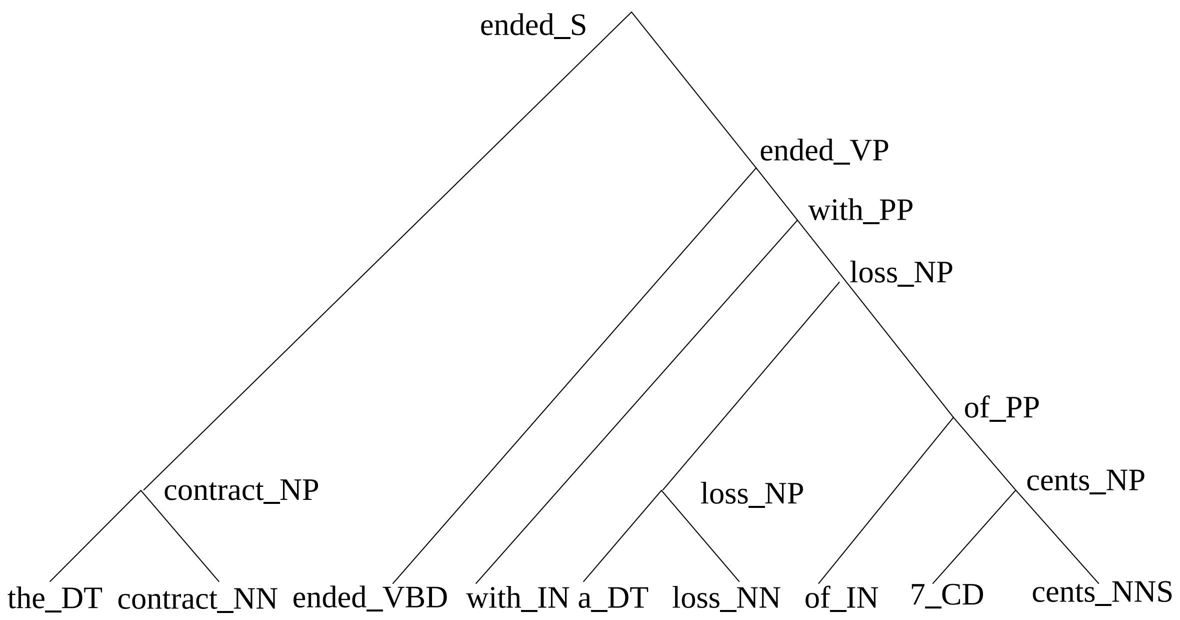
Hierarchical Structure in Natural Language



- Words are hierarchically organized in syntactic constituents — tree structure
- Part of Speech(POS) and Non-Terminal(NT) tags identify the type of constituent
- Lexicalized annotation of intermediate nodes in the tree

Identifying the syntactic structure $\equiv$ Parsing

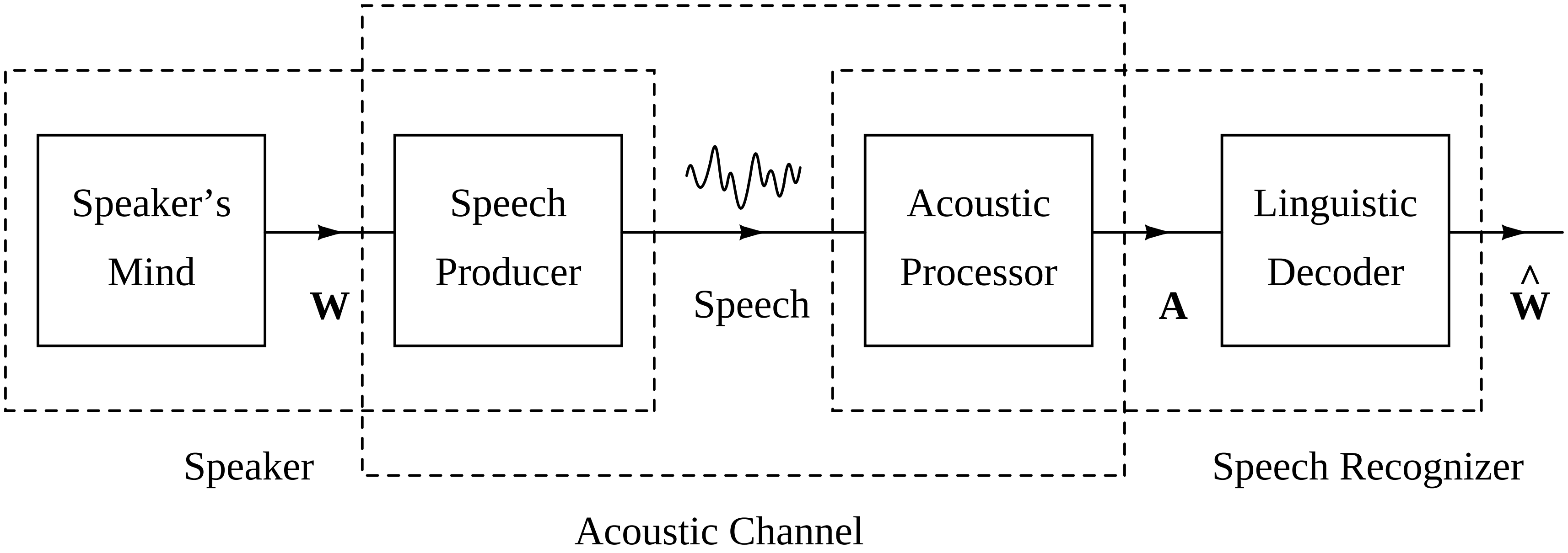
✔ Automatic parsing of natural language text is an area of active research

Exploiting Syntactic Structure for Language Modeling

Ciprian Chelba, Frederick Jelinek

- Hierarchical Structure in Natural Language
- Speech Recognition: Statistical Approach
- Basic Language Modeling:
  - Measures for Language Model Quality
  - Current Approaches to Language Modeling
- A Structured Language Model:
  - Language Model Requirements
  - Word and Structure Generation
  - Research Issues
  - Model Performance: Perplexity results on UPenn-Treebank
  - Model Performance: Perplexity and WER results on WSJ/SWB/BN
- Any Future for the Structured Language Model?
  - Richer Syntactic Dependencies
  - Syntactic Structure Portability
  - Information Extraction from Text

Speech Recognition — Statistical Approach



$$\hat{W} = \text{argmax}_W P(W|A) = \text{argmax}_W P(A|W) \cdot P(W)$$

- $P(A|W)$ *acoustic model*: channel probability;
- $P(W)$ *language model*: source probability;
- *search* for the most likely word string $\hat{W}$.

✔ due to the large vocabulary size — tens of thousands of words — an exhaustive search is intractable.

Basic Language Modeling

Estimate the source probability

$$P(W), \qquad W = w_1, w_2, \ldots, w_n$$

from a training corpus — millions of words of text chosen for its similarity to the expected utterances.

Parametric conditional models:

$$P_\theta(w_i/w_1 \ldots w_{i-1}), \theta \in \Theta, w_i \in \mathcal{V}$$

- Θ parameter space
- $\mathcal{V}$ source alphabet (vocabulary)

✔ Source Modeling Problem

Measures for Language Model Quality

Word Error Rate (WER)

```
TRN: UP UPSTATE NEW YORK SOMEWHERE UH     OVER OVER HUGE AREAS
HYP:    UPSTATE NEW YORK SOMEWHERE UH ALL ALL  THE  HUGE AREAS
     1  0       0   0    0         0  1   1    1    0    0

  :4 errors per 10 words in transcription; WER = 40%
```

Evaluating WER reduction is computationally expensive.

Perplexity(PPL)

$$PPL(M) = \exp\left(-\frac{1}{N}\sum_{i=1}^{N} \ln\left[P_M(w_i|w_1 \ldots w_{i-1})\right]\right)$$

- ✔ different than maximum likelihood estimation: the test data is not seen during the model estimation process;
- ✔ good models are smooth:

$$P_M(w_i|w_1 \ldots w_{i-1}) > \epsilon$$

Current Approaches to Language Modeling

Assume a Markov source of order n; equivalence classification of a given context:

$$[w_1 \dots w_{i-1}] = w_{i-n+1} \dots w_{i-1} = h_n$$

Data sparseness: 3-gram model $(w_i | w_{i-2}, w_{i-1})$

- approx. 70% of the trigrams in the training data have been seen once.
- the rate of new (unseen) trigrams in test data relative to those observed in a training corpus of size 38 million words is 32% for a 20,000-words vocabulary;

Smoothing: recursive linear interpolation among relative frequency estimates of different orders $f_k(\cdot), k = 0 \dots n$ using a recursive mixing scheme:

$$\begin{aligned} &P_n(u|z_1, \dots, z_n) = \\ &\quad \lambda(z_1, \dots, z_n) \cdot P_{n-1}(u|z_1, \dots, z_{n-1}) + (1 - \lambda(z_1, \dots, z_n)) \cdot f_n(u|z_1, \dots, z_n), \\ &P_{-1}(u) = uniform(\mathcal{U}) \end{aligned}$$

Parameters:

$$\theta = \{\lambda(z_1, \dots, z_n); count(u|z_1, \dots, z_n), \forall (u|z_1, \dots, z_n) \in \mathcal{T}\}$$

Exploiting Syntactic Structure for Language Modeling

- Hierarchical Structure in Natural Language
- Speech Recognition: Statistical Approach
- Basic Language Modeling:

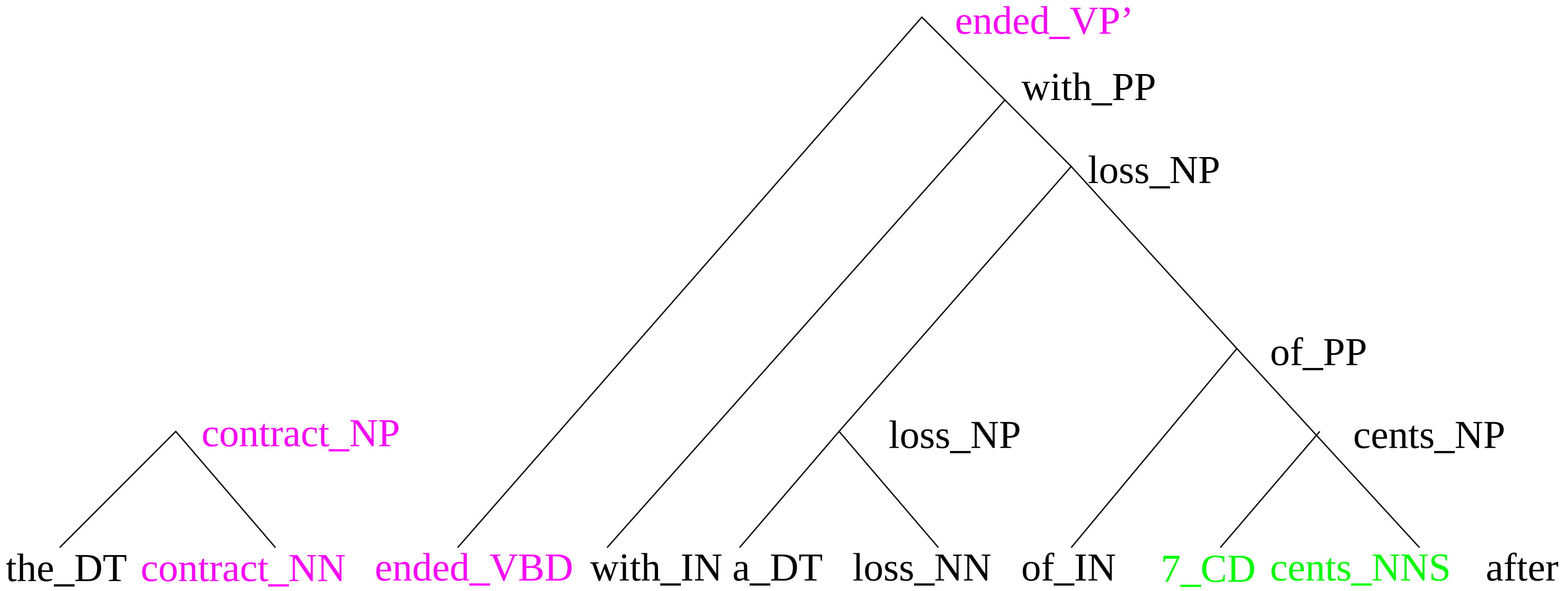
☞ A Structured Language Model:
  - Language Model Requirements
  - Word and Structure Generation
  - Research Issues:
    * Model Component Parameterization
    * Pruning Method
    * Word Level Probability Assignment
    * Model Statistics Reestimation
  - Model Performance: Perplexity results on UPenn-Treebank
  - Model Performance: Perplexity and WER results on WSJ/SWB/BN

A Structured Language Model

- Generalize trigram modeling (local) by taking advantage of sentence structure (influence by more distant past)
- Use exposed heads h (words w and their corresponding non-terminal tags l) for prediction:

$$P(w_i|\mathbf{T}_i) = P(w_i|h_{-2}(\mathbf{T}_i), h_{-1}(\mathbf{T}_i))$$

$\mathbf{T}_i$ is the partial hidden structure, with head assignment, provided to $\mathbf{W}_i$



Language Model Requirements

- Model must operate left-to-right: $P(w_i/w_1 \ldots w_{i-1})$
- In hypothesizing hidden structure, the model can use only word-prefix $\mathbf{W}_i$, *i.e.*, **not** the complete sentence $w_0, ..., w_i, ..., w_{n+1}$ as all conventional parsers do!
- Model complexity must be limited; even trigram model faces critical data sparseness problems
- Model will assign joint probability to sequences of words and hidden parse structure:

$$P(\mathbf{T}_i, \mathbf{W}_i)$$

contract_NP loss_NP

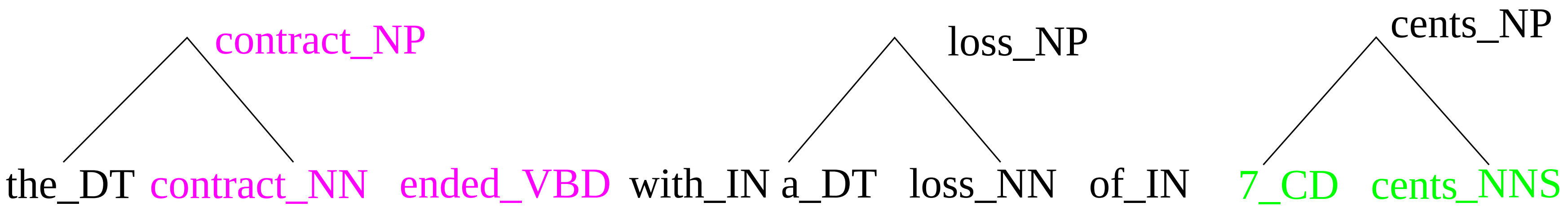
the_DT contract_NN ended_VBD with_IN a_DT loss_NN of_IN 7_CD cents

. . .; **null**; predict `cents`;

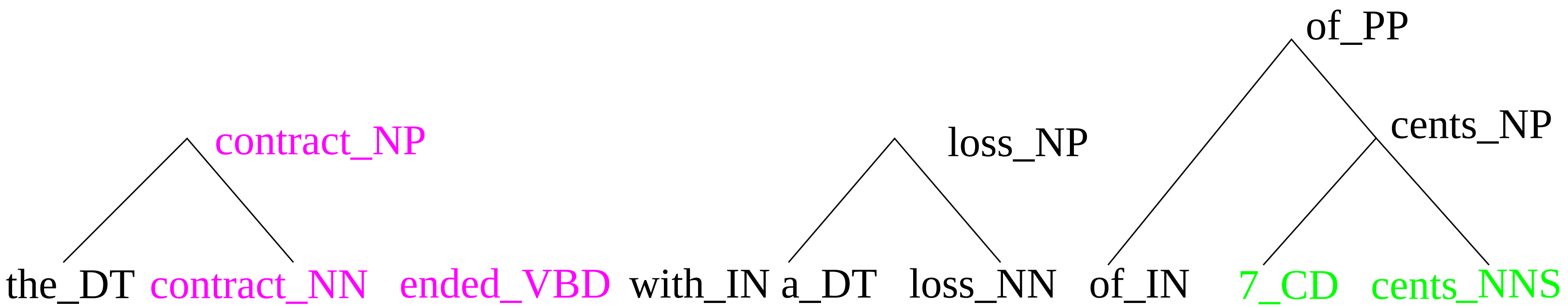
contract_NP loss_NP

the_DT contract_NN ended_VBD with_IN a_DT loss_NN of_IN 7_CD cents_NNS

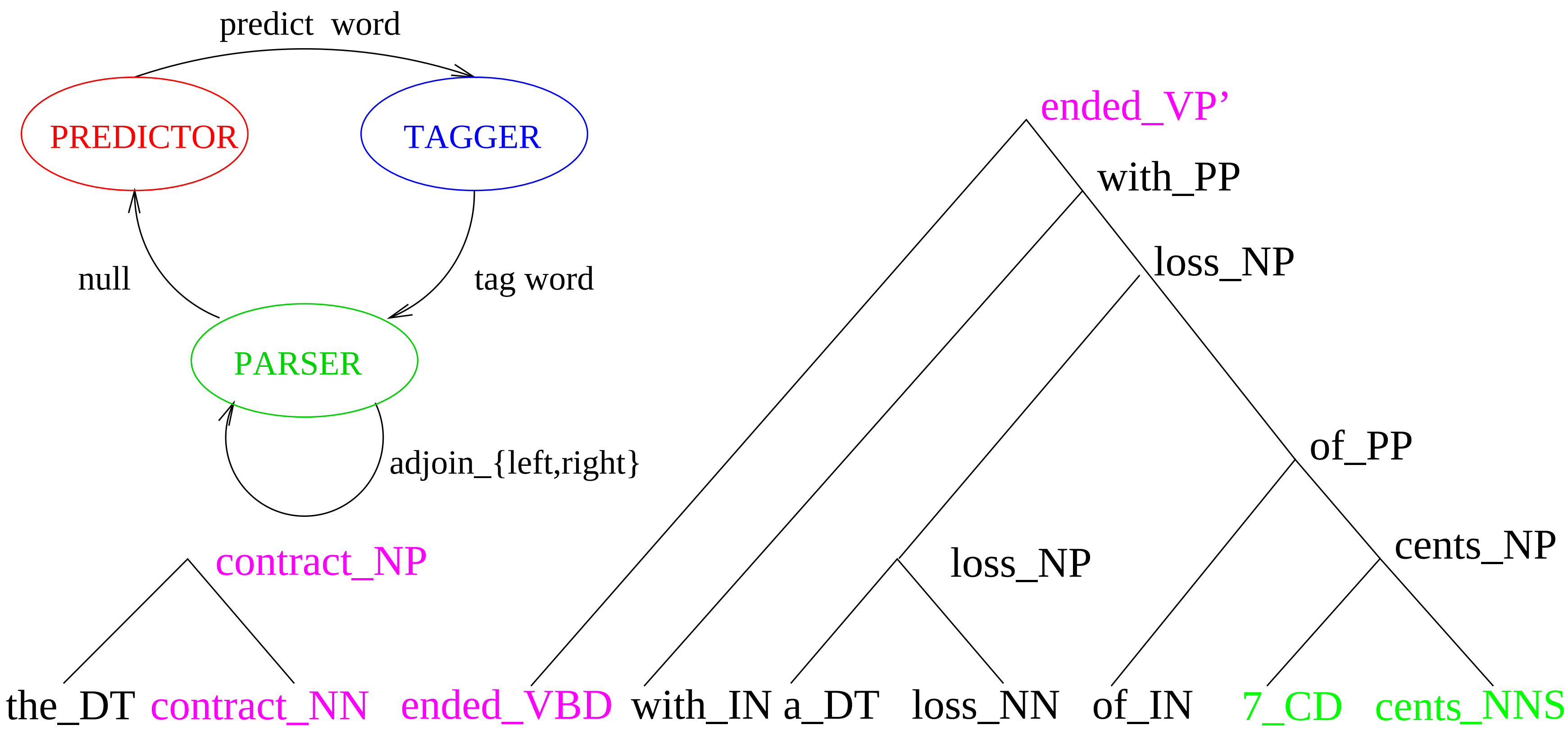
. . .; **null**; predict `cents`; POStag `cents`;



. . .; **null**; predict `cents`; POStag `cents`; **adjoin-right-NP**;



...; **null**; predict `cents`; POStag `cents`; **adjoin-right-NP**; **adjoin-left-PP**;



. . .; **null**; predict `cents`; POStag `cents`; **adjoin-right-NP**; **adjoin-left-PP**; . . .; **adjoin-left-VP'**; **null**; . . .;

Word and Structure Generation

$$P(\mathbf{T}_{n+1}, \mathbf{W}_{n+1}) =$$

$$\prod_{i=1}^{n+1} \underbrace{P(w_i|h_{-2}, h_{-1})}_{\text{predictor}} \underbrace{P(g_i|w_i, h_{-1}.tag, h_{-2}.tag)}_{\text{tagger}} \underbrace{P(\mathbf{T}_i|w_i, g_i, \mathbf{T}_{i-1})}_{\text{parser}}$$

- The **predictor** generates the next word w_i with probability $P(w_i = v|h_{-2}, h_{-1})$
- The **tagger** attaches tag g_i to the most recently generated word w_i with probability $P(g_i|w_i, h_{-1}.tag, h_{-2}.tag)$
- The **parser** builds the partial parse $\mathbf{T}_i$ from $\mathbf{T}_{i-1}, w_i$, and g_i in a series of *moves* ending with **null**, where a parser move a is made with probability $P(a|h_{-2}, h_{-1})$; $a \in \{\texttt{(adjoin-left, NTtag)}, \texttt{(adjoin-right, NTtag)}, \texttt{null}\}$

Research Issues

- Model component parameterization — equivalence classifications for model components:

$$P(w_i = v|h_{-2}, h_{-1}), P(g_i|w_i, h_{-1}.tag, h_{-2}.tag), P(a|h_{-2}, h_{-1})$$

- Huge number of hidden parses — need to prune it by discarding the unlikely ones
- Word level probability assignment — calculate $P(w_i/w_1 \ldots w_{i-1})$
- Model statistics estimation — unsupervised algorithm for maximizing $P(W)$ (minimizing perplexity)

Pruning Method

Number of parses T_k for a given word prefix W_k is $|\{T_k\}| \sim O(2^k)$;

Prune most parses without discarding the most likely ones for a given sentence

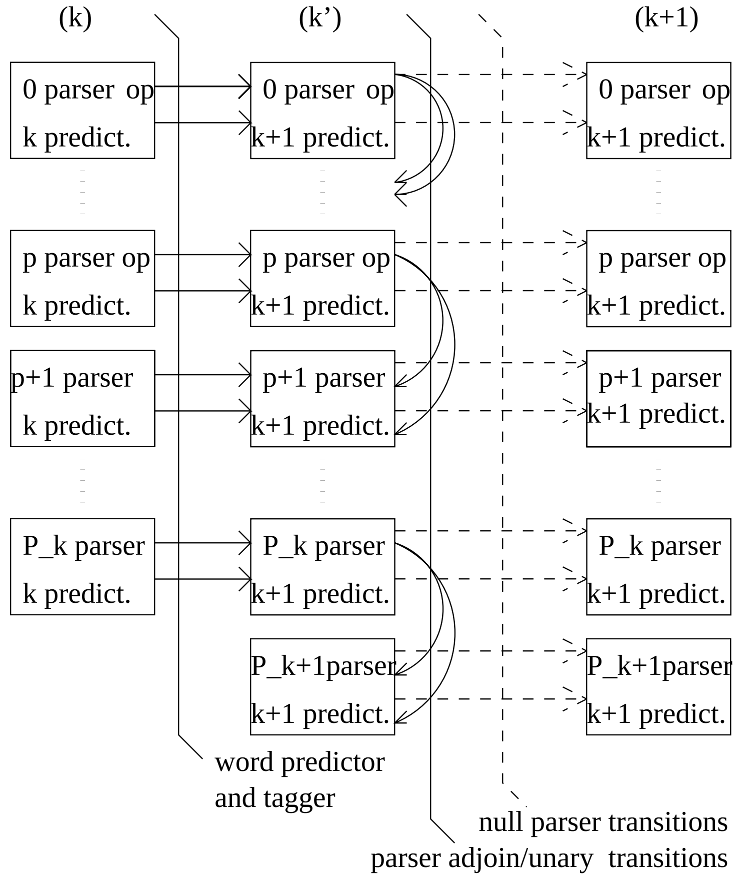
Synchronous Multi-Stack Pruning Algorithm

- the hypotheses are ranked according to $\ln(P(W_k, T_k))$
- each stack contains partial parses constructed by *the same number of parser operations*

The width of the pruning is controlled by:

- maximum number of stack entries
- log-probability threshold

Pruning Method



Word Level Probability Assignment

The probability assignment for the word at position $k+1$ in the input sentence must be made using:

$$P(w_{k+1}/W_k) = \sum_{T_k \in S_k} P(w_{k+1}/W_k T_k) \cdot \rho(W_k, T_k)$$

- S_k is the set of all parses present in the stacks at the current stage k
- interpolation weights $\rho(W_k, T_k)$ must satisfy:

$$\sum_{T_k \in S_k} \rho(W_k, T_k) = 1$$

  in order to ensure a proper probability over strings W^*:

$$\rho(W_k, T_k) = P(W_k T_k) / \sum_{T_k \in S_k} P(W_k T_k)$$

Model Parameter Reestimation

Need to re-estimate model component probabilities such that we decrease the model perplexity.

$$P(w_i = v|h_{-2}, h_{-1}), P(g_i|w_i, h_{-1}.tag, h_{-2}.tag), P(a|h_{-2}, h_{-1})$$

Modified Expectation-Maximization(EM) algorithm:

- We retain the N “best” parses $\{\mathbf{T}^1, \ldots, \mathbf{T}^N\}$ for the complete sentence $\mathbf{W}$
- The hidden events in the EM algorithm are restricted to those occurring in the N “best” parses
- We seed re-estimation process with statistics gathered from manually parsed sentences

Language Model Performance — Perplexity

- Training set: UPenn Treebank text; 930Kwds; manually parsed;
- Test set: UPenn Treebank text; 82Kwds;
- Vocabulary: 10K — out of vocabulary words are mapped to <unk>
- incorporate trigram in word PREDICTOR:

$$P(w_i|W_i) = (1-\lambda) \cdot P(w_i|h_{-2}, h_{-1}) + \lambda \cdot P(w_i|w_{i-1}, w_{i-2}), \lambda = 0.36$$

Language Model		L2R Perplexity		
		DEV set	TEST set	
			no int	3-gram int
Trigram	$P(w_i\|w_{i-2}, w_{i-1})$	21.20	167.14	167.14
Seeded with Treebank	$P_0(w_i\|h_{i-2}, h_{i-1})$	24.70	167.47	152.25
Reestimated	$P(w_i\|h_{i-2}, h_{i-1})$	20.97	158.28	148.90

Language Model Performance — Wall Street Journal

- Training set: WSJ0 "Treebank"-ed text; $\approx$ 20Mwds
  automatically parsed using Ratnaparkhi's MaxEnt parser trained on UPenn-Treebank text (mismatch);
- Test set: DARPA'93 HUB1 3.4kwds, 213 sentences;
- Vocabulary: 20k open, standard
- incorporate trigram in word PREDICTOR:

$$P(w_i|W_i) = \lambda \cdot P(w_i|w_{i-1}, w_{i-2}) + (1-\lambda) \cdot P(w_i|h_{-2}, h_{-1}), \lambda = 0.4$$

  3gram trained on CSR text, 40Mwds
- A^* lattice decoder

Language Model	L2R Perplexity			WER
	DEV set	TEST set		TEST set
		no int	3-gram int	3-gram-int
Trigram	33	147.8	147.8	13.7%
Initial SLM (E0)	39.1	151.9	135.9	13.0%
Reestimated SLM (E3)	34.6	144.1	132.8	13.2%

Language Model Performance — Switchboard

- Training set: Switchboard "Treebank"-ed text; 2.29 Mwds; automatically parsed using SLM;
- Test set: Switchboard "Treebank"-ed text; 28 Kwds (WS97 DevTest), 2427 sentences;
- Vocabulary: 22K closed over test set;
- incorporate trigram in word PREDICTOR:

$$P(w_i|W_i) = \lambda \cdot P(w_i|w_{i-1}, w_{i-2}) + (1-\lambda) \cdot P(w_i|h_{-2}, h_{-1}), \lambda = 0.6$$

- A^* lattice decoder

Language Model	L2R Perplexity			WER
	DEV set	TEST set		TEST set
		no int	3-gram int	3-gram-int
Trigram	22.53	68.56	68.56	41.3%
Initial SLM (E0)	23.94	72.09	65.80	40.6% †
Reestimated SLM (E3)	22.70	71.04	65.35	40.7%

†The WER improvement is significant at level 0.008 according to a sign test at sentence level

25-best rescoring WER was 40.6%

Language Model Performance — Broadcast News

- Training set: $\approx$ 14Mwds;
  automatically parsed using Ratnaparkhi's MaxEnt parser trained on UPenn-Treebank text (mismatch);
- Test set: DARPA'96 HUB4 devtest;
- Vocabulary: 64K open
- incorporate trigram in word PREDICTOR:

$$P(w_i|W_i) = \lambda \cdot P(w_i|w_{i-1}, w_{i-2}) + (1-\lambda) \cdot P(w_i|h_{-2}, h_{-1}), \lambda = 0.4$$

  3gram trained on CSR text, 100Mwds

- A^* lattice decoder

Language Model	L2R Perplexity			WER-F0
	DEV set	TEST set		TEST set
		no int	3-gram int	3-gram-int
Trigram	35.4	217.8	217.8	13.0%
Initial SLM (E0)	57.7	231.6	205.5	12.5%
Reestimated SLM (E3)	40.1	221.7	202.4	12.2%

Exploiting Syntactic Structure for Language Modeling

Ciprian Chelba, Frederick Jelinek

Acknowledgments:

- this research was funded by the NSF grant IRI-19618874 (STIMULATE);
- thanks to Eric Brill, William Byrne, Sanjeev Khudanpur, Harry Printz, Eric Ristad, Andreas Stolcke and David Yarowsky for useful comments, discussions on the model and programming support
- also thanks to:
  Bill Byrne, Sanjeev Khudanpur, Mike Riley, Murat Saraclar for help in generating the SWB, WSJ and BN lattices;

  Adwait Ratnaparkhi for making available the MaxEnt WSJ parser;

  Vaibhava Goel, Harriet Nock and Murat Saraclar for useful discussions about lattice decoding

Exploiting Syntactic Structure for Language Modeling

- Hierarchical Structure in Natural Language
- Speech Recognition: Statistical Approach
- Basic Language Modeling:
  - Measures for Language Model Quality
  - Current Approaches to Language Modeling
- A Structured Language Model:
  - Language Model Requirements
  - Word and Structure Generation
  - Research Issues
  - Model Performance: Perplexity results on UPenn-Treebank
  - Model Performance: Perplexity and WER results on WSJ/SWB/BN

☞ Any Future for the Structured Language Model?

- Richer Syntactic Dependencies
- Syntactic Structure Portability
- Information Extraction from Text

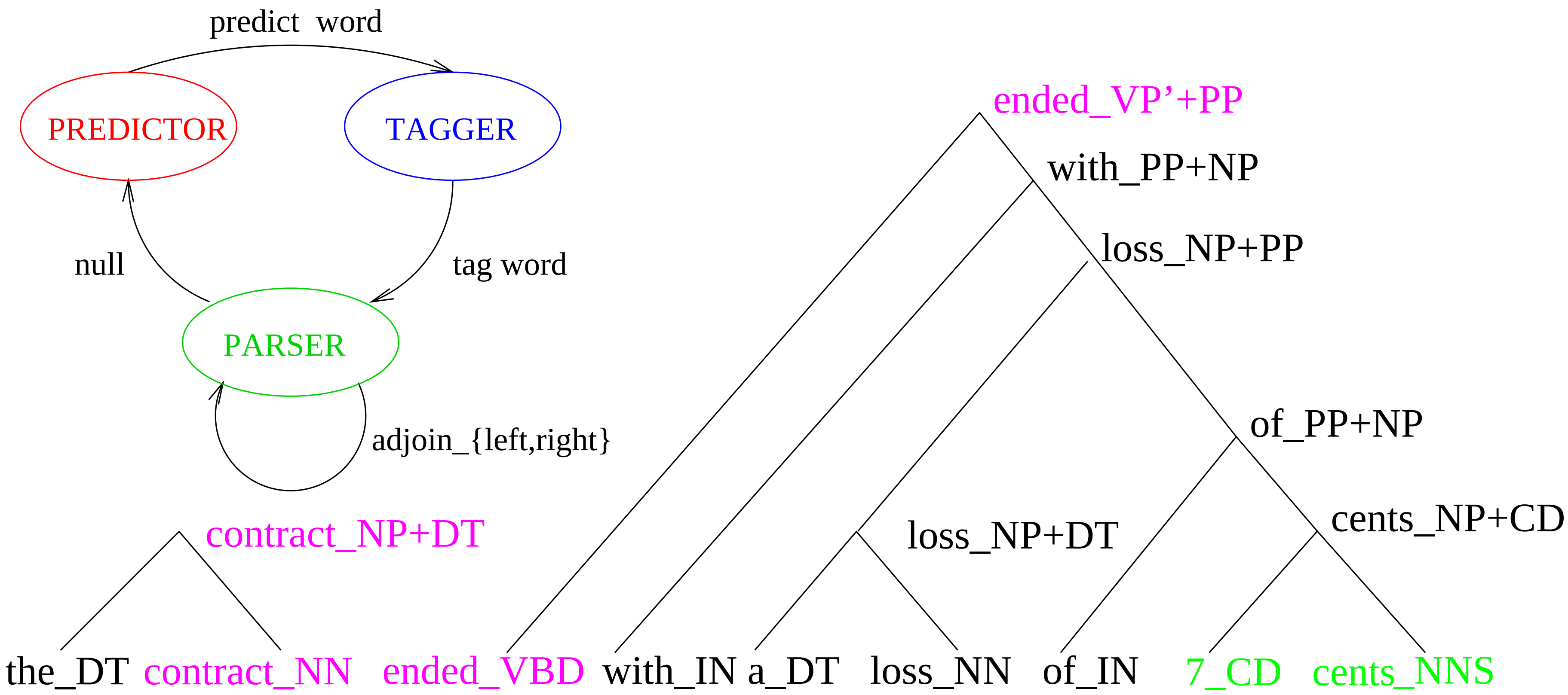
Richer Syntactic Dependencies

Ciprian Chelba, Peng Xu(CLSP)

☞ Is it beneficial to enrich the syntactic dependencies in the SLM?

- 3 simple ways to enrich the syntactic dependencies by modifying the binarization of parse trees:
  - opposite
  - same
  - both
- perplexity and WER results on UPenn Treebank and Wall Street Journal

“Opposite” Enriching Scheme



. . .; **null**; predict `cents`; POStag `cents`; **adjoin-right-NP+CD**; **adjoin-left-PP+NP**; . . .; **adjoin-left-VP’+PP**; **null**; . . .;

Enriched Language Model Performance — Perplexity

- Training set: UPenn Treebank text; 930Kwds; manually parsed;
- Test set: UPenn Treebank text; 82Kwds;
- Vocabulary: 10K — out of vocabulary words are mapped to <unk>
- incorporate trigram in word PREDICTOR:

$$P(w_i|W_i) = (1-\lambda)\cdot P(w_i|h_{-2},h_{-1}) + \lambda \cdot P(w_i|w_{i-1},w_{i-2}), \lambda = 0.6$$

Model	Iter	λ = 0.0	λ = 0.6	λ = 1.0
baseline	3	158.75	148.67	166.63
opposite	3	150.83	<u>144.08</u>	166.63
same	3	155.29	146.39	166.63
both	3	153.30	144.99	166.63
opposite+$h_{-3}.NT$	3	153.60	144.40	166.63

Enriched Language Model Performance — WER

- Training set: WSJ0 "Treebank"-ed text; $\approx$ 20Mwds automatically parsed using Ratnaparkhi's MaxEnt parser trained on UPenn-Treebank text (mismatch);
- Initial parses binarized and enriched using the **opposite** scheme
- Enrich CONSTRUCTOR context with the $h_{-3}.NT$ tag
- Test set: DARPA'93 HUB1 3.4kwds, 213 sentences;
- Vocabulary: 20k open, standard
- incorporate trigram in word PREDICTOR:

$$P(w_i|W_i) = \lambda \cdot P(w_i|w_{i-1}, w_{i-2}) + (1 - \lambda) \cdot P(w_i|h_{-2}, h_{-1})$$

  3gram trained on CSR text, 40Mwds
- N-best rescoring

Model	Iter	Interpolation weight					
		0.0	0.2	0.4	0.6	0.8	1.0
baseline SLM WER, %	0	13.1	13.1	13.1	13.0	13.4	**13.7**
opposite SLM WER, %	0	12.7	12.8	12.7	12.7	13.1	**13.7**
opposite+$h_{-3}.NT$ SLM WER, %	0	12.3	12.4	12.6	12.7	12.8	**13.7**

Syntactic Structure Portability

☞ Is the knowledge of syntactic structure as embodied in the SLM parameters portable accross domains?

- ATIS-III corpus
- Training set: 76k words
- Test set: 9.6k words
- Vocabulary: 1k; OOV rate: 0.5%

Initial Statistics:

- parse the training data (approximatively 76k words) using Microsoft's NLPwin and then intialize the SLM from these parse trees
- use the limited amount of manually parsed ATIS-3 data (approximatively 5k words)
- use the manually parsed data in the WSJ section of the Upenn Treebank.

Syntactic Structure Portability: Perplexity Results

✔ regardless of initialization method, further N-best EM reestimation iterations are carried out on the entire training data (76k wds)

- incorporate trigram in word PREDICTOR:

$$P(w_i|W_i) = \lambda \cdot P(w_i|w_{i-1}, w_{i-2}) + (1-\lambda) \cdot P(w_i|h_{-2}, h_{-1}), \lambda = 0.6$$

Initial Stats	Iter	λ = 0.0	λ = 0.6	λ = 1.0
NLPwin parses	0	21.3	16.7	16.9
NLPwin parses	13	17.2	15.9	16.9
SLM-atis parses	0	64.4	18.2	16.9
SLM-atis parses	13	17.8	15.9	16.9
SLM-wsj parses	0	8311	22.5	16.9
SLM-wsj parses	13	17.7	15.8	16.9

Syntactic Structure Portability: WER Results

- rescoring N-best (N=30) lists generated by the Microsoft Whisper speech recognizer. The 1-best WER —baseline — is 5.8%. The best achievable WER on the N-best lists generated this way is 2.1% — ORACLE WER

Initial Stats	Iter	$\lambda = 0.0$	$\lambda = 0.6$	$\lambda = 1.0$
NLPwin parses	0	6.4	5.6	5.8
NLPwin parses	13	6.4	5.7	5.8
SLM-atis parses	0	6.5	5.6	5.8
SLM-atis parses	13	6.6	5.7	5.8
SLM-wsj parses	0	12.5	6.3	5.8
SLM-wsj parses	13	6.1	5.4	5.8

✔ The model initialized on WSJ parses outperforms the other initialization methods based on in-domain annotated data, achieving a significant 0.4% absolute and 7% relative reduction in WER

Conclusions

- ✔ original approach to language modeling that takes into account the hierarchical structure in natural language
- ✔ devised an algorithm to reestimate the model parameters such that the perplexity of the model is decreased
- ✔ showed improvement in both perplexity and word error rate over current language modeling techniques
- ✔ model initialization is very important
- ✔ code and data is available at `http://www.research.microsoft.com/~chelba`

Future Work

- ✘ better parametrization/statistical modeling tool in model components, especially PREDICTOR and PARSER; potential improvement in PPL from guessing the final best parse is large.

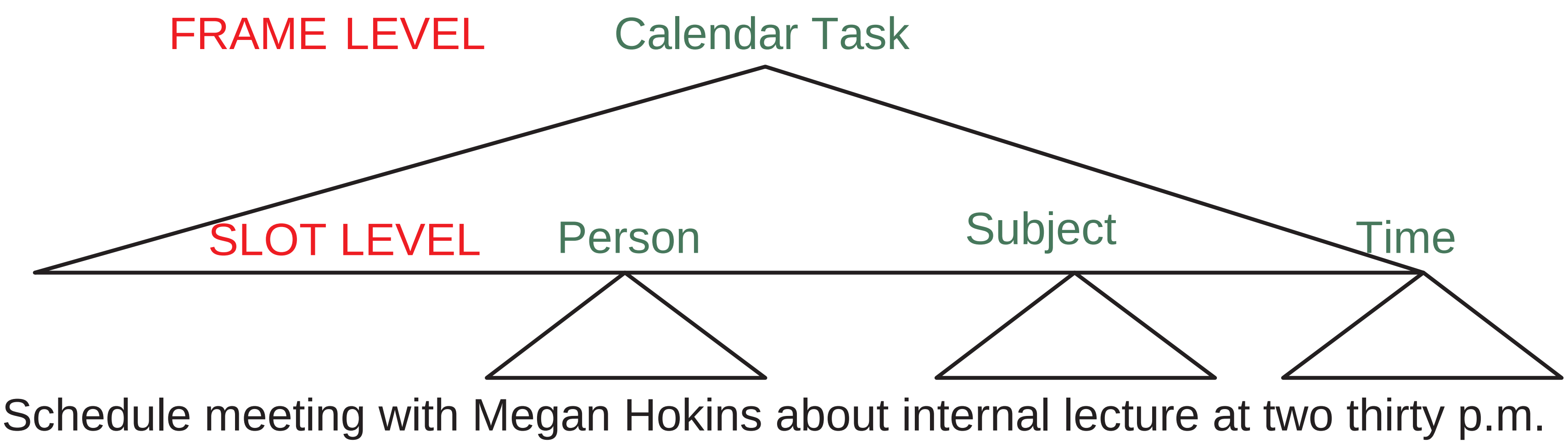
Information Extraction Using the Structured Language Model

Ciprian Chelba, Milind Mahajan

- Information Extraction from Text
- SLM for Information Extraction
- Experiments

Information Extraction from Text

- Information extraction viewed as the recovery of a two level semantic parse S for a given word sequence W
- Sentence independence assumption: the sentence W is sufficient for identifying the semantic parse S



☞ GOAL: Data driven approach with minimal annotation effort: clearly identifiable semantic slots and frames

SLM for Information Extraction

☞ Training:

initialization Initialize SLM as a syntactic parser from treebank

syntactic parsing Train SLM as a matched constrained parser and parse the training data: boundaries of semantic constituents are matched

augmentation Enrich the non/pre-terminal labels in the resulting treebank with semantic tags

syntactic+semantic parsing Train SLM as an L-matched constrained parser: boundaries and tags of the semantic constituents are matched

☞ Test:

- **syntactic+semantic parsing** of test sentences; retrieve the semantic parse by taking the semantic projection of the most likely parse:

$$S = SEM(\arg\max_{T_i} P(T_i, W))$$

Experiments

MiPad data (personal information management)

- training set: 2,239 sentences (27,119 words) and 5,431 slots
- test set: 1,101 sentences (8,652 words) and 1,698 slots
- vocabularies: WORD: 1,035wds, closed over test data; FRAME: 3; SLOT: 79;

Training Iteration		Error Rate (%)			
		Training		Test	
Stage 2	Stage 4	Slot	Frame	Slot	Frame
Baseline		43.41	7.20	57.36	14.90
0, MiPad/NLPwin	0	9.78	1.65	37.87	21.62
1, UPenn Trbnk	0	8.44	2.10	36.93	16.08
1, UPenn Trbnk	1	7.82	1.70	36.98	16.80
1, UPenn Trbnk	2	7.69	1.50	36.98	16.80

- baseline is a semantic grammar developed manually that makes no use of syntactic information
- initialize the syntactic SLM from in-domain MiPad treebank (NLPwin) and out-of-domain Wall Street Journal treebank (UPenn)
- 3 iterations of N-best EM parameter reestimation algorithm

Conclusions

- ✔ Presented a data driven approach to information extraction that outperforms a manually written semantic grammar
- ✔ Coupling of syntactic and semantic information improves information extraction accuracy, as shown previously by Miller et al., NAACL 2000

Future Work

- ✘ Use a statistical modeling technique that makes better use of limited amounts of training data and rich conditioning information — maximum entropy
- ✘ Aim at information extraction from speech: treat the word sequence as a hidden variable, thus finding the most likely semantic parse given a speech utterance