Domain Adaptation for and Tree Blazing

Andrew MacKinlay, Rebecca Dridan, Dan Flickinger and Tim Baldwin

June 27, 2011

New Domains for Parse Selection

- With most grammars, a statistical parse selection model trained on one domain performs less well over a different domain.
- The ERG is different to other grammars – manually constructed, not induced from a treebank, so the effect may be less pronounced.
- But the size of this effect hasn't been quantified for the ERG and other DELPH-IN grammars.
- One reason is that we haven't had enough data – we need large quantities of high quality treebanks.

Adapting to New Domains Effectively

- We can do experiments training on in-domain, out-of-domain and mixed domain training data.
- This give us an idea how robust the grammar is over new domains.
- But it is also of practical use to downstream grammar users:
  - We have some idea how much accuracy we can expect out of the box on a new domain
  - We have an idea how many sentences we should try and treebank for a new domain to get reasonable performance
  - We may get some idea how to make best use of what limited in-domain data we have, in terms of combining it with out-of-domain data.

Corpus Summary

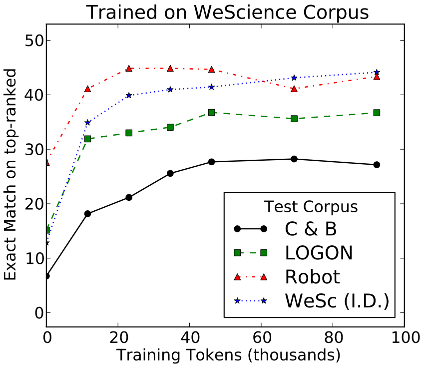
Corpus Statistics

Corpus	Description	Sentences (train/test)	Sent. length	Parses /sent.
WeScience	Wikipedia	6149/1482	18.1	271.9
LOGON	Hiking	6823/1727	14.2	229.9
C&B	Linux essay	0/567	21.6	323.8
Robot1	Dialog	768/535	6.7	97.2

Evaluating the size of the penalty

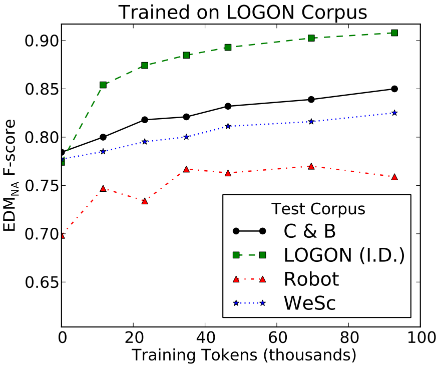
- We would like an idea of how the different training data performs on different test corpora
- With the 2 training corpora and 4 test corpora, this gives us 8 combinations to test:
  - 2 with purely in-domain training data
  - 6 with purely out-of-domain training data
- Using subsets of the training corpora, we can also create learning curves

Learning curves – exact match

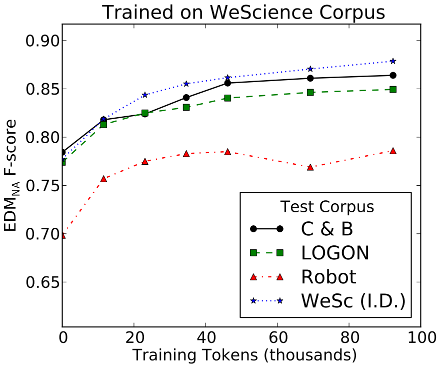


(a) WeScience (b) LOGON

Learning curves – EDM



(c) WeScience

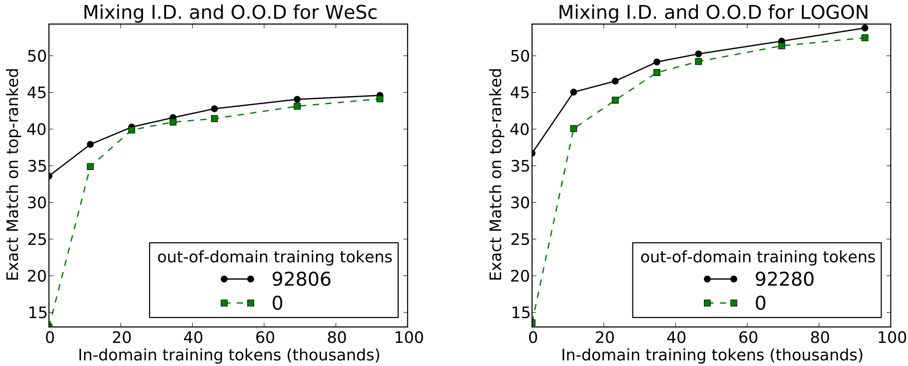


(d) LOGON

Domain Mixing Experiments

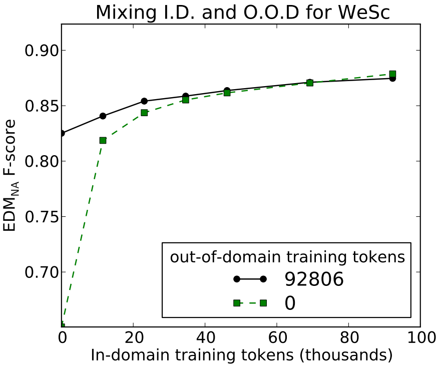
- How much of an improvement in accuracy can we get by treebanking some new sentences in the target domain?
- We use either none or all of the out-of-domain data
- And combine this with varying quantities of "newly treebanked" data in the target domain
- This simulates treebanking new sentences and combining with existing data
- We train a maxent model from concatenated training data – which we call CONCAT.

Mixing training corpora: Concat – exact match

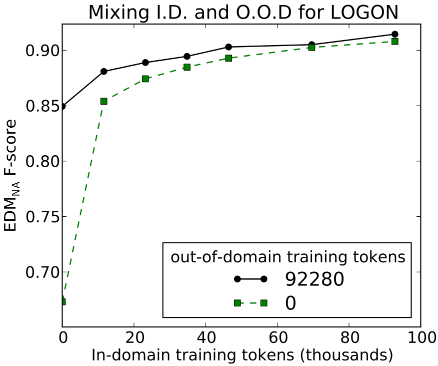


(e) WeScience (f) LOGON

Mixing training corpora: CONCAT – EDM



(g) WESCIENCE



(h) LOGON

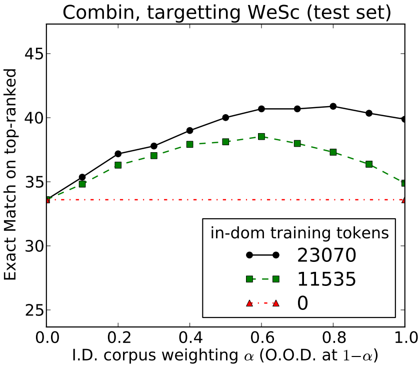
Methods for combining training data

- We now have an idea how much value we can get out of treebanking.
- And also some idea about using as much out-of-domain data as possible
- But can we get better "value" from some given small quantity of treebanked data when combining it with the out-of-domain data?
- We may wish to weight the in-domain data more heavily, since we know it's a good fit

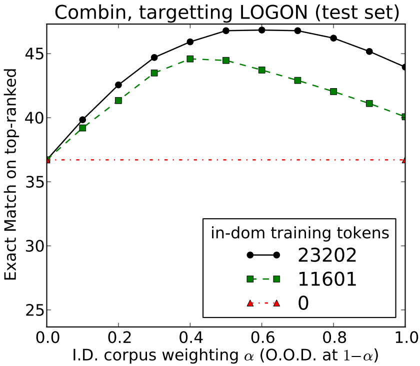
Methods for combining training data (cont.)

- Previously mentioned: CONCAT – simply treat all data as one monolithic block of training data.
- COMBIN – train a model separately using the data from each domain and combine using linear interpolation with some weighting
- DUPLIC – duplicate the data from one of the domains an integral number of times

Mixing training corpora: COMBIN: exact match

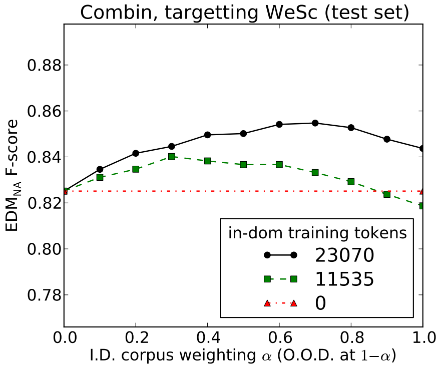


(i) WESCIENCE

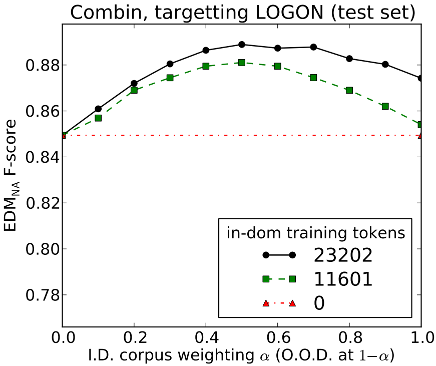


(j) LOGON

Mixing training corpora: COMBIN: EDM

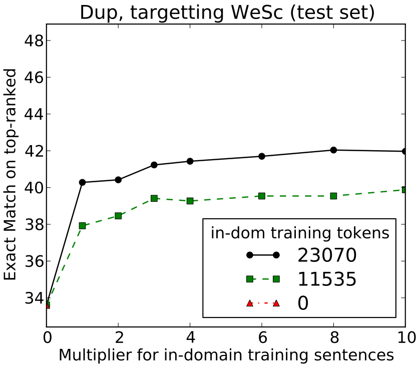


(k) WESCIENCE

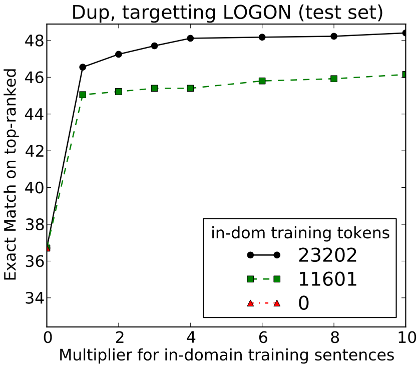


(l) LOGON

Mixing training corpora: DUPLIC: exact match

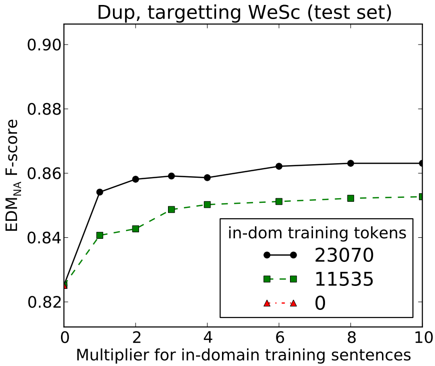


(m) WESCIENCE

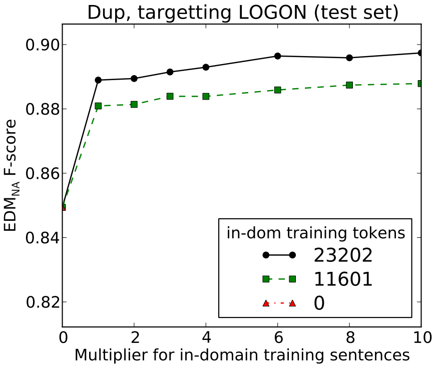


(n) LOGON

Mixing training corpora: DUPLIC: EDM



(o) WESCIENCE



(p) LOGON

Findings

- The ERG does reasonably well with only out-of-domain training data
- But unsurprisingly, in-domain data is much more valuable than out-of-domain.
- On new domains, the choice of training domain matters – some corpora may match better than others.
- EDM scores look good out of the box – this may reflect utility for downstream applications.
- Consequently we see smaller relative changes in EDM scores under different conditions.

More findings

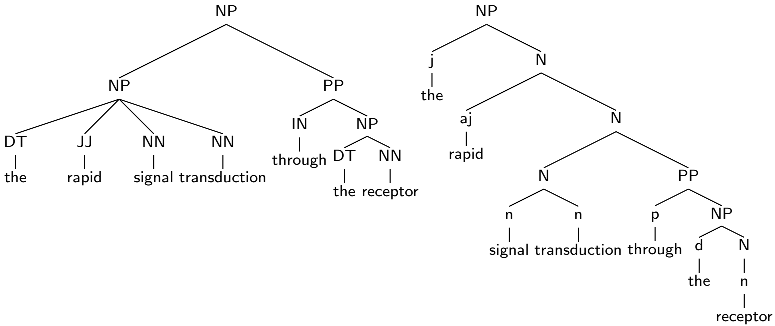
- The relatively modest effort to treebank 750-1500 sentences has a huge payoff
- Simply concatenating this with available out-of-domain data works reasonably
- But by upweighting it, particularly by duplicating the smaller corpus, we get improvements – often significant

Reusing existing treebank annotations

- It sometimes occurs that there *is* a treebank for a new domain/language, it's just not in the right formalism
- Assuming a constituency (PTB-style) treebank, can we use the trees for domain adaptation? What is the relative gain in parse selection accuracy? What is the relative impact on treebanking vs. parse selection?
- Extend earlier work on POS blazing

Methodology

- Translate trees to discriminants, and use to:
  - (in case of parse selection) partition set of analyses into "silver" (possible) and incorrect analyses
  - (in case of treebanking) reduce the set of discriminants directly
- Dealing with systematic differences in parsing style:



Methodology (cont.)

- Perform blazing by:
  1. ignoring cross-bracketing within "embedded" phrases but otherwise use trees verbatim [**IEP**]
  2. binarising trees and reattaching phrases (except parens, commas, conjunctions) [**RP**]
- Select preferred analysis from "silver" analyses via parse selection
- For treebanking, additionally:
  - left-bracket NPs in case of doubt

Setup

- Evaluate over GENIA Treebank, using a new mini-treebank of ~1000 items
- ERG with POS-conditioned unknown word handling via GENIA tagger (incl. NE handling)
- First parse with WeScience parse selection model, and selectively unpack top-500 parses
- Out-of-domain baseline: WeScience parse selection model
- In-domain baseline: self-trained parse selection model

Parse selection

Config	Gold Added	Acc A_1 / A_{10}	EDM_{NA} P / R / F
(WeSc only)	WeSc	12.3 / 39.2	82.4 / 79.2 / 80.7
Random	WeSc	6.1 / 20.0	70.7 / 70.2 / 70.5
Self-train	WeSc	12.9 / 39.2	82.4 / 80.3 / 81.3 *
IEP + S-T	WeSc	12.9 / 39.2	83.5 / 80.9 / 82.2 *** ††
RP + S-T	WeSc	13.3 / 40.1	83.8 / 81.2 / 82.5 *** †††

Parse selection findings

- Self-training is a high baseline, but blazing improves over it (when combined with a self-trained parse selection model)
- Greater improvements for $\mathrm{EDM}_{\mathrm{NA}}$
- Poor results when we treat all silver trees as correct; marginal results without self-trained parse selection

Treebanking

		Standard		Blazed	
Annotator A	Decisions	6.25	7	3.51	4
	Time (sec)	150	144	113	107
Annotator D	Decisions	6.42	7	4.68	4
	Time (sec)	105	101	96	80

Finding

- Treebankers work faster and agree (somewhat) more reliably with tree blazing

Wrap-up

- Moving to a new domain definitely drives down parse selection accuracy, but small amounts of in-domain data (combined with out-of-domain data) lead to significant gains
- Also possible to "recycle" in-domain treebank data in the form of "tree blazing" for both domain tuning and treebanking purposes