Home Page

Title Page

Contents

Page 1 of 48

Go Back

Full Screen

Close

Quit

State of the art NLParsing

Razvan Popescu
{popescu@di.unipi.it}
Computer Sciences Department, University of P*i*sa

25th May 2004

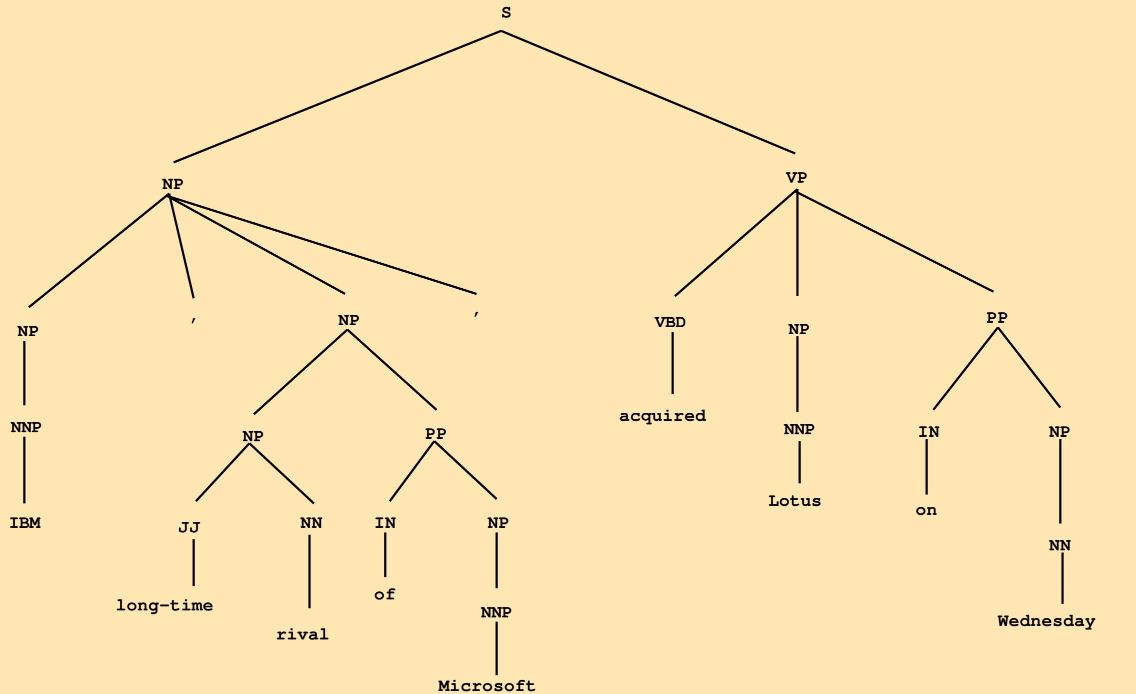
CONTENTS

PARSING

- Fundamental NLP problem
- Sentence $\rightsquigarrow$ **Parser** $\rightsquigarrow$ *ParseTree*
- State of the art NLParsing systems $\equiv$ `ML Probabilistic Parsing Techniques`
  - Training sets $\equiv \{(Sentence1,\ Tree1);\ (Sentence2,\ Tree2);\ \ldots;\ (SentenceN,\ TreeN)\} \Rightarrow$ Parameter estimates
  - Test sets $\equiv$ *model evaluation*

Home Page
Title Page
Contents
Page 4 of 48
Go Back
Full Screen
Close
Quit

PARSE TREE EXAMPLE



- POS
- ParseTree
- Applications ≡ Information Extraction/Retrieval, Machine Translation, Speech Recognition

NL PARSING PROBLEMS

- Ambiguity
  - POS ambiguity (e.g. `saw` (verb) vs. `saw` (noun))
  - PP attachment ambiguity
  - Coordination btw different words in a sentence
- WSJ statistics
  - average sentence length: 23 words
  - sentences over 30 words: 26%
  - sentences over 40 words: 7%

APPROACHING THE NL PARSING PROBLEM

- Standard Approaches (Rule-Based)
  - hand-crafted grammar + lexically specific info
  - selectional restrictions (e.g. `eat` & *+food*[a] and `apple` $\ni$ *+food*) $\leadsto$ disambiguation
  - problems with selectional restrictions
    - vocabulary and grammar size(e.g. $\geq$ 24,444 distinct words in 40,000 sentences of WSJ)
    - theoretical problems
  - (MUC-6, 1995) – none of the five best systems used full-parsing

[a] word feature

APPROACHING THE NL PARSING PROBLEM (cont'd)

- Machine-Learning Approaches (Statistical Methods)
  - `treebank`: {(sentence, parse-tree), ...}
  - PCFG systems ⇒ disappointing results
  - Other directions
    - increased structural sensitivity models
    - partially supervised training algs.
    - probabilistic versions of lexicalised grammars
    - history-based models
  - state of the art[a]: SPATTER parser
    - tested on WSJ
    - no hand-crafted grammar; treebank trained
    - 84.5/84.0% LP/LR – section 23 of the Penn WSJ treebank (non-lexicalised PCFGs ≈ 72% avg. LP/LR)
    - params conditioned on lexical information

---
[a] Magerman, 1995

APPROACHING THE NL PARSING PROBLEM (cont'd)

- Machine-Learning Approaches (Statistical Methods[a])
  - D. Magerman's "Statistical decision tree models for parsing" (1995)
  - M. Collins' "Three generative lexicalised models for statistical parsing" (1997)
  - BBN's "SIFT" system (1998, derived from Model 2 of Collins)
  - M. Collins' "Head-Driven Statistical Models for Natural Language Parsing" (1999)
  - E. Charniak's "Maximum entropy-inspired parser" (2000)
  - D. Chiang's "Stochastic TAG parser" (2000)
  - LCC's parser (2002)
  - D. Bikel's "Multilingual Statistical Parsing Engine" (2002)
  - ...

[a] 1995 – nowadays

A STATISTICAL MODEL FOR PARSING AND WORD-SENSE DISAMBIGUATION[a]

- first attempt (2000)
- performance metrics: 84.0/67.3% LR/LP vs. 78.6% inter-annotator agreement (gold standard)

[a] the core of Bikel's parser

MOTIVATION FROM EXAMPLES

1. [IBM bought [Lotus for $200 million]].
2. Sony **widened** its product line [$_{PP}$ with personal computers].
3. The **bank** issued a **check** for $100,000.
4. Apple is **expecting** [$_{NP}$ strong results].
5. IBM **expected** [$_{SBAR}$ each employee to wear a shirt and tie].

For 3.-5. syntactic context $\rightsquigarrow$ word meanings

MOTIVATION FROM PREVIOUS WORK (PARSING)

- previous parsers success
  - computing machinery development
  - treebanks (e.g. Penn Treebank)
  - ML techniques for NLP (POS tagging and PP attachment)
  - **probabilistically modeled lexicalisation of grammar formalisms**
  - all recent successful parsers use "bilexical dependencies"
    - attach probabilities to parser moves (Magerman 1997, Ratnaparkhi 1997)
    - lexicalised PCFG variety (Collins 1997, Charniak 1997)
    - involve "head-modifier relations"
    - **[lexical head] semantics reduces parsing ambiguity!**
  - WordNet + hypernym structure ⇒ soft-clusters

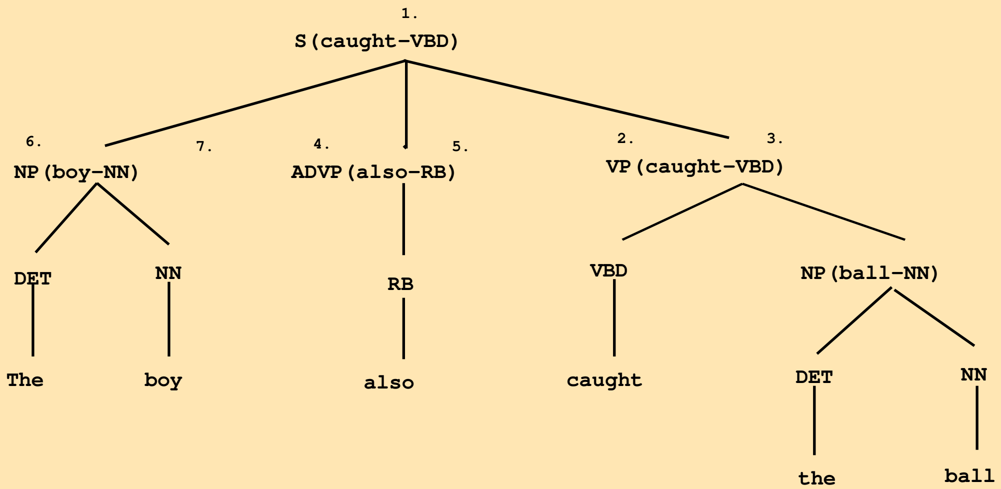
MOTIVATION FROM PREVIOUS WORK (WSD)

- syntactic context + dependency structures $\Rightarrow$ WSD
- unsupervised WSD + WordNet based similarity heuristic $\Rightarrow$ PP attachment (88.1%)
- head-driven bilexical dependencies + syntactic relations $\Rightarrow$ generalised WSD (Stetina, 1998)

Home Page
Title Page
Contents
Go Back
Full Screen
Close
Quit

THE MODEL

- core $\equiv$ BBN's SIFT



- generation order: head then L/R modifiers from the head outward[a]
- recursive process
- many words are generated high up in the tree

[a](e.g. 1.-7.)

THE MODEL (cont'd)

- formally
  - $P \rightarrow L_n L_{n-1} \ldots L_1 H R_1 \ldots R_{n-1} R_n$
  - P, H, L_i and R_i are lexicalised nonterminals of the form $X < w, t, f >$:
    - $X \equiv$ traditional CFG nonterminal
    - $< w, t, f > \equiv$ `word - POS - word-feature` (the head of X)
- $H \equiv$ head constituent of P
- $L_i / R_i \equiv$ left/right modifier constituents of P w.r.t. H

PROBABILITY STRUCTURE OF THE ORIGINAL MODEL

- p, l_i, r_i and $h \equiv$ unlexicalised nonterminals corresp. to P, L_i, R_i and H
- top-level generation probabilities[a]
  - Probability of generating p as root:

$$P(p|+TOP+), e.g. P(S|+TOP+)^{b} \tag{1}$$

  - Probability of generating a head node h with a parent p:

$$P(h|p), e.g. P(VP|S) \tag{2}$$

  - Probability of generating a left-modifier l_i:

$$P_L(l_i|l_{i-1}, p, h, w_h), e.g. P_L(NP|ADVP, S, VP, caught)^{c} \tag{3}$$

  - Probability of generating a right-modifier r_i:

$$P_R(r_i|r_{i-1}, p, h, w_h), e.g. P_R(NP|+BEGIN+, VP, VBD, caught)^{d} \tag{4}$$

---

[a] omitting the smoothing details of BBN's model
[b] $+TOP+ \equiv$ the hidden root of all parse trees
[c] when generating the NP for NP(boy-NN)
[d] when generating the NP for NP(ball-NN); $+BEGIN+ \equiv$ hidden nonterminal for determining the initial probability

PROBABILITY STRUCTURE OF THE ORIGINAL MODEL (cont'd)

- probabilities for generating lexical elements

  1. for the POS tag of the head of the entire sentence t_h:

  $$P(t_h|p) \tag{5}$$

  2. for the POS tags of modifier constituents t_{l_i} and t_{r_i}:

  $$P(t_{l_i}|l_i, t_h, w_h) \textit{ and } P(t_{r_i}|r_i, t_h, w_h) \tag{6}$$

  3. for the head word of the entire sentence w_h:

  $$P(w_h|t_h, p) \tag{7}$$

PROBABILITY STRUCTURE OF THE ORIGINAL MODEL (cont'd)

- probabilities for generating lexical elements

4. for head words of modifier constituents w_{l_i} and w_{r_i}:

$$P(w_{l_i}|t_{l_i}, l_i, t_h, w_h) \textit{ and } P(w_{r_i}|t_{r_i}, r_i, t_h, w_h) \tag{8}$$

5. for the word feature of the head of the entire sentence f_h:

$$P(f_h|w_h, t_h, p) \tag{9}$$

6. for the word features of the head words of modifier constituents f_{l_i} and f_{r_i}:

$$P(f_{l_i}|known(w_{l_i}), t_{l_i}, l_i, t_h, w_h) \textit{ and } P(f_{r_i}|known(w_{r_i}), t_{r_i}, r_i, t_h, w_h)^a \tag{10}$$

- probability of the entire parse tree:

$$P(parse\_tree) = \prod_{i \in I} P_i \tag{11}$$

, where $I \equiv$ the set of all elements of the $parse\_tree$

- training data $\Rightarrow$ maximum-likelihood estimates of the params

[a] $known(x) \Rightarrow true$ iff $observed(x) \geq 4$ in the training data

WORD-SENSE EXTENSIONS TO THE LEXICAL MODEL

- parser output $\equiv$ standard Treebank-style parse tree; (words + POS tags + WordNet synsets)
- **Q: synset is to be generated but WHEN?**
  - generation of $< w, t, f > \equiv$ three steps $\Rightarrow$ four possible computation points
  - soft clustering of synsets $\Rightarrow$ add specificity to ambiguous lexical items + cluster lexical items with similar meanings
  - noun + vb synsets $\Rightarrow$ *concept taxonomy* + hypernym $\Rightarrow$ partial ordering over WordNet lemmas
- **A: after generating the POS tag, before generating the word**

UPDATED PROBABILITY STRUCTURES OF THE MODEL

- the probabilities for generating
  1. the synset of the head of the entire sentence s_h:

$$P(s_h|t_h, p) \tag{12}$$

  2. the head word of the entire sentence w_h becomes[a]:

$$P(w_h|t_h, p) \tag{13}$$

  3. synsets of modifier constituents s_{m_i}[b]:

$$\begin{aligned} \hat{P}(s_{m_i}|t_{m_i}, m_i, w_h, s_h) = \\ \lambda_0 \hat{P}(s_{m_i}|t_{m_i}, m_i, w_h, s_h) \\ +\lambda_1 \hat{P}(s_{m_i}|t_{m_i}, m_i, s_h) \\ +\lambda_2 \hat{P}(s_{m_i}|t_{m_i}, m_i, @^1(s_h)) \\ + \ldots \\ +\lambda_{n+1} \hat{P}(s_{m_i}|t_{m_i}, m_i, @^n(s_h)) \\ +\lambda_{n+2} \hat{P}(s_{m_i}|t_{m_i}, m_i) \\ +\lambda_{n+3} \hat{P}(s_{m_i}|t_{m_i}) \end{aligned} \tag{14}$$

  where $@^i(s_h)$ is the i^{th} hypernym of s_h

[a] (7)++
[b] complete with smoothing components

UPDATED PROBABILITY STRUCTURES OF THE MODEL (cont'd)

- WordNet hypernym rels form DAG $\Rightarrow$ uniformly-weighted mean over the probabilities conditioning on each of the hypernyms:

$$\hat{P}(s_{m_i}|t_{m_i}, m_i, @^j(s_h)) = \frac{1}{n}\sum_{k=1}^{n}\hat{P}(s_{m_i}|t_{m_i}, m_i, @^j_k(s_h)) \tag{15}$$

when $@^j(s_h) = \{@^j_1(s_h), \ldots, @^j_n(s_h)\}$

UPDATED PROBABILITY STRUCTURES OF THE MODEL (cont'd)

- the probability for generating
  - head words of modifier constituents w_{m_i}[a] becomes[b]:

$$\begin{aligned}\hat{P}(w_{m_i}|s_{m_i}, t_{m_i}, m_i, w_h, s_h) = \\ \lambda_0 \hat{P}(w_{m_i}|s_{m_i}, t_{m_i}, m_i, w_h) \\ +\lambda_1 \hat{P}(w_{m_i}|s_{m_i}, t_{m_i}, m_i, s_h) \\ +\lambda_2 \hat{P}(w_{m_i}|s_{m_i}, t_{m_i}, m_i, @^1(s_h)) \\ + \dots \\ +\lambda_{n+1} \hat{P}(w_{m_i}|s_{m_i}, t_{m_i}, m_i, @^n(s_h)) \\ +\lambda_{n+2} \hat{P}(w_{m_i}|s_{m_i}, t_{m_i}, m_i) \\ +\lambda_{n+3} \hat{P}(w_{m_i}|s_{m_i}, t_{m_i}) \\ +\lambda_{n+4} \hat{P}(w_{m_i}|s_{m_i})\end{aligned} \tag{16}$$

where $@^i(s_h)$ is the i^{th} hypernym of s_h

[a] complete with smoothing components

[b] (8)++

UPDATED PROBABILITY STRUCTURES OF THE MODEL (cont'd)

- observations
  - train with $[_{VP}$ `strike the target]`; test with $[_{VP}$ `attack the target]` $\Rightarrow$ **OK** (`attack` $=$ **hypernym(**`strike`**)**)
  - **only** 2-4 back-off levels $\rightsquigarrow$ negligible difference in parsing performance

A NEW APPROACH, A NEW DATASET

- Penn Treebank $\not\supseteq$ word-sense annotated corpus
- meet **SemCor**
  + 455k word portion of the Brown Corpus
  + every noun, verb, adjective, adverb + WordNet synset
  – Brown Corpus Treebank I style annotation
  + part of Brown Corpus Treebank II style annotation
- {Treebank II annotated Brown} $\cap$ {SemCor} $\approx$ 220k words

Step 1 synchronising the 220k words

- hyphenates + word senses
  1. word sense of the head (e.g. `twelve-foot` $\rightsquigarrow$ `foot_1`)
  2. if no clear head then word sense of the hypernym (e.g. `U.S.-Soviet` $\rightsquigarrow$ `country_2`)
  3. if 1. & 2. fail, then split hyphenate in the Treebank II file
  4. if hyphenate $\in$ {`"non-XYZ"`,`"anti-XYZ"`}, then annotate with the word sense of `XYZ`

A NEW APPROACH, A NEW DATASET (cont'd)

Step 2 SemCor $\cup$ Treebank II Brown issues

- keep only the first synset
- collocations[a]: WordNet+, Treebank- $\Rightarrow$ reanalyze collocations as a seq. of separate words with the same synset

[a] patterns of words appearing together (e.g. "apple pie", "apple tree" – "apple" collocates with "pie" and "tree")

TRAINING AND DECODING

- the hypernym chain of the parent head used for the computation of back-off levels
- plug-'n'-play lexical model system
- top-down model, bottom-up parsing
- rank candidate parse trees
- (unextended parsing model) every possible tag t for a word $w \rightsquigarrow < w, t, f >$ (f is computed deterministically) $\Rightarrow$ 1st degree of ambiguity in decoding
- (WordNet extended model) every possible synset s for a word-tag pair $\rightsquigarrow < w, t, f, s >$
- forms of pruning during decoding
  - parse tree ranking score $\geq$ factor of e^{-k} of the top ranked parse
  - keep the n top-ranked subtrees
- "out-of-the-box" BBN ($k = -5$ and $n = 25$)
- Bikel's model ($k = -9$ and $n = 50$)

Home Page

Title Page

Contents

◀◀ ▶▶

◀ ▶

Go Back

Full Screen

Close

Quit

EXPERIMENTS AND RESULTS

- PARSING (1% of the 220K word corpus)

Test 1 the last 117 sentences (section "r")

- Disappointing results ← *"our initial test corpus was literally a joke"*[a]

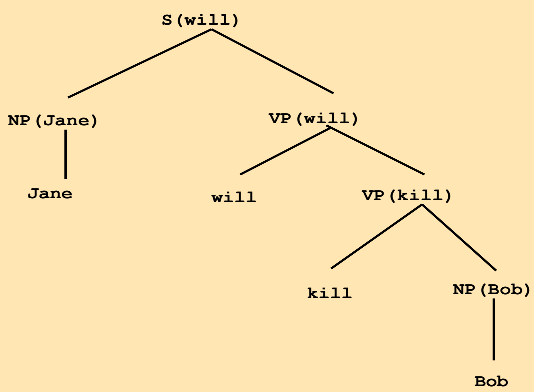
Test 2 sample every 100 sentences ⇒ 117 sentences

Model, test set	≤40 words				
	LR	LP	$\overline{\text{CB}}$	0CB	≤2CB
Norm, "r"*	69.7	72.6	2.93	31.9	55.0
WN-ext, "r"	**69.7**	**72.7**	**2.86**	**30.8**	**56.0**
Norm, bal	83.1	85.0	0.82	75.9	85.7
WN-ext, bal	**82.9**	**84.0**	**1.02**	**70.5**	**81.3**
	All sentences				
	LR	LP	$\overline{\text{CB}}$	0CB	≤2CB
Norm, "r"*	68.6	71.2	3.83	25.9	44.8
WN-ext, "r"	**69.7**	**71.5**	**3.77**	**25.0**	**45.7**
Norm, bal	82.0	84.4	1.00	73.5	83.8
WN-ext, bal	**80.5**	**82.2**	**1.43**	**68.4**	**78.6**

[a] humor writing section

EXPERIMENTS AND RESULTS (cont'd)

- PARSING OBSERVATIONS
  - roughly similar results
  - WN-ext $\rightsquigarrow$ intermediate ambiguity during decoding
  - trilexical/tertalexical dependencies $\rightsquigarrow$ synset advantages
  - `[`**bought** `company [for` **million**`]]` – no dependency
  - soft clustering the synsets $\rightsquigarrow$ offset the sparse data problem
  - head rules are tuned for syntax, not semantics:



EXPERIMENTS AND RESULTS (cont'd)

- WORD SENSE DISAMBIGUATION
  - results[a]:

	Recall	Precision
Noun	86.5%	70.9%
Verb	84.0%	59.5%
Adj	80.2%	70.4%
Adv	78.5%	75.8%
Total	**84.0%**	**67.3%**

[a] on the balanced test set

Home Page

Title Page

Contents

◀◀ ▶▶

◀ ▶

Go Back

Full Screen

Close

Quit

EXPERIMENTS AND RESULTS (cont'd)

- WSD OBSERVATIONS

Others distinguish homonyms

Bikel WordNet $\Rightarrow$ fine-grained distinctions

Others disambiguate a small set of homonyms

Bikel attacks generalised WSD

- SemCor's inter-annotator agreement is 78.6% overall and 70% for words with polysemy $\geq 8 \Rightarrow$ **precision upper bound**

Bikel considers **exact synset matches** only

Others paradoxically Stetina reported 79.4% overal accuracy (1998)

Others Stetina ranks with heuristics

Bikel ranks with maximum-likelihood probability estimates

Bikel 50-odd Treebank vs. 4 WordNet POS $\Rightarrow$ (output == synset, for a WordNet POS diff. from the gold file) $\equiv$ **recall error**

FUTURE WORK

- toward a state-of-the-art model (Collins' Model 2/3 based)
- experiment with radical model where nonterminals only have synsets as their heads and words are generated at the leaves
- add word-distance context to help WSD
- investigate unsupervised methods for WSD (e.g. Stetina's heuristics)

Bikel's "Multilingual Statistical Parsing Engine"

- limitations of previous parsers
  - fairly fixed probabilistic structure ⇒ re-coding
  - hard-coded English language features
  - hard-coded Penn Treebank features
  - designed for uniprocessor environment
- characteristics of Bikel's parser
  - head-driven, chart parsing engine
  - language/treebank portable
  - "plug-'n'-play" lexical probability structures
  - multiprocessor/multi-host support; multi-threaded sentence server ⇒ parallelism at the sentence level

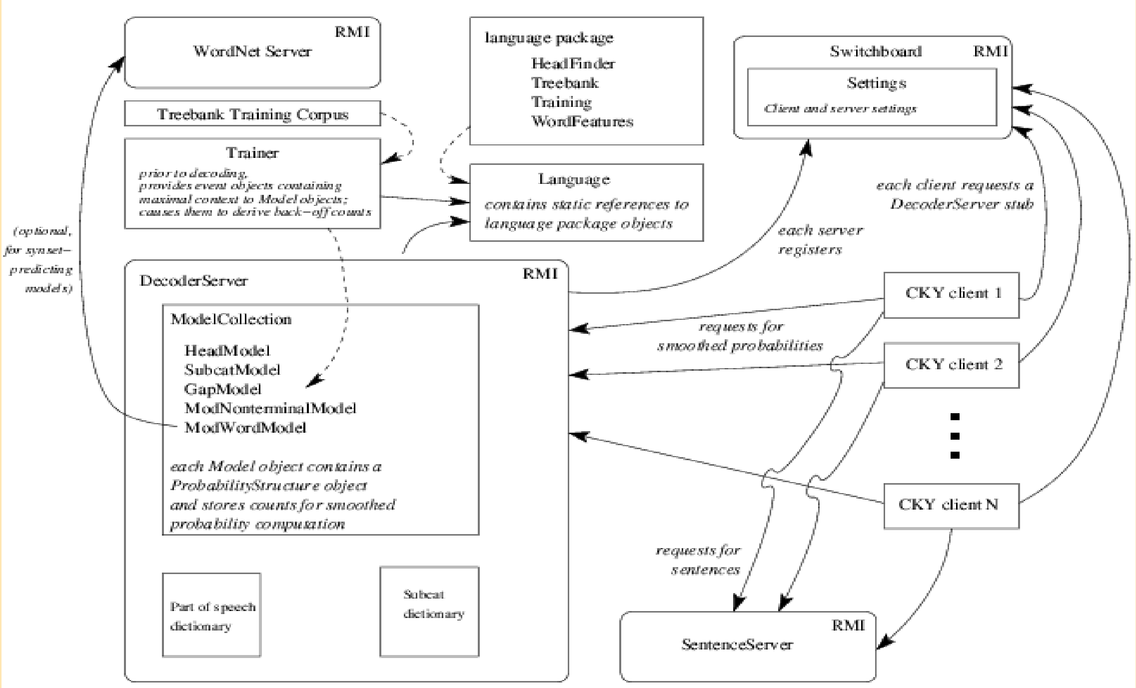
LANGUAGE INDEPENDENCE

- Testing
  - BBN's SIFT derived parser
  - Chiang's Stochastic TAG parser
- on English and Chinese
  - $\approx$ 100k words of WSJ text from the English Penn Treebank
  - $\approx$ 100k words of Xinhua text from the Chinese Treebank[a]
- resulted in
  - Chiang's Stochastic TAG $\Rightarrow$ 77%/78% LP/LR on Xinhua compared to 79%/80% on WSJ

[a]consisting of 4185 sentences

PARSER DESIGN

- emulates BBN's SIFT & Model 2,3 of Collins



Arrow ≡ functional relationship
Solid arrow ≡ the direction of request from a client to a server
Dashed arrow ≡ the flow of information

LANGUAGE PACKAGE

- java package
- required classes
  - `Treebank` – data and methods specific to a particular treebank
  - `Training` – —#— to preprocessing training trees
  - `HeadFinder` – I: text file with head rules specific to a language treebank; O: head-finding method
  - `WordFeatures` – mapping of lexical items from a language to an orthographic/morphological word feature vectors
- unicode I/O files
- does not include WordNet
- **GOAL: creation of a new language package in 1-2 weeks**

PROBABILITY STRUCTURE OBJECTS

- output element + [`ProbabilityStructure` object $\equiv$ data objects representing the future and history of all possible back-off levels]
- `TrainerEvent` object $\equiv$ history and future for a specified back-off level

PROBABILITY-LEVEL PARALLELISM

- **basic idea:** $(FILE, EXEC, HOST) \equiv (f_1, e_1, h_1), \ldots, (f_n, e_n, h_n)$**, where** $FILE = f_1 \cup f_2 \cup \ldots \cup f_n$**,** $EXEC = e_1 = e_2 = \ldots = e_n$ **and** $HOST \rightsquigarrow h_1 \cup \ldots \cup h_n$
- distributed computing parsing engine
- Sentence Server
- Separate parsers (clients) on each host
- Probability Server $\equiv$ `DecoderServer` object + multi-proc + large RAM $\Rightarrow$ smoothed top-level probability estimates to multiple small-chart parsing clients
- architecture features
  - load-balancing
  - fault-tolerant parsing engine w.r.t.
    - `DecoderServer`
    - `Switchboad`
  - Java RMI based architecture
  - copes with Solaris, Linux, Windows and MacOS X

BUILT FOR SPEED

- parser optimizations
  - log-probability estimates and log-lambdas precomputing
  - hash maps
  - **(new chart item == 0 probability)? short-circuit decoding ops**
  - object pool
  - smaller optimizations based on profiling

REPLICATING COLLINS' MODEL 2

	$\leq$ 40 words		$\leq$ 100 words	
Parser	**LR**	**LP**	**LR**	**LP**
Collins	89.75	90.19	88.47	89.30
Bikel	89.89	90.14	88.72	89.03

Tests carried out on Section 00 of the Penn Treebank

DEVELOPING A LANGUAGE PACKAGE FOR CHINESE

- implementation time: **one and a half days!**
- state of the art results: On sentences $\leq$40 words $\Rightarrow$ 77.0/81.6% LR/LP

REFERENCES

- Daniel M. Bikel. Design of a Multi-lingual, Parallel-processing Statistical Parsing Engine, HLT 2002 proceedings
- Daniel M. Bikel. A statistical model for parsing and word-sense disambiguation. In *Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, Hong Kong, October 2000.
- Daniel M. Bikel and David Chiang. Two statistical parsing model applied to the Chinese Treebank. In Martha Palmer, Mitch Marcus, Aravind Joshi, and Fei Xia, editors, *Proceedings of the Second Chinese Language Processing Workshop*, pages 1-6, Hong Kong, 2000.
- Eugene Charniak. Statistical Parsing with a Context-free Grammar and Word Statistics. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, Menlo Park. AAAI Press/MIT Press. 1997.
- Michael John Collins. *Head-Driven Statistical Models for Natural Language Parsing.* PhD thesis, University of Pennsylvania, 1999, Chapter 1, Pages 1-30.
- Mitchell P. Marcus, Beatrice Santorini, and Marry Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. Computational Linguistics, 19:313-330, 1993.
- George A. Miller, Richard T. Beckwith, Christiane D. Fellbaum, Derek Gross, and Katherine J. Miller. 1990. WordNet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235-244.

APPENDIX

TREEBANK – SAMPLE TAGGED TEXT[a]

Battle-tested/NNP*/JJ industrial/JJ managers/NNS here/RB always/RB buck/VB*/VBP up/IN*/RP nervous/JJ newcomers/NNS with/IN the/DT tale/NN of/IN the/DT first/JJ of/IN their/PP$ countrymen/NNS*/FW warriors/NNS blown/VBN ashore/RB 375/CD years/NNS ago/RB ./. "/" From/IN the/DT beginning/NN ,/, it/PRP took/VBD a/DT man/NN with/IN extraordinary/JJ qualities/NNS to/TO succeed/VB in/IN Mexico/NNP ,/, "/" says/VBZ Kimihide/NNP Takimura/NNP ,/, president/NN of/IN Mitsui/NNS*/NNP group/NN 's/POS Kensetsu/NNP Engineering/NNP Inc./NNP unit/NN ./.[b]

---

[a] after correction

[b] "*" marks multiple POS tags

TREEBANK – SAMPLE BRACKETED TEXT[a]

```
( (S
   (NP Battle-tested industrial managers
       here)
    always
   (VP buck
       up
       (NP nervous newcomers)
       (PP with
           (NP the tale
               (PP of
                   (NP (NP the
                           (ADJP first
                                 (PP of
                                     (NP their countrymen)))
                           (S (NP *)

                              to
                              (VP visit
                                  (NP Mexico))))
                       ,
                       (NP (NP a boatload
                               (PP of
                                   (NP (NP warriors)
                                   (VP-1 blown
                                       ashore
                                       (ADVP (NP 375 years)
                                             ago)))))
                           (VP-1 *pseudo-attach*))))))))
```

[a]after correction; simplified

WORD FEATURE EXAMPLE

- `WordFeatures` ≡ orthographic and morphological features of words. It encodes:
  1. capitalization
  2. hyphenation
  3. inflection
  4. derivation
  5. numeric
- Example: ”C3H0I0D3N0” stands for `Geography` (non-sentence-initial capitalised, no hyphenation, no inflection, `"graphy"` derivation and non-numeric)

BIKEL'S PARSER – PACKAGES

Packages	
danbikel.lisp	Provides classes to create, read and manipulate symbolic expressions (S-expressions), including interned symbols.
danbikel.parser	Provides the core framework of this extensible statistical parsing engine.
danbikel.parser.arabic	Provides language-specific classes necessary to parse Arabic.
danbikel.parser.chinese	Provides language-specific classes necessary to parse Chinese.
danbikel.parser.constraints	Provides interfaces and classes to allow constrain-parsing.
danbikel.parser.english	Provides language-specific classes necessary to parse English.
danbikel.parser.lang	Provides default abstract base classes for the required interfaces of a language package.
danbikel.parser.ms	Default package for model structure classes (subclasses of `ProbabilityStructure`).
danbikel.parser.util	Utility classes for displaying and manipulating parse trees.
danbikel.switchboard	Provides classes to implement a distributed client-server environment, with a central switchboard responsible for assigning clients to servers and for doling out objects to clients for processing.
danbikel.util	Provides some basic utility classes.
danbikel.util.proxy	Contains various `InvocationHandler` objects with static factory methods to provide *proxy instances*.

BIKEL'S PARSER – LANGUAGE PACKAGE

Class Summary	
AbstractHeadFinder	Provides a default abstract implementation of the HeadFinder interface.
AbstractHeadFinder.HeadFindInstruction	Data structure for specifying a way to search for a head in a grammar production: a set of symbols to scan for and the direction of that scan.
AbstractTraining	Provides methods for language-specific preprocessing of training parse trees.
AbstractTreebank	A collection of mostly-abstract methods to be implemented by a langauge-specific subclass.
AbstractWordFeatures	Provides a default abstract implementation of the WordFeatures interface.

IMPLEMENTATION METRICS

- 12 packages
- $\approx$ 240 classes
- $\approx$ 4800 methods ?!?

THANK YOU!