NIST08: +1.65 BLEU

Discussion

- Missing details:
  - Prior weight γ
  - No IR/SVM evaluation on NIST?
  - Example sentence showing improvement
  - Explicit comparison with sentence-level feature approach
- Analysis on how approach performs as a function of dataset size
- Is uniform prior the best choice?
- Is it necessary to have a two-stage model?
- A better way to incorporate Gigaword corpora?

Thank you!