Part C: Parsing the web and domain adaptation
In-domain vs out-domain

- Annotated data in Domain A
  - Training
  - A Parser
    - Parsing texts in Domain A
    - Parsing texts in Domain B

In-domain
Out-domain
Motivation

- Few or no labeled resources exist for parsing text of the target domain.

- Unsupervised grammar induction?
  - Lots of work
  - Accuracies significantly lag behind those of supervised systems
  - Only on short sentences or assuming the existence of gold POS tags

- Build strong parsers by exploring labeled resources of existing domains plus unlabeled data for the target domain.
Outline

- Three shared tasks for parsing out-domain text
- Approaches for parsing out-domain text
  - news domain
  - web data
Shared tasks

- CoNLL 2007 shared task on domain adaptation
- CoNLL 2009 shared task on domain adaptation
- SANCL 2012 parsing the web
CoNLL 2007 shared task on domain adaptation

• Setup for the domain adaptation track
  • Data
    • Train: Large-scale labeled data for the source domain (WSJ)
    • Development: labeled data for biomedical abstracts
    • Test: labeled data for chemical abstracts
    • Unlabeled: large-scale unlabeled data for each train/dev/test.
  • The goal is to use the labeled data of the source domain, plus any unlabeled data, to produce accurate parsers for the target domains.
CoNLL 2009 shared task on domain adaptation

- Setup for the domain adaptation track
  - Czech, German, English (Brown corpus)
  - No unlabeled data
- Provide initial out-of-domain results for the three languages.
SANCL 2012: Parsing the web

- Data Setup (Petrov and McDonald, 2012)
  - Labeled data
    - Train: WSJ-train
    - Development: emails, weblogs, WSJ-dev
    - Test: answers, newsgroups, reviews, WSJ-test
  - Unlabeled data
    - Large-scale unlabeled data for all domains
- The goal is to build a single system that can robustly parse all domains.
# Data sets for SANCL 2012

## Training vs. Development vs. Evaluation

<table>
<thead>
<tr>
<th></th>
<th>WSJ-train</th>
<th>Emails</th>
<th>Weblogs</th>
<th>WSJ-dev</th>
<th>Answers</th>
<th>Newsgroups</th>
<th>Reviews</th>
<th>WSJ-eval</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sentences</td>
<td>30,060</td>
<td>2,450</td>
<td>1,016</td>
<td>1,336</td>
<td>1,744</td>
<td>1,195</td>
<td>1,906</td>
<td>1,640</td>
</tr>
<tr>
<td>Tokens</td>
<td>731,678</td>
<td>29,131</td>
<td>24,025</td>
<td>32,092</td>
<td>28,823</td>
<td>20,651</td>
<td>28,086</td>
<td>35,590</td>
</tr>
<tr>
<td>Types</td>
<td>35,933</td>
<td>5,478</td>
<td>4,747</td>
<td>5,889</td>
<td>4,370</td>
<td>4,924</td>
<td>4,797</td>
<td>6,685</td>
</tr>
<tr>
<td>OOV</td>
<td>0.0%</td>
<td>30.7%</td>
<td>19.6%</td>
<td>11.8%</td>
<td>27.7%</td>
<td>23.1%</td>
<td>29.5%</td>
<td>11.5%</td>
</tr>
</tbody>
</table>

## Sentences, Tokens, Types

<table>
<thead>
<tr>
<th></th>
<th>Emails</th>
<th>Weblogs</th>
<th>Answers</th>
<th>Newsgroups</th>
<th>Reviews</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sentences</td>
<td>1,194,173</td>
<td>524,834</td>
<td>27,274</td>
<td>1,000,000</td>
<td>1,965,350</td>
</tr>
<tr>
<td>Tokens</td>
<td>17,047,731</td>
<td>10,356,284</td>
<td>424,299</td>
<td>18,424,657</td>
<td>29,289,169</td>
</tr>
<tr>
<td>Types</td>
<td>221,576</td>
<td>166,515</td>
<td>33,325</td>
<td>357,090</td>
<td>287,575</td>
</tr>
</tbody>
</table>
Approaches for parsing canonical out-domain text (CoNLL07)

- Feature-based approaches
  - Only include features that transfer well (Dredze+, 07)
  - Structural corresponding learning: transform features from source domain to target domain (Shimizu and Nakagawa, 07)

- Ensemble-based approaches
  - Stacking (Dredze+, 07)
  - Co-training (Sagae and Tsujii, 07)
  - Variant of self-training (Watson and Briscoe, 07)
Approaches for parsing canonical out-domain text (CoNLL07)

- Other approaches
  - Tree revision rules for target domain (Attardi+, 07)
  - Training instance weighting (Dredze+, 07)
  - Hybrid: use the output of a Constraint Grammar parser (Bick, 07)
  - Use collocations and relational nouns from unlabeled target domain data (Schneider+, 07)
Frustratingly hard domain adaptation (Dredze+, 2007)

- Theoretical work on domain adaptation attributes adaptation loss to two sources (Ben-David+, 2006)
  - Difference in the distribution between domains
  - Difference in labeling functions
- The error analysis of Dredze+ (2007) suggests that the primary source of errors is the difference in annotation guidelines between treebanks.
Frustratingly hard domain adaptation (Dredze+, 2007)

- Challenges for adaptation from WSJ (90%) to BIO (84%)
  - Annotation divergences between BIO and WSJ
  - Unlike WSJ, BIO contains many long sequence of digits.
  - Complex noun phrases
  - Appositives
  - WSJ uses fine-grained POS tags such as NNP, while BIO uses NN.
- Long list of failed attempts
Frustratingly hard domain adaptation (Dredze+, 2007)

- Feature manipulation
  - Remove features less likely to transfer
  - Add features more likely to transfer
  - Using word clustering features

- Parser diversity
  - Ensemble of parsers (similar to stacking and bagging)

- Target focused learning
  - Assign higher weights to instances similar to the target when training
Domain + non-canonical text differences

Domain A
Canonical text

Domain B

Domain B
Non-canonical text
Parsing non-canonical out-domain text (SANCL)

- What is new?
  - Inconsistent usage of punctuation and capitalization
  - Lexical shift due to increased use of slang, technical jargon, or other phenomena.
  - Spelling mistakes and ungrammatical sentences
  - Some syntactic structures are more frequently used in web texts than in newswire
    - Questions, imperatives, long lists of names, sentence fragments…
Examples

- Plz go there.
- I like it very much!!!!!!
- Gooooooooo
- ...
Approaches for parsing non-canonical out-domain text (SANCL)
Approaches

- Domain B Non-canonical text
- Domain B Canonical text
- Domain A Canonical text

Text normalization

Domain adaptation
Approaches for parsing non-canonical out-domain text (SANCL)

- Main approaches
  - Text normalization (preprocessing)
  - Ensemble of parsers
  - Self-training for constituent parsing
  - Word clustering/embedding
  - Co/tri-training (unsuccessful)
  - Instance weighting and genre classification
Text normalization

- Preprocessing the data leads to better POS tagging and parsing performance. (Foster, 2010; Gadde+, 2011; Roux and Foster+, 2012)
Text normalization

- The preprocessing rules of (Roux, Foster+, 2012)
  - Emoticon => comma or full stop
  - Email address, URL => generic strings
  - Uppercased words => lowercased
  - Abbreviations, spelling variants (plz, ppl) => standard form
  - nt; s => n’t; ’s
  - Repeated punctuation (!!!) => collapsed into one
  - List items (# 2) => removed
Text normalization

- The preprocessing rules of (Seddah+, 2012)
  - An Ontonote/PTB token normalization stage
  - Smileys, URLs, email addresses, similar entities
  - Correct tokens or token sequences
    - Spelling error patterns
    - Lowercasing
    - Rewriting rules for dealing with frequent amalgams (gonna or im)

<table>
<thead>
<tr>
<th></th>
<th>Ontonotes dev</th>
<th>e-mail dev</th>
<th>weblog dev</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>all</td>
<td>unk</td>
<td>all</td>
</tr>
<tr>
<td>-corr</td>
<td>96.5</td>
<td>92.3</td>
<td>88.9</td>
</tr>
<tr>
<td>+corr</td>
<td>96.5</td>
<td>92.9</td>
<td>90.4</td>
</tr>
</tbody>
</table>
Text normalization

- The preprocessing rules of (McClosky+, 2012)
  - High-precision text replacements
    - 1,057 spelling auto-correction rules (yuo => you) from Pidgin instant messaging client
    - 151 common Internet abbreviations (LOL => “laughing out loud”)
  - Limited gain
    - Such spelling errors are infrequent in the unlabeled data.
Ensemble of parsers

- Product-of-experts (Alpage, DCU-Paris13)
- Stacking (IMS, Stanford, UPenn)
- Voting (CPH-Trento, DCU-Paris, HIT)
- Bagging (HIT)
- Up-training (IMS)
- Re-ranking (DUC-Paris13, IMS, Stanford)
- Model merging (OHSU, Stanford)

- Obtain large improvement gain.
  - More like improvement in in-domain parsing
  - Contribution to domain adaptation?
Exploring unlabeled data

- Self-training (successful for constituent parsers)
  - Two-stage generative model and reranker (Charniak and Johnson, 2005)
  - Generative PCFG-LA model (Petrov and Klein, 2007)
- Word clusters or embeddings
- Co/tri-training (unsuccessful for dependency parsers)
Why self-training is unsuccessful for dependency parsing?

- Generative models suffer less from the over-fitting problem during training.
- Current dependency parsing models are commonly discriminative.
  - Linear models with online training, no probabilistic explanation.
  - Generative models leads to unsatisfactory accuracy.
### Evaluation results

- Top 4 systems of SANCL on POS tagging
  - Tagging performance is very important

<table>
<thead>
<tr>
<th>Team</th>
<th>Answers</th>
<th>Newsgroups</th>
<th>Reviews</th>
<th>WSJ</th>
<th>Averaged</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCU-Paris (Roux, Foster+)</td>
<td>91.79</td>
<td>93.81</td>
<td>93.11</td>
<td>97.29</td>
<td>92.90 (1)</td>
</tr>
<tr>
<td>HIT (Zhang+)</td>
<td>90.99</td>
<td>93.32</td>
<td>90.65</td>
<td>97.76</td>
<td>91.32 (2)</td>
</tr>
<tr>
<td>IMS (Bohnet+)</td>
<td>91.07</td>
<td>91.70</td>
<td>90.01</td>
<td>97.57</td>
<td>90.93 (3)</td>
</tr>
<tr>
<td>Stanford (McClosky+)</td>
<td>90.30</td>
<td>91.49</td>
<td>90.46</td>
<td>95.00</td>
<td>90.75 (4)</td>
</tr>
</tbody>
</table>
Which one is the best/most important?

- Main approaches
  - Text normalization (preprocessing)
  - Ensemble of parsers
  - Self-training for constituent parsing
  - Word clustering/embedding
  - Co/tri-training (unsuccessful)
  - Instance weighting and genre classification
End of Part C
References


References


References

References

- Djame Seddah, Benoit Sagot, and Marie Candito. 2012. Robust pre-processing and semi-supervised lexical bridging for user-generated content parsing. In Notes of the First Workshop on SANCL.
References

- Meishan Zhang, Wanxiang Che, Yijia Liu, Zhenghua Li, Ting Liu. 2012. HIT dependency parsing: Bootstrap aggregating heterogeneous parsers. In Notes of the First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL)