Domain Adaptation in Machine Translation

Marine Carpuat
National Research Council Canada

Marine.Carpuat@nrc.gc.ca
<table>
<thead>
<tr>
<th>Old Domain (Parliament)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Original</strong></td>
<td>monsieur le président, les pêcheurs de homard de la région de l'atlantique sont dans une situation catastrophique.</td>
</tr>
<tr>
<td><strong>Reference</strong></td>
<td>mr. speaker, lobster fishers in atlantic canada are facing a disaster.</td>
</tr>
<tr>
<td><strong>System</strong></td>
<td>mr. speaker, the lobster fishers in atlantic canada are in a mess.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>New Domain</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Original</strong></td>
<td>comprimés pelliculés blancs pour voie orale.</td>
</tr>
<tr>
<td><strong>Reference</strong></td>
<td>white film-coated tablets for oral use.</td>
</tr>
<tr>
<td><strong>System</strong></td>
<td>white pelliculés tablets to oral.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>New Domain</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Original</strong></td>
<td>mode et voie(s) d'administration</td>
</tr>
<tr>
<td><strong>Reference</strong></td>
<td>method and route(s) of administration</td>
</tr>
<tr>
<td><strong>System</strong></td>
<td>fashion and voie(s) of directors</td>
</tr>
</tbody>
</table>
Domain adaptation in MT

• Translating across domains is hard, but often necessary

• Lots of interest in domain adaptation driven by
  – Increasing amounts of parallel training data
  – Increasing diversity of data sources
What is a domain?

• No clear definition of domain
  – Related to topic, genre, register

• Defined in practice by datasets/tasks
  • Single homogeneous domain
    e.g. Parliament proceedings
  • Large old domain & small new domain
    e.g. Parliament + News or Science
  • Large data collection from various sources
    e.g. NIST OpenMT, DARPA BOLT, WMT gigafren ...
What is domain adaptation?

From classical “single-domain” learning...

- predict $x \rightarrow y$
- *training* and *test* data generated from the same *distribution* $(x, y) \sim \Pr[x, y]$

... to **Domain Adaptation**

- Port system trained on *old* (aka source) domain to *new* (aka target) domain

$$(x, y) \sim \Pr_S[x, y] \quad (x, y) \sim \Pr_T[x, y]$$
No “one size fits all” approach

• Lots of domain adaptation work in Machine Learning
  – see [Blitzer & Daumé III, ICML 2010] for an overview

• But not directly applicable to MT
  – heterogeneous components trained independently
  – large variety of settings
Addressing domain shift in MT

• General approach
  – adjust MT parameters to optimize performance for a test set, based on some knowledge of its domain

• Various settings
  – amount of in-domain training data: small, dev-sized, none (just source text)
  – nature of out-of-domain data: size, diversity, labeling
  – monolingual resources: source and target, in-domain or not, comparable or not
  – latency: offline, tuning, dynamic, online, (interactive)
What to adapt?

- **Language model (LM)**
  - Effective and simple
  - Previous work from speech
  - Perplexity-based interpolation popular

- **Translation model (TM)**
  - Most popular target
  - Gains can be elusive

- **Distortion/Reordering model (DM)**

- **Log-linear model**
  - limited scope if in-domain dev set available
How to adapt to a new domain?

• Filter training data
  – Select from out-of-domain data based on similarity to test domain

• Corpus weighting
  – At sub-corpora, sentence or phrase-pair level

• Model combination
  – Train submodels on different subcorpora

• Self training
  – Use MT to generate new parallel data

• Latent semantics
  – Exploit latent topic structure

• Mining comparable corpora
Domain adaptation in MT

• Lots of recent work, but still many open questions

• I’ll focus on 2 of them today
  – What goes wrong when porting a MT system to a new domain?
  – What does “domain adaptation” mean in more heterogeneous data settings?
I. WHAT GOES WRONG WHEN PORTING MT TO A NEW DOMAIN?
When porting a machine translation system to a new domain...

1. **what goes wrong?**
   analysis of lexical choice errors
   [Irvine, Morgan, Carpuat, Daumé III, Munteanu, TACL 2013]

2. **how can we fix common errors?**
   new task to address under-studied “sense” errors
   [Carpuat, Daumé III, Henry, Irvine, Jagarlamudi, Rudinger,.. ACL 2013]
# S⁴ Taxonomy of Adaptation Errors

**New Domain (Medical)**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Original</strong></td>
<td>mode et voie(s) d’ administration</td>
</tr>
<tr>
<td><strong>Reference</strong></td>
<td>method and route(s) of administration</td>
</tr>
<tr>
<td><strong>System</strong></td>
<td>fashion and voie(s) of directors</td>
</tr>
</tbody>
</table>

**Seen:** Never seen this word before “voie(s)”

**Sense:** Never seen this word used in this way “mode” → “method”

**Score:** Wrong output is scored higher “administration” → “administration” or “directors”?

**Search:** Decoding/Search erred
Measuring impact of S4 errors

• We port MT system to new domain
  – Assumption: no new domain training data
  – **Old domain** resources
    • Large parallel training set
  – **New domain** resources
    • Tuning + test set

![Diagram showing Hansard and Medical resources](image-url)
Measuring impact of S4 errors

• Compare translation quality with “oracle”
  – Trained on
    • large old domain corpus
    • large new domain corpus
  – new domain tuning set
Measuring SEEN effects

Add all phrase pairs with previously unseen F side
Measuring SENSE effects

Add all phrase pairs with previously seen F side, but unseen translation
Measuring SCORE effects

Add all phrase pairs, period (and keep new domain scores)
Impact of fixing $S^4$ errors on BLEU

**News**
- OLD: 15
- +Seen: 20
- +Sense: 25
- +Score: 30
- Mixed: 35

**Medical**
- OLD: 15
- +Seen: +8%
- +Sense: +7%
- +Score: +5%
- Mixed: +28%

**Science**
- OLD: 15
- +Seen: +6%
- +Sense: +4%
- +Score: +10%
- Mixed: +23%

**Subtitles**
- OLD: 15
- +Seen: +6%
- +Sense: +9%
- +Score: +8%
- Mixed: +22%
How to fix the $S^4$ errors (without new domain parallel data)

**Seen:** Dictionary mining for unseen terms


**Score:** Existing domain adaptation techniques

[Blitzer et al. 2006, Bickel et al. 2007, inter alia]

**Sense: SenseSpotting + {dictionary mining, active learning}**

[Bloodgood & CCB 2010]
SenseSpotting

• **Why?** MT performance across domains degrades due to lexical choice errors

• **What?** New task to identify word occurrences (tokens) that gain a new sense in new domains

• **How?** Automatic annotation from parallel text + supervised learning
SenseSpotting task definition

Old domain translation lexicon

| rapport || report || 0.8
| rapport || connection || 0.1
| rapport || study || 0.05
| rapport || relationship || 0.05

New domain sentences

- ces données sont basées sur le rapport d’ étude clinique
  - this data is based on clinical study report
- le rapport cholestérol total / hdlc est resté stable
  - the ratio of total cholesterol : hdlc was unchanged
Key aspects of SenseSpotting

• Sense inventory is defined by the MT lexicon [Chan et al. 2007, Carpuat & Wu, 2007, inter alia]

• New Senses are detected at the token-level
Data requirements

Hansard

Extract candidate terms and statistics

Medical

Extract useful statistics

Train model parameters
Classification set-up

Logistic regression model trained with VW
- L1 or L2 regularized based on tuning data

16-fold cross validation at the type level
- Never test on type seen in training!
- E.g., train on “mode”, “administration”; test on “rapport”

Evaluation metric: AUC
- area under the ROC curve
- Pr(a true positive outranks a true negative)
Indicators of new sense

New senses alter corpus-level word frequency
New senses alter document-level context
  • topic distribution
New senses alter local context
  • n-gram language model
  • distributional similarity
  • context-dependent translation model

Computed at both type and token levels
SenseSpotting results

- **Medical**
  - 52% positive, 35k tokens

- **Subtitles**
  - 43% positive, 23k tokens

- **Science**
  - 24% positive, 8k tokens

Area Under the ROC Curve (cross-validation)
Part I: Summary

We used **automatic annotation** derived from parallel corpora to address key questions

- what goes wrong when translating across domains?
  - All errors categories (seen, sense, score) matter

- how can we fix common errors?
  - proposed new task to address under-studied “sense” errors
II. WHAT DOES "DOMAIN ADAPTATION" MEAN IN MORE HETEROGENEOUS DATA SETTINGS?
How to estimate MT models from heterogeneous data?

• So far we have studied clear cut domain adaptation tasks (Europarl -> Medical)

• But we often train on more heterogeneous data

• How to robustly estimate models
  • from heterogeneous data
  • to achieve good translation quality on various test domains?

[Carpuat, Goutte and Foster, WMT 2014]
Estimating MT Models From Heterogeneous Data

Approaches

– Data selection
  [Moore & Lewis 2010, Axelrod et al. 2011... ]

– Data weighting based on provenance
  [Chiang et al. 2011, Eidelman et al. 2012,...]

– Linear mixture models
  [Foster & Kuhn 2007, Foster et al. 2010, Sennrich 2012, ...]

– Finer grained instance weighting
  [Foster et al. 2010, Hasler et al. 2014...]

...
Defining Linear Mixtures With Heterogeneous Data

• We focus on translation probabilities
• Given K subsets of the training corpus

\[ P(t|s) = \sum_{k=1}^{K} \lambda_k P_k(t|s) \]

– How to define mixture components?
– How to learn mixture weights?
Mixture Models for Robust MT

• We empirically study impact on BLEU of
  – Component definitions
  – Mixture weights

• Key findings
  – All mixture models improve BLEU
  – Surprisingly, domain knowledge is not necessary
How to set mixing weights?

$$P(t|s) = \sum_{k=1}^{K} \lambda_k P_k(t|s)$$

2 methods:

- **Maximum likelihood weights**
  - Requires dev data representative of test domain
  - Estimate joint distribution $\tilde{p}(s, t)$ from dev
  - Optimize ML objective using EM

$$\hat{\lambda} = \operatorname{argmax}_\lambda \sum_{s,t} \tilde{p}(s, t) \log \sum_{k=1}^{K} \lambda_k p_k(s|t)$$
How to set mixing weights?

\[ P(t|s) = \sum_{k=1}^{K} \lambda_k P_k(t|s) \]

2 methods:

- **Maximum likelihood weights**
  - Requires dev data representative of test domain

- **Uniform weights**
  - Domain agnostic
How to define mixture components?

\[ P(t|s) = \sum_{k=1}^{K} \lambda_k P_k(t|s) \]

We partition training data

- By hand, using domain knowledge
- By automatic clustering, to learn data-driven domain distinctions
- Randomly
  - Random partition
  - Random sample (with replacement)
Domain knowledge in linear mixture models

<table>
<thead>
<tr>
<th>Corpus Components</th>
<th>Max Likelihood Weights</th>
<th>Uniform Weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manual partition</td>
<td>Dev + Train</td>
<td>Train</td>
</tr>
<tr>
<td>Automatic partition</td>
<td>Dev</td>
<td>None</td>
</tr>
<tr>
<td>Random partition</td>
<td>Dev</td>
<td>None</td>
</tr>
<tr>
<td>Random sample</td>
<td>Dev</td>
<td>None</td>
</tr>
</tbody>
</table>
Experiments:
2 lang. pairs & 2 test domains

<table>
<thead>
<tr>
<th>Arabic-English Training Conditions</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>segs</td>
<td>src</td>
<td>en</td>
<td></td>
</tr>
<tr>
<td>train</td>
<td>8.5M</td>
<td>262M</td>
<td>207M</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test Domain 1: Webforum</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>segs</td>
<td>src</td>
<td>en</td>
<td></td>
</tr>
<tr>
<td>dev (tune)</td>
<td>4.1k</td>
<td>66k</td>
<td>72k</td>
</tr>
<tr>
<td>web1 (eval)</td>
<td>2.2k</td>
<td>35k</td>
<td>38k</td>
</tr>
<tr>
<td>web2 (eval)</td>
<td>2.4k</td>
<td>37k</td>
<td>40k</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test Domain 2: News</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>segs</td>
<td>src</td>
<td>en</td>
<td></td>
</tr>
<tr>
<td>dev (tune)</td>
<td>1664</td>
<td>54k</td>
<td>51k</td>
</tr>
<tr>
<td>news (eval)</td>
<td>813</td>
<td>32k</td>
<td>29k</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chinese-English Training Conditions</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>segs</td>
<td>src</td>
<td>en</td>
<td></td>
</tr>
<tr>
<td>train</td>
<td>11M</td>
<td>234M</td>
<td>253M</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test Domain 1: Webforum</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>segs</td>
<td>src</td>
<td>en</td>
<td></td>
</tr>
<tr>
<td>dev (tune)</td>
<td>2.7k</td>
<td>61k</td>
<td>77k</td>
</tr>
<tr>
<td>web1 (eval)</td>
<td>1.4k</td>
<td>31k</td>
<td>38k</td>
</tr>
<tr>
<td>web2 (eval)</td>
<td>1.2k</td>
<td>29k</td>
<td>36k</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test Domain 2: News</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>segs</td>
<td>src</td>
<td>en</td>
<td></td>
</tr>
<tr>
<td>dev (tune)</td>
<td>1.7k</td>
<td>39k</td>
<td>24k</td>
</tr>
<tr>
<td>news (eval)</td>
<td>0.7k</td>
<td>19k</td>
<td>19k</td>
</tr>
</tbody>
</table>
Experiments: defining mixture components

- Split training set into homogeneous components
  - Same provenance, epoch, dialect, genre

- Arabic
  - 47 files, 15 genres, 4 dialects
  - 82 basic components
  - grouped into $K = 10$ components

- Chinese
  - 101 basic components
  - grouped into $K = 17$ components
Experiments: Phrase-based MT system

• Features
  – 4 phrase-table scores
    • Kneser-Ney smoothed translation probabilities x 2 [Chen et al. 2011]
    • Lexical weights x 2 [Zens & Ney 2004]
    • Counts summed across several word alignments (IBM2, HMM, IBM4)
  – hierarchical reordering, word penalty, distortion penalty [Galley & Manning 2008, Cherry 2013]
  – 3 5-gram language models
    • All training set, Gigaword, webforum or news only
  – Sparse features [Hopkins & May, 2011]

• Loglinear weights learned with batch lattice MIRA
Findings: linear mixtures significantly improve BLEU

Arabic-English

Chinese-English
ar-en: all mixture components improve BLEU

Explicitly modeling domain in mixture components does not help!
ar-en: mixing weights only have a small impact on BLEU

domain knowledge in mixing weights does not clearly help
zh-en: no consistent advantage from domain knowledge
Why doesn’t domain knowledge help more?

• Hypothesis: mixture models
  – don’t capture domain specific translations
  – smooth translation distributions toward “general language” instead
  – learn more robust translation probabilities
    • Random sampling + averaging = bagging
      [Breiman 94]
Part II: Domain Adaptation in heterogeneous data settings

When learning mixture models from heterogeneous data

• should mixture components represent domains?
• should weights reflect proximity between components and test domain?
Part II: Domain Adaptation in heterogeneous data settings

Findings

• All mixtures improve BLEU
• Domain knowledge is not necessary
• Are mixture models just a form of smoothing toward “general language”? 
Conclusion

• There’s no data like more relevant data
  – Handling data heterogeneity matters

• Lots of “domain adaptation” results in the literature, but no clear picture yet
  – various data settings, targets for adaptation, approaches

• Key open questions remain
  – How exactly does translation quality degrade in new domains?
  – What domain knowledge do domain adaptation techniques actually capture?
  – …
Domain Adaptation in Machine Translation

Marine Carpuat
National Research Council Canada

Marine.Carpuat@nrc.gc.ca