Domain Adaptation Using Domain Similarity- and Domain Complexity-based Instance Selection for Cross-domain Sentiment Analysis

Robert Remus  
rrremus@informatik.uni-leipzig.de

Natural Language Processing Group  
Department of Computer Science  
University of Leipzig, Germany

IEEE ICDM SENTIRE-2012 — December 10, 2012
Sentiment analysis and its subtasks are domain-dependent

- To overcome domain dependencies, a lot of NLP and ML research focuses on domain adaptation (DA): transfer a model from a source domain $d_{src}$ to a target domain $d_{tgt}$ with minimal performance loss.

- We consider a domain as a genre attribute, that describes the topics sth. deals with, e.g.
  - news articles (= genre) of different sections, e.g.
  - sports or politics (= domains)
[Ponomareva & Thelwall, 2012] hypothesized, that the optimal parameter setting of their DA algorithm is related to the notions of domain similarity and domain complexity

- domain similarity = corpus similarity
- domain complexity = corpus complexity

Our idea: “Tailor” a $d_{src}$ training set to a given $d_{tgt}$ based on their similarity and complexity
Method — Measuring Domain Similarity

- Similarity of domains \(d_{src}, d_{tgt}\) is measured as Jensen-Shannon (JS) divergence between \(d_{src}, d_{tgt}\)'s term unigram distributions
  - Unigram probabilities are estimated via relative frequencies
- JS divergence \(D_{JS}\) is based on Kullback-Leibler divergence \(D_{KL}\):

\[
D_{KL}(Q||R) = \sum_{w \in W} Q(w) \log \frac{Q(w)}{R(w)}
\]

where \(Q, R\) are probability distributions over a finite set \(W\), e.g. words.

\[
D_{JS}(Q||R) = \frac{1}{2} [D_{KL}(Q||M) + D_{KL}(R||M)]
\]

where \(M = \frac{1}{2}(Q + R)\) is the average distribution of \(Q\) and \(R\) and \(0 \leq D_{JS}(Q||R) \leq 1\)
Method — Measuring Domain Complexity

- Domain complexity is measured according to a procedure proposed by [Kilgarriff & Rose, 1998]:
  1. Shuffle corpus
  2. Split corpus into 2 equally-sized sub-corpora
  3. Measure similarity between sub-corpora
  4. Iterate and calculate mean similarity over all (here: 10) iterations

- Again, our similarity measure is JS divergence
Goal: Automatically select $d_{src}$ training instances, that are likely to help in estimation of a more accurate $d_{tgt}$ model

- How many/which $d_{src}$ training instances to select?

Assumptions:

- The more similar $d_{src}$ and $d_{tgt}$ are, the more ...
- The more the complexity varies among $d_{src}$ and $d_{tgt}$, the less ...

...the $d_{src}$ training data helps to estimate a more accurate $d_{tgt}$ model &

- The more similar a single $d_{src}$ training instance is to a $d_{tgt}$, the more it helps to estimate a more accurate $d_{tgt}$ model
Method — DA via Instance Selection II

1. $d_{src}$ training instances are ranked acc. to their similarity to the $d_{tgt}$
2. A training set size reduction factor $r_{d_{src},d_{tgt}}$ is estimated as

$$\tilde{r}_{d_{src},d_{tgt}} = 1.0 - (\alpha \cdot s_{d_{src},d_{tgt}} + \beta \cdot |\Delta c_{d_{src},d_{tgt}}|)$$ (3)

where

- $s_{d_{src},d_{tgt}}$ is the domain similarity
- $\Delta c_{d_{src},d_{tgt}} = c_{d_{src}} - c_{d_{tgt}}$ is the domain complexity variance
- $\alpha, \beta$ are scaling parameters

3. Top $100 \cdot \tilde{r}_{d_{src},d_{tgt}}$ % instances are kept while the rest is discarded
Evaluation — Setup I

- Task: Document-level cross-domain polarity classification in a semi-supervised setting
- Classifier: SVMs
  - Linear “kernel”
  - Cost $C$ fixed to $2.0$, no further optimization
- Features encode word unigram absence/presence
  - No feature selection
  - No feature weighting
  - No further pre-processing
- Gold standard: Reviews from 10 domains of [Blitzer et al., 2007]’s Multi-domain Sentiment Dataset v2.0
- For each $d_{src} - d_{tgt}$ pair:
  - 2,000 labeled $d_{src}$ instances, 200 labeled $d_{tgt}$ instances for training
  - 1,800 labeled $d_{tgt}$ instances for testing
  - 2,000 unlabeled $d_{tgt}$ instances for training (if required)
Evaluation — Setup II

- Instance selection IS
- Baselines:
  - “SrcOnly”, “TgtOnly” and “All”
  - EA and EA++ [Daumé III, 2007, Daumé III et al., 2010]
- IS combined with EA/++: IS-EA, IS-EA++
- “Sanity checks”
  - $IS_{r=0.8}$: fixed $\tilde{r}_{d_{src},d_{tgt}}$ of 0.8 (= average “optimal” $r$)
  - $IS_{\text{random}}$: random $\tilde{r}_{d_{src},d_{tgt}}$; instance selection without ranking
We experimented with different scaling parameter settings (Recall $\alpha$ scales domain similarity measure, $\beta$ scales domain complexity variance):

- $\alpha \in [0,1]$ (step size .1) and $\beta \in [0,6]$ (step size .5)
- Best overall result when $\alpha = 0.2$, $\beta = 5.5$
- “Stable” results when $\alpha \in [0.2, 0.4] \& \beta \in [0.5, 5.5]$
- IS outperforms strongest baseline (“All”) for when $\alpha \in [0.1, 0.8]$

IS is successful without fine-tuning $\alpha, \beta$!
Evaluation — Results II

- Evaluation on all \( \frac{10!}{(10-2)!} = 90 \) possible \( d_{src} - d_{tgt} \) pairs
- Averaged accuracy \( A \):

<table>
<thead>
<tr>
<th>Method</th>
<th>( A )</th>
</tr>
</thead>
<tbody>
<tr>
<td>SrcOnly</td>
<td>72.2</td>
</tr>
<tr>
<td>TgtOnly</td>
<td>68.43</td>
</tr>
<tr>
<td>All</td>
<td>74.25</td>
</tr>
<tr>
<td>IS</td>
<td>74.68&amp;</td>
</tr>
<tr>
<td>EA</td>
<td>74.02</td>
</tr>
<tr>
<td>EA++</td>
<td>74.5</td>
</tr>
<tr>
<td>IS-EA</td>
<td>73.74</td>
</tr>
<tr>
<td>IS-EA++</td>
<td>74.28</td>
</tr>
</tbody>
</table>

- IS is significantly better \( (p < 0.005) \) than all “SrcOnly”, “TgtOnly”, “All”, IS\(_{random} \) (71.47), IS\(_{r=0.8} \) (74.31)
  - Level of statistical significance is determined by “stratified shuffling”
Conclusions & Future Work

■ We proposed an approach to DA via instance selection, that is . . .
  □ based on similarity and complexity variance of $d_{src}$ and $d_{tgt}$
  □ a pre-processing step before learning a model

■ Future work: Apply IS to other cross-domain tasks, e.g. parsing, to answer whether . . .
  □ IS is general?
  □ IS is task-bound or feature-specific?
Thanks!

Any questions?
Appendix — Literature I

(2007).
Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL).

Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification.
In [acl, 2007], (S. 440–447).

Frustratingly easy domain adaptation.
In [acl, 2007], (S. 256–263).
Appendix — Literature II


Appendix — Literature III

Computational Natural Language Learning (CoNLL) (S. 655–665).