Linguistics? Or Computer Science?

My Personal Experience of Learning Computational Linguistics

Weiwei Sun

Institute of Computer Science and Technology
Peking University

November 23, 2012

PART I: BLAH BLAH BLAH

My story

- Bachelor of Arts in Applied Linguistics (2002-2006)
- Bachelor of Science in Computer Science (2003-2006)
- Master of Science in Computer Science (2006-2009)
- Doctor of Engineering (2009-2012)

What does a linguist care about?

What does a computer scientist care about?

- Elegant mathematical model
- Computationality

What does a computer scientist care about?

- Elegant mathematical model
- Computationality

Frederick Jelinek

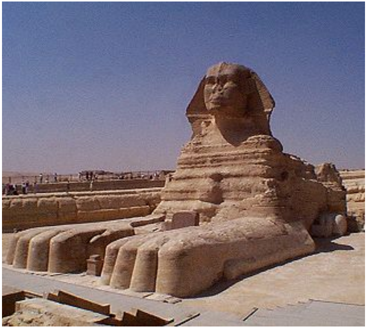
Every time I fire a linguist, the performance of the speech recognizer goes up.

Computational linguistics

Linguistics + Computer Science?

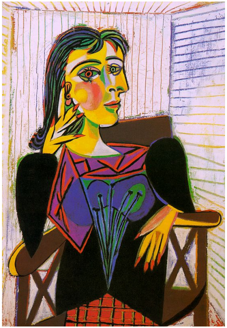
Computational linguistics

Linguistics + Computer Science?



Computational linguistics

Linguistics + Computer Science?



Computational linguistics

Linguistics + Computer Science?

Core tasks in NLP

Raw sentence

警察正在详细调查事故原因

Word segmentation

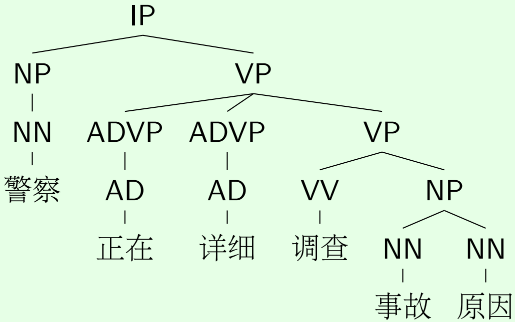
警察 / 正在 / 详细 / 调查 / 事故 / 原因

POS tagging

警察/NN 正在/AD 详细/AD 调查/VV 事故/NN 原因/NN

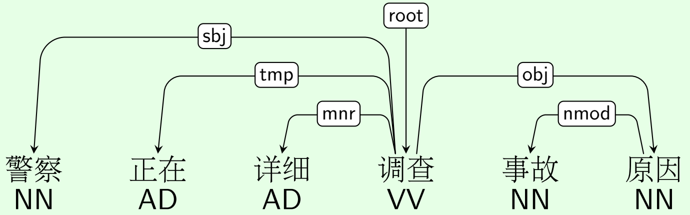
Core tasks in NLP

Constituent tree



Core tasks in NLP

Dependency tree



Not even a full list

- Discourse, Dialogue, and Pragmatics
- Information Extraction
- Information Retrieval
- Language Resources
- Lexical Semantics
- Lexicon and ontology development
- Linguistic Creativity
- Machine Translation
- Multilinguality
- Multimodal representations and processing
- NLP for Web 2.0

Not even a full list (cont)

- NLP in vertical domains, such as biomedical, chemical and legal text
- Natural Language Processing Applications
- Phonology/Morphology, Tagging and Chunking, Word Segmentation
- Question Answering
- Sentiment Analysis and Opinion Mining
- Spoken Language Processing
- Statistical and Machine Learning Methods
- Summarization and Generation
- Syntax and Parsing
- Text Classification
- Text Mining
- User Studies and Evaluation Methods

PART II: A CASE EXAMPLE

A case study

- Chinese POS tagging has been proven to be very chanllenging.
  - Per-word accuracy: 93-94%
- Requiring sophisticated techniques ⇒ drawing inferences from subtle linguistic knowledge.
- The *value* of a word is determined by
  - paradigmatic lexical relations
  - syntagmatic lexical relations
- Towards accurate Chinese POS tagging:
  - Capturing paradigmatic relations: unsupervised word clustering
  - Capturing syntagmatic relations: model ensemble
- Advance the state-of-the-art.
  - Per-word accuracy: 95+%

Outline

Motivating analysis

Capturing paradigmatic lexical relations

Capturing syntagmatic lexical relations

Combining both

Outline

Motivating analysis

Capturing paradigmatic lexical relations

Capturing syntagmatic lexical relations

Combining both

State-of-the-art methods

Discriminative sequence labeling based methods achieve the state-of-the-art of English POS tagging. (ACL wiki)

Averaged Perceptron	Averaged Perception discriminative sequence model	Collins (2002)
Maxent easiest-first	Maximum entropy bidirectional easiest-first inference	Tsuruoka and Tsujii (2005)
SVMTool	SVM-based tagger and tagger generator	Giménez and Márquez (2004)
Morče/COMPOST	Averaged Perceptron	Spoustová et al. (2009)
Stanford Tagger 1.0	Maximum entropy cyclic dependency network	Toutanova et al. (2003)
Stanford Tagger 2.0	Maximum entropy cyclic dependency network	Manning (2011)
Stanford Tagger 2.0	Maximum entropy cyclic dependency network	Manning (2011)
LTAG-spinal	Bidirectional perceptron learning	Shen et al. (2007)

State-of-the-art methods

Computational solution

Probabilistic model:

$$p(\mathbf{t}|\mathbf{x};\theta) = \frac{\exp(\theta^\top \Phi(\mathbf{t},\mathbf{x}))}{\sum_{\mathbf{t}\in\mathcal{T}^n} \exp(\theta^\top \Phi(\mathbf{t},\mathbf{x}))}$$

Combinatorial optimization:

$$\hat{\mathbf{t}} = \arg\max_{\mathbf{t}\in\mathcal{T}^n} \theta^\top \Phi(\mathbf{t},\mathbf{x})$$

$\Phi(\mathbf{t},\mathbf{x})$ represents rich features:

- Word form features
- Morphological features

How can I find good a θ?

A state-of-the-art system

Features

- Word uni-grams
- Word bi-grams
- Prefix strings
- Suffix strings

A state-of-the-art system

Features for w_i in $\ldots w_{i-2} w_{i-1} w_i w_{i+1} w_{i+2} \ldots$:

- Word uni-grams
- Word bi-grams
- Prefix strings
- Suffix strings

A state-of-the-art system

Features for w_i in $\ldots w_{i-2} w_{i-1} w_i w_{i+1} w_{i+2} \ldots$:

- Word uni-grams: w_{i-2}, w_{i-1}, w_i, w_{i+1}, w_{i+2}
- Word bi-grams
- Prefix strings
- Suffix strings

A state-of-the-art system

Features for w_i in $\ldots w_{i-2} w_{i-1} w_i w_{i+1} w_{i+2} \ldots$:

- Word uni-grams: w_{i-2}, w_{i-1}, w_i, w_{i+1}, w_{i+2}
- Word bi-grams: $w_{i-2}w_{i-1}$, $w_{i-1}w_i$, w_iw_{i+1}, $w_{i+1}w_{i+2}$
- Prefix strings
- Suffix strings

A state-of-the-art system

Features for $w_i = c_1 ... c_n$ in $... w_{i-2} w_{i-1} w_i w_{i+1} w_{i+2} ...$:

- Word uni-grams: w_{i-2}, w_{i-1}, w_i, w_{i+1}, w_{i+2}
- Word bi-grams: $w_{i-2}w_{i-1}$, $w_{i-1}w_i$, w_iw_{i+1}, $w_{i+1}w_{i+2}$
- Prefix strings
- Suffix strings

A state-of-the-art system

Features for $w_i = c_1 ... c_n$ in $... w_{i-2} w_{i-1} w_i w_{i+1} w_{i+2} ...$:

- Word uni-grams: w_{i-2}, w_{i-1}, w_i, w_{i+1}, w_{i+2}
- Word bi-grams: $w_{i-2}w_{i-1}$, $w_{i-1}w_i$, w_iw_{i+1}, $w_{i+1}w_{i+2}$
- Prefix strings: c_1, c_1c_2, $c_1c_2c_3$
- Suffix strings

A state-of-the-art system

Features for $w_i = c_1 ... c_n$ in $... w_{i-2} w_{i-1} w_i w_{i+1} w_{i+2} ...$:

- Word uni-grams: w_{i-2}, w_{i-1}, w_i, w_{i+1}, w_{i+2}
- Word bi-grams: $w_{i-2}w_{i-1}$, $w_{i-1}w_i$, w_iw_{i+1}, $w_{i+1}w_{i+2}$
- Prefix strings: c_1, c_1c_2, $c_1c_2c_3$
- Suffix strings: c_n, $c_{n-1}c_n$, $c_{n-2}c_{n-1}c_n$

A state-of-the-art system

Features for $w_i = c_1 \ldots c_n$ in $\ldots w_{i-2} w_{i-1} w_i w_{i+1} w_{i+2} \ldots$:

- Word uni-grams: w_{i-2}, w_{i-1}, w_i, w_{i+1}, w_{i+2}
- Word bi-grams: $w_{i-2}w_{i-1}$, $w_{i-1}w_i$, w_iw_{i+1}, $w_{i+1}w_{i+2}$
- Prefix strings: c_1, c_1c_2, $c_1c_2c_3$
- Suffix strings: c_n, $c_{n-1}c_n$, $c_{n-2}c_{n-1}c_n$

Discriminative sequential tagging achieves the state-of-the-art of Chinese POS tagging.

System	Acc.
CRF	94.69%

A state-of-the-art system

Features for $w_i = c_1 ... c_n$ in $... w_{i-2} w_{i-1} w_i w_{i+1} w_{i+2} ...$:

- Word uni-grams: w_{i-2}, w_{i-1}, w_i, w_{i+1}, w_{i+2}
- Word bi-grams: $w_{i-2}w_{i-1}$, $w_{i-1}w_i$, $w_i w_{i+1}$, $w_{i+1}w_{i+2}$
- Prefix strings: c_1, c_1c_2, $c_1c_2c_3$
- Suffix strings: c_n, $c_{n-1}c_n$, $c_{n-2}c_{n-1}c_n$

Discriminative sequential tagging achieves the state-of-the-art of Chinese POS tagging.

System	Acc.
CRF	94.69%

A state-of-the-art system

Example

刘华清
副总理
的
这
次
来访

A state-of-the-art system

Example

刘华清
副总理
的
这
次
来访

A state-of-the-art system

Example

刘华清
副总理
的
这
次
来访

A state-of-the-art system

Example

刘华清
副总理
的
这
次
来访

A state-of-the-art system

Example

	Prefix	Suffix
刘华清		
副总理	P1:副;P2:副总;P3:副总理	S1:理;S2:总理;S3:副总理
的		
这		
次		
来访		

A state-of-the-art system

Example

	Prefix	Suffix	POS
刘华清			
副总理	P1:副;P2:副总;P3:副总理	S1:理;S2:总理;S3:副总理	NN
的			
这			
次			
来访			

A state-of-the-art system

Example

	Prefix	Suffix	POS
刘华清	P1:刘;P2:刘华;P3:刘华清	S1:清;S2:华清;S3:刘华清	
副总理	P1:副;P2:副总;P3:副总理	S1:理;S2:总理;S3:副总理	
的	P1:的	S1:的	
这	P1:这	S1:这	
次	P1:次	S1:次	
来访	P1:来;P2:来访	S1:访;S2:来访	

A state-of-the-art system

Example

	Prefix	Suffix	POS
刘华清	P1:刘;P2:刘华;P3:刘华清	S1:清;S2:华清;S3:刘华清	NR
副总理	P1:副;P2:副总;P3:副总理	S1:理;S2:总理;S3:副总理	NN
的	P1:的	S1:的	DEG
这	P1:这	S1:这	DT
次	P1:次	S1:次	M
来访	P1:来;P2:来访	S1:访;S2:来访	NN

Error analysis I

Word frequency	Acc.
0 [Unknown word]	83.55%
1-5	89.31%
6-10	90.20%
11-100	94.88%
101-1000	96.26%
1001-	93.65%

Tagging accuracies relative to word frequency.

Error analysis I

Word frequency	Acc.
0 [Unknown word]	83.55%
1-5	89.31%
6-10	90.20%
11-100	94.88%
101-1000	96.26%
1001-	93.65%

Classifiction of low-frequency words is hard.

Error analysis I

Word frequency	Acc.
0 [Unknown word]	83.55%
1-5	89.31%
6-10	90.20%
11-100	94.88%
101-1000	96.26%
1001-	93.65%

Classifiction of very high-frequency words is hard too.

Error analysis II

- A word projects its grammatical property to its maximal projection.
- A maximal projection syntactically governs all words under it.
- The words under the span of current token thus reflect its syntactic behavior and are good clues for POS tagging.

Length of span	Acc.
1-2	93.79%
3-4	93.39%
5-6	92.19%

Tagging accuracies relative to span length.

Error analysis II

- A word projects its grammatical property to its maximal projection.
- A maximal projection syntactically governs all words under it.
- The words under the span of current token thus reflect its syntactic behavior and are good clues for POS tagging.

Length of span	Acc.	
1-2	93.79%	
3-4	93.39%	$\downarrow$
5-6	92.19%	$\downarrow$

$\#\{$words governed by a word$\}\uparrow$;

Error analysis II

- A word projects its grammatical property to its maximal projection.
- A maximal projection syntactically governs all words under it.
- The words under the span of current token thus reflect its syntactic behavior and are good clues for POS tagging.

Length of span	Acc.	
1-2	93.79%	
3-4	93.39%	$\downarrow$
5-6	92.19%	$\downarrow$

$\#\{$words governed by a word$\}\uparrow$; the prediction difficulty $\uparrow$

What a linguist say?

- Meaning arises from the differences between linguistic units.
- These differences are of two kinds:
  - paradigmatic: concerning substitution
  - syntagmatic: concerning positioning
- Functions:
  - paradigmatic: differentiation
  - syntagmatic: possibilities of combination
- The distinction is a key one in structuralist semiotic analysis.

What a linguist say?

- The *value* of a word is determined by both paradigmatic and syntagmatic lexical relations.
- Both relations have a great impact on POS tagging.

What a linguist say?

- The *value* of a word is determined by both paradigmatic and syntagmatic lexical relations.
- Both relations have a great impact on POS tagging.

Low tagging accuracy of low-frequency words

Lack of knowledge about paradigmatic lexical relations.

What a linguist say?

- The *value* of a word is determined by both paradigmatic and syntagmatic lexical relations.
- Both relations have a great impact on POS tagging.

Low tagging accuracy of low-frequency words

Lack of knowledge about paradigmatic lexical relations.

Low tagging accuracy of words governing long spans

Lack of information about syntagmatic lexical relations.

Outline

Motivating analysis

Capturing paradigmatic lexical relations

Capturing syntagmatic lexical relations

Combining both

Word clustering

Word clustering

Partitioning sets of words into subsets of syntactically or semantically similar words.

- A very useful technique to capture paradigmatic or substitutional similarity among words.
  - Unsuperivsed word clustering explores paradigmatic lexical relations encoded in unlabeled data.
    - A great quantity of unlabeled data can be used $\Rightarrow$ We can automatically acquire a large lexicon
- To bridge the gap between high and low frequency words, word clusters are utilized as features.

Clustering algorithms

Distributional word clustering

Words that appear in similar contexts tend to have similar meanings.

Based on the word bi-gram context:

- Brown clustering

$$P(w_i|w_1, ...w_{i-1}) \approx p(C(w_i)|C(w_{i-1}))p(w_i|C(w_i))$$

- MKCLS clustering

$$P(w_i|w_1, ...w_{i-1}) \approx p(C(w_i)|w_{i-1})p(w_i|C(w_i))$$

Brown and MKCLS Clustering

- Hard clustering: each word belongs to exactly one cluster.
- Good open source tools.
- Successful application to boost named entity recognition and dependency parsing.

Main results

Features	Brown	MKCLS
Supervised	94.48%	
+ #100	94.82%	94.93%
+ #500	94.92%	94.99%
+ #1000	94.90%	95.00%

Main results

Features	Brown	MKCLS
Supervised	94.48%	
+ #100	94.82%↑	94.93%↑
+ #500	94.92%↑	94.99%↑
+ #1000	94.90%↑	95.00%↑

Consistently improved.

Main results

Features	Brown	MKCLS
Supervised	94.48%	
$+$ #100	94.82%$\uparrow$	94.93%$\uparrow$
$+$ #500	94.92%$\uparrow$	94.99%$\uparrow$
$+$ #1000	94.90%$\uparrow$	95.00%$\uparrow$

Consistently improved.

The granularities do not affect much.

Supervised or semi-supervised word segmentation

To cluster Chinese words, we must segment raw texts first.

- Supervised segmenter: a traditional character-based segmenter.
- Semi-supervised segmenter: a character-based segmenter with
  - string knowledges that are automatically induced from unlabeled data.

Features	Segmenter	MKCLS
+ #100	Supervised	94.83%
+ #500	Supervised	94.93%
+ #1000	Supervised	94.95%
+ #100	Semi-supervised	94.97%
+ #500	Semi-supervised	94.88%
+ #1000	Semi-supervised	94.94%

Supervised or semi-supervised word segmentation

To cluster Chinese words, we must segment raw texts first.

- Supervised segmenter: a traditional character-based segmenter.
- Semi-supervised segmenter: a character-based segmenter with
  - string knowledges that are automatically induced from unlabeled data.

Features	Segmenter	MKCLS
+ #100	Supervised	94.83%
+ #500	Supervised	94.93%
+ #1000	Supervised	94.95%
+ #100	Semi-supervised	94.97%
+ #500	Semi-supervised	94.88%
+ #1000	Semi-supervised	94.94%

No significant difference.

Learning curves

Size	Baseline	+Cluster
4.5K	90.10%	91.93%
9K	92.91%	93.94%
13.5K	93.88%	94.60%
18K	94.24%	94.77%
22K	94.48%	95.00%

Learning curves

Size	Baseline	+Cluster
4.5K	90.10%	91.93% $\uparrow$
9K	92.91%	93.94% $\uparrow$
13.5K	93.88%	94.60% $\uparrow$
18K	94.24%	94.77% $\uparrow$
22K	94.48%	95.00% $\uparrow$

Consistently improved.

Two-fold contribution

- Word clustering abstracts context information.
  - This linguistic knowledge is helpful to better correlate a word in a certain context to its POS tag.
- The clustering of the unknown words fights the sparse data.
  - Correlate an unknown word with known words through their classes.

Supervised	94.48%
+Known words' clusters	94.70%
+All words' clusters	95.02%

Evaluation

Two-fold contribution

- Word clustering abstracts context information.
  - This linguistic knowledge is helpful to better correlate a word in a certain context to its POS tag.
- The clustering of the unknown words fights the sparse data.
  - Correlate an unknown word with known words through their classes.

Supervised	94.48%	
+Known words' clusters	94.70%	↑0.22
+All words' clusters	95.02%	

Useful linguistic knowledge.

Two-fold contribution

- Word clustering abstracts context information.
  - This linguistic knowledge is helpful to better correlate a word in a certain context to its POS tag.
- The clustering of the unknown words fights the sparse data.
  - Correlate an unknown word with known words through their classes.

Supervised	94.48%	
+Known words' clusters	94.70%	↑0.22
+All words' clusters	95.02%	↑0.32

Fight the data sparse problem.

Tagging recall of unknown words

	Baseline	+Clustering	Δ
AD	33.33%	42.86%	
CD	97.99%	98.39%	
JJ	3.49%	26.74%	
NN	91.05%	91.34%	
NR	81.69%	88.76%	
NT	60.00%	68.00%	
VA	33.33%	53.33%	
VV	67.66%	72.39%	

Tagging recall of unknown words

	Baseline	+Clustering	Δ
AD	33.33%	42.86%	$<$
CD	97.99%	98.39%	$<$
JJ	3.49%	26.74%	$<$
NN	91.05%	91.34%	$<$
NR	81.69%	88.76%	$<$
NT	60.00%	68.00%	$<$
VA	33.33%	53.33%	$<$
VV	67.66%	72.39%	$<$

The recall of all *unknown words* is improved.

Outline

Capturing syntagmatic lexical relations

- Syntax-free discriminative sequential tagging:
  - Flexible to integrate multiple informance sources.
    - Like word clustering.
  - Reach state-of-the-art [94.48%]
- Syntax-based generative chart parsing:
  - Rely on treebanks.
  - Close to state-of-the-art [93.69%]
- Syntactic structures $\Rightarrow$ Syntagmatic lexical relations

Complementary strengths

A comparative analysis illuminates more precisely the contribution of full syntactic information in Chinese POS tagging.

☺Tagger > ☹Parser	☹Tagger < ☺Parser
open classes	close classes
content words	function words
local disambiguation	global disambiguation

Empirical comparison

	Parser<Tagger		Parser>Tagger
AD	94.15<94.71	AS	98.54>98.44
CD	94.66<97.52	BA	96.15>92.52
CS	91.12<92.12	CC	93.80>90.58
ETC	99.65<100.0	DEC	85.78>81.22
JJ	81.35<84.65	DEG	88.94>85.96
LB	91.30<93.18	DER	80.95>77.42
LC	96.29<97.08	DEV	84.89>74.78
M	95.62<96.94	DT	98.28>98.05
NN	93.56<94.95	MSP	91.30>90.14
NR	89.84<95.07	P	96.26>94.56
NT	96.70<97.26	VV	91.99>91.87
OD	81.06<86.36		
PN	98.10<98.15		
SB	95.36<96.77		
SP	61.70<68.89		
VA	81.27<84.25		
VC	95.91<97.67		
VE	97.12<98.48		

Overall	
Tagger:	94.48%
Parser:	93.69%

- Open classes vs.close classes

Empirical comparison

	Parser<Tagger		Parser>Tagger
AD	94.15<94.71	AS	98.54>98.44
CD	94.66<97.52	BA	96.15>92.52
CS	91.12<92.12	CC	93.80>90.58
ETC	99.65<100.0	DEC	85.78>81.22
JJ	81.35<84.65	DEG	88.94>85.96
LB	91.30<93.18	DER	80.95>77.42
LC	96.29<97.08	DEV	84.89>74.78
M	95.62<96.94	DT	98.28>98.05
NN	93.56<94.95	MSP	91.30>90.14
NR	89.84<95.07	P	96.26>94.56
NT	96.70<97.26	VV	91.99>91.87
OD	81.06<86.36		
PN	98.10<98.15		
SB	95.36<96.77		
SP	61.70<68.89		
VA	81.27<84.25		
VC	95.91<97.67		
VE	97.12<98.48		

Overall

Tagger:	94.48%
Parser:	93.69%

- Open classes vs.close classes

Empirical comparison

Parser<Tagger		Parser>Tagger	
AD	94.15<94.71	AS	98.54>98.44
CD	94.66<97.52	BA	96.15>92.52
CS	91.12<92.12	CC	93.80>90.58
ETC	99.65<100.0	DEC	85.78>81.22
JJ	81.35<84.65	DEG	88.94>85.96
LB	91.30<93.18	DER	80.95>77.42
LC	96.29<97.08	DEV	84.89>74.78
M	95.62<96.94	DT	98.28>98.05
NN	93.56<94.95	MSP	91.30>90.14
NR	89.84<95.07	P	96.26>94.56
NT	96.70<97.26	VV	91.99>91.87
OD	81.06<86.36		
PN	98.10<98.15		
SB	95.36<96.77		
SP	61.70<68.89		
VA	81.27<84.25		
VC	95.91<97.67		
VE	97.12<98.48		

Overall

Tagger:	94.48%
Parser:	93.69%

- Open classes vs.close classes

	Known	Unknown
Tagger	95.22%	81.59%
Parser	95.38%	64.77%

Empirical comparison

	Parser<Tagger		Parser>Tagger
AD	94.15<94.71	AS	98.54>98.44
CD	94.66<97.52	BA	96.15>92.52
CS	91.12<92.12	CC	93.80>90.58
ETC	99.65<100.0	DEC	85.78>81.22
JJ	81.35<84.65	DEG	88.94>85.96
LB	91.30<93.18	DER	80.95>77.42
LC	96.29<97.08	DEV	84.89>74.78
M	95.62<96.94	DT	98.28>98.05
NN	93.56<94.95	MSP	91.30>90.14
NR	89.84<95.07	P	96.26>94.56
NT	96.70<97.26	VV	91.99>91.87
OD	81.06<86.36		
PN	98.10<98.15		
SB	95.36<96.77		
SP	61.70<68.89		
VA	81.27<84.25		
VC	95.91<97.67		
VE	97.12<98.48		

Overall

Tagger:	94.48%
Parser:	93.69%

- Open classes vs.close classes

	Known	Unknown
Tagger	95.22%	81.59%
Parser	95.38%	64.77%

Empirical comparison

Parser<Tagger		Parser>Tagger	
AD	94.15<94.71	AS	98.54>98.44
CD	94.66<97.52	BA	96.15>92.52
CS	91.12<92.12	CC	93.80>90.58
ETC	99.65<100.0	DEC	85.78>81.22
JJ	81.35<84.65	DEG	88.94>85.96
LB	91.30<93.18	DER	80.95>77.42
LC	96.29<97.08	DEV	84.89>74.78
M	95.62<96.94	DT	98.28>98.05
NN	93.56<94.95	MSP	91.30>90.14
NR	89.84<95.07	P	96.26>94.56
NT	96.70<97.26	VV	91.99>91.87
OD	81.06<86.36		
PN	98.10<98.15		
SB	95.36<96.77		
SP	61.70<68.89		
VA	81.27<84.25		
VC	95.91<97.67		
VE	97.12<98.48		

Overall

Tagger:	94.48%
Parser:	93.69%

- Open classes vs.close classes

	Known	Unknown
Tagger	95.22%	81.59%
Parser	95.38%	64.77%

- Content words vs. function words

Empirical comparison

Parser<Tagger		Parser>Tagger	
AD	94.15<94.71	AS	98.54>98.44
CD	94.66<97.52	BA	96.15>92.52
CS	91.12<92.12	CC	93.80>90.58
ETC	99.65<100.0	DEC	85.78>81.22
JJ	81.35<84.65	DEG	88.94>85.96
LB	91.30<93.18	DER	80.95>77.42
LC	96.29<97.08	DEV	84.89>74.78
M	95.62<96.94	DT	98.28>98.05
NN	93.56<94.95	MSP	91.30>90.14
NR	89.84<95.07	P	96.26>94.56
NT	96.70<97.26	VV	91.99>91.87
OD	81.06<86.36		
PN	98.10<98.15		
SB	95.36<96.77		
SP	61.70<68.89		
VA	81.27<84.25		
VC	95.91<97.67		
VE	97.12<98.48		

Overall

Tagger:	94.48%
Parser:	93.69%

- Open classes vs.close classes

	Known	Unknown
Tagger	95.22%	81.59%
Parser	95.38%	64.77%

- Content words vs. function words
- Local disambiguation vs. global disambiguation

Model ensemble

- Model ensemble: voting?

Model ensemble

- Model ensemble: voting?
- Oops! Only two systems.

Model ensemble

- Model ensemble: voting?
- Oops! Only two systems.
- Let's generate more sub-models.

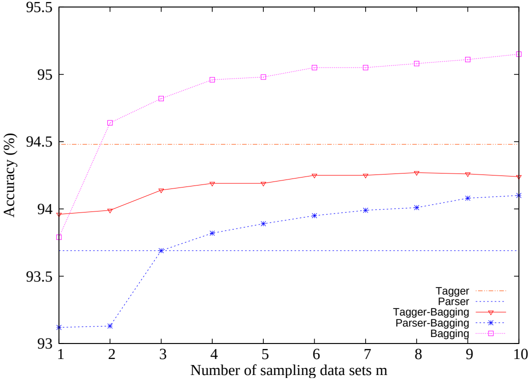
Model ensemble

- Model ensemble: voting?
- Oops! Only two systems.
- Let's generate more sub-models.

A Bagging model

- Generating m new training sets D_i by sampling. [Bootstrap]
- Each D_i is separately used to train a tagger and a parser.
- In the test phase, $2m$ models outputs $2m$ tagging results
- The final prediction is the voting result. [Aggregating]

Results

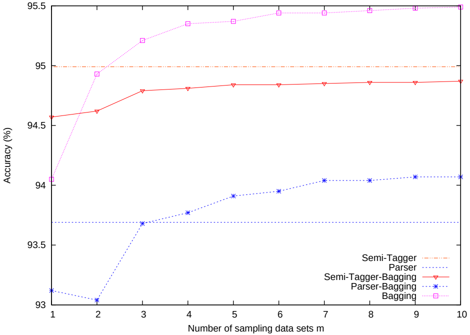


95.5
95
94.5
94
93.5
93
Accuracy (%)
1
2
3
4
5
6
7
8
9
10
Number of sampling data sets m
Tagger
Parser
Tagger-Bagging
Parser-Bagging
Bagging

Outline

Combining both

- Two distinguished improvements: capturing different types of lexical relations
- Further improvement: combining both



Final results

Tagger	94.33%
Tagger+Parser	94.96%
Tagger[+cluster]	94.85%
Tagger[+cluster]+Parser	95.34%

Evaluation

Final results

Tagger	94.33%
Tagger+Parser	94.96%
Tagger[+cluster]	94.85%
Tagger[+cluster]+Parser	95.34%

Baseline achieves state-of-the-art

Final results

Tagger	94.33%
Tagger+Parser	94.96%
Tagger[+cluster]	94.85%
Tagger[+cluster]+Parser	95.34%

Model ensemble helps capture syntagmatic lexical relations

Final results

Tagger	94.33%
Tagger+Parser	94.96%
Tagger[+cluster]	94.85%
Tagger[+cluster]+Parser	95.34%

Learning ensemble helps capture paradigmatic lexical relations

Final results

Tagger	94.33%
Tagger+Parser	94.96%
Tagger[+cluster]	94.85%
Tagger[+cluster]+Parser	95.34%

Two enhancements are not much overlapping

Conclusion I

An interesting question

Part-of-Speech Tagging from 97% to 100%: Is It Time for Some Linguistics?

Christopher D. Manning

Departments of Linguistics and Computer Science
Stanford University

Conclusion I

An interesting question

Part-of-Speech Tagging from 97% to 100%: Is It Time for Some Linguistics?

Christopher D. Manning

Departments of Linguistics and Computer Science
Stanford University

What we've done here

Chinese POS tagging from 94% to 95%: We are inspired by linguistics.

Conclusion I

An interesting question

Part-of-Speech Tagging from 97% to 100%: Is It Time for Some Linguistics?

Christopher D. Manning

Departments of Linguistics and Computer Science
Stanford University

What we've done here

Chinese POS tagging from 94% to 95%: We are inspired by linguistics.

- Paradigmatic lexical relations have a great impact on POS tagging.
- Syntagmatic lexical relations have a great impact on POS tagging.

Conclusion II

- Where is my linguistic knowledge?
- Where is my mathematical knowledge?
- Am I an empiricist?

Game over