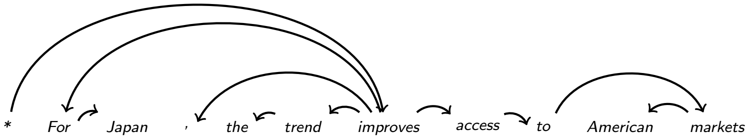
Ensembles of Diverse Cluster-based Discriminative Dependency Parsers

Marzieh Razavi

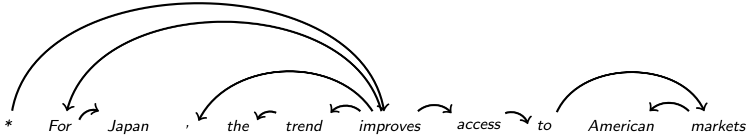
Simon Fraser University

16 August 2012

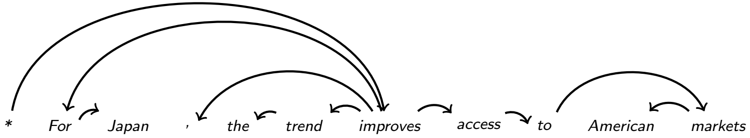
Discriminative Dependency Parsing



Discriminative Dependency Parsing



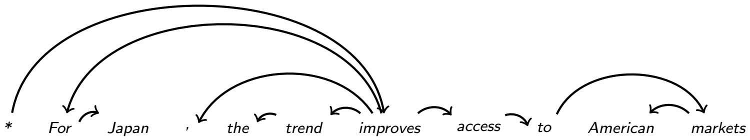
- Structured Linear Model



- Structured Linear Model

$$\mathrm{PARSE}(\mathbf{s}) = \arg\max_{\mathbf{t} \in \mathcal{T}(\mathbf{s})} \mathbf{w} \cdot \mathbf{f}(\mathbf{s}, t) \qquad (1)$$

Discriminative Dependency Parsing

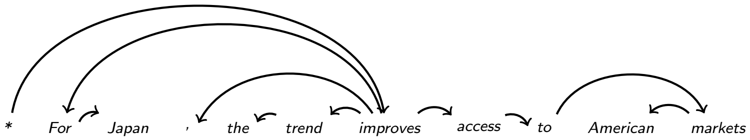


- Structured Linear Model

$$\mathrm{PARSE}(\mathbf{s}) = \arg\max_{\mathbf{t} \in \mathcal{T}(\mathbf{s})} \mathbf{w} \cdot \mathbf{f}(\mathbf{s}, t) \tag{1}$$

- Arc-factored Models

Discriminative Dependency Parsing



- Structured Linear Model

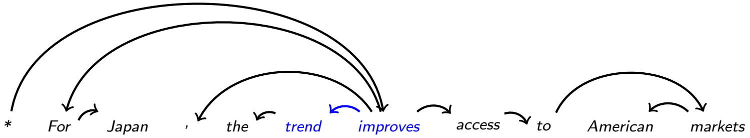
$$\text{PARSE}(\mathbf{s}) = \arg\max_{\mathbf{t} \in \mathcal{T}(\mathbf{s})} \mathbf{w} \cdot \mathbf{f}(\mathbf{s}, t) \tag{1}$$

- Arc-factored Models

$$\text{PARSE}(\mathbf{s}) = \arg\max_{\mathbf{t} \in \mathcal{T}(\mathbf{s})} \sum_{r \in \mathbf{t}} \mathbf{w} \cdot \mathbf{f}(\mathbf{s}, r) \tag{2}$$

Discriminative Dependency Parsing

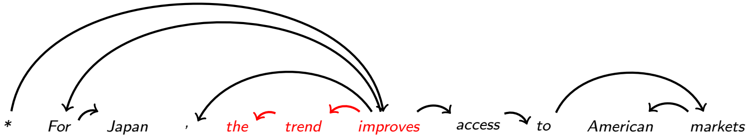
First-order & Second-order Factorization:



- Individual parts (h, m)

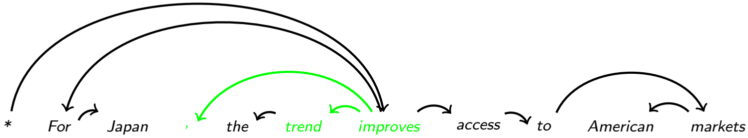
Discriminative Dependency Parsing

First-order & Second-order Factorization:



- grandchild parts (h, m, c)

First-order & Second-order Factorization:



- sibling parts (h, m, si)

Discriminative Dependency Parsing

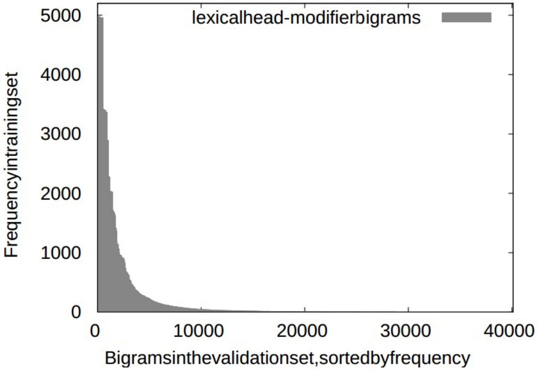
- Flexible feature vector representation
  - Lexical information (e.g. improves, trend)
  - POS tags (e.g. VBZ, NN)

Discriminative Dependency Parsing

- Flexible feature vector representation
  - Lexical information (e.g. improves, trend)
  - POS tags (e.g. VBZ, NN)
- Problem : Sparsity of lexicalized statistics

Discriminative Dependency Parsing

- Flexible feature vector representation
  - Lexical information (e.g. improves, trend)
  - POS tags (e.g. VBZ, NN)
- Problem : Sparsity of lexicalized statistics

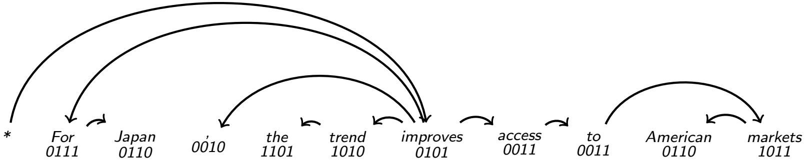


Cluster-based Discriminative Dependency Parsing

- (Koo, Carreras, and Collins, 2008):
  - A simple semi-supervised method
  - Use unlabeled data to extract **word clusters**
  - (Brown et al., 1992) clustering algorithm
  - Incorporate word clusters as features

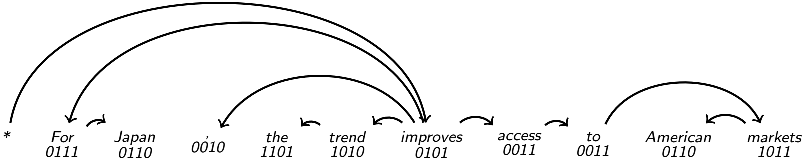
Cluster-based Discriminative Dependency Parsing

- (Koo, Carreras, and Collins, 2008):
  - A simple semi-supervised method
  - Use unlabeled data to extract **word clusters**
  - (Brown et al., 1992) clustering algorithm
  - Incorporate word clusters as features



Cluster-based Discriminative Dependency Parsing

- (Koo, Carreras, and Collins, 2008):
  - A simple semi-supervised method
  - Use unlabeled data to extract **word clusters**
  - (Brown et al., 1992) clustering algorithm
  - Incorporate word clusters as features



- Can inform the parser about **unknown words** in test data that are clustered with known words

Contribution

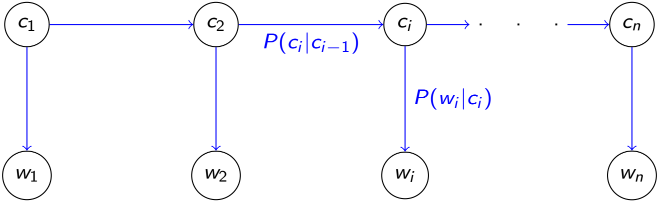
- **Extend** the framework of (Koo, Carreras, and Collins, 2008): use **multiple diverse** clustering methods
- **Ensemble model**: **exact inference** on the **shared hypothesis** space of base models
- **Improving** unlabeled dependency accuracy from 90.82% to 92.46% on Sec. 23 of PTB
- Significant improvements in **domain adaptation** to the Switchboard and Brown corpora

Multiple Clustering Methods

Multiple Clustering Methods

Brown Algorithm:

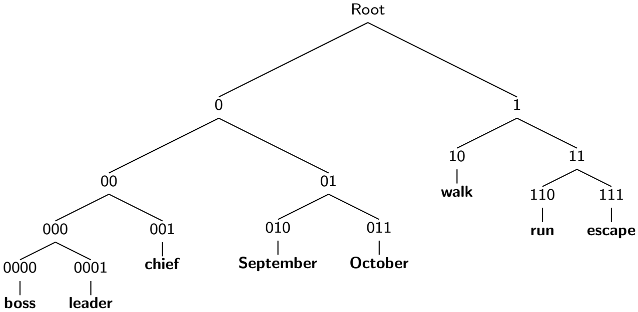
- Bottom-up Algorithm
- Repeatedly merges the two clusters that maximize the likelihood of the input according to class-based bigram language model:



- By tracing the merges we can obtain a binary tree

Multiple Clustering Methods

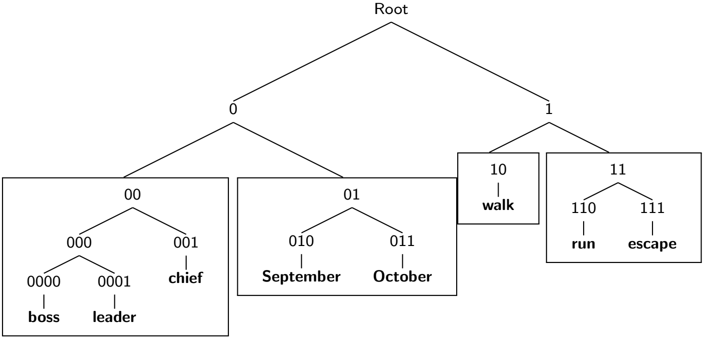
Brown Algorithm:



- Word clusters can be obtained by selecting the nodes at certain depths from the root
  - Determines the granularity of the clustering

Multiple Clustering Methods

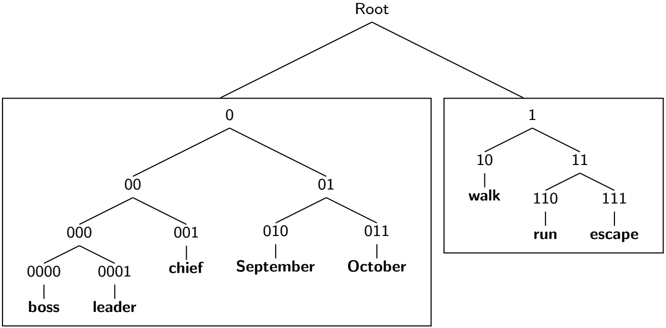
Brown Algorithm:



- Word clusters can be obtained by selecting the nodes at certain depths from the root
  - Determines the granularity of the clustering

Multiple Clustering Methods

Brown Algorithm:



- Word clusters can be obtained by selecting the nodes at certain depths from the root
  - Determines the granularity of the clustering

Multiple Clustering Methods

HMM State Splitting

- The clusters can be viewed as hidden states in HMM
- Brown Algorithm : **Hard** clustering of words
- Maximize the likelihood function using EM: **Soft** clustering of words
- Hierarchical Split-Merge Technique:

Multiple Clustering Methods

HMM State Splitting

- The clusters can be viewed as hidden states in HMM
- Brown Algorithm : **Hard** clustering of words
- Maximize the likelihood function using EM: **Soft** clustering of words
- Hierarchical Split-Merge Technique:

C_1

C_2

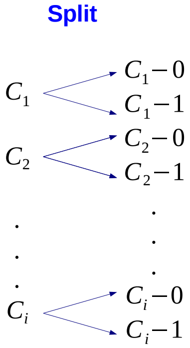
.
.
.

C_i

Multiple Clustering Methods

HMM State Splitting

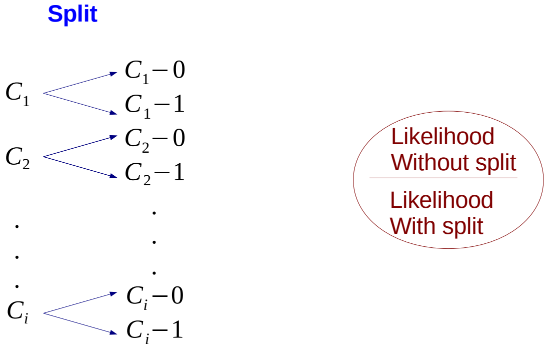
- The clusters can be viewed as hidden states in HMM
- Brown Algorithm : **Hard** clustering of words
- Maximize the likelihood function using EM: **Soft** clustering of words
- Hierarchical Split-Merge Technique:



Multiple Clustering Methods

HMM State Splitting

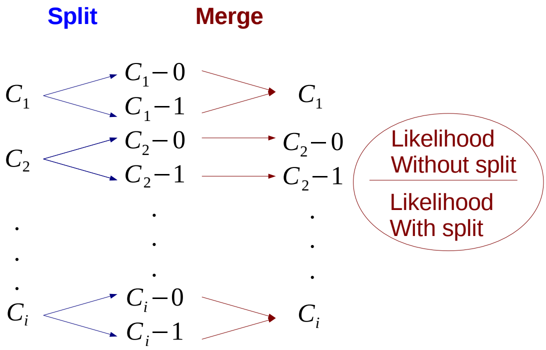
- The clusters can be viewed as hidden states in HMM
- Brown Algorithm : **Hard** clustering of words
- Maximize the likelihood function using EM: **Soft** clustering of words
- Hierarchical Split-Merge Technique:



Multiple Clustering Methods

HMM State Splitting

- The clusters can be viewed as hidden states in HMM
- Brown Algorithm : **Hard** clustering of words
- Maximize the likelihood function using EM: **Soft** clustering of words
- Hierarchical Split-Merge Technique:



Multiple Clustering Methods

- Context specific clustering of HMM vs. hard clustering of Brown:
- Brown Clustering :

1010
milk
pick up
juice
drink
...

- HMM Clustering:

10101111111 **milk (v)** service transport fuel ...	**1010**011111 **milk (n)** cheese fruit water ...

Multiple Clustering Methods

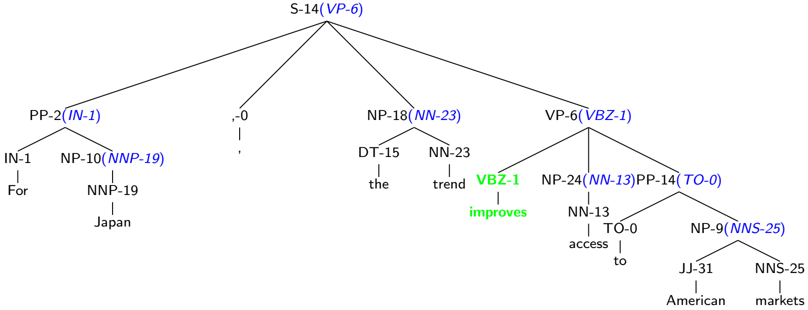
State splitting in PCFGs

- Berkeley Parser : Employs split-merge training of PCFGs (Petrov et al., 2006)

Multiple Clustering Methods

State splitting in PCFGs

- Berkeley Parser : Employs split-merge training of PCFGs (Petrov et al., 2006)

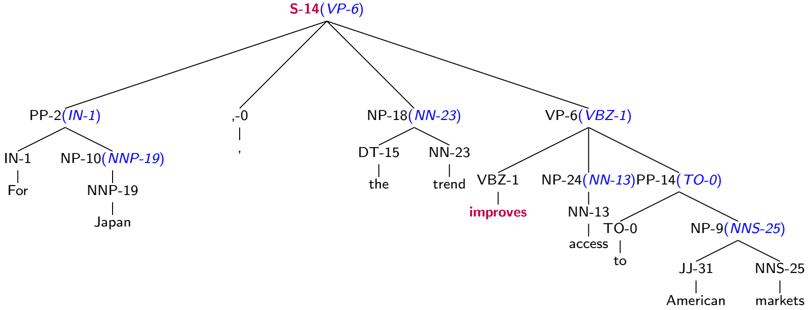


- Syn-Low: Split POS tag from Berkeley parser
  - e.g., improves : VBZ-1

Multiple Clustering Methods

State splitting in PCFGs

- Berkeley Parser : Employs split-merge training of PCFGs (Petrov et al., 2006)



- Syn-High: Split non-terminals from Berkeley parser
  - e.g., improves : S-14

Diversity Among Clustering Annotations

- different HMM Clustering with various randomization seeds:

HMM1	HMM2	HMM3	HMM4	HMM5
stuff	mess	problem	idea	story
list	impression	stuff	stuff	stuff
mess	problem	chore	limitation	limitation
pool	stuff	triangle	argument	mentality

- Diversity of the words clustered with *stuff* in each model
- Intuition: Ensemble of HMMs can lead to a better model
  - different clusterings can be informative to the parser

Diversity Among Clustering Annotations

- different HMM Clustering with various randomization seeds:

HMM1	HMM2	HMM3	HMM4	HMM5
[verb]_	**make_**	**pick_**	**bring_**	**tell_**
stuff	mess	problem	idea	story
list	impression	stuff	stuff	stuff
mess	problem	chore	limitation	limitation
pool	stuff	triangle	argument	mentality

- Diversity of the words clustered with *stuff* in each model
- Intuition: Ensemble of HMMs can lead to a better model
  - different clusterings can be informative to the parser

Ensemble Model

Ensemble Model

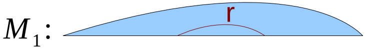
- Linear combination of base models:

$$PARSE(\mathbf{s}) = \arg\max_{\mathbf{t}\in\mathcal{T}(\mathbf{s})} \sum_k \underbrace{\alpha_k}_{\text{model weight}} \sum_{r\in\mathbf{t}} \underbrace{\mathbf{w}_k \cdot \mathbf{f}_k(\mathbf{s}, r)}_{k\text{th base model}}$$

Ensemble Model

- Linear combination of base models:

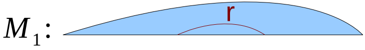
$$PARSE(\mathbf{s}) = \arg\max_{\mathbf{t}\in\mathcal{T}(\mathbf{s})} \sum_k \underbrace{\alpha_k}_{\text{model weight}} \sum_{r\in\mathbf{t}} \underbrace{\mathbf{w}_k \cdot \mathbf{f}_k(\mathbf{s}, r)}_{sc_k(s,r)}$$



$$sc_1(s,r) = w_1 . f_1(s,r)$$

Ensemble Model

- Linear combination of base models:
$$PARSE(\mathbf{s}) = \arg\max_{\mathbf{t}\in\mathcal{T}(\mathbf{s})} \sum_k \underbrace{\alpha_k}_{\text{model weight}} \sum_{r\in\mathbf{t}} \underbrace{\mathbf{w}_k \cdot \mathbf{f}_k(\mathbf{s}, r)}_{sc_k(s,r)}$$



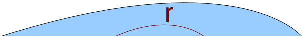
$$sc_1(s,r) = w_1 . f_1(s,r)$$

...

Ensemble Model

- Linear combination of base models:
  $PARSE(\mathbf{s}) = \arg\max_{\mathbf{t}\in\mathcal{T}(\mathbf{s})} \sum_k \underbrace{\alpha_k}_{\text{model weight}} \sum_{r\in\mathbf{t}} \underbrace{\mathbf{w}_k \cdot \mathbf{f}_k(\mathbf{s}, r)}_{sc_k(s,r)}$

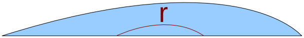
M_1:



$$sc_1(s,r) = w_1 . f_1(s,r)$$

...

M_K:



$$sc_K(s,r) = w_K . f_K(s,r)$$

Ensemble Model

Related Works on ensemble learning:

- Combine **different** dependency parsing systems
- Combining base parsers at parsing time using **voting** (Sagae and Lavie, 2006)
- Integrate base parsers at training time using **stacking** (Nivre and McDonald, 2008)

Ensemble Model

Related Works on ensemble learning:

- Combine **different** dependency parsing systems
- Combining base parsers at parsing time using **voting** (Sagae and Lavie, 2006)
- Integrate base parsers at training time using **stacking** (Nivre and McDonald, 2008)

Alternative Approach

- Concatenate feature sets of the base parsing models
- Train one **joint model** in a discriminative parsing approach

Ensemble Model

Related Works on ensemble learning:

- Combine **different** dependency parsing systems
- Combining base parsers at parsing time using **voting** (Sagae and Lavie, 2006)
- Integrate base parsers at training time using **stacking** (Nivre and McDonald, 2008)

Alternative Approach

- Concatenate feature sets of the base parsing models
- Train one **joint model** in a discriminative parsing approach

Our Approach:

- Combine various **graph-based** models
- **Online** combination of clustering-based models

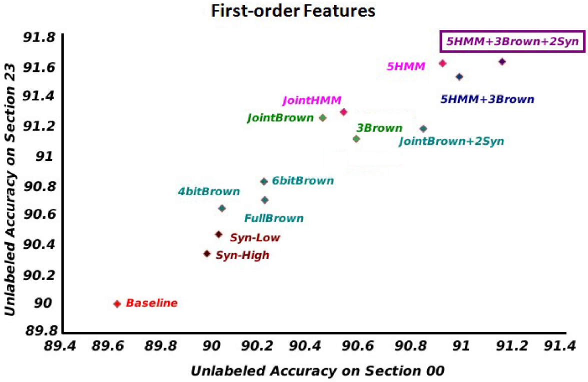
Experiments

In-domain Experimental Set-up

- MSTParser framework (McDonald, Crammer, and Pereira, 2005)
  - Inference : (Eisner, 1996)
  - Extend the feature set to incorporate cluster based features from (Koo, Carreras, and Collins, 2008)
  - Implementation of the ensemble and joint model
- Data : English Penn Treebank
  - Training set : Section 2-21
  - Dev set : section 22
  - Test set : Section 0, 1, 23, 24
  - POS tags : MXPOST (Ratnaparkhi, 1996)
    - Training data created by "leave one section out" method
- Clusters
  - Brown Clustering: Liang's implementation [1] on BLLIP Corpus
  - HMM Clustering: On-House implementation on BLLIP corpus
  - Syntactic Clustering: Berkeley Parser [2]

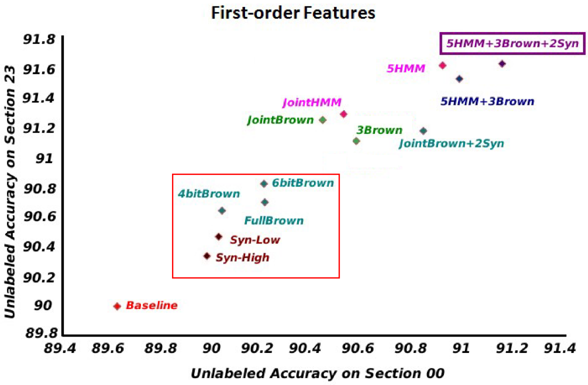
[1] *cs.stanford.edu/∼pliang/software/brown-cluster-1.2.zip*
[2] *code.google.com/p/berkeleyparser*

1^{st} Order Results: Consistent Improvement in Accuracy



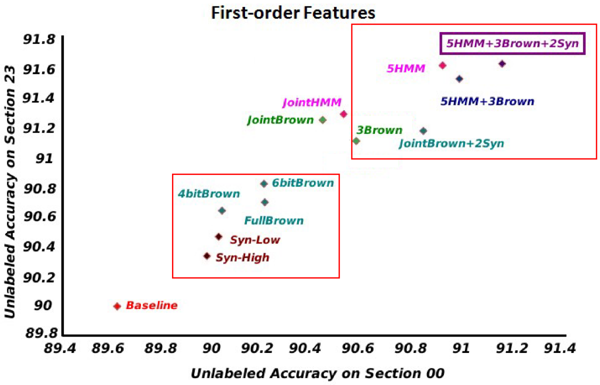
- The ensemble outperforms the baseline and the individual models in all cases
- The ensemble outperforms the joint model in almost all cases

1^{st} Order Results: Consistent Improvement in Accuracy



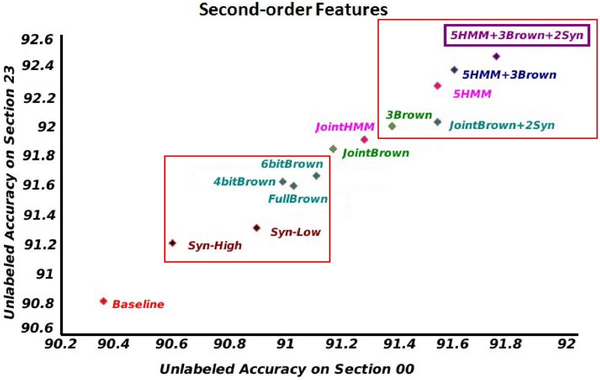
- The ensemble outperforms the baseline and the individual models in all cases
- The ensemble outperforms the joint model in almost all cases

1^{st} Order Results: Consistent Improvement in Accuracy



- The ensemble outperforms the baseline and the individual models in all cases
- The ensemble outperforms the joint model in almost all cases

2^{nd} Order Results: Consistent Improvement in Accuracy



- Incrementally adding number of models
  - Further improvement in accuracy

Out-of-domain Experiments

Set-up:

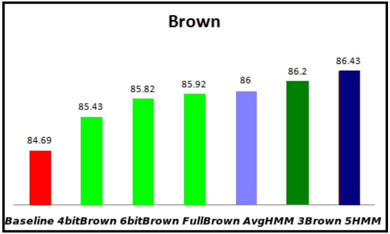
- Brown Corpora : Press reviews
- SwitchBoard Corpora : Phone conversations
  - Larger domain divergence to WSJ

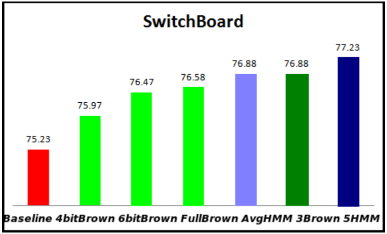
Out-of-domain Experiments

Set-up:

- Brown Corpora : Press reviews
- SwitchBoard Corpora : Phone conversations
  - Larger domain divergence to WSJ

Experiments:





- the ensemble performance improvement for the SwitchBoard is more than that for the Brown corpus

Error Analysis

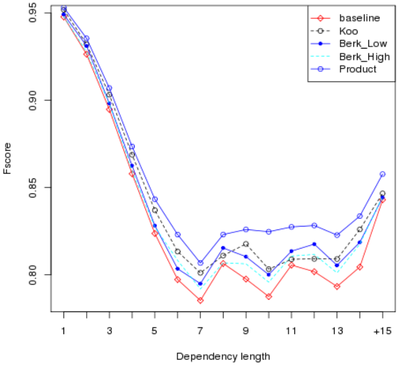


Figure : F-score for each dependency length for in-domain setting.

- Ensemble combines each model's expertise and does best at each dependency length

Error Analysis

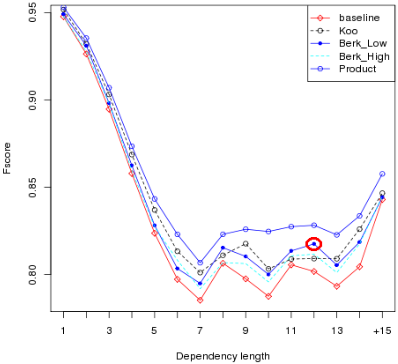


Figure : F-score for each dependency length for in-domain setting.

- Ensemble combines each model's expertise and does best at each dependency length

Error Analysis

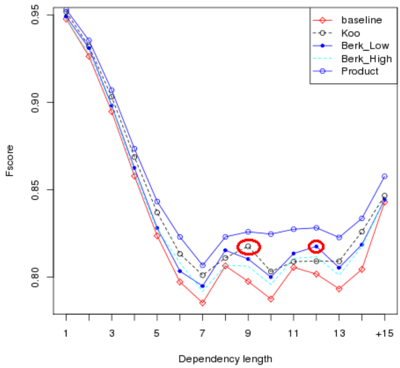


Figure : F-score for each dependency length for in-domain setting.

- Ensemble combines each model's expertise and does best at each dependency length

Error Analysis

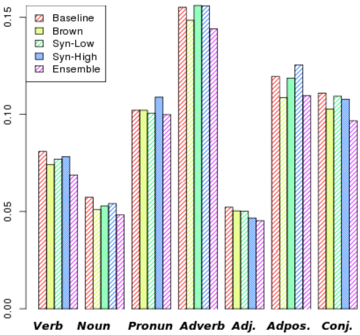


Figure : Error rate of the head attachment for different types of modifier categories for in-domain setting.

- The ensemble always does best in every grammatical category

Error Analysis

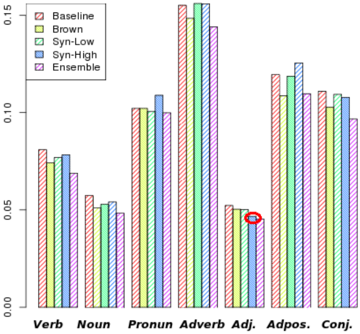


Figure : Error rate of the head attachment for different types of modifier categories for in-domain setting.

- The ensemble always does best in every grammatical category

Error Analysis

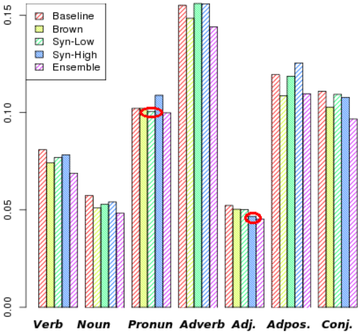


Figure : Error rate of the head attachment for different types of modifier categories for in-domain setting.

- The ensemble always does best in every grammatical category

Learning the models weights

- **Motivation** : benefit from each model's strength on a specific dependency type

Learning the models weights

- **Motivation** : benefit from each model's strength on a specific dependency type
- **Set-up**
  - Modified-MIRA: update the weights based on a specific dependency type
  - dev set : Section 0
  - test set : Section 22, 23

Learning the models weights

- **Motivation** : benefit from each model's strength on a specific dependency type
- **Set-up**
  - Modified-MIRA: update the weights based on a specific dependency type
  - dev set : Section 0
  - test set : Section 22, 23
- **Experimental Results:**
  - Verb and Pronoun : Effective in improving the accuracy (5.6% error reduction)

sec	uniform	Verb
22	91.31	**91.33**
	40.55	**40.96**

sec	uniform	Pronoun
23	91.12	**91.14**
	38.73	**38.77**

  - Learning the model weights can lead to improvements in some cases,but it does not have an important contribution to the overall accuracy

Conclusion and Future directions

Conclusion:

- Ensemble of different parsing models:
  - More Powerful model : improving unlabeled accuracy from 90.82% to 92.46% on Sec. 23 of PTB
  - Strength in Domain Adaptation Scenario : from 75.23% to 77.23% on SwitchBoard data
- Simple uniform ensemble model performs essentially as well as other more complex models

Conclusion and Future directions

Conclusion:

- Ensemble of different parsing models:
  - More Powerful model : improving unlabeled accuracy from 90.82% to 92.46% on Sec. 23 of PTB
  - Strength in Domain Adaptation Scenario : from 75.23% to 77.23% on SwitchBoard data
- Simple uniform ensemble model performs essentially as well as other more complex models

Future directions:

- Experimenting Ensemble on other languages
  - word clusters are shown to be useful for a diverse set of languages (Täckström, McDonald, and Uszkoreit, 2012)
- Using other word representations
  - distributed word representations (Turian, Ratinov, and Bengio, 2010)

Questions?

Bibliography I

Brown, P. F., P. V. deSouza, R. L. Mercer, T. J. Watson, V. J. Della Pietra, and J. C. Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4).

Eisner, J. 1996. Three new probabilistic models for dependency parsing: an exploration. In *COLING*.

Koo, T., X. Carreras, and M. Collins. 2008. Simple semi-supervised dependency parsing. In *Proc. of ACL/HLT*.

McDonald, R., K. Crammer, and F. Pereira. 2005. Online large-margin training of dependency parsers. In *Proc. of ACL*.

Nivre, J. and R. McDonald. 2008. Integrating graph-based and transition-based dependency parsers. In *Proc. of ACL*.

Petrov, S., L. Barrett, R. Thibaux, and D. Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proc. COLING-ACL*.

Bibliography II

Ratnaparkhi, A. 1996. A maximum entropy model for part-of-speech tagging. In *Proc. of EMNLP*.

Sagae, K. and A. Lavie. 2006. Parser combination by reparsing. In *Proc. of NAACL-HLT*.

Täckström, Oscar, Ryan T. McDonald, and Jakob Uszkoreit. 2012. Cross-lingual word clusters for direct transfer of linguistic structure. In *HLT-NAACL*, pages 477–487.

Turian, J., L. Ratinov, and Y. Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proc. of ACL*.