Enhancing Unlexicalized Parsing Performance using a Wide Coverage Lexicon, Fuzzy Tag-set Mapping, and EM-HMM-based Lexical Probabilities

Yoav Goldberg   Reut Tsarfaty
Meni Adler   Michael Elhadad

Ben Gurion University

University of Amsterdam

EACL 2009, Athens

Unlexicalized Hebrew Parsing

Learning

- Start with a Treebank

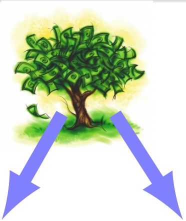
Learning

- Start with a Treebank
- Extract a Grammar

S → NP VP
NP → DT NN
VP → VB NP
. . .
DT → the
NN → cat
NN → cake
NN → dog
VB → ate
VB → kicked

Learning

- Start with a Treebank
- Extract a Grammar
- Assign probabilities to rules



S → NP VP	0.2
NP → DT NN	0.04
VP → VB NP	0.5
...	
DT → the	0.1
NN → cat	0.002
NN → cake	0.005
NN → dog	0.003
VB → ate	0.08
VB → kicked	0.09

Parsing with PCFGs

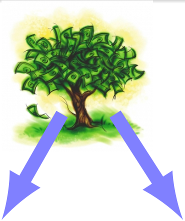
Basic stuff you probably already know

Learning

- Start with a Treebank
- Extract a Grammar
- Assign probabilities to rules

Inference

Standard CKY stuff

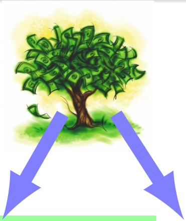


S	$\rightarrow$ NP VP	0.2
NP	$\rightarrow$ DT NN	0.04
VP	$\rightarrow$ VB NP	0.5
...		
DT	$\rightarrow$ the	0.1
NN	$\rightarrow$ cat	0.002
NN	$\rightarrow$ cake	0.005
NN	$\rightarrow$ dog	0.003
VB	$\rightarrow$ ate	0.08
VB	$\rightarrow$ kicked	0.09

Parsing with PCFGs
Two kinds of rules

Syntactic Rules
- Finite (small) set of symbols
- Relative frequency estimates + some smoothing works fine

Lexical Rules
- Huge set of terminal symbols
- Problem with rare events
  - Sparsity
  - Overfitting



S → NP VP	0.2
NP → DT NN	0.04
VP → VB NP	0.5
…	
DT → the	0.1
NN → cat	0.002
NN → cake	0.005
NN → dog	0.003
VB → ate	0.08
VB → kicked	0.09

Syntactic Rules

- Finite (small) set of symbols
- Relative frequency estimates + some smoothing works fine

Lexical Rules

- Huge set of terminal symbols
- Problem with rare events
  - Sparsity
  - Overfitting

S → NP VP	0.2
NP → DT NN	0.04
VP → VB NP	0.5

...

DT → the	0.1
NN → cat	0.002
NN → cake	0.005
NN → dog	0.003
VB → ate	0.08
VB → kicked	0.09

Focus of this work

- Affixation:
  - *and, from, to, the, which, as, in* are prefixes
  - *possessives* are suffixed to nouns

- Affixation:
  - *and, from, to, the, which, as, in* are prefixes
  - *possessives* are suffixed to nouns

In her net ⇒ inhernet

A piece of Hebrew
In (mostly) English words

- Affixation:
  - *and, from, to, the, which, as, in* are prefixes
  - *possessives* are suffixed to nouns

In her net ⇒ inhernet

- Unvocalized writing system
  - most vowels are “dropped” in writing

A piece of Hebrew

In (mostly) English words

- Affixation:
  - *and, from, to, the, which, as, in* are prefixes
  - *possessives* are suffixed to nouns

In her net ⇒ inhernet

- Unvocalized writing system
  - most vowels are “dropped” in writing

in her net ⇒ inhernet ⇒ inhrnt

A piece of Hebrew
In (mostly) English words

- Affixation:
  - *and, from, to, the, which, as, in* are prefixes
  - *possessives* are suffixed to nouns

In her net ⇒ inhernet

- Unvocalized writing system
  - most vowels are “dropped” in writing

in her net ⇒ inhernet ⇒ inhrnt

in her net?
in her note?
in her night?
inherent?

A piece of Hebrew
In (mostly) English words

- Affixation:
  - *and, from, to, the, which, as, in* are prefixes
  - *possessives* are suffixed to nouns

In her net ⇒ inhernet

- Unvocalized writing system
  - most vowels are "dropped" in writing

in her net ⇒ inhernet ⇒ inhrnt

in her net?
in her note?
in her night?
inherent?

- Rich morphology
  - *inherent* could be inflected into different forms according to sing/pl, masc/fem properties

inhrnt, inhrnti, inhrntit, inrntiot, inhrntim

- Affixation:
  - *and, from, to, the, which, as, in* are prefixes
  - *possessives* are suffixed to nouns

In her net ⇒ inhernet

- Unvocalized writing system
  - most vowels are “dropped” in writing

in her net ⇒ inhernet ⇒ inhrnt

in her net?
in her note?
in her night?
inherent?

- Rich morphology
  - *inherent* could be inflected into different forms according to sing/pl, masc/fem properties

inhrnt, inhrnti, inhrntit, inrntiot, inhrntim

- Especially complex verb morphology
  - Root + template morphology for verbs

ktb ⇒ ktb mktyb ywktb htktb kwtb yktwb ykwtb

. . .

Tying it together . . .

The situation in Hebrew

- Complex, productive morphology
- Many word forms (487K distinct tokens in a 34M words corpus)
- High level of ambiguity
  - **2.7** tags/token, vs. 1.4 in English
- POS carries a lot of information
  - gender, number, tense, possesiveness, status,. . .

Tying it together . . .

The situation in Hebrew

- Complex, productive morphology
- Many word forms (487K distinct tokens in a 34M words corpus)
- High level of ambiguity
  - **2.7** tags/token, vs. 1.4 in English
- POS carries a lot of information
  - gender, number, tense, possesiveness, status,. . .

which means

Treebank derived lexicon is inadequate

- Low coverage ⇒ Many unseen events
- Hard to guess POS of unknown words

some baseline parsing performance

but first…

Our parsing setup

Data: Hebrew Treebank V2 ($\sim$ 6000 sentences)

Our parsing setup

Data: Hebrew Treebank V2 ($\sim$ 6000 sentences)

Syntactic Rules (Goldberg and Tsarfaty 2008)

- Parent annotation
- Linguistically motivated state splits
- $p(X \rightarrow Y)$: relative frequency estimate (unsmoothed)

Our parsing setup

Data: Hebrew Treebank V2 ($\sim$ 6000 sentences)

Syntactic Rules (Goldberg and Tsarfaty 2008)

- Parent annotation
- Linguistically motivated state splits
- $p(X \rightarrow Y)$: relative frequency estimate (unsmoothed)

Stable lexical items (seen $\geq K$ times in treebank)

Rare/unseen lexical items (seen $< K$ times)

Our parsing setup

Data: Hebrew Treebank V2 ($\sim$ 6000 sentences)

Syntactic Rules (Goldberg and Tsarfaty 2008)

- Parent annotation
- Linguistically motivated state splits
- $p(X \rightarrow Y)$: relative frequency estimate (unsmoothed)

Stable lexical items (seen $\geq K$ times in treebank)

$p(tag \rightarrow word) = p_{rf}(word|tag)$

Rare/unseen lexical items (seen $< K$ times)

Our parsing setup

Data: Hebrew Treebank V2 ($\sim$ 6000 sentences)

Syntactic Rules (Goldberg and Tsarfaty 2008)

- Parent annotation
- Linguistically motivated state splits
- $p(X \rightarrow Y)$: relative frequency estimate (unsmoothed)

Stable lexical items (seen $\geq K$ times in treebank)

$p(tag \rightarrow word) = p_{rf}(word|tag)$

Fixed

Rare/unseen lexical items (seen $< K$ times)

Our parsing setup

Data: Hebrew Treebank V2 ($\sim$ 6000 sentences)

Syntactic Rules (Goldberg and Tsarfaty 2008)

- Parent annotation
- Linguistically motivated state splits
- $p(X \rightarrow Y)$: relative frequency estimate (unsmoothed)

Stable lexical items (seen $\geq K$ times in treebank)

$p(tag \rightarrow word) = p_{rf}(word|tag)$

Rare/unseen lexical items (seen $< K$ times)

???

Fixed

Varies

Is the low-coverage of the TB lexicon really a problem?

Easy baseline: assuming a segmentation Oracle

Input Sentence:	`inhrnt`
Parser sees:	`in hr nt`

Model

- rare/unknown items replaced with RARE token
- $p(tag \rightarrow word)$ = distribution over rare words:

$$p(word|tag) = \begin{cases} p_{rf}(RARE|tag) & rare \\ p_{rf}(word|tag) & otherwise \end{cases}$$

Is the low-coverage of the TB lexicon really a problem?

Easy baseline: assuming a segmentation Oracle

Input Sentence:	`inhrnt`
Parser sees:	`in hr nt`

Model

- rare/unknown items replaced with RARE token
- $p(tag \rightarrow word)$ = distribution over rare words:

$$p(word|tag) = \begin{cases} p_{rf}(RARE|tag) & rare \\ p_{rf}(word|tag) & otherwise \end{cases}$$

 **72.24 F** (evalb score)

Is the low-coverage of the TB lexicon really a problem?

Realistic baseline: no Oracles

Input Sentence: `inhrnt`
Parser sees: `inhrnt`

Is the low-coverage of the TB lexicon really a problem?

Realistic baseline: no Oracles

Input Sentence:	`inhrnt`
Parser sees:	`inhrnt`

Model

Model of Goldberg and Tsarfaty (2008)

- lattice parser
- non-trivial treebank-based morphological analyzer
  - extended with a spellchecker wordlist
- for details, see paper

Is the low-coverage of the TB lexicon really a problem?

Realistic baseline: no Oracles

Input Sentence: `inhrnt`
Parser sees: `inhrnt`

Model

Model of Goldberg and Tsarfaty (2008)

- lattice parser
- non-trivial treebank-based morphological analyzer
  - extended with a spellchecker wordlist
- for details, see paper

72.24 F (evalb score)
67.02 F (generalized evalb score)

What can we do?

What can we do?

Look outside of the treebank

Dictionary Base Morphological Analyzer

(Developed and maintained by the Knowledge center for processing Hebrew)

What can we do?

Look outside of the treebank

Dictionary Base Morphological Analyzer

(Developed and maintained by the Knowledge center for processing Hebrew)

כתבתי

Noun$_{f,s+gen/b/s/1st}$

Verb$_{b,s,1st,past,PAAL}$

maps word forms to their possible analyses

Treebank vs. Dictionary

- Low Lexical Coverage
  - 6,219 sentences
  - 17,731 unique (non-affixed) word forms
  - 28,349 unique tokens

- High Lexical Coverage
  - 25k lemmas
  - 562,439 (non-prefixed) word forms
  - 73 prefixes and prefixation rules
  - + smart heuristic for unknown words (Adler et al 2008)

Resource Incompatibility

Let's use the Dictionary
for rare words!

Resource Incompatibility

Let's use the Dictionary for rare words!

But the tagsets are different. . .

Resource Incompatibility

Treebank and Dictionary use different tagsets

NN NNT NNP PRP JJ
JJT RB RBR MOD VB
VBMD VBINF AUX AGR
IN COM REL CC QW
HAM WDT DT CD CDE
CDT AT POS

Noun NounC Proper
Pron Adj AdjC Adv Exist
Copula Conj Pref Verb
Beinoni Modal Infinitive
Prep QW Det Num
NumExp NumC At Pos

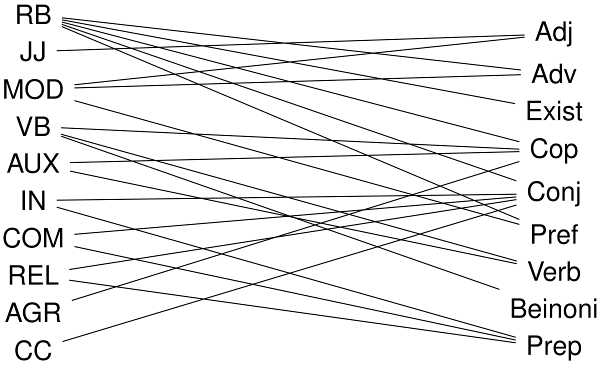
Resource Incompatibility

Treebank and Dictionary use different tagsets

NN	Noun
NNT	NounC
NNP	Proper
AT	At
…	…
POS	Pos

Resource Incompatibility

Treebank and Dictionary use different tagsets



Differences in annotation perspectives

- Syntactic annotation scheme
- “If a word modifies a verb and can be replaced with an adverb, it’s an adverb”

- Lexicographic guidelines
- “If a word can have this inflection, it can be a verb”

Retag the treebank with the dictionary tagset?

Retag the treebank with the dictionary tagset?

A lesson from Arabic

- Arabic TB originally constructed with lexicon-based tags
- Switching to more syntactic tags improved results by $\sim$ 2F-points

(Maamouri et.al 2008)

Hurt parser performance

Retag the treebank with the dictionary tagset?

And in Hebrew

- We re-tagged the treebank
  - $\sim$ 90% automatically, $\sim$ 10% manually
- Gold-morphology Oracle experiment

  Input Sentence: `inhrnt`

  Parser sees: $\texttt{IN}$ $\texttt{PRP}_{f,p}$ $\texttt{NN}_{f,s}$

Resource Incompatibility
Conversion?

Retag the treebank with the dictionary tagset?

And in Hebrew

- We re-tagged the treebank
  - $\sim$ 90% automatically, $\sim$ 10% manually
- Gold-morphology Oracle experiment

Input Sentence:	`inhrnt`	$\Rightarrow$		83.29 F
Parser sees:	`IN` $\texttt{PRP}_{f,p}$ $\texttt{NN}_{f,s}$			81.29 F

Hurt parser performance

Notice – same grammar:

Gold morphology	83.29
Gold segmentation	72.24
Full ambiguity	67.02

– morphology is informative!
– morphology is ambiguous!
– morphology is hard!

Retag the treeb

And in Hebrew

- We re-tagg
  - ∼ 90%
- Gold-morphology Oracle experiment

Input Sentence: `inhrnt`
Parser sees: IN $PRP_{f,p}$ $NN_{f,s}$ ⇒

 83.29 F

 81.29 F

Hurt parser performance

Retag the treebank with the dictionary tagset?

And in Hebrew

- We re-tagged the treebank
  - $\sim$ 90% automatically, $\sim$ 10% manually
- Gold-morphology Oracle experiment

Input Sentence: `inhrnt`
Parser sees: `IN PRP`$_{f,p}$ `NN`$_{f,s}$

$\Rightarrow$


83.29 F


81.29 F

Hurt parser performance

Fuzzy Map

Retag the treebank with the dictionary tagset?

Hurt parser performance

We would like to

- Keep syntactic hints of TB tagging
- Benefit from the large coverage of the Dictionary

Probabilistic Fuzzy Mapping

- Take the best of both worlds
- Define a probabilistic mapping function between the tagsets:

$$p(T_{Dict}|T_{TB})$$

- "*sometimes, demonstrative pronouns function as adjective*"

Layered Trees

The fuzzy map gives rise to a simple generative process:

$$T_{TB} \longrightarrow T_{Dict} \longrightarrow Word$$

Layered Trees

 **TB**	 **Dict**	 **Layered**
⋮ JJ-ZY | זה this	⋮ Pron-M-S-3-DEM | זה this	⋮ JJ-ZY | Pron-M-S-3-DEM | זה this
⋮ IN | במסגרת “inside”	⋮ ⋮ Prep Noun-F-S | | ב מסגרת in frame	⋮ IN / \\ Prep Noun-F-S | | ב מסגרת in frame

Layered Trees

		
TB	**Dict**	**Layered**
⋮ JJ-ZY │ זה this	⋮ Pron-M-S-3-DEM │ זה this	⋮ JJ-ZY │ Pron-M-S-3-DEM │ זה this
⋮ IN │ במסגרת “inside”	⋮ ⋮ Prep Noun-F-S │ │ ב מסגרת in frame	⋮ IN / \\ Prep Noun-F-S │ │ ב מסגרת in frame

Mapping layer

Combining fuzzy-mapping in a parser

New lexical model

Stable words (seen $\geq$ 2 in training)
estimated as usual:

$$p(T_{TB} \rightarrow word) = p_{rf}(word|T_{TB})$$

Rare/unseen words:

$$p(T_{TB} \rightarrow word) = p(T_{TB} \rightarrow T_{Dict})p(T_{Dict} \rightarrow word)$$

Combining fuzzy-mapping in a parser

New lexical model

Stable words (seen ≥ 2 in training) estimated as usual:

$$p(T_{TB} \rightarrow word) = p_{rf}(word | T_{TB})$$

Rare/unseen words:

$$p(T_{TB} \rightarrow word) = p(T_{TB} \rightarrow T_{Dict}) p(T_{Dict} \rightarrow word)$$

But . . . what is $p(T_{Dict} \rightarrow word)$?

Option 1: LexFilter

Use the tag-distribution over rare-words in training, but zero out analyses incompatible with the lexicon:

$$p(T_{Dict} \longrightarrow w_{rare}) =$$

$$p(w_{rare}|T_{Dict}) = \begin{cases} \frac{count(RARE, T_{Dict})}{count(T_{Dict})} & T_{Dict} \in Dict(w_{rare}) \\ 0 & T_{Dict} \notin Dict(w_{rare}) \end{cases}$$

Results

	 Segmentation Oracle	 No Oracle
Baseline 	72.24	67.02
LexFilter  + 	**76.54**	

Results

	 Segmentation Oracle	 No Oracle
Baseline 	72.24	67.02
LexFilter  + 	**76.54**	**68.84**

Results

	Segmentation Oracle 	No Oracle 
Baseline 	72.24	67.02
LexFilter  + 	**76.54**	**68.84**

Realistic performance still low...
can we do better?

Hope in the face of uncertainty

Option 2: LexProb

Consider the familiar HMM Tagging model:

$$p(t_1, \ldots, t_n, w_1, \ldots, w_n) = \prod p(t_i | t_{i-1}, t_{i-2}) p(w_i | t_i)$$

Option 2: LexProb

Consider the familiar HMM Tagging model:

$$p(t_1, \ldots, t_n, w_1, \ldots, w_n) = \prod p(t_i|t_{i-1}, t_{i-2})p(w_i|t_i)$$

Can be estimated from raw text using EM

Option 2: LexProb

Dictionary

Raw Text

Smart Thing

(Adler and Elhadad 2006, Goldberg et.al 2008)

$$P(t|t_{-1}, t_{-2})$$

$$P(w|t)$$

$>$ 92% accuracy

Estimating $p(T_{Dict} \longrightarrow w_{rare})$

Semi-supervised estimation

Option 2: LexProb

Dictionary

Raw Text

Smart Thing

(Adler and Elhadad 2006, Goldberg et.al 2008)

Ignore

$P(t|t_{-1}, t_{-2})$

$P(w|t)$

$>$ 92% accuracy

Use as $P(T_{Dict} \longrightarrow word)$

Results

	Segmentation Oracle	No Oracle
		
Baseline 	72.24	67.02
LexFilter  + 	76.54	68.84
LexProb  +  + 	76.64	

Results

	Segmentation Oracle	No Oracle
Baseline 	72.24	67.02
LexFilter  + 	76.54	68.84
LexProb  +  + 	76.64	**73.69**

Results

	 Segmentation Oracle	 No Oracle
Baseline 	72.24	67.02
LexFilter 	76.54	68.84
LexProb 	76.64	**73.69**

We’re happy

(. . . at least until next year)

Take home message

- Treebank derived lexicons are sparse
  - ⇒ Use an external dictionary / morphological analyzer
- Tagsets may differ
  - ⇒ That's OK. Tagsets may (and should) differ
  - ⇒ Use a fuzzy map
- Dictionaries don't provide probabilities
  - ⇒ Semi-supervised estimation using dictionary and raw text