A Single Generative Model for Joint Morphological Segmentation and Syntactic Parsing

Yoav Goldberg | Computer Science Department | Ben Gurion University of the Negev
P.O.B 653 Be'er Sheva 84105, Israel | yoavg@cs.bgu.ac.il

Reut Tsarfaty | Institute for Logic, Language and Computation | University of Amsterdam
Plantage Muidergracht 24, 1018TV Amsterdam, Netherlands | rtsarfat@science.uva.nl

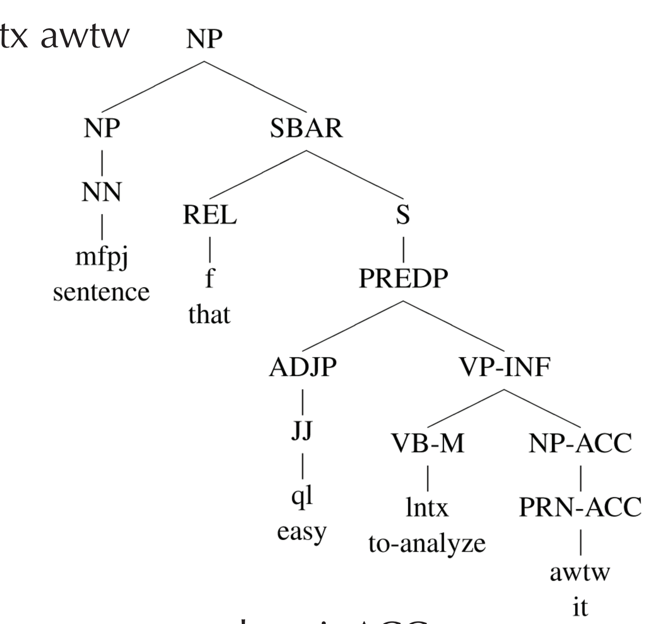
ACL-HLT
Columbus, Ohio, USA
15-20 June 2008

Background

- Word formation processes in morphologically rich languages deliver space-delimited words which introduce multiple, distinct, syntactic units into the syntactic parse tree.
- Morphological Segmentation of space-delimited words to morphemes in Semitic languages is highly ambiguous (Adler and Elhadad 2006, Habash and Rambow 2006, Bar-Haim et al. 2007).
- Correct disambiguation may be facilitated by syntactic context and long distance dependencies (Tsarfaty 2006, Cohen and Smith 2007).

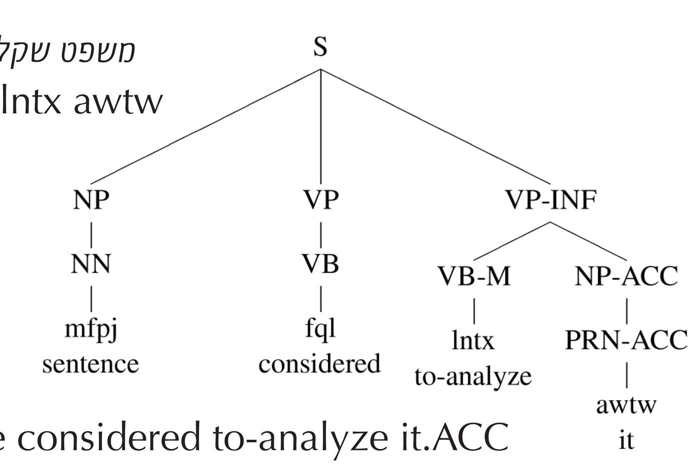
(1) *משפט שקל לנתחו*
mfpj fql lntxw

(a) *משפט ש קל לנתח אותו*
mfpj f ql lntx awtw



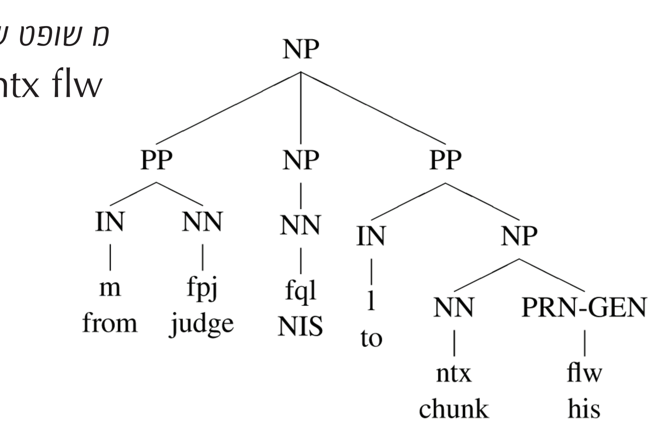
sentence that easy to-analyze it.ACC
A sentence that is easy to analyze

(b) *משפט שקל לנתח אותו*
mfpj fql lntx awtw



sentence considered to-analyze it.ACC
A sentence considered to analyze it

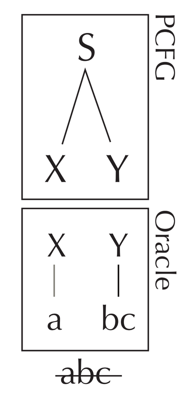
(c) *מ שופט שקל ל נתח שלו*
m fpj fql l ntx flw



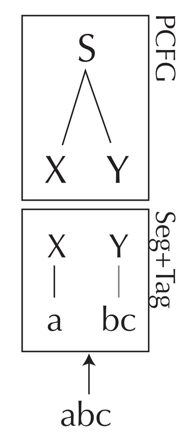
from judge NIS for chunk he.GEN
one NIS from a judge to his chunk

Approaches

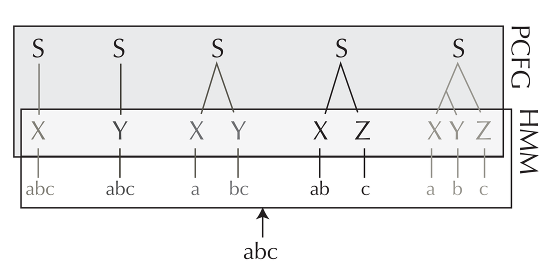
Current Arabic Parsers:
Segmentation/Tagging Oracle



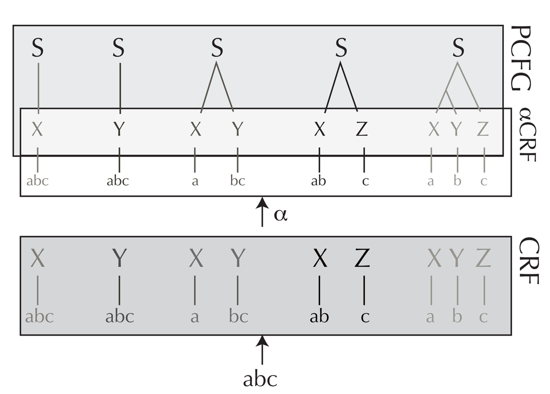
The Naive Solution:
Pipeline



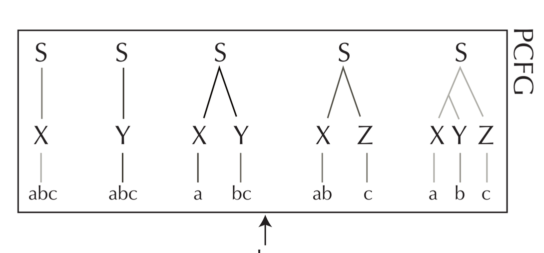
Tsarfaty 2006:
An Integrated Model



Cohen and Smith 2007:
A Factored Model



This work:
A Joint Model



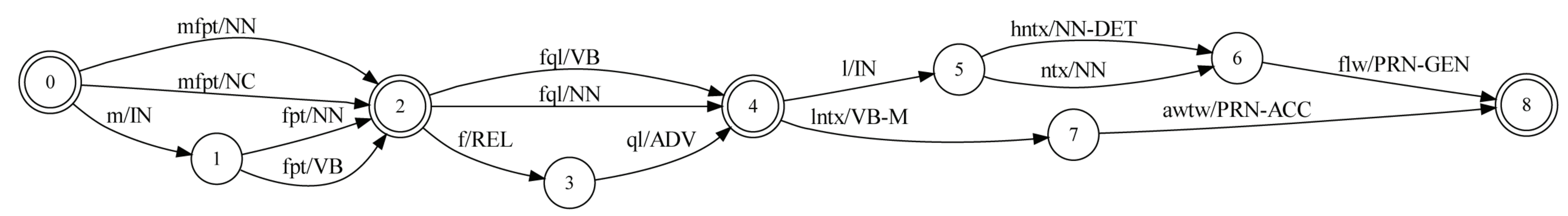
Our Model

A Lattice Representation

The Lattice

The Input — A sequence of surface words $w \in \mathcal{W}, W = w_1, w_2, w_3, ..., w_n$

A morphological analyzer maps sentences to lattices $\mathcal{M} : \mathcal{W} \to \mathcal{L}$



Each token is mapped to a lattice representing its morphological analyses. $\mathcal{M}(w_i) = L_i$

Our lattice is a concatenation of the different word-graphs. $L \in \mathcal{L}, L = L_1\hat{}L_2\hat{}L_3\hat{}...\hat{}L_n$

All segmentation possibilities are represented as lattice paths. $SEG = \{\langle l_1, l_2, ..., l_k \rangle | \langle l_1, l_2, ..., l_k \rangle \in L\}$

Each arc in the lattice corresponds to a tagged segment. $\forall l_i \in L : l_i = \langle s_i, p_i \rangle$

We assume a lexeme-based lexicon consisting of tagged lexemes. $LEX = \{l | l = \langle s_i, p_i \rangle\}$

We assume all lattice path are a-priori equally likely.

A Generative Model

The Grammar

A probabilistic lexeme-based context-free grammar read off of the Modern Hebrew Treebank (Simaan et al. 2001).

Three types of rules:

- **Syntactic rules:** $S \longrightarrow NP \quad VP$
  non-terminal --> a sequence of non-terminals
- **Pre-Terminal Rules:** $VP \longrightarrow Verb$
  non-terminal --> pre-terminal
- **Lexical rules:** $Verb \longrightarrow \langle$ **fkl**, $Verb \rangle$
  pre-terminal --> a lexeme (corresponding to a lattice arc)

The Parser

We look for the most probable parse given the surface forms and morphological analyses. $\pi = argmax_\pi P(\pi | W, \mathcal{M})$

The lattice L is determined by W,M. That is, $P(L | W, \mathcal{M} \approx 1)$

We therefore remain with a model familiar as lattice parsing (cf. Chappelier et al. 1999). $\pi = argmax_\pi P(\pi | L)$

In our model, the most probable parse induces a specific morphological segmentation (cf. PoS tagging Charniak et al. 1996).

The Main Point

When modelling the different lexeme probabilities, we do not treat inter-token lexeme sequences as complex tags, and do not take linear context into account.

Instead, the different lexemes are generated independently based on their corresponding PoS tags.

The context is modeled via the PCFG (sub)derivation resulting in the different lexemes.

For example, we model the probability of the event *fql* resulting in the morpheme sequence $f|$REL $ql|$JJ as:

$$P(\texttt{REL} \longrightarrow f|\texttt{REL}) \times P(\texttt{JJ} \longrightarrow ql|\texttt{JJ})$$

Unknown Tokens Handling

The Problem

Unknown Tokens in Hebrew are doubly unknown:

Unknown token $\mathcal{M}(w) = \emptyset$
Unknown lexeme $l \notin LEX$

Our Data-Driven Solution

1. The Treatment:

a. For Unknown tokens | Propose possible segmentations for an unknown token by chopping off all seen prefixes.
b. For Unknown lexemes | Assign a tag distribution learned for rare-words (#1 occurrence).

2. Lexical Constraints:

Use an external lexical resource (HSPELL) to prune lexically improper segments.

mfpt $\longrightarrow$ **{mfpt, m fpt, ~~m f pt~~}**

3. Gramatical Constraints:

Token-internal collocations unseen in the training data are pruned away.

m fpt $\longrightarrow$ **{~~m fpt/verb,~~ m fpt/noun}**

Experiments and Results

- We tested our system with increasingly complex grammars.
- We investigated the effect of lexical pruning for unknown tokens.

Model	U	SEG_{Tok} / no H	SEG_F	$CPOS$	$FPOS$	SYN / SYN^{CS}	$GS\ SYN$
$\mathbf{GT}_{\mathbf{nohsp/pln}}$	7	89.77 / 93.18	91.80	80.36	76.77	60.41 / 61.66	65.00
···+vpi	7	89.80 / 93.18	91.84	80.37	76.74	61.16 / 62.41	66.70
···+ppp	7	89.79 / 93.20	91.86	80.43	76.79	61.47 / 62.86	67.22
···+nph	7	89.78 / 93.20	91.86	80.43	76.87	61.85 / 63.06	68.23
···+v=2	9	89.12 / 92.45	91.77	82.02	77.86	64.53 / 66.02	70.82
$\mathbf{GT}_{\mathbf{hsp/pln}}$	11	92.00 / 94.81	94.52	82.35	78.11	62.10 / 64.17	65.00
···+vpi	11	92.03 / 94.82	94.58	82.39	78.23	63.00 / 65.06	66.70
···+ppp	11	92.02 / 94.85	94.58	82.48	78.33	63.26 / 65.42	67.22
···+nph	11	92.14 / 94.91	94.73	82.58	78.47	63.98 / 65.98	68.23
···+v=2	13	91.42 / 94.10	94.67	84.23	79.25	66.60 / 68.79	70.82

Table 1: Segmentation, tagging and parsing results on the Standard dev/train Split, for all Sentences.

Model	SEG_{Tok}	$CPOS$	$FPOS$	SYN^{CS}
$\mathbf{GT}_{\mathbf{nohsp/pln}}$	89.50	81.00	77.65	62.22
$\mathbf{GT}_{\mathbf{nohsp/\cdots+nph}}$	89.58	81.26	77.82	64.30
CS_{pln}	*91.10*	*80.40*	*75.60*	*64.00*
$CS_{v=2}$	*90.90*	*80.50*	*75.40*	*64.40*
$\mathbf{GT}_{\mathbf{hsp/pln}}$	93.13	83.12	79.12	64.46
$\mathbf{GT}_{\mathbf{nohsp/\cdots+v=2}}$	89.66	82.85	78.92	66.31
Oracle CS_{pln}	*91.80*	*83.20*	*79.10*	*66.50*
Oracle $CS_{v=2}$	*91.70*	*83.00*	*78.70*	*67.40*
$\mathbf{GT}_{\mathbf{hsp/\cdots+v=2}}$	93.38	85.08	80.11	69.11

Non-lexically pruned
Lexically pruned
Previous work
Upper-Bound/Oracle

Table 2: Segmentation, Parsing and Tagging Results using the Setup of (Cohen and Smith, 2007) (sentence length ≤ 40). The Models are Ordered by Performance.

Analysis

- Our best model **without** lexical pruning outperforms S&C non-oracle results.
- All lexically pruned models outperform S&C non-oracle results.
- Our best lexically-pruned model outperforms S&C **oracle** results.
- Our model doesnt require tuning of hyper-parameters.

Conclusions

- Better grammars yield better results on all tasks (in line with Tsarfaty 2006).
- Parsing and Segmentation, should support, rather than compete with, one another (cf. Cohen and Smith 2007).

To Sum Up

- we propose a single, clean generative model that outperforms previous models on the joint task.
- We present a motivated unknown handling technique based on lexical and grammatical constraints.
- We achieve the best realistic parsing results for Modern Hebrew so far (~70%).

Try this at home! Parse Arabic this way!

References

Meni Adler and Michael Elhadad. 2006. An Unsupervised Morpheme-Based HMM for Hebrew Morphological Disambiguation. In Proceeding of COLING-ACL-06.
Roy Bar-Haim, Khalil Sima'an, and Yoad Winter. 2007. Part-of-speech tagging of Modern Hebrew text. Natural Language Engineering, 14(02):223-251.
J. Chappelier, M. Rajman, R. Aragues, and A. Rozenknop. 1999. Lattice Parsing for Speech Recognition.
Eugene Charniak, Glenn Carroll, John Adcock, Anthony R. Cassandra, Yoshihiko Gotoh, Jeremy Katz, Michael L. Littman, and John McCann. 1996. Taggers for Parsers. AI, 85(1-2):45-57.
David Chiang, Mona Diab, Nizar Habash, Owen Rambow, and Safiullah Shareef. 2006. Parsing Arabic Dialects. In Proceedings of EACL-06.
Shay B. Cohen and Noah A. Smith. 2007. Joint morphological and syntactic disambiguation. In Proceedings of EMNLP-CoNLL-07.
Nizar Habash and Owen Rambow. 2005. Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In Proceeding of ACL-05.
Nadav Har'el and Dan Kenigsberg. 2004. HSpell - the free Hebrew Spell Checker and Morphological Analyzer. Israeli Seminar on Computational Linguistics.
Helmut Schmid. 2004. Efficient Parsing of Highly Ambiguous Context-Free Grammars with Bit Vector. In Proceedings of COLING-04.
Danny Shacham and Shuly Wintner. 2007. Morphological Disambiguation of Hebrew: A Case Study in Classifier Combination. In Proceedings of EMNLP-CoNLL-07.
Khalil Sima'an, Alon Itai, Yoad Winter, Alon Altman, and Noa Nativ. 2001. Building a Tree-Bank for Modern Hebrew Text. In Traitement Automatique des Langues, volume 42.
Noah A. Smith, David A. Smith, and Roy W. Tromble. 2005. Context-based morphological disambiguation with random fields. In Proceedings of HLT-05.
Reut Tsarfaty and Yoav Goldberg. 2008. Word-Based or Morpheme-Based? Annotation Strategies for Modern Hebrew Clitics. In Proceedings of LREC-08.
Reut Tsarfaty. 2006. Integrated Morphological and Syntactic Disambiguation for Modern Hebrew. In Proceedings of ACL-SRW-06.

Acknowledgments

We thank Meni Adler and Michael Elhadad (BGU) and Khalil Simaan (ILLC-UvA) for feedback and discussion.
The work of Y. Goldberg and R. Tsarfaty was supported by the Lynn and William Frankel Center for Computer Sciences and The Netherlands Organization for Scientific Research (NWO) respectively.
Graphic design courtesy of Noa Tsarfaty.