ERG Tokenization and Lexical Categorization

A sequence labeling approach

Murhaf Fares

University of Oslo

June 17, 2013

Outline

Terminology

- English Resource Grammar (ERG)
- Tokenization
- Lexical categorization
- Syntactic analysis

Terminology

- English Resource Grammar (ERG)
- Tokenization
- Lexical categorization
- Syntactic analysis

Terminology

- English Resource Grammar (ERG)
- Tokenization
- Lexical categorization
- Syntactic analysis

Terminology

- English Resource Grammar (ERG)
- Tokenization
- Lexical categorization
- Syntactic analysis

Overarching Goal

Improve ERG syntactic analysis through improving tokenization and lexical categorization

Why?

- Improve `ERG` syntactic analysis
- Through improving tokenization and lexical categorization

Why?

- Improve ERG syntactic analysis
- Through improving tokenization and lexical categorization

Some Research Questions

(1) Tokenization

(a) Apply sequence labeling techniques to approach tokenization

(b) CRF sequence labeling for PTB & ERG tokenization

(2) Lexical Categorization

(c) Features to model ERG lexical categories

(d) Accuracy vs. linguistic granularity in lexical categories

(3) Integration

(e) Parsing efficiency, coverage and accuracy when using our lexical categorization and tokenization models

(f) Linguistic granularity in lexical categories vs. parsing efficiency

Some Research Questions

(1) Tokenization
(a) Apply sequence labeling techniques to approach tokenization
(b) `CRF` sequence labeling for `PTB` & `ERG` tokenization

(2) Lexical Categorization
(c) Features to model `ERG` lexical categories
(d) Accuracy vs. linguistic granularity in lexical categories

(3) Integration
(e) Parsing efficiency, coverage and accuracy when using our lexical categorization and tokenization models
(f) Linguistic granularity in lexical categories vs. parsing efficiency

Some Research Questions

(1) Tokenization

(a) Apply sequence labeling techniques to approach tokenization

(b) `CRF` sequence labeling for `PTB` & `ERG` tokenization

(2) Lexical Categorization

(c) Features to model `ERG` lexical categories

(d) Accuracy vs. linguistic granularity in lexical categories

(3) Integration

(e) Parsing efficiency, coverage and accuracy when using our lexical categorization and tokenization models

(f) Linguistic granularity in lexical categories vs. parsing efficiency

ERG Parsing Pipeline

Raw text	String normalization PTB-like token boundary detection
Sequence of initial tokens	PoS tagging Normalization & Classification Lightweight NE Recognition Mapping to internal tokens
Lattice of internal tokens	Morphological segmentation Lexical instantiation
Lattice of lexical entries	Lexical parsing Lexical filtering
Lattice of lexical items	Syntactic parsing

ERG Parsing Pipeline

Raw text	String normalization **PTB-like token boundary detection**
Sequence of initial tokens	PoS tagging Normalization & Classification Lightweight NE Recognition **Mapping to internal tokens**
Lattice of internal tokens	Morphological segmentation Lexical instantiation
Lattice of lexical entries	Lexical parsing **Lexical filtering**
Lattice of lexical items	Syntactic parsing

A Sequence Labeling Approach

- Labeling (Classification)
- Sequence Labeling
- Conditional Random Fields (CRF)
  - Discriminative model
  - Proved powerful
  - No in-depth investigation of CRF for ERG lexical categorization

A Sequence Labeling Approach

- Labeling (Classification)
- Sequence Labeling
- Conditional Random Fields (CRF)
  - Discriminative model
  - Proved powerful
  - No in-depth investigation of CRF for ERG lexical categorization

A Sequence Labeling Approach

- Labeling (Classification)
- Sequence Labeling
- Conditional Random Fields (`CRF`)
    - Discriminative model
    - Proved powerful
    - No in-depth investigation of `CRF` for `ERG` lexical categorization

Tokenization

Definition

- Breaking up "natural language text ... into distinct **meaningful units** (or tokens)" *(Kaplan 2005)*
- Punctuation ambiguity
  - Periods
    - The luxury auto maker last year sold 1,214 cars in the U.S.
  - Parentheses and commas
    - `'Ca(2+)'` `'390,926'`

Definition

- Breaking up "natural language text ... into distinct **meaningful units** (or tokens)" *(Kaplan 2005)*
- Punctuation ambiguity
  - Periods
    - The luxury auto maker last year sold 1,214 cars in the U.S.
  - Parentheses and commas
    - `'Ca(2+)'` `'390,926'`

Definition

- Breaking up "natural language text ... into distinct **meaningful units** (or tokens)" *(Kaplan 2005)*
- Punctuation ambiguity
  - Periods
    - The luxury auto maker last year sold 1,214 cars in the U.S.
  - Parentheses and commas
    - '`Ca(2+)`' '`390,926`'

Two Tokenization Schemes, Two Experimental Setups

1. Penn Treebank `PTB`
2. English Resource Grammar `ERG`

Two Tokenization Schemes

<table>
<tr><td></td><td colspan="8">Sun-filled Mountain View didn’t collapse.</td></tr>
<tr><td>PTB</td><td colspan="2">Sun-filled</td><td>Mountain</td><td>View</td><td>did</td><td>n’t</td><td>collapse</td><td>.</td></tr>
<tr><td>ERG</td><td>Sun-</td><td>filled</td><td colspan="2">Mountain View</td><td colspan="2">didn’t</td><td colspan="2">collapse.</td></tr>
</table>

Two Tokenization Schemes

<table>
<tr><td></td><td colspan="8">Sun-filled Mountain View didn’t collapse.</td></tr>
<tr><td>PTB</td><td colspan="2">Sun-filled</td><td>Mountain</td><td>View</td><td>did</td><td>n’t</td><td>collapse</td><td>.</td></tr>
<tr><td>ERG</td><td>Sun-</td><td>filled</td><td colspan="2">Mountain View</td><td colspan="2">didn’t</td><td colspan="2">collapse.</td></tr>
</table>

Tokenization as a Sequence Labeling Problem

- **Target tokenization scheme** Such as `PTB` and `ERG`
- **Basic processing unit** The smallest unit that can make up a single token
- **Tokenization labels** The set of classification labels
- **Machine learning models and features** Such as CRFs and HMMs
- **Data split** The train-development-test data split

Basic Processing Unit

- Character-based

Basic Processing Unit

- Character-based
- Character classes

Character Class	**Description**
alpha	Alphabetical characters
num	Numerical characters
SQ	Single quote
OQ	Open quote

Basic Processing Unit

PC shipments total some $38.3 billion world-wide.

alphaC	*alpha*	*alpha*	*alpha*	*dollar*	*num*	*dot*	*num*	*alpha*	*alpha*	*hyphen*	*alpha*	*dot*
PC	*shipments*	*total*	*some*	$	38	.	3	*billion*	*world*	-	*wide*	.
1	2	3	4	5	6	7	8	9	10	11	12	13

- Token: one or more sub-tokens
- Candidate token boundary between each pair of sub-tokens

Basic Processing Unit

PC shipments total some $38.3 billion world-wide.

$$\underbrace{\overbrace{\textit{PC}}^{\textit{alphaC}}}_{1}\ \underbrace{\overbrace{\textit{shipments}}^{\textit{alpha}}}_{2}\ \underbrace{\overbrace{\textit{total}}^{\textit{alpha}}}_{3}\ \underbrace{\overbrace{\textit{some}}^{\textit{alpha}}}_{4}\ \underbrace{\overbrace{\$}^{\textit{dollar}}}_{5}\ \underbrace{\overbrace{38}^{\textit{num}}}_{6}\ \underbrace{\overbrace{.}^{\textit{dot}}}_{7}\ \underbrace{\overbrace{3}^{\textit{num}}}_{8}\ \underbrace{\overbrace{\textit{billion}}^{\textit{alpha}}}_{9}\ \underbrace{\overbrace{\textit{world}}^{\textit{alpha}}}_{10}\ \underbrace{\overbrace{\text{-}}^{\textit{hyphen}}}_{11}\ \underbrace{\overbrace{\textit{wide}}^{\textit{alpha}}}_{12}\ \underbrace{\overbrace{.}^{\textit{dot}}}_{13}$$

- Token: one or more sub-tokens
- Candidate token boundary between each pair of sub-tokens

PTB-Style — Experimental Setup

- **Target tokenization scheme** PTB
- **Basic processing unit** Sub-tokens
- **Tokenization labels** Binary (SPLIT, NONSPLIT)
- **Machine learning models and features** CRFs
- **Data** PTB WSJ

  PoS tagging 'standard' split (0–18, 19–21, 22–24)

PTB-Style — Features

30 features exploiting lexical and orthographic information

Feature	**Feature**	**Feature**
W_i	W_i & W_{i-1} & W_{i-2} & W_{i-3}	W_{i+1} & CC_{i+1}‡
W_{i+1}‡	W_i & W_{i+1} & W_{i+2} & W_{i+3}	FC_i
W_{i+2}‡	$Space_i$†	LC_i
W_{i+3}‡	W_i & $Space_i$	FC_i & FC_{i+1}
W_{i-1}‡	$Space_i$ & $Space_{i+1}$†	FC_i & FC_{i-1}
W_{i-2}‡	$Space_i$ & $Space_{i-1}$†	LC_{i-1} & FC_i
W_{i-3}‡	CC_i‡	LC_i & FC_{i+1}

PTB-Style — Evaluation

- Performance measured on sentence level
- REPP *(Dridan and Oepen 2012)*

PTB-Style — PTB Results

	REPP	**PTB model**
Accuracy	98.60%	99.07%

Tokenization accuracy on PTB WSJ sections 22–24

- 45% of our PTB model's errors are due to tokenization inconsistencies
  - The 'U.S.' idiosyncrasy: 30%
  - Inconsistencies in splitting hyphens ⟨trade, -, ethnic⟩: 4%
  - Splitting periods from acronyms: 11%

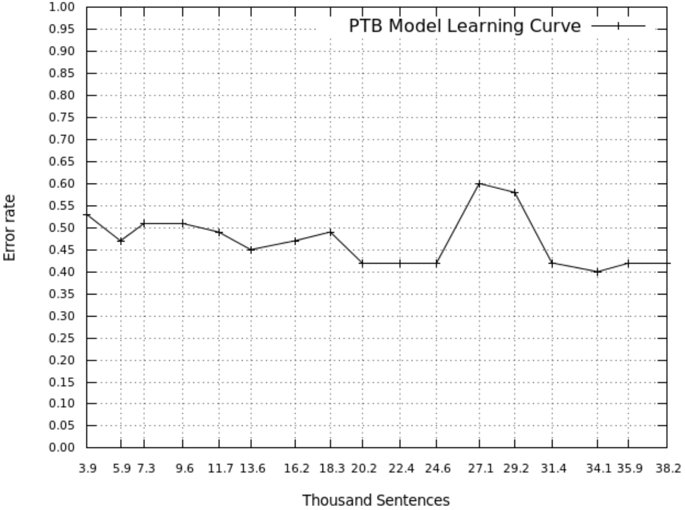
PTB-Style — PTB Results

	REPP	**PTB model**
Accuracy	98.60%	99.07%

Tokenization accuracy on PTB WSJ sections 22–24

- 45% of our PTB model's errors are due to tokenization inconsistencies
  - The 'U.S.' idiosyncrasy: 30%
  - Inconsistencies in splitting hyphens $\langle$trade, -, ethnic$\rangle$: 4%
  - Splitting periods from acronyms: 11%

PTB-Style — Learning Curve



PTB-Style — Genre & Domain Variation

- Brown & GENIA follow the PTB tokenization scheme
- Tested our PTB model and REPP on Brown and GENIA

- Both are resilient to genre variation
- On GENIA, REPP outperforms our PTB model
- With only 1000 sentences in-domain our PTB-adapted model substantially outperforms REPP

Machine Learning for High-Quality Tokenization — Replicating Variable Tokenization Schemes. Fares et al. 2013

PTB-Style — Genre & Domain Variation

- Brown & GENIA follow the PTB tokenization scheme
- Tested our PTB model and REPP on Brown and GENIA

- Both are resilient to genre variation
- On GENIA, REPP outperforms our PTB model
- With only 1000 sentences in-domain our PTB-adapted model substantially outperforms REPP

Machine Learning for High-Quality Tokenization — Replicating Variable Tokenization Schemes. Fares et al. 2013

ERG-Style — Terminology

- Initial tokens
- Lexical tokens
    - ⟨ad, hoc⟩
    - ⟨New, Year's, Eve⟩
    - ⟨as, such⟩
    - ⟨e-, mail⟩
- 10% of ERG 1212 lexicon (38,500 lemmata) are multi-word lexical entries

ERG-Style — Terminology

- Initial tokens
- Lexical tokens
    - $\langle$ad, hoc$\rangle$
    - $\langle$New, Year's, Eve$\rangle$
    - $\langle$as, such$\rangle$
    - $\langle$e-, mail$\rangle$
- 10% of ERG 1212 lexicon (38,500 lemmata) are multi-word lexical entries

ERG-Style — Experimental Setup

- **Target tokenization scheme** ERG
- **Basic processing unit** Initial tokens
- **Tokenization labels** Binary (SPLIT, NONSPLIT)
- **Machine learning models and features** CRFs & PTB model features +2
- **Data** DeepBank

ERG-Style — Results

N	Accuracy
1	94.69%
2	99.15%
3	99.57%
4	99.64%
5	99.85%

n-best ERG tokenization on DeepBank 21

- Hyphenated multi-word lexical units ‘south-west’
- Ambiguous multi-word lexical units ‘as well as’

ERG-Style — Results

N	Accuracy
1	94.69%
2	99.15%
3	99.57%
4	99.64%
5	99.85%

n-best ERG tokenization on DeepBank 21

- Hyphenated multi-word lexical units 'south-west'
- Ambiguous multi-word lexical units 'as well as'

ERG-Style — Results

N	Accuracy
1	94.69%
2	99.15%
3	99.57%
4	99.64%
5	99.85%

n-best ERG tokenization on DeepBank 21

- Hyphenated multi-word lexical units ‘south-west’
- Ambiguous multi-word lexical units ‘as well as’

Reflections

- PTB
  - Our sequence labeling approach outperforms state-of-the-art rule-based systems
  - Domain-adaptable models can achieve very high accuracies
- ERG
  - How good? To be decided later

Reflections

- PTB
  - Our sequence labeling approach outperforms state-of-the-art rule-based systems
  - Domain-adaptable models can achieve very high accuracies
- ERG
  - How good? To be decided later

Reflections

- PTB
  - Our sequence labeling approach outperforms state-of-the-art rule-based systems
  - Domain-adaptable models can achieve very high accuracies
- ERG
  - How good? To be decided later

Lexical Categorization

Background

- Assigning lexical categories to words
- Lexical categories: PoS tags or supertags (linguistically rich PoS tags)

PoS tags vs. 'Supertags'

- Cray Computer will be a concept stock, he said.
- Cray$_{\texttt{NNP}}$ Computer$_{\texttt{NNP}}$ will$_{\texttt{MD}}$ be$_{\texttt{VB}}$ a$_{\texttt{DT}}$ concept$_{\texttt{NN}}$ stock$_{\texttt{NN}}$,$_{\texttt{,}}$ he$_{\texttt{PRP}}$ said$_{\texttt{VBD}}$.$_{\texttt{.}}$
- Cray$_{\texttt{n_-_pn_le}}$ Computer$_{\texttt{n_-_pn_le}}$ will$_{\texttt{v_vp_will-p_le}}$ be$_{\texttt{v_np_be_le}}$ a$_{\texttt{d_-_sg-nmd_le}}$ concept$_{\texttt{n_-_c_le}}$ stock,$_{\texttt{n\_-\_mc\_le}}$ he$_{\texttt{n_-_pr-he_le}}$ said.$_{\texttt{v\_pp*-cp\_fin-imp\_le}}$

PoS tags vs. 'Supertags'

- Cray Computer will be a concept stock, he said.
- Cray$_{\texttt{NNP}}$ Computer$_{\texttt{NNP}}$ will$_{\texttt{MD}}$ be$_{\texttt{VB}}$ a$_{\texttt{DT}}$ concept$_{\texttt{NN}}$ stock$_{\texttt{NN}}$,$_{\texttt{,}}$ he$_{\texttt{PRP}}$ said$_{\texttt{VBD}}$.$_{\texttt{.}}$
- Cray$_{\texttt{n_-_pn_le}}$ Computer$_{\texttt{n_-_pn_le}}$ will$_{\texttt{v_vp_will-p_le}}$ be$_{\texttt{v_np_be_le}}$ a$_{\texttt{d_-_sg-nmd_le}}$ concept$_{\texttt{n_-_c_le}}$ stock,$_{\texttt{n\_-\_mc\_le}}$ he$_{\texttt{n_-_pr-he_le}}$ said.$_{\texttt{v\_pp*-cp\_fin-imp\_le}}$

'Our' ERG Lexical Categories

- Lexical type

  e.g. `v_pp_e_le`

  ⟨syntactic-cat⟩_⟨subcategorization⟩_⟨description⟩_le
- Major syntactic categories
- Relation between linguistic granularity and accuracy
- Scalability of CRF to large-scale tagging tasks
- Impact of linguistic granularity on syntactic parsing

'Our' ERG Lexical Categories

- Lexical type

  e.g. v_pp_e_le

  ⟨syntactic-cat⟩_⟨subcategorization⟩_⟨description⟩_le
- Major syntactic categories

- Relation between linguistic granularity and accuracy
- Scalability of CRF to large-scale tagging tasks
- Impact of linguistic granularity on syntactic parsing

'Our' ERG Lexical Categories

- Lexical type

  e.g. `v_pp_e_le`

  `⟨syntactic-cat⟩_⟨subcategorization⟩_⟨description⟩_le`
- Major syntactic categories
- Relation between linguistic granularity and accuracy
- Scalability of CRF to large-scale tagging tasks
- Impact of linguistic granularity on syntactic parsing

'Our' ERG Lexical Categories

- Lexical type

  e.g. `v_pp_e_le`

  `⟨syntactic-cat⟩_⟨subcategorization⟩_⟨description⟩_le`
- Major syntactic categories

- Relation between linguistic granularity and accuracy
- Scalability of CRF to large-scale tagging tasks
- Impact of linguistic granularity on syntactic parsing

'Our' ERG Lexical Categories

- Lexical type

  e.g. v_pp_e_le

  ⟨syntactic-cat⟩_⟨subcategorization⟩_⟨description⟩_le
- Major syntactic categories

- Relation between linguistic granularity and accuracy
- Scalability of CRF to large-scale tagging tasks
- Impact of linguistic granularity on syntactic parsing

Experimental Setup

	Dridan (2009)	Ytrestøl (2012)	Our experiments
Grammar	ERG 2009	ERG 2011	ERG 2012
Observations	Initial tokens	Lexical tokens	Lexical tokens
Lexical categories	letype et al.	letype	letype & MSC
Learning model	HMM & MaxEnt	MaxEnt & SVM	CRFs
Data set	Redwoods 2009	Redwoods 2011 WikiWoods	DeepBank
Train set (# tokens)	157,920	141,893,437	656,507

2 Types of Lexical Categories, 3 Experimental Setups

1. Lexical types (letype)
2. Major syntactic categories (MSC)
3. Specified lexical types (specified letype)

1. Lexical Types — Feature Ablation Study

Lexical	**M**orphosyntactic	**M**orphological	**O**rthographic
W_i	T_i	5-prefix$_i$	Cap$_i$ & W_i
W_{i-1}	W_i & T_i	5-suffix$_i$	Cap$_i$ & Cap $_{i-1}$
W_{i+1}	T_i & T_{i+1}	4-prefix$_i$	Hyph$_i$
W_i & W_{i-1} & W_{i-2}	T_i & T_{i-1}	4-suffix$_i$	
W_i & W_{i+1} & W_{i+2}	T_i & T_{i+2}	3-prefix$_i$	
	T_i & T_{i-2}	3-suffix$_i$	
	T_i & T_{i+3}	2-prefix$_i$	
	T_i & T_{i-3}	2-suffix$_i$	
	T_i & T_{i+1} & T_{i-1}	1-prefix$_i$	
		1-suffix$_i$	

Candidate features to learn ERG lexical types

1. Lexical Types — Feature Ablation Study

Model	**Accuracy**	**Features size** GB	**Training time** hours
L	90.37%	6.83	15.24^{γ}
MS	90.57%	0.68	15.55^{γ}
MS+O	90.73%	0.92	16.77^{α}
L+O	91.35%	7.06	18.46^{α}
MS+M	91.37%	1.17	15.59^{α}
L+M	92.09%	7.31	17.64^{γ}
L+MS	92.52%	7.52	20.14^{α}
L+M+O	92.33%	7.55	17.45^{γ}
L+MS+O	92.70%	7.75	17.11^{γ}
L+MS+M	93.48%	8.00	16.58^{γ}
L+MS+M+O	**93.54%**	8.24	49.08^{β}

Features ablation experiments on `DeepBank` 20 — α=8, β=4, γ=10 threads

1. Lexical Types — Results & Error Analysis

N	Accuracy
1	92.84%
2	94.21%
3	95.15%
4	95.64%
5	96.12%

L+MS+M+O on DeepBank 21

- 18% unseen words
- Manual assessment of 5%
  - PTB PoS tag errors 8%
  - Inconsistency errors 9%
  - Classification errors 83%

1. Lexical Types — Results & Error Analysis

N	Accuracy
1	92.84%
2	94.21%
3	95.15%
4	95.64%
5	96.12%

L+MS+M+O on DeepBank 21

- 18% unseen words
- Manual assessment of 5%
  - PTB PoS tag errors 8%
  - Inconsistency errors 9%
  - Classification errors 83%

1. Lexical Types — Error Analysis

- PTB PoS tag errors
  - Consumers may want to move their telephones a little closer to the TV **set**$_{\mathtt{VBD}}$.
    Model: `v_np*_le`. Gold: `n_-_c_le`
- Inconsistency errors
  - . . . viewers of several NBC **daytime**$_{\mathtt{JJ}}$ consumer segments . . .
    Model: `aj_-_i_le`. Gold: `n_-_c_le`
- Classification errors
  - "**The** Well-Tempered Clavier."
    Model: `d_-_the_le`. Gold: `n_-_pn_le`

1. Lexical Types — Error Analysis

- PTB PoS tag errors
  - Consumers may want to move their telephones a little closer to the TV **set**$_{\mathtt{VBD}}$.
    Model: `v_np*_le`. Gold: `n_-_c_le`
- Inconsistency errors
  - ... viewers of several NBC **daytime**$_{\mathtt{JJ}}$ consumer segments ...
    Model: `aj_-_i_le`. Gold: `n_-_c_le`
- Classification errors
  - "**The** Well-Tempered Clavier."
    Model: `d_-_the_le`. Gold: `n_-_pn_le`

1. Lexical Types — Error Analysis

- PTB PoS tag errors
  - Consumers may want to move their telephones a little closer to the TV **set**$_{\texttt{VBD}}$.
    Model: `v_np*_le`. Gold: `n_-_c_le`
- Inconsistency errors
  - ... viewers of several NBC **daytime**$_{\texttt{JJ}}$ consumer segments ...
    Model: `aj_-_i_le`. Gold: `n_-_c_le`
- Classification errors
  - "**The** Well-Tempered Clavier."
    Model: `d_-_the_le`. Gold: `n_-_pn_le`

2. Major Syntactic Categories

- Lexical categories: major syntactic categories 11
- Learning models & features: CRFs & letype model features

2. Major Syntactic Categories

- Lexical categories: major syntactic categories 11
- Learning models & features: CRFs & letype model features

N	**Accuracy**
1	98.01%
2	98.97%
3	99.36%
4	99.46%
5	99.57%

N-best list results for MSC tagging on `DeepBank` section 21

3. Specified Lexical Types

- Dividing the lexical types into 11 sub-sets
- Cray$_{\texttt{n_-_pn_le}}$ Computer$_{\texttt{n_-_pn_le}}$ will$_{\texttt{v_vp_will-p_le}}$ be$_{\texttt{v_np_be_le}}$ a$_{\texttt{d_-_sg-nmd_le}}$ concept$_{\texttt{n_-_c_le}}$ stock,$_{\texttt{n\_-\_mc\_le}}$ he$_{\texttt{n_-_pr-he_le}}$ said.$_{\texttt{v\_pp*-cp\_fin-imp\_le}}$
- **n model:** Cray$_{\texttt{n_-_pn_le}}$ Computer$_{\texttt{n_-_pn_le}}$ will$_{\texttt{v}}$ be$_{\texttt{v}}$ a$_{\texttt{d}}$ concept$_{\texttt{n_-_c_le}}$ stock,$_{\texttt{n\_-\_mc\_le}}$ he$_{\texttt{n_-_pr-he_le}}$ said.$_{\texttt{v}}$
- **v model** Cray$_{\texttt{n}}$ Computer$_{\texttt{n}}$ will$_{\texttt{v_vp_will-p_le}}$ be$_{\texttt{v_np_be_le}}$ a$_{\texttt{d}}$ concept$_{\texttt{n}}$ stock,$_{\texttt{n}}$ he$_{\texttt{n}}$ said.$_{\texttt{v\_pp*-cp\_fin-imp\_le}}$

3. Specified Lexical Types

- Dividing the lexical types into 11 sub-sets
- Cray$_{\texttt{n_-_pn_le}}$ Computer$_{\texttt{n_-_pn_le}}$ will$_{\texttt{v_vp_will-p_le}}$ be$_{\texttt{v_np_be_le}}$ a$_{\texttt{d_-_sg-nmd_le}}$ concept$_{\texttt{n_-_c_le}}$ stock,$_{\texttt{n\_-\_mc\_le}}$ he$_{\texttt{n_-_pr-he_le}}$ said.$_{\texttt{v\_pp*-cp\_fin-imp\_le}}$
- **n model:** Cray$_{\texttt{n_-_pn_le}}$ Computer$_{\texttt{n_-_pn_le}}$ will$_{\texttt{v}}$ be$_{\texttt{v}}$ a$_{\texttt{d}}$ concept$_{\texttt{n_-_c_le}}$ stock,$_{\texttt{n\_-\_mc\_le}}$ he$_{\texttt{n_-_pr-he_le}}$ said.$_{\texttt{v}}$
- **v model** Cray$_{\texttt{n}}$ Computer$_{\texttt{n}}$ will$_{\texttt{v_vp_will-p_le}}$ be$_{\texttt{v_np_be_le}}$ a$_{\texttt{d}}$ concept$_{\texttt{n}}$ stock,$_{\texttt{n}}$ he$_{\texttt{n}}$ said.$_{\texttt{v\_pp*-cp\_fin-imp\_le}}$

3. Specified Lexical Types

- Dividing the lexical types into 11 sub-sets
- Cray$_{\texttt{n_-_pn_le}}$ Computer$_{\texttt{n_-_pn_le}}$ will$_{\texttt{v_vp_will-p_le}}$ be$_{\texttt{v_np_be_le}}$ a$_{\texttt{d_-_sg-nmd_le}}$ concept$_{\texttt{n_-_c_le}}$ stock,$_{\texttt{n\_-\_mc\_le}}$ he$_{\texttt{n_-_pr-he_le}}$ said.$_{\texttt{v\_pp*-cp\_fin-imp\_le}}$
- **n model:** Cray$_{\texttt{n_-_pn_le}}$ Computer$_{\texttt{n_-_pn_le}}$ will$_{\texttt{v}}$ be$_{\texttt{v}}$ a$_{\texttt{d}}$ concept$_{\texttt{n_-_c_le}}$ stock,$_{\texttt{n\_-\_mc\_le}}$ he$_{\texttt{n_-_pr-he_le}}$ said.$_{\texttt{v}}$
- **v model** Cray$_{\texttt{n}}$ Computer$_{\texttt{n}}$ will$_{\texttt{v_vp_will-p_le}}$ be$_{\texttt{v_np_be_le}}$ a$_{\texttt{d}}$ concept$_{\texttt{n}}$ stock,$_{\texttt{n}}$ he$_{\texttt{n}}$ said.$_{\texttt{v\_pp*-cp\_fin-imp\_le}}$

3. Specified Lexical Types

- Dividing the lexical types into 11 sub-sets
- Cray$_{\texttt{n_-_pn_le}}$ Computer$_{\texttt{n_-_pn_le}}$ will$_{\texttt{v_vp_will-p_le}}$ be$_{\texttt{v_np_be_le}}$ a$_{\texttt{d_-_sg-nmd_le}}$ concept$_{\texttt{n_-_c_le}}$ stock,$_{\texttt{n\_-\_mc\_le}}$ he$_{\texttt{n_-_pr-he_le}}$ said.$_{\texttt{v\_pp*-cp\_fin-imp\_le}}$

- **n model:** Cray$_{\texttt{n_-_pn_le}}$ Computer$_{\texttt{n_-_pn_le}}$ will$_{\texttt{v}}$ be$_{\texttt{v}}$ a$_{\texttt{d}}$ concept$_{\texttt{n_-_c_le}}$ stock,$_{\texttt{n\_-\_mc\_le}}$ he$_{\texttt{n_-_pr-he_le}}$ said.$_{\texttt{v}}$

- **v model** Cray$_{\texttt{n}}$ Computer$_{\texttt{n}}$ will$_{\texttt{v_vp_will-p_le}}$ be$_{\texttt{v_np_be_le}}$ a$_{\texttt{d}}$ concept$_{\texttt{n}}$ stock,$_{\texttt{n}}$ he$_{\texttt{n}}$ said.$_{\texttt{v\_pp*-cp\_fin-imp\_le}}$

3. 11 Specified Lexical Types Models

Specified letype	Per token accuracy	Training time
x-letype	98.34%	9 mins
cm-letype	98.32%	13 mins
d-letype	98.33%	48 mins
c-letype	98.27%	1.75 hours
pt-letype	98.26%	6 mins
pp-letype	98.27%	17 mins
av-letype	98.15%	2.58 hours
aj-letype	98.21%	**2.71** hours
p-letype	97.06%	1.60 hours
n-letype	96.87%	1.88 hours
v-letype	96.58%	2.20 hours

3. Combining the Outputs

Model	**Per token accuracy**	**Decoding time**
Specified letype	92.29%	69s
letype	92.84%	240s

Combining the specified lexical type outputs — `DeepBank` section 21

Integration

Introduction

- Hard constraints: restrict the parser search space
- Token boundaries (94.69%)
- Lexical categories: major syntactic categories (98.01%) & lexical types (92.84%)

Introduction

- Hard constraints: restrict the parser search space
- Token boundaries (94.69%)
- Lexical categories: major syntactic categories (98.01%) & lexical types (92.84%)

Introduction

- Hard constraints: restrict the parser search space
- Token boundaries (94.69%)
- Lexical categories: major syntactic categories (98.01%) & lexical types (92.84%)

Evaluation

- Coverage
- Efficiency
- Accuracy: exact matches & PARSEVAL
- DeepBank 21

Evaluation

- Coverage
- Efficiency
- Accuracy: exact matches & PARSEVAL
- DeepBank 21

Evaluation

- Coverage
- Efficiency
- Accuracy: exact matches & PARSEVAL
- DeepBank 21

Evaluation

- Coverage
- Efficiency
- Accuracy: exact matches & PARSEVAL
- `DeepBank` 21

Token Boundaries Integration

	Efficiency	Coverage	Accuracy	
	Seconds	%	Exact matches	PARSEVAL
All	20.61	97.3	339	87.2
Gold TB	19.00	97.6	345	87.8
TB	18.81	97.3	339	87.5

Parsing evaluation using ambiguous token boundaries, gold-standard token boundaries and automatically assigned token boundaries

- Reduction(s) of parsing time by:
  - 7.8% gold-standard token boundaries
  - 8.7% automatically assigned token boundaries

Lexical Categories Integration

- Single tag
- Multiple tags
- Selective tags

n-best Major Syntactic Categories

	Efficiency	Coverage	Accuracy	
	Seconds	%	Exact matches	PARSEVAL
Unrestricted	19.00	97.6	345	87.8
1-best	4.00	91.6	305	84.0
2-best	4.67	95.5	333	86.3
5-best	6.59	98.3	352	87.3

Parsing efficiency, coverage and accuracy with *n*-best major syntactic categories

- Reduction(s) of parsing time by:
  - 5-best: 65%

Selective Major Syntactic Categories

	Efficiency	Coverage	Accuracy	
	Seconds	%	Exact matches	PARSEVAL
Unrestricted	19.00	97.6	345	87.8
β=0.80	4.87	96.4	340	86.9
β=0.85	5.11	97.0	340	87.0
β=0.90	5.39	97.6	349	87.4
β=0.95	6.34	98.6	351	87.8

Parsing efficiency, coverage and accuracy with selective major syntactic categories

- Reduction(s) of parsing time by:
  - β=0.95: 66%

Selective Lexical Types

	Efficiency	Coverage	Accuracy	
	Seconds	%	Exact matches	PARSEVAL
Unrestricted	19.00	97.6	345	87.8
β=0.80	1.35	89.3	366	84.2
β=0.85	1.54	92.2	374	85.7
β=0.90	1.97	94.7	383	86.8
β=0.95	3.01	97.8	395	88.3

Parsing efficiency, coverage and accuracy with selective lexical types

- Reduction(s) of parsing time by:
  - β=0.95: 84%

Conclusion

Answers for Research Questions

(a) Apply sequence labeling techniques to approach tokenization
(b) `CRF` sequence labeling for `PTB` & `ERG` tokenization

(c) Features to model `ERG` lexical categories
(d) Accuracy vs. linguistic granularity in lexical categories

(e) Parsing efficiency, coverage and accuracy when using our lexical categorization and tokenization models
(f) Linguistic granularity in lexical categories vs. parsing efficiency

Answers for Research Questions

(a) Apply sequence labeling techniques to approach tokenization

(b) CRF sequence labeling for PTB & ERG tokenization

(c) Features to model ERG lexical categories

(d) Accuracy vs. linguistic granularity in lexical categories

(e) Parsing efficiency, coverage and accuracy when using our lexical categorization and tokenization models

(f) Linguistic granularity in lexical categories vs. parsing efficiency

Answers for Research Questions

(a) Apply sequence labeling techniques to approach tokenization
(b) `CRF` sequence labeling for `PTB` & `ERG` tokenization

(c) Features to model `ERG` lexical categories
(d) Accuracy vs. linguistic granularity in lexical categories

(e) Parsing efficiency, coverage and accuracy when using our lexical categorization and tokenization models
(f) Linguistic granularity in lexical categories vs. parsing efficiency

Thanks!

Thanks!

End-to-end Integration

	Efficiency	Coverage	Accuracy	
	Seconds	%	Exact matches	PARSEVAL
All	20.61	97.3	339	87.2
β=0.95	8.06	98.6	348	87.6

- Reduction(s) of parsing time by:
  - β=0.95: 52%