WS 2003

Syntax for Statistical Machine Translation

WS 2003

Syntax for Statistical Machine Translation

WS 2003

Syntax for Statistical Machine Translation

WS 2003

Syntax for Statistical Machine Translation

WS 2003

Syntax for Statistical Machine Translation

WS 2003

Syntax for Statistical Machine Translation

The Team (12 + 1)

- Franz Josef Och - ISI
- Daniel Gildea - Upenn
- Anoop Sarkar - SFU
- Kenji Yamada -XRCE
- Sanjeev Khudanpur - JHU
- + Dragomir Radev - Univ. of Michigan
- Alex Fraser - ISI
- Shankar Kumar - JHU
- David Smith - JHU
- Libin Shen - Upenn
- Viren Jain - Upenn
- Katherine Eng - Stanford
- Zhen Jin - Mt. Holyoke

Statistical Machine Translation

- Enormous progress in MT in recent years due to **statistical approaches**
  - Verbmobil: German-English (speech-to-speech)
  - TIDES project: Chinese, Arabic, Hindi
- Advantages:
  - **Better** quality
  - **Faster** (rapid) development cycle
  - **Cheaper**

SMT - Modeling

- **Modeling Pr(elf)**: describing the relevant dependencies between e and f
- Here: log-linear model that combines feature functions depending on Chinese string (f), English string (e) (+ word alignment)

$$\Pr(e|f) = p_{\lambda_1^M}(e|f) \propto \exp[\sum_{m=1}^{M} \lambda_m h_m(e,f)]$$

SMT - Training

- **Training** of model parameters
- Here: maximum BLEU training
  - Discriminative training with BLEU as objective function

$$\hat{\lambda}_1^M = \text{argmax BLEU}(r_1^S, \hat{e}(f_1^S; \lambda_1^M))$$

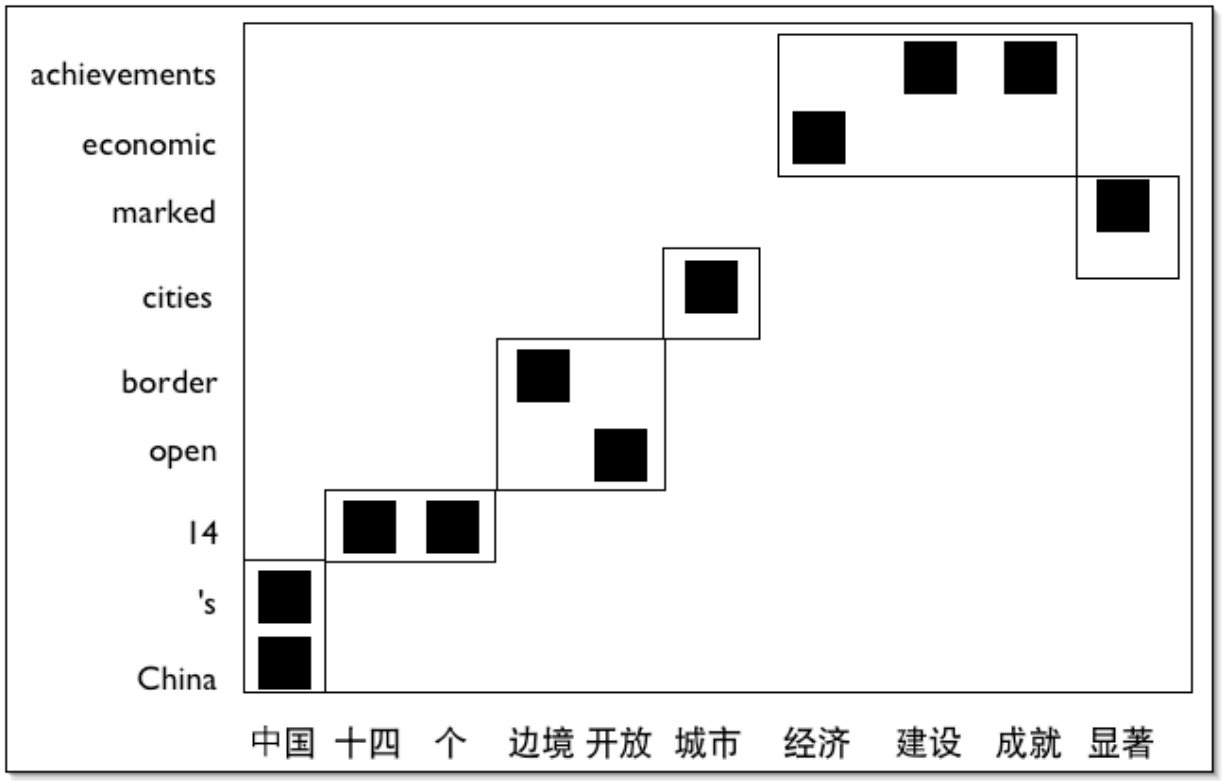
- Algorithm: greedy descent with optimal line search
- Advantage: directly optimizes evaluation criterion
- Problem: danger of overfitting to training data

SMT - Search

- Search: minimize expected loss
- Standard approach: 0-1 loss function
  - Later: Minimum Bayes risk with different loss functions (e.g. BLEU loss(!))
- Log-linear model: simple decision rule

$$\hat{e}(f) = \text{argmax}_e \left\{ \sum_{m=1}^{M} \lambda_m h_m(f, e) \right\}$$

Baseline: Alignment Templates



Basic idea: learn **all** aligned phrase pairs (= Alignment Templates) seen in training data

Baseline: Features

- Product of alignment template probabilities
- Product of word translation probabilities
- (4) Trigram language models
- Number of produced words
- Number of produced alignment templates
- … Some other feature functions

Chinese-English NIST Eval

System	2002 NIST score	2003 NIST score
AlTemp system	7.6 (RWTH)	9.0 (ISI)
Best competing	7.3	7.9
Best COTS	6.1	6.6

- 2003: Similar results for human evaluation

But…

- SMT often makes stupid errors
  - Missing content words:
    - MT: Condemns US interference in its internal affairs.
    - Human: Ukraine condemns US interference in its internal affairs
  - Verb phrase:
    - MT: Indonesia that oppose the presence of foreign troops.
    - Human: Indonesia reiterated its opposition to foreign military presence.

But…….

- Wrong dependencies
  - MT: …, particularly those who cheat the audience the players.
  - Human: …, particularly those **players who cheat the audience**.
- Missing articles:
  - MT: …, he is fully able to activate team.
  - Human: … he is fully able to activate **the** team.

But……

- Word salad:
  - the world arena on top of the u . s . sampla competitors , and since mid - july has not appeared in the sports field , the wounds heal go back to the situation is very good , less than a half hours in the same score to eliminate 6:2 in light of the south african athletes to the second round .

But……

- State-of-the-art SMT often makes ‘stupid’ mistakes
- Many problems are ‘syntactic’
- Possible reason:
  - State-of-the-art MT does not know enough about what is important to humans
- Idea:
  - Take into account syntactic dependencies more explicitly

The Plan

- Starting Point: state-of-the-art phrase-based statistical MT system
  - Here: Alignment Template system from ISI
  - Purely data-driven
- Error Analysis: What goes wrong?
- Develop syntactically motivated feature functions for specific problems

The Plan - Feature Functions

- Refined feature functions depend on:
  - Standard: English string, Chinese string, word alignment, phrase alignment
  - POS tag sequence
  - Chunk segmentation for Chinese/English
  - Parse trees for Chinese/English
  - Dependency parses for Chinese/English
  - …

The Plan - Major Tasks

1. Error Analysis/Feature Hunting
   - Contrastive error analysis
2. **Development of Feature Functions**
3. Discriminative Training Techniques
   - Maximum-BLEU, Perceptron
4. Search Approaches
   - Minimum Bayes risk with syntactic loss functions

The Plan - Framework (1)

- **Chinese-English large data track**
  - Training data for Training of Baseline FF
    - 150M words per language
  - Chinese treebank available
  - **Dev** data for Maximum BLEU training
    - NIST eval-01: 993 sentences
  - **Test** data
    - NIST eval-02: 878 sentences
  - Blind test data: (for after-workshop evaluation)
    - NIST eval-03: 929 sentences
  - Prepared also larger sets of development/test data
    - Dev: 4830 sentences, Test: 1813 sentences

The Plan - Framework (2)

- During workshop: **Rescoring of n-best lists**
  - Advantages:
    - No need to integrate FF in dynamic programming search
    - FF can depend arbitrarily on full Chinese/English sentence/parse tree/…
    - Simple software architecture
    - Using fixed set of n-best alternatives: FF = vector of numbers
  - (Disadvantage: limits improvements to n-best lists)
  - N=16384 alternative translations for Dev/Test
    - Precomputed before workshop

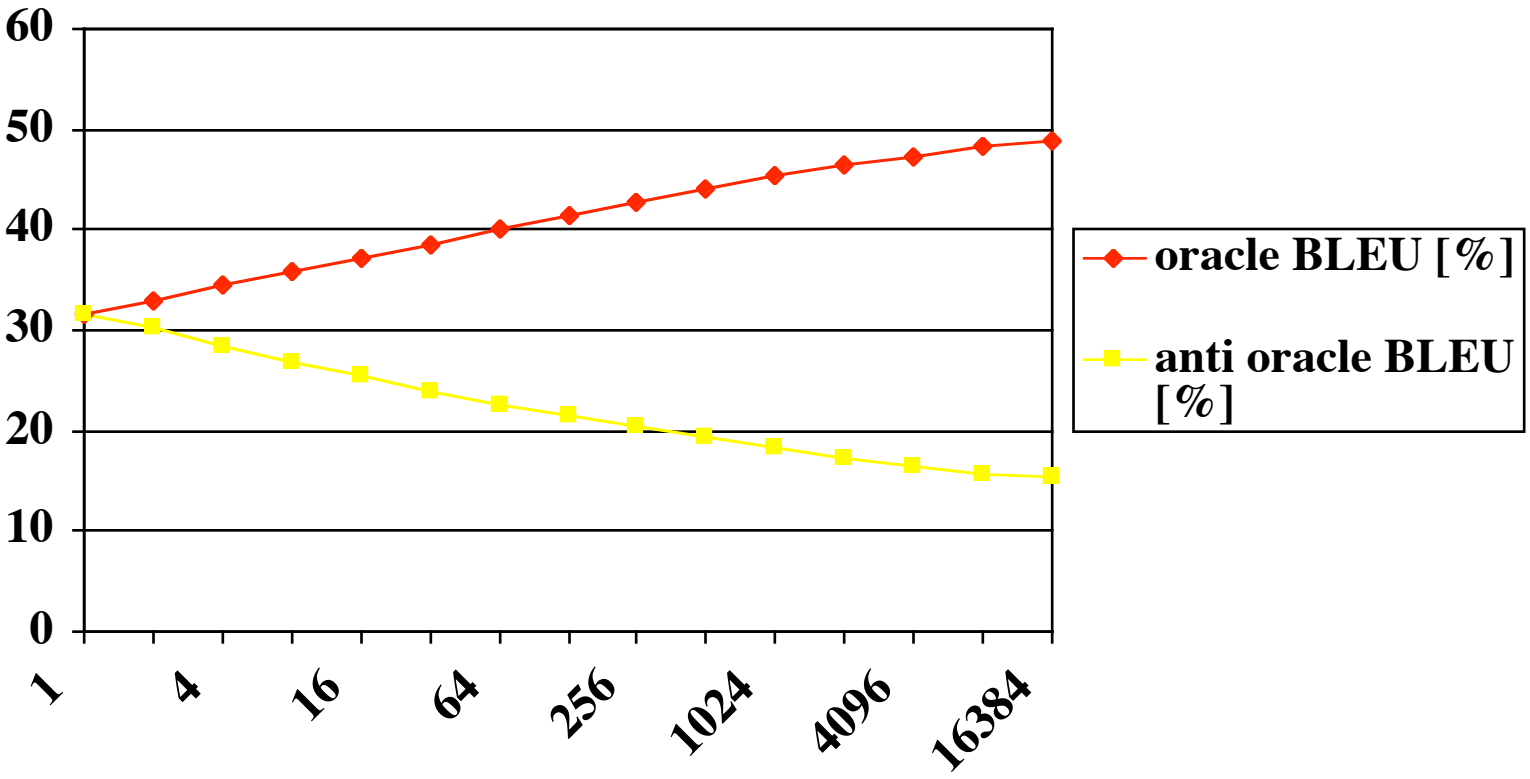
Evaluation metric: BLEU

- Automatic evaluation via BLEU
  - Compute precision of n-grams (here: 1,2,3,4-grams)
  - Geometric mean of n-gram precision
  - Multiply a penalty factor for too short translations
- Shown in recent evaluations to correlate with humans
  - But: MT performance looks too good compared to human performance measured by BLEU
- Important question for workshop: Is BLEU sensitive to (subtle) syntactic changes?
- After-workshop: subjective evaluation of final system

Oracle BLEU

- How large are potential improvements?
  - Oracle(-Best) translations: set of translations from n-best list that give best score
  - Oracle-Worst translations: set of worst scoring translations
- Computation for BLEU non-trivial
  - Greedy search algorithm

Oracle vs. anti-Oracle



How good is the oracle?

- Average human score
  - BLEU: 37.9% (3 refs)
- Average first-best score
  - BLEU: 27.6% (3 refs)
  - Relative to human BLEU score: 72.9 %
- Average oracle of 1000 best BLEU score
  - BLEU: 39.8% (3 refs)
  - Relative to human BLEU score: 105.0%
- But: quality of oracle is still bad…
  - Hence: reaching oracle is (very) unrealistic
  - Important to note: references used were taken into account during selection process of oracle
  - This experiment does not show that BLEU is not a good measure to assess MT quality

Contrastive error analysis

- Compare produced (first-best) translation with oracle translation
  - Quantitative: how often certain properties (features) hold (fire) for produced or oracle
  - Qualitative: what problems can be fixed in the n-best list
- What are the next low-hanging fruit?

Some Highlights

- Improved on best known Chinese-English MT result known so far…
- 450 syntactic (and other) new feature functions
  - Low-level, shallow, deep, tricky syntactic FF
- 7,965,393 million sentences parsed (a few times…)
  - Collins parser, Minipar, Dan Bikels parser, …
- 219 million alternative translations produced
- The first 100 GByte filled up after five days

Presentation Outline

- Approach/Motivation (Franz)
- Syntactic Framework and Preprocessing (Dan)
- Features
  - Implicit Syntax (David)
  - Shallow Syntax (Alex, Zhen)
  - Deep Syntax (Kenji, Katherine)
  - Tricky Syntax (Anoop, Viren)
- Rescoring Techniques
  - Perceptron Learning (Libin)
  - Minimum Bayes Risk (Shankar)
- Feature Combination and Evaluation (Sanjeev)
- Student Proposals (Viren, Libin)
- Conclusions (Franz)

Outline

- Approach/Motivation (Franz)
- Syntactic Framework and Preprocessing (Dan)
- Features
  - Implicit Syntax (David)
  - Shallow Syntax (Alex, Zhen)
  - Deep Syntax (Kenji, Katherine)
  - Tricky Syntax (Anoop, Viren)
- Rescoring Techniques
  - Perceptron Learning (Libin)
  - Minimum Bayes Risk (Shankar)
- Feature Combination and Evaluation (Sanjeev)
- Student Proposals (Libin, Viren)
- Conclusion (Franz)

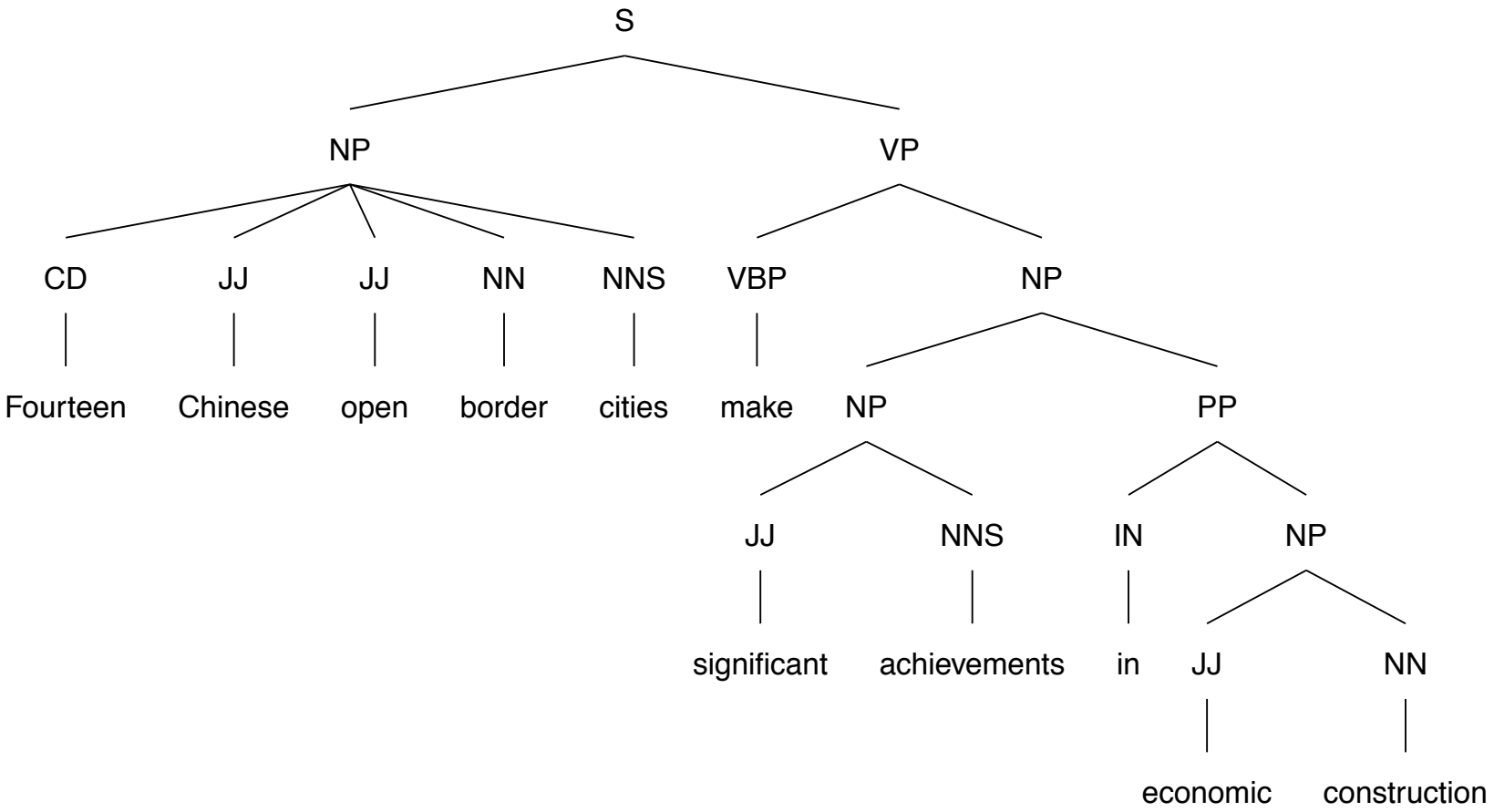
Data Processing

- POS tagging, chunking, parsing for both Chinese and English
- Case and tokenization issues
- Challenges: parsing and tagging MT output

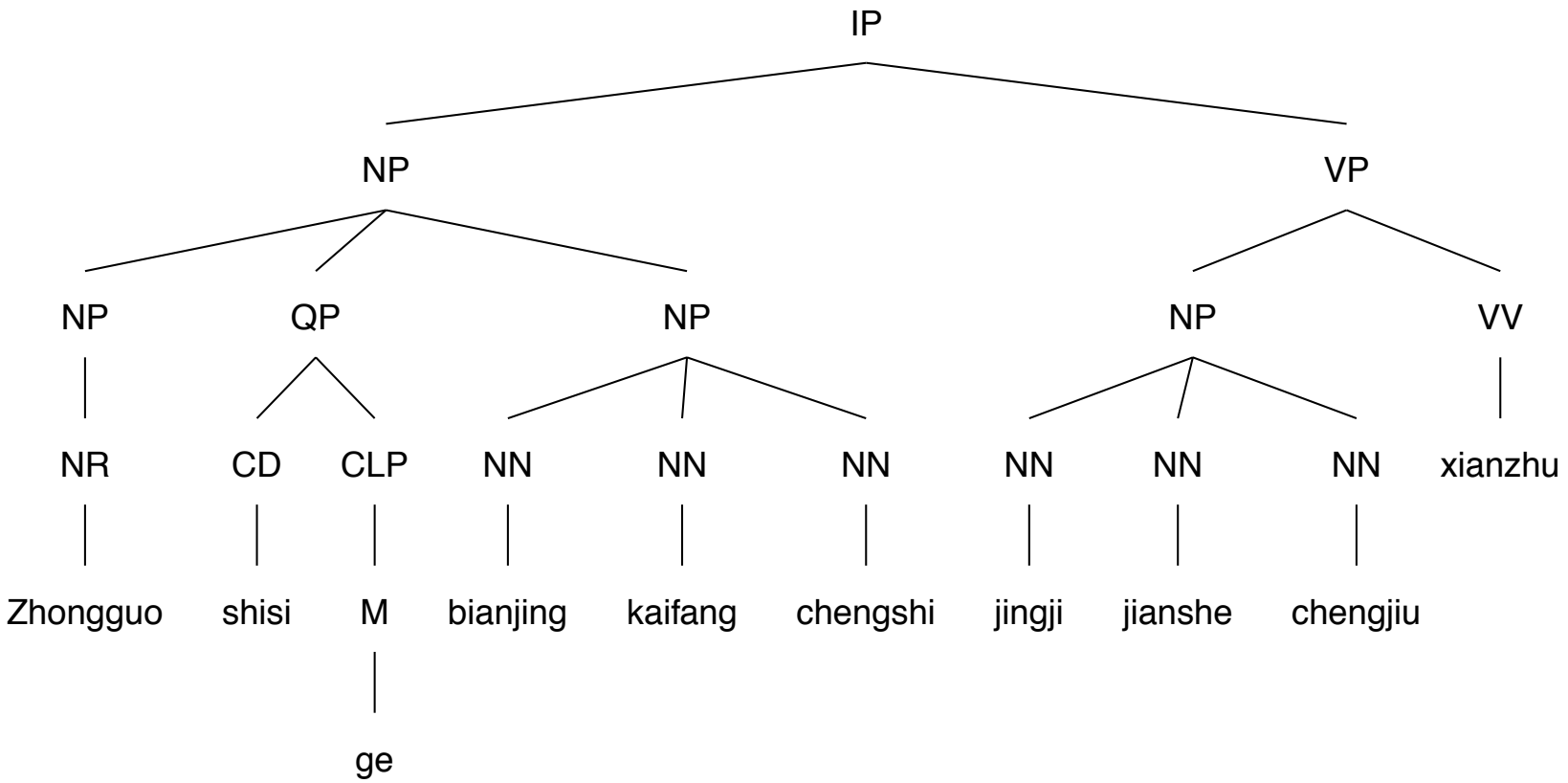
Tools

- Chinese segmenter: LDC, Nianwen Xue
- POS tagger: Ratnaparkhi, Nianwen Xue
- English parser: Collins/Charniak
- Chinese parser: Bikel (UPenn)
- Chunker: fnTBL (Ngai, Florian)

A Parse Tree


S
NP
VP
CD
JJ
JJ
NN
NNS
VBP
NP
Fourteen
Chinese
open
border
cities
make
NP
PP
JJ
NNS
IN
NP
significant
achievements
in
JJ
NN
economic
construction

A Chinese Parse Tree



Chunks

[$_{NP}$ Fourteen Chinese open border cities] [$_{VP}$ make] [$_{NP}$ significant achievements] [$_{PP}$ in] [$_{NP}$ economic construction] [$_{NP}$ Zhongguo]

[$_{QP}$ shisi ge] [$_{NP}$ bianjing kaifang] [$_{NN}$ chengshi] [$_{NP}$ jingji jianshe chengjiu] [$_{VP}$ xianzhu]

- Data Processing: Training data
  - English 1M sents (chunked all, parsed all)
  - Chinese 1M sents (chunked all, parsed 100K sents)
  - English/Chinese FBIS parsed: 70K sents
- Data Processing: n-best lists
  - English 5000 sents, 1000 nbest (tagging, chunking, parsing)
  - Chinese 5000 sents (segmentation, tagging, chunking, parsing)

Data Processing - Case Issues

- POS taggers, parsers expect mixed case input, MT system produces lower-case only text.
- Wrote a true-caser using SRI LM toolkit: 3.36% WER (Zhen, Katherine)
- Parsing tokenization vs. MT tokenization issues,
  e.g. *high␣@-@␣tech* $\longrightarrow$ *high-tech*;
  and␣/␣or $\longrightarrow$ *and\/or*

Word-Level Alignments

- Alignments from MT system do not match parse tokenization.
- Solved by writing a min edit-distance program to align MT with parsed text. (Viren)
- Composed the alignment produced by the MT system with the MT-to-parser token alignment.
- Some word level alignment information missing from n-best lists due to rule-based translations. Alignments added using separate alignment tool. (Zhen)

Processing Noisy Data

Tagger tries to “fix up” ungrammatical sentences:

> China_NNP 14_CD open_JJ border_NN cities_NNS achievements_VBZ remarkable_JJ

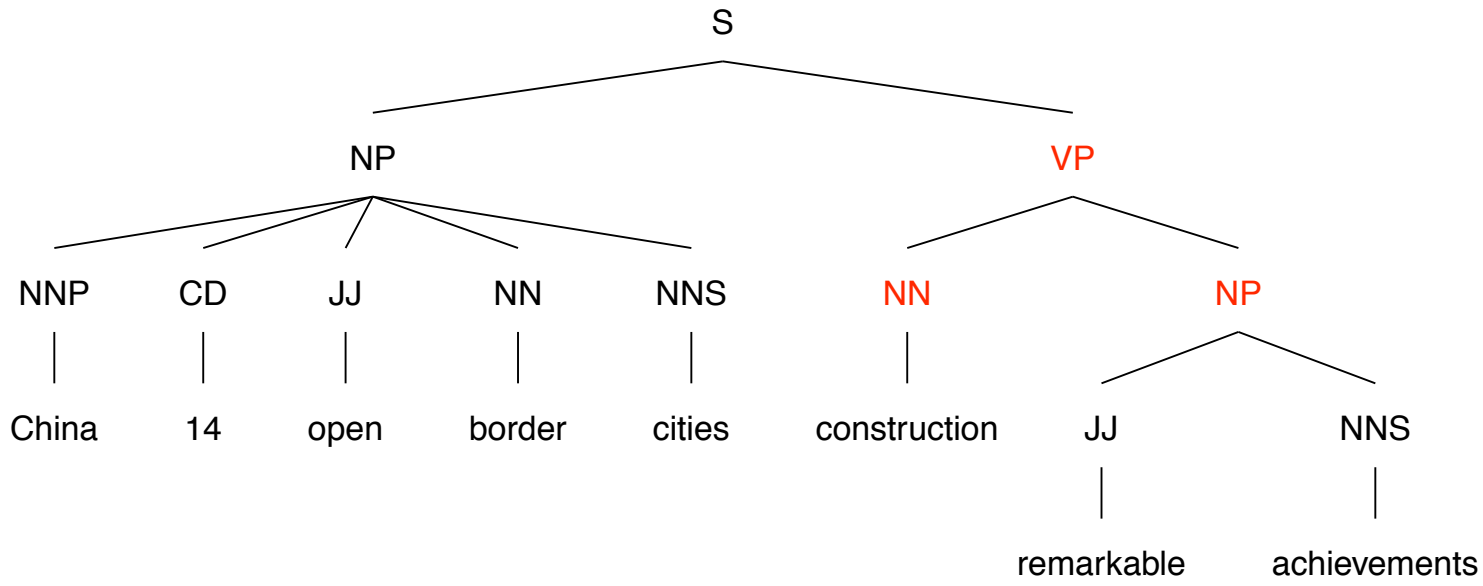
MT data include headlines with no verb.

Tagger trained on full sentences with normalized punctuation:

> China_NNP ’s_POS economic_JJ development_NN and_CC opening_VBG up_RP 14_CD border_NN cities_NNS remarkable_JJ achievements_.

Processing Noisy Data

Parser can create verb phrases where none exist:



Processing Noisy Data

Implications

- Features such “is there a verb phrase” may not do what you expect
- Possible solution: features involving probabilities of parse/tag sequence - “how good a verb phrase?”

Chinese Parsing

Although our Chinese data are “clean”

- Parsing highly dependent on segmentation
- Chinese parsing accuracy is lower than English even with perfect segmentation (82% parseval vs 90% for English)
- Also slower!

3% of English candidates and 4% of Chinese sentences in development set had no parse.

Outline

- Approach/Motivation (Franz)
- Syntactic Framework and Preprocessing (Dan)
- Features
  - Implicit Syntax (David)
  - Shallow Syntax (Alex, Zhen)
  - Deep Syntax (Kenji, Katherine)
  - Tricky Syntax (Anoop, Viren)
- Rescoring Techniques
  - Perceptron Learning (Libin)
  - Minimum Bayes Risk (Shankar)
- Feature Combination and Evaluation (Sanjeev)
- Student Proposals (Libin, Viren)
- Conclusion (Franz)

Presentation Outline

- Approach/Motivation (Franz)
- Syntactic Framework and Preprocessing (Dan)
- Features
  - Implicit Syntax (David)
  - Shallow Syntax (Alex, Zhen)
  - Deep Syntax (Kenji, Katherine)
  - Tricky Syntax (Anoop, Viren)
- Rescoring Techniques
  - Perceptron Learning (Libin)
  - Minimum Bayes Risk (Shankar)
- Feature Combination and Evaluation (Sanjeev)
- Student Proposals (Viren, Libin)
- Conclusions (Franz)

Implicit Syntax

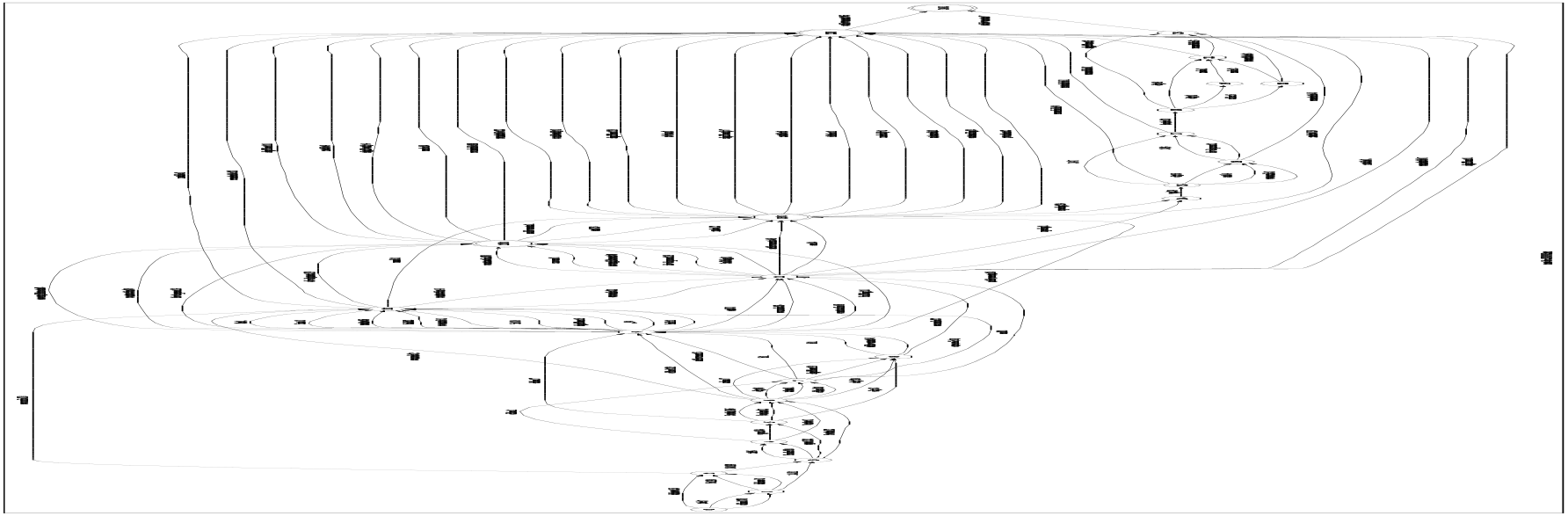
- **Exploiting sequence data not in baseline translation system**
- Other language modeling
- Multi-sequence alignment
- Evidence for alignment template ordering
  - AT reordering model
  - Overlapping ATs
- Content word translation and word triggering

Simple Language Modeling

- Skip language model
  - Interpolates immediate context and one-word-prior context
  - Relax for disfluent hypotheses
- Counts for frequently omitted function words (Katherine)
  - E.g., that, a, the, …
- Word “popularity” (Drago)
  - Consistent word choice in multiple-reference training
- Probability of word triggering its repetition
  - Neighboring templates may duplicate words
  - Produced: …multinational corporations in shandong has the exploratory stage to investment scale investment
  - Feature: … multinational corporations in shandong from investment exploratory stage to large-scale investment

Multi-Sequence Alignment

- Find sentence pairs with smallest edit distance
- Iteratively merge alignments
- Arcs weighted by consensus



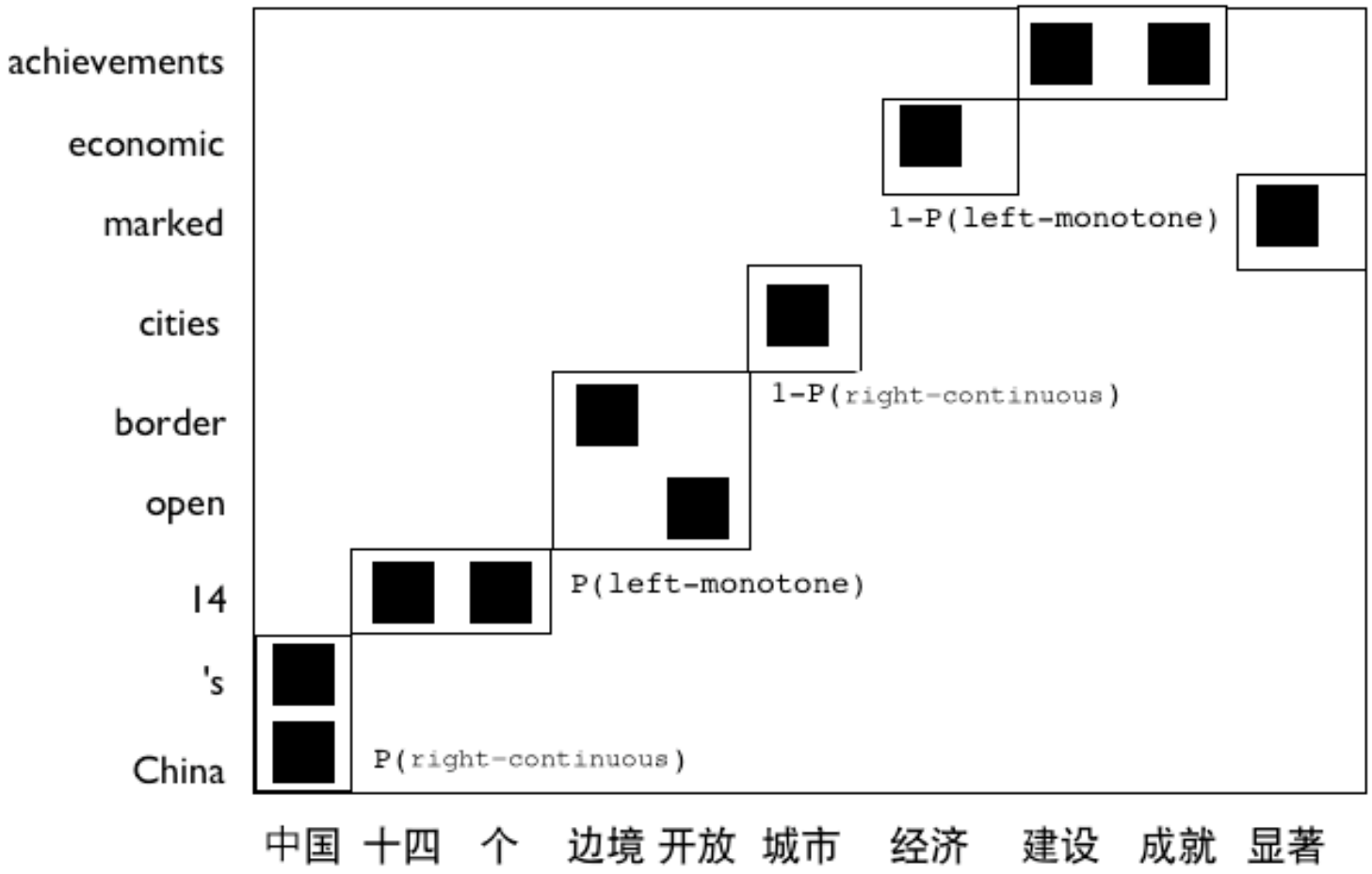
Multi-Sequence Alignment

- Features
  - Path weight of each hypothesis
  - Binary feature: agrees with majority arcs?
    - Feature: it was learned that chinese eiderdown products quality has shown a steady rise .
    - Baseline: it was learned that chinese eiderdown products quality has been a steadily increasing .
  - How many majority arcs?
- Augmenting hypothesis lists
  - Recomputing features, cf. Zhen

Alignment Template Ordering

- **Motivation:** Boundaries of alignment templates not always well founded
- **Idea:** Use probability of non-reordering joins, and multiple-AT support for joins
- **Implementation:** Feature function as product of join probabilities, or count of overlapping ATs (Anoop and Franz)

Alignment Template Ordering



Word Translation Models

- **Motivation:** Content word omissions
- Missing content word translation (Franz)
- IBM Model 1 (Kenji)
  - Additive translation probabilities
    - For each Chinese word C
      - Sum probabilities that each English word translates C
    - Multiply all these sums
  - Triggering effect among words
    - Among the best single features

Implicit Syntax Results

Feature	Test BLEU[%]
IBM Model 1	32.5
Right-continuous ATs	32.0
Left-monotone ATs	31.9
Missing content words	31.9
Overlapping ATs	31.9
Consensus path score	31.6
Skip language model	31.6

Syntax for Statistical Machine Translation

Outline

- Approach/Motivation (Franz)
- Syntactic Framework and Preprocessing (Dan)
- Features
  - Implicit Syntax (David)
  - Shallow Syntax (Alex, Zhen)
  - Deep Syntax (Kenji, Katherine)
  - Tricky Syntax (Anoop, Viren)
- Rescoring Techniques
  - Perceptron Learning (Libin)
  - Minimum Bayes Risk (Shankar)
- Feature Combination and Evaluation (Sanjeev)
- Student Proposals (Viren, Libin)
- Conclusions (Franz)

Syntax for Statistical Machine Translation

Shallow Syntactic Feature Functions

- Overview
- Part-of-Speech and Chunk Tag Counts
- Tag Fertility Models
- Lexicalized Chunk Features
- Projected POS Language Model
- Alignment Templates using POS Features
- Results
- A Trio of Approaches to Improve Punctuation

Alex Fraser / Zhenzhen Jin

Syntax for Statistical Machine Translation

Overview

- Scaling from Implicit Syntax to Shallow Syntax
  - Use linguistic resources to improve generalization
- Advantages
  - Part-of-Speech Taggers and Chunkers efficient
  - Decisions are local, may function better than parsers on noisy MT hypotheses
  - 1.3 M tagged parallel sentences for training
  - Simple models allow quick reaction to problems evident from contrastive error analysis
- Disadvantages
  - POS information implicit in baseline system
  - Chunks roughly at Alignment Template granularity
  - Generalization power less than full syntax

POS and Chunk Tag Counts

- Motivation: baseline low-level syntactic problems
  - Overgeneration of article, comma, singular nouns
  - Undergeneration of pronouns, past tense, coordination, plural nouns, etc
- Idea: reranker can learn to favor more balanced distribution of tags
- Implementation and Examples:
  - tag count and tag count difference (POS, Chunk)
    - Difference in number of NPs from Chinese to English
  - tag translation counts using word alignment (POS)
    - Number of Chinese N tags translated to only non-N tags

Tag Fertility Models

- Motivation: baseline tag distribution is wrong
- Idea: improve English tag distribution using Chinese tags
- Implementation and Examples:
  - Bag of Tags
  - Similar to IBM Model 1, but explicitly models tags with zero counts
  - Conditional probability of number of English tags given number of Chinese tags (POS, Chunk)
    - $P(NP_{english} = 2 \mid NP_{chinese} = 1)$
    - $P(CC_{english} = 1 \mid CC_{chinese} = 0)$
    - $P(CD_{english} = 1 \mid M_{chinese} = 1)$
  - Log-linear combination of conditional probabilities
    - Various combinations tried, smoothing is large issue
    - Parameterized distribution on ratio might be better

Lexicalized Chunk Features

- Motivation: Part of speech information is too local, but not all chunks reflect longer dependencies
- Idea: Lexicalized chunking uses NP, VP, ADJP chunk labels; words otherwise
- Implementation:
  - Process English and/or Chinese into chunk sequence.
  - Build Language Model or Translation Model.
  - Feature Function is log probability of the model.
- Examples:
  - Bad: NP NP and VP up NP
  - Good: NP NP VP up NP

Projected POS Language Model

- Motivation: Baseline system has weak model of word movement
- Idea: use Chinese POS tag sequences as surrogates for Chinese words to model movement
- Implementation:
  - Chinese POS sequences projected to English using the word alignment. Relative positions indicated for each Chinese tag.
  - Feature function is log probability of trigram language model built on projected Chinese tags and positions
  - Similar to HMM Alignment model (Vogel), but on POS
- Example:

CD_+0_M_+1	NN_+3	NN_-1	NN_+2_NN_+3
14 (measure)	open	border	cities

Alignment Templates Using POS Features

- Motivation: Alignment Templates may be too lexically dependent
- Idea: Probabilities of consecutive-word-phrase Alignment Templates could be smoothed using POS tag phrases
- Implementation and Examples:
  - Sequences of Chinese or English POS tags in templates
    - (C:NN), (E:JJ_NN)
  - Pairs of Chinese – English POS sequences in the aligned templates
    - (C:NN – E:JJ_NN), (C:NN – E:JJ_JJ_NN)
  - Lexicalized POS sequences and pairs
    - (E:JJ_NN#market), (C:NN – E:JJ_NN#market)
- Unigram language model computes generative probability
- Combined with the conditional probability of a pair given the Chinese POS sequence or English POS sequence

Syntax for Statistical Machine Translation

Results

Feature Name	BLEU[%]
Number of VBP (not sing. present 3rd person) (Alex)	31.7
V tag to V tag translation (Alex)	31.6
Pronoun to Pronoun translation (Alex)	31.6
Chunk to Chunk Tag Fertility (Alex)	31.5
POS to POS Tag Fertility (POS equivalences) (Alex)	31.4
Lexical Chunk Translation Model (David)	31.7
Lexical Chunk Language Model (David)	31.5
Projected POS Language Model (Alex)	31.8
POS Alignment Template (AT) Language Model (Libin)	31.6
POS AT Given Chinese POS Sequence (Libin)	31.4
And many, many more! (244 feature functions tried)	

A Trio for Punctuation

- Motivation:

  Frequent appearances of ungrammatical punctuations, especially)(s and “”s

- Ideas:
  - Feature function that penalizes the occurances of unbalanced parens
  - Correction of the wrong parens in the n-best hypotheses — adding new hypotheses
  - Feature function that checks the alignment of punctuation-grouped English and Chinese words

FF: Matching Parens

- Implementation:

  h-PAREN(e) = the number of occurances of)***(, (***(, (***)***), (), etc.

- Example:

  B: *news in brief) (korean high @-@ ranking officials from april to visit the democratic people 's republic of korea*

  B+F: *(bulletin) korean high @-@ ranking officials from april to visit the democratic people 's republic of korea*

  REF: *(news in brief) south korean high @-@ ranking officials to visit north korea in april*

- Yet Another Problem:

  Little discrimination power when most of the nbests for a single sentence make the same mistake

Parentheses Correction Based on Word Alignment

- Implementation:
  - Delete unaligned parentheses
  - Insert opening paren before the first word aligned to the first Chinese word inside the parentheses
  - Insert closing paren after the last word aligned to the last Chinese word inside the parentheses
- Challenges of adding new n-bests
  - feature function values
  - n-best score
- BLEU score evaluation:

  Trivial improvement to the baseline score

FF: Punctuation Restrain

- Implementation:

  h-PuncRestrain(e,f)= the percentage overlap of punctuation-grouped English and Chinese words

  - Dislike word movement around punctuations (10/11)

    Chinese words in Chinese position 01 (345)78

    English words in Chinese position 01 (34)**5**7778

  - Abhor missing punctuations (6/10)

    Chinese words in Chinese position 01 (345)78

    English words in Chinese position 01 (345**7778**

- Example:

  REF: *sharon will make a televised speech to the nation at 8:30 p.m. local time on the 31 st (1:30 on april 1 , beijing time) and explain to the people the government 's position on handling the conflict between israel and palestine .*

  B: *sharon will be in the 31 st 8:30 evening local time (beijing time) on april 1 1:30 ...*

  B+F: *sharon in the 31 st 8:30 evening local time (beijing time on april 1 1:30) ...*

- BLEU score 31.4

- Human Evaluation:

Total	Better Baseline Output	Better B+F Output	As Bad
30	3	7	20

Outline

- Approach/Motivation (Franz)
- Syntactic Framework and Preprocessing (Dan)
- Features
  - Implicit Syntax (David)
  - Shallow Syntax (Alex, Zhen)
  - Deep Syntax (Kenji, Katherine)
  - Tricky Syntax (Anoop, Viren)
- Rescoring Techniques
  - Perceptron Learning (Libin)
  - Minimum Bayes Risk (Shankar)
- Feature Combination and Evaluation (Sanjeev)
- Student Proposals (Libin, Viren)
- Conclusion (Franz)

Feature Functions: Deep Syntax

Kenji Yamada / Katherine Eng

Deep Syntax

- What is deep? — use of parser output
- Why parser? — grammaticality can be measured by parse trees
- How to use parser output?
  - simple features
  - model-based features
  - dependency-based features
  - other complex (tricky) features — next section

Simple features: Parser score

Motivation: grammatical sentences should have higher parse prob.

Feature Functions:

- $log(\text{parseProb})$ (Alex)
- $log(\text{parseProb}/\text{trigramProb})$ (Anoop)

Result: **worse than baseline**

Does Parser give high probability for grammatical sentence?

Parser LogProb for produced/oracle/reference sentences (Shankar)

	$log(\text{parseProb})$
produced	-147.2
oracle	-148.5
ref 1	-148.0
ref 2	-157.5
ref 3	-155.6
ref 4	-158.6

Other simple parse-tree features

Motivation: grammatical sentences should have specific tree shape.

Feature Functions: (Anoop)

- right branching factor
- tree depth
- num. of PPs
- VP probs
- ...

Model-based features

Translation Model as Feature Function

- Originally developed as a standalone model $P(f|e)$
  - Syntax-based model for parse trees

- $P(f|e)$ can be used as a feature value
  - Tree-based models represent systematic difference between two languages' grammar
    - e.g. SVO vs. verb-final word order
    - constituents (e.g. NP) tend to move as a unit

- Better translation should yield higher probs

- featureVal = $log[P(f|e)]$

Syntax-based Translation Model

Tree-based probability model for translation

- Early work:
  - Inversion Transduction Grammar [Wu 1997]
  - Bilingual Head Automata [Alshawi, et. al 2000]
- Tree-to-String [Yamada & Knight 2001]
- Tree-to-Tree [Gildea 2003]

Syntax-based Translation Model (cont)

Probabilistic operation on parse tree:

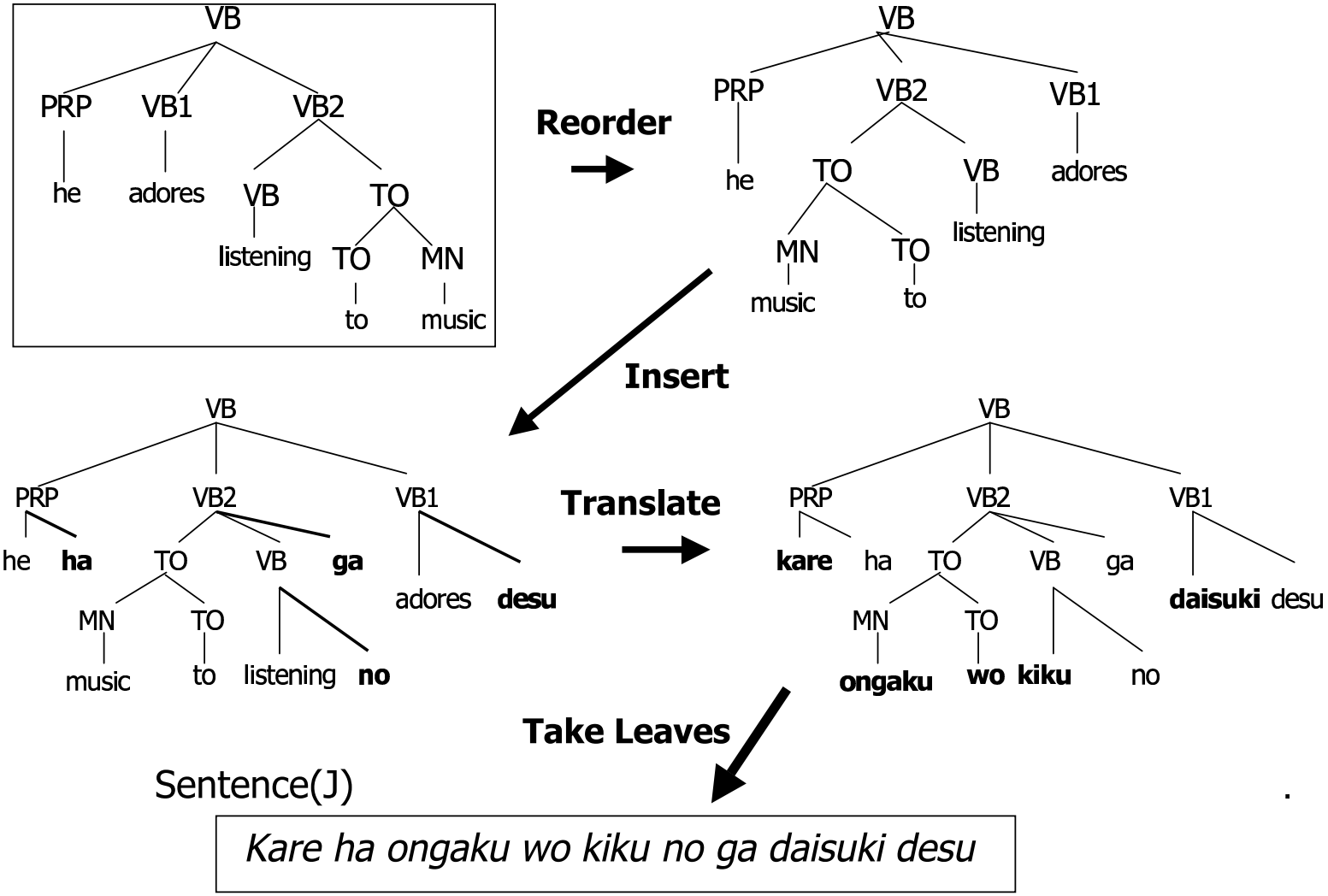
- Reorder
- Insert
- Translate

- Merge
- Clone

Parameters are estimated from training pairs (Tree/Tree, Tree/String) using EM algorithm.

Parse Tree(E) → Sentence (J)

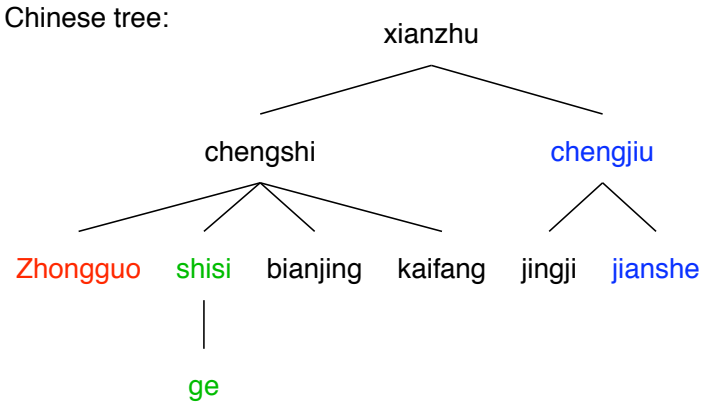
Parse Tree(E)

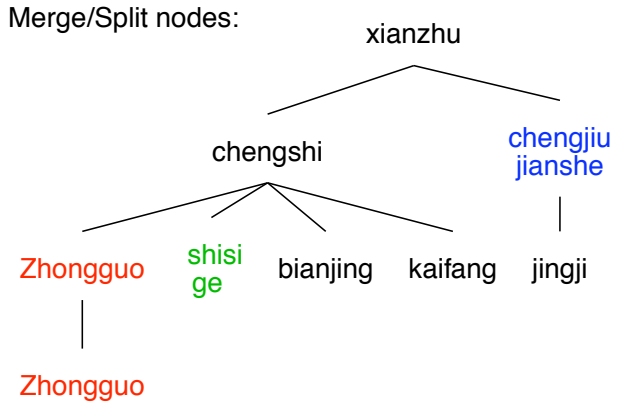


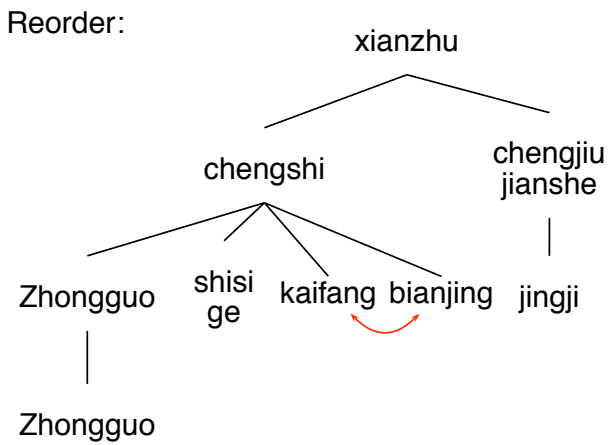
Sentence(J)

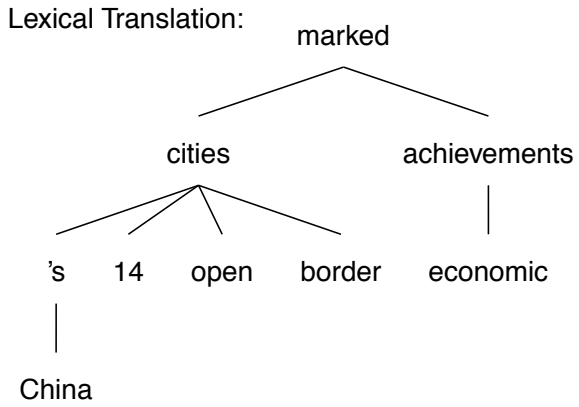
Kare ha ongaku wo kiku no ga daisuki desu

Tree-to-Tree Alignment

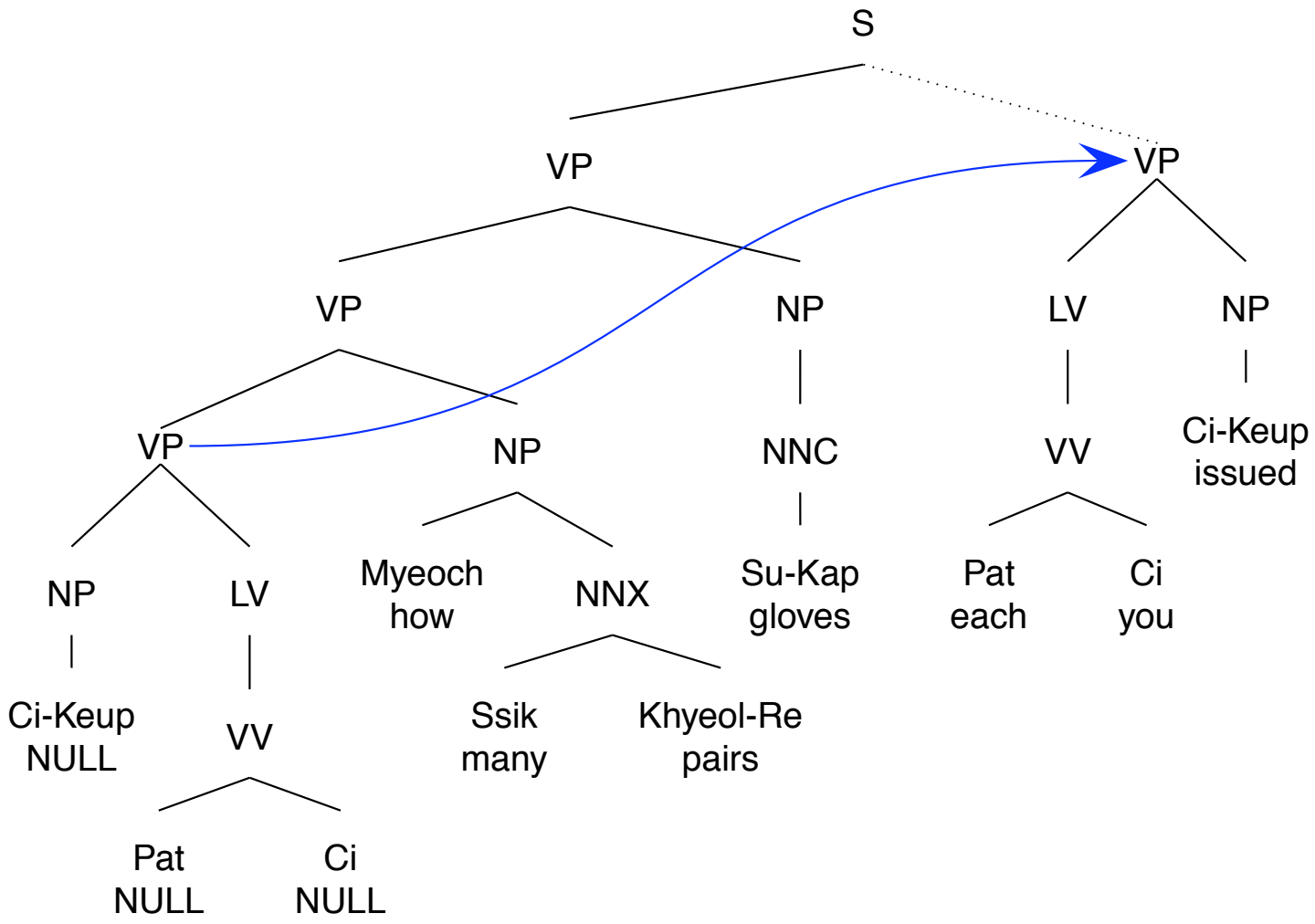








Cloning example



Problems

- n-best list doesn't contain big word jump
  - reordering at upper node is useless
- English/Chinese word-order is almost the same
  - both SVO in general
  - but relative clause comes before noun
- Computationally expensive
  - use word-level alignment from MT output
  - limit by sentence length and fanout
  - break up long sentences into small fragments (machete)

Experiments

Tree-to-String (Kenji, Anoop)

- Trained on 3M words of parallel text
  - English side parsed by Collins
- Max sentence length 20 Chinese characters
  - 273/993 sentences covered

Tree-to-Tree (Dan, Katherine)

- Trained on 40,000 biparsed FBIS sentences
- Max fan-out 6, max sentence length 60
  - 525/993 sentences covered

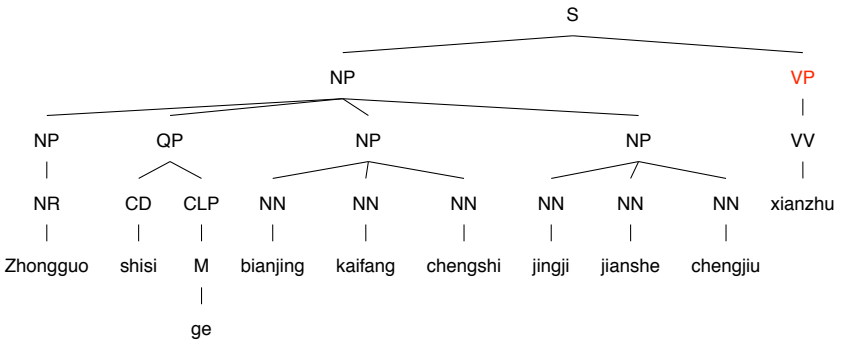
Results

	BLEU%	
Baseline	31.6	
ParseProb	31.6	
ParseProbDivLM	31.0	
RightBranching	31.6	
TreeDepth	31.5	
numPPs	31.3	
VPProb	31.3	
Tree-to-String	31.7	(32.0)
Tree-to-Tree	31.6	

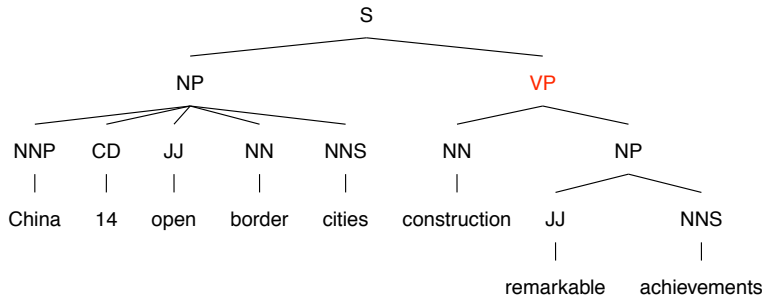
Dependency Tree Features

- **Motivation**
  - Constituency trees can be unreliable due to bad parsing
  - Difference in tree depth makes finding a good tree-to-tree alignment difficult even though trees are syntactically similar

chinese constituency tree



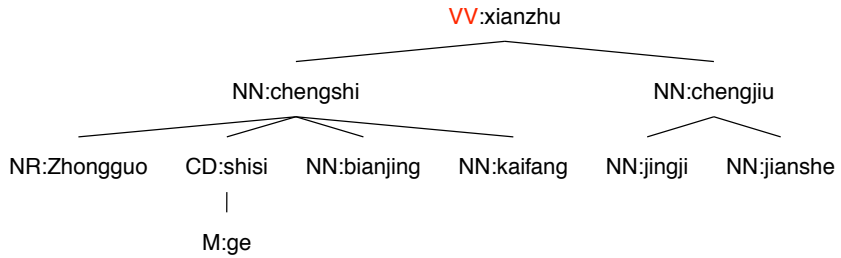
english constituency tree



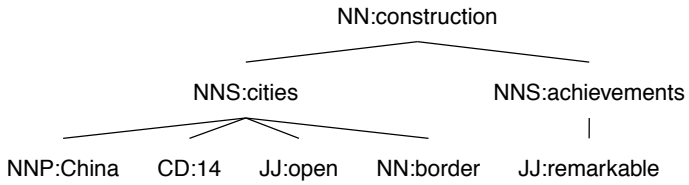
Dependency Trees: Feature Extraction

- **Idea**
  - Use dependency trees to eliminate non-terminal super-structure above lexical items
  - Represent more directly the relationships between words

chinese dependency tree



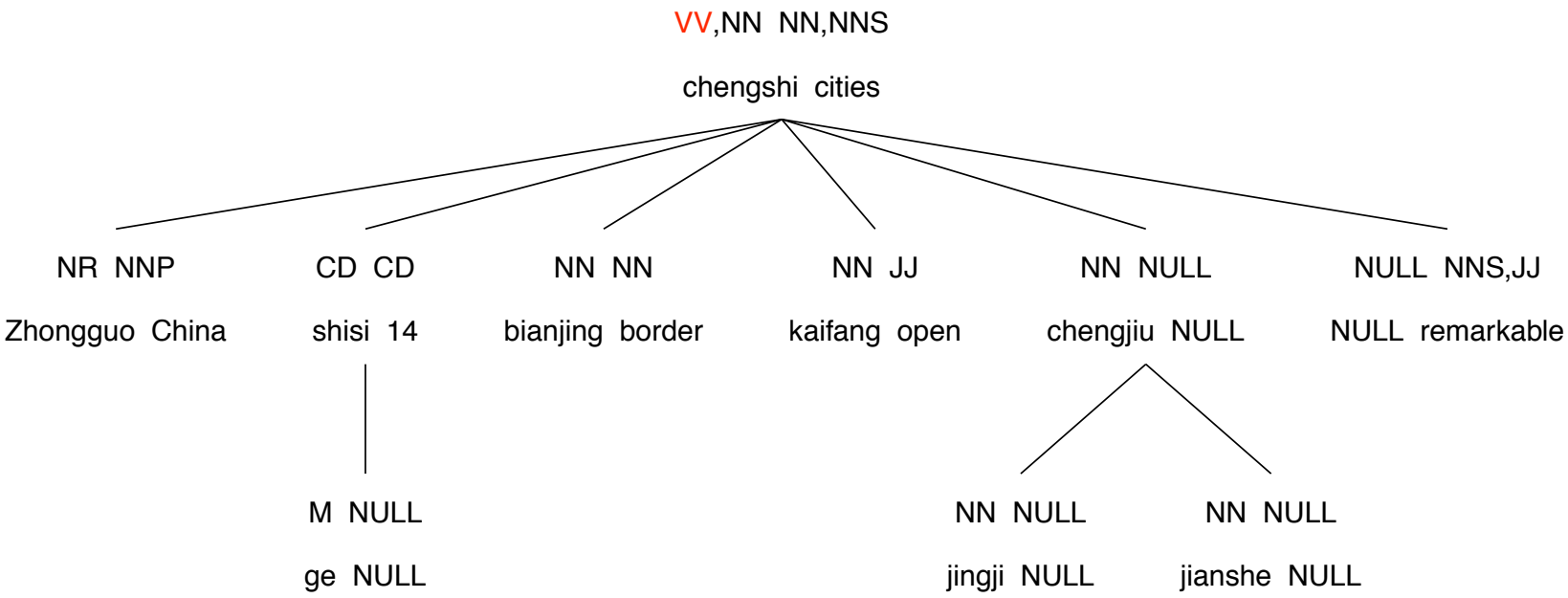
english dependency tree



Aligned Dependency Tree Features

- Modify regular tree-to-tree alignment to align dependency trees
  - Consider translation occuring at every node
  - Introduce lexical reordering probability
- Train aligner on about 40,000 pairs of FBIS Chinese/English sentences converted to dependency trees
- Extract features from alignments
  - Alignment score
  - Number of similar nodes (e.g. verbs) that align

Sample Alignment



$h\_ALIGNSCORE(e, f)$ = alignment probability (3.940690245314e-26)

$h\_VERBALIGN(e, f)$ = Number of times a verb aligns with another verb (0)

Dependency Projection Features

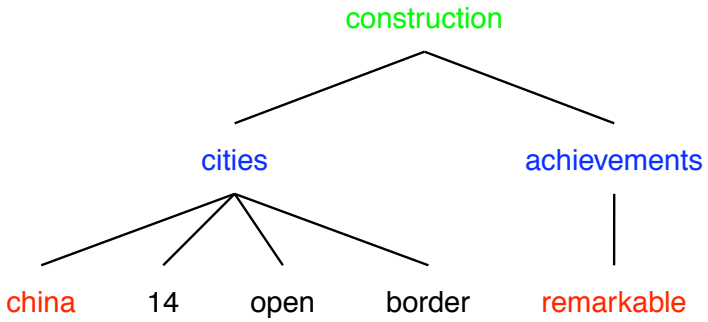
- **Motivation:** Dependency relationships tend to hold in both a sentence and its translation (e.g., Hwa et al., 2002)
- **Implementation:** Count each dependency in Chinese where the aligned English words also have a dependency

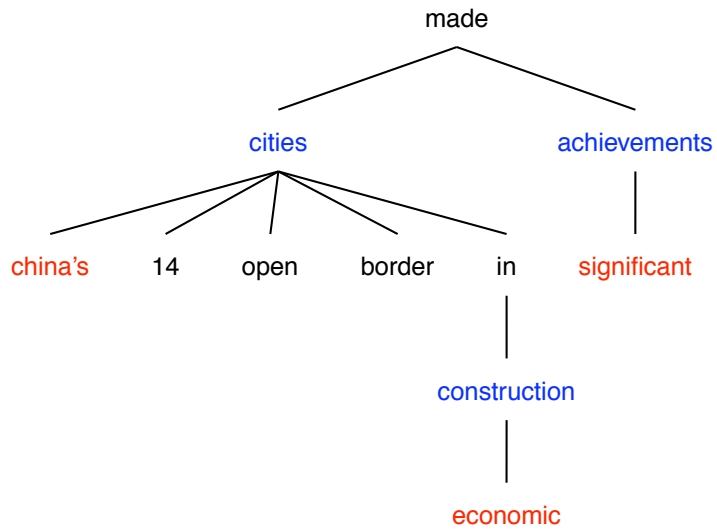
Examining the Plausibility of Lexical Dependencies

Intuition: Lexical dependencies proposed by the parser for a English hypothesis should be evidenced in fluent English text — e.g. *strong* tea and ∗*powerful* tea.

One Hypothesis: china 14 open border cities construction remarkable achievements

A Better Hypothesis: china's 14 open border cities in economic construction made significant achievements





Reference: fourteen chinese open border cities make significant achievements in economic construction

Dependency Trigram LM Probability Ratio

Idea: A trigram language model on dependencies extracted from fluent English.

$$P(\mathbf{e}) = P(e_1, e_2, \ldots, e_m) \approx \prod_{i=1}^{m} P_E\left(e_i \,|\, \text{parent}(e_i), \text{grandparent}(e_i)\right)$$

Concern: Content words tend to have low LM probability — the model may prefer hypotheses with fewer content words.

Solution: Normalize the probability of e_i by the probability of the Chinese word f_j of which it is (presumably) the translation. Use word-alignments $\mathbf{a} = \{a_{ij}\}$.

$$\text{DependencyLMratio}(\mathbf{e}, \mathbf{f} \,|\, \mathbf{a}) = \prod_{i=1}^{m} \left| \frac{P_E\left(e_i | \text{parent}(e_i), \text{grandparent}(e_i)\right)}{P_C\left(f_j | \text{parent}(f_j), \text{grandparent}(f_j)\right)} \right|$$

Remark: DependencyLMratio$(\mathbf{e}, \mathbf{f} \,|\, \mathbf{a})$ is likely to detect "rare" Chinese words mistakenly translated to "frequent" English words and vice versa.

Translation Results for Dependency Derived Features

Investigated many variants on dependency-derived features. Combined them one-at-a-time with features from the baseline system (baseline: 31.6%).

$$P(\mathbf{e} \mid \mathbf{f}) \propto \exp\left\{\sum_{m=1}^{M} \lambda_m h_m(\mathbf{e},\mathbf{f}) + \lambda_{M+1} h_{M+1}(\mathbf{e},\mathbf{f})\right\}$$

Feature Name	BLEU
Both parses have verbs	31.6%
Verb Alignment	**31.6**%
Difference in # Nouns	31.4%
Dependency Projection	**31.6**%
Doubly Transitive Dependencies	31.4%
Maximum LM Ratio	31.6%
Average LM Ratio	31.3%
Minimum LM Ratio	31.5%

Feature Name	BLEU
Number of args of main verb	31.6%
Noun Alignment	31.4%
Difference in # Verbs	31.4%
Transitive Eng Dependencies	31.7%
Labeled-Dep Model-1 Prob	31.5%
Alignment Score	**31.6**%
Dependency LM Score	31.7%
Dependency LM Ratio	**31.5**%

Outline

- Approach/Motivation (Franz)
- Syntactic Framework and Preprocessing (Dan)
- Features
  - Implicit Syntax (David)
  - Shallow Syntax (Alex, Zhen)
  - Deep Syntax (Kenji, Katherine)
  - Tricky Syntax (Anoop, Viren)
- Rescoring Techniques
  - Perceptron Learning (Libin)
  - Minimum Bayes Risk (Shankar)
- Feature Combination and Evaluation (Sanjeev)
- Student Proposals (Libin, Viren)
- Conclusion (Franz)

Syntax-based Alignment Features

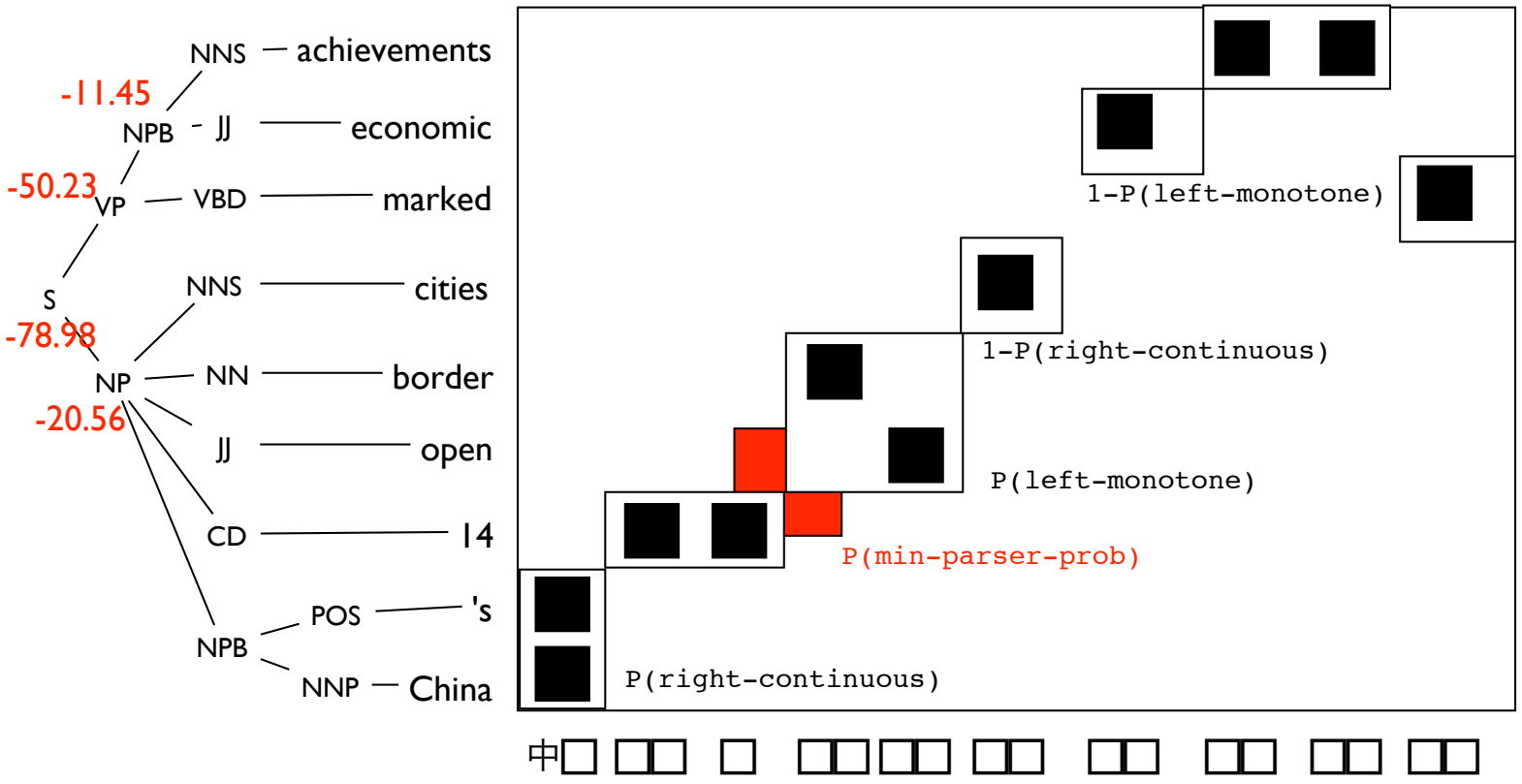
Without aligning full parse trees
aka “tricky syntax”

Anoop Sarkar/Viren Jain

Feature Functions

- Parser probabilities over alignment template boundaries
- Hacking up the parse tree: a Markov assumption for tree alignments
- Using trees for scoring word alignments
- **Viren**: scoring constituents based on word alignments

Parser probabilities over alignment template boundaries

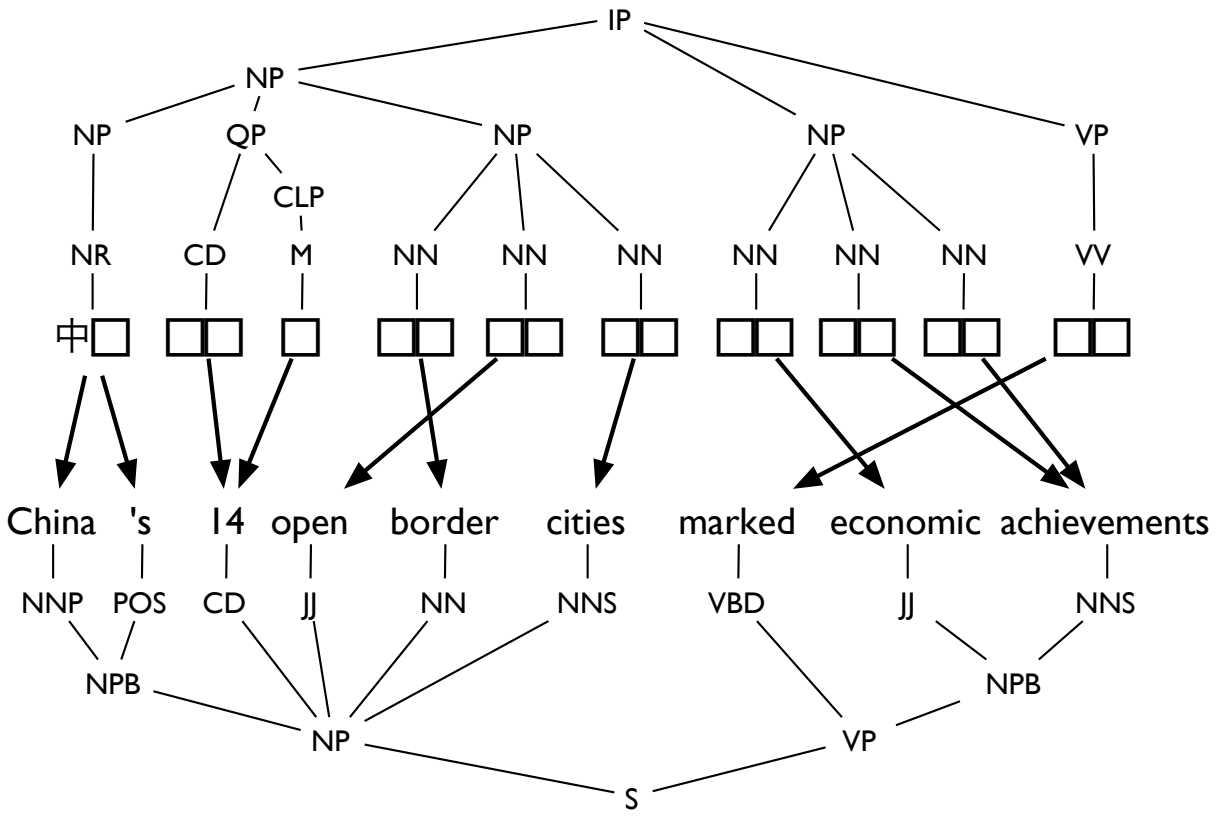


$h_{sentATParserOverlap}$ = -20.56 -50.23 -11.45 ... = -105.78

Tricky Syntax

- Parser probabilities over alignment template boundaries
- Hacking up the parse tree: a Markov assumption for tree alignments
- Using trees for scoring word alignments
- **Viren**: scoring constituents based on word alignments

Hacking up the parse tree



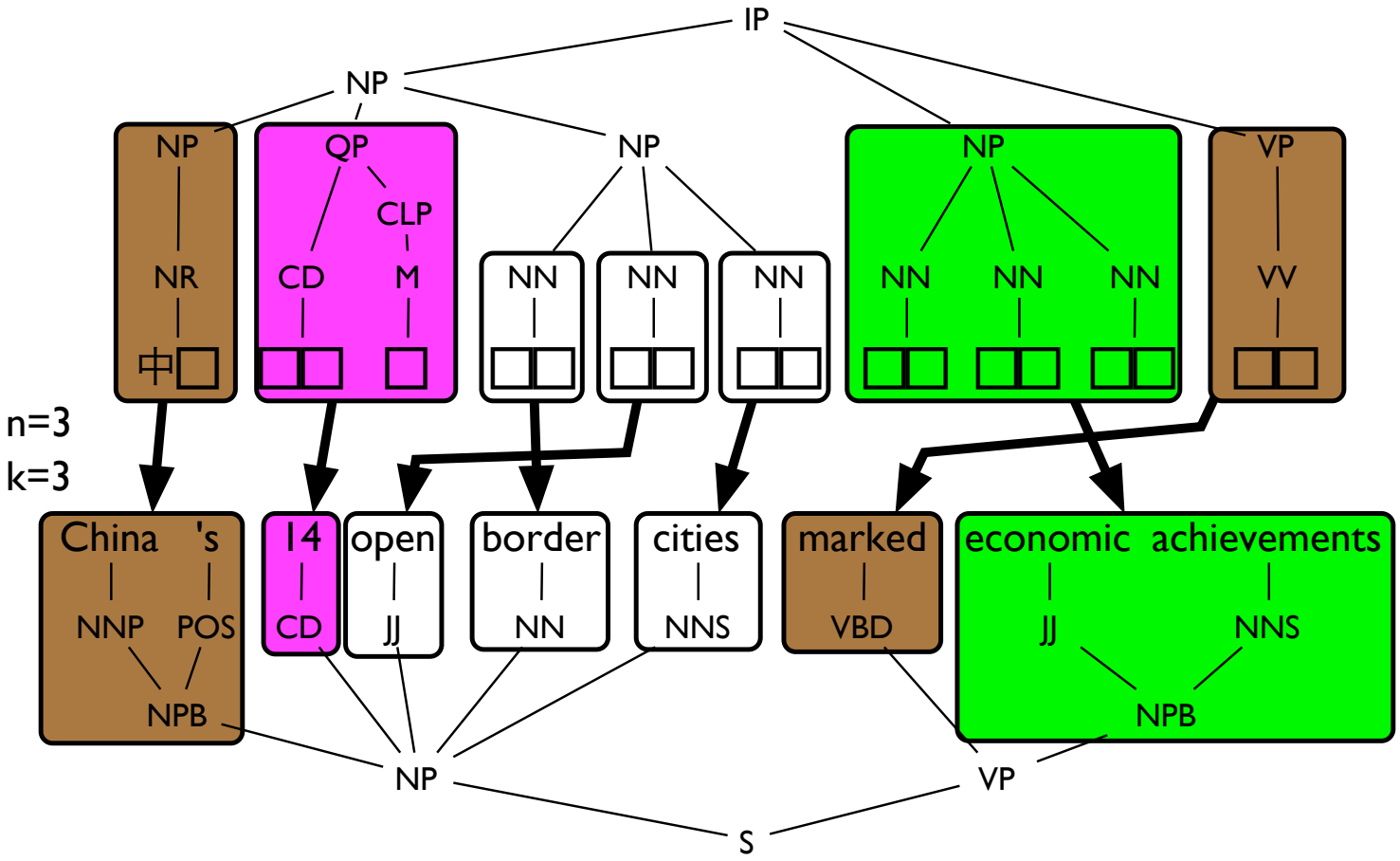
Hacking up the parse tree

- Full parse tree models are expensive to compute for long sentences and for trees with flat constituents

- Limited reordering in the nbest lists: higher levels of parse tree rarely reordered

- Algorithm for hacking into tree fragments: start with word alignments, two parameters: n for maximum number of words in tree fragment and k for maximum height of tree fragment

- Advantage: a simple Markov model for tree-based alignments, tractability (covers 98% of the nbest list) and more robust to bad parses

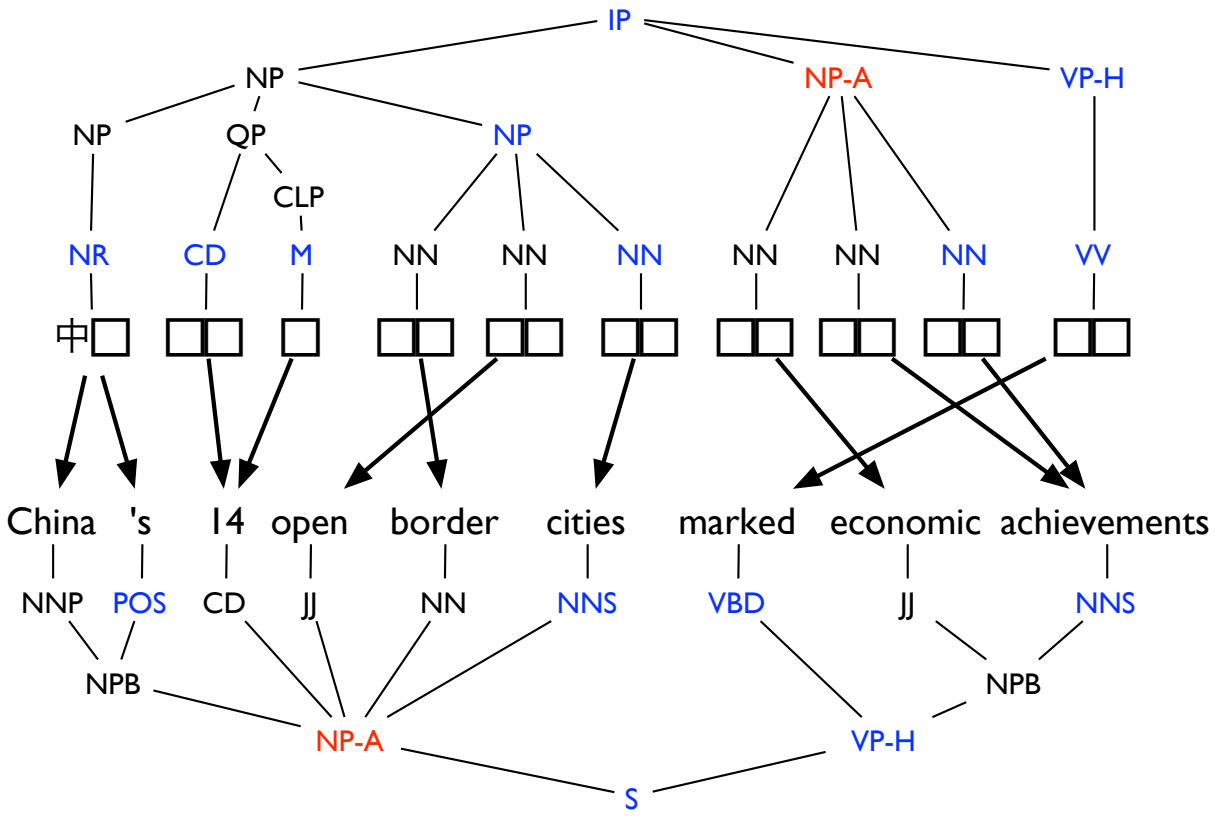


$h_{\text{TreeToStringMachete}}$ = log(P_TreeToString(Frag0)) + log(P_TreeToString(Frag1)) + ...

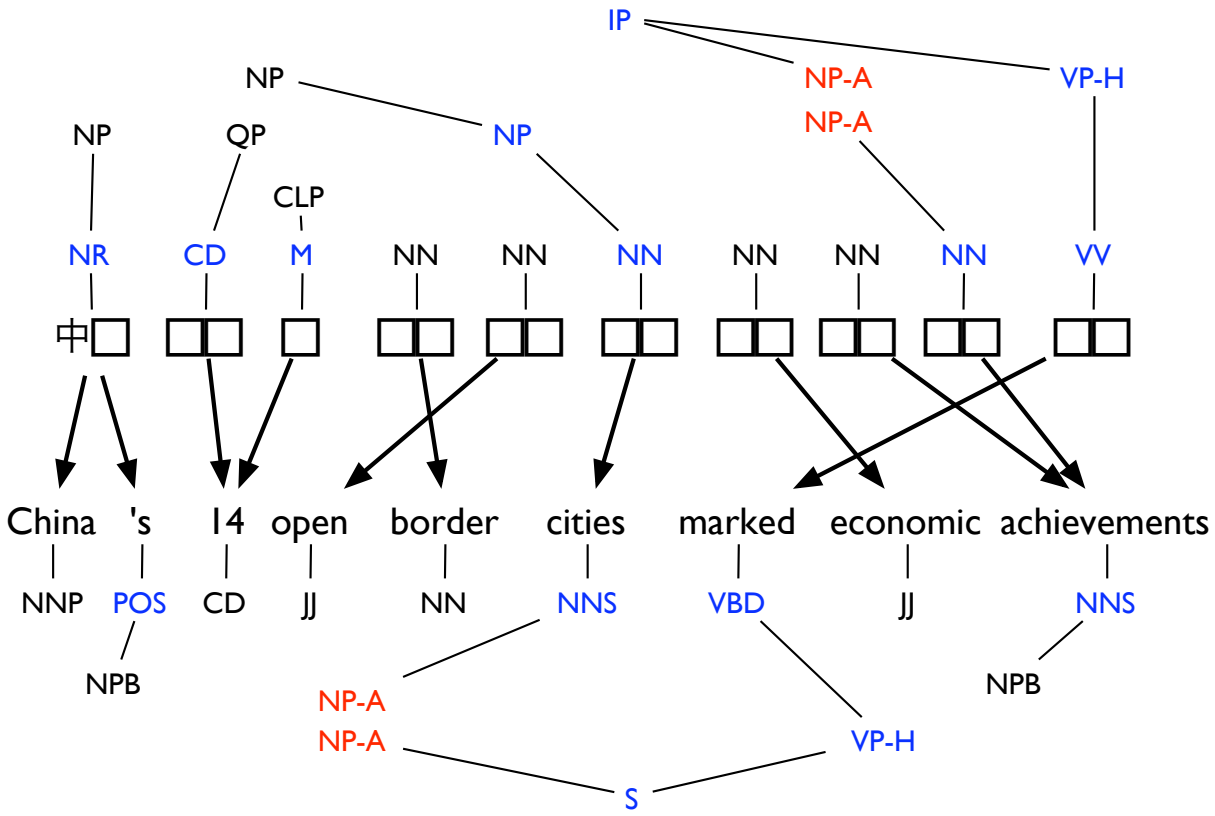
Tricky Syntax

- Parser probabilities over alignment template boundaries
- Hacking up the parse tree: a Markov assumption for tree alignments
- Using trees for scoring word alignments
- **Viren**: scoring constituents based on word alignments

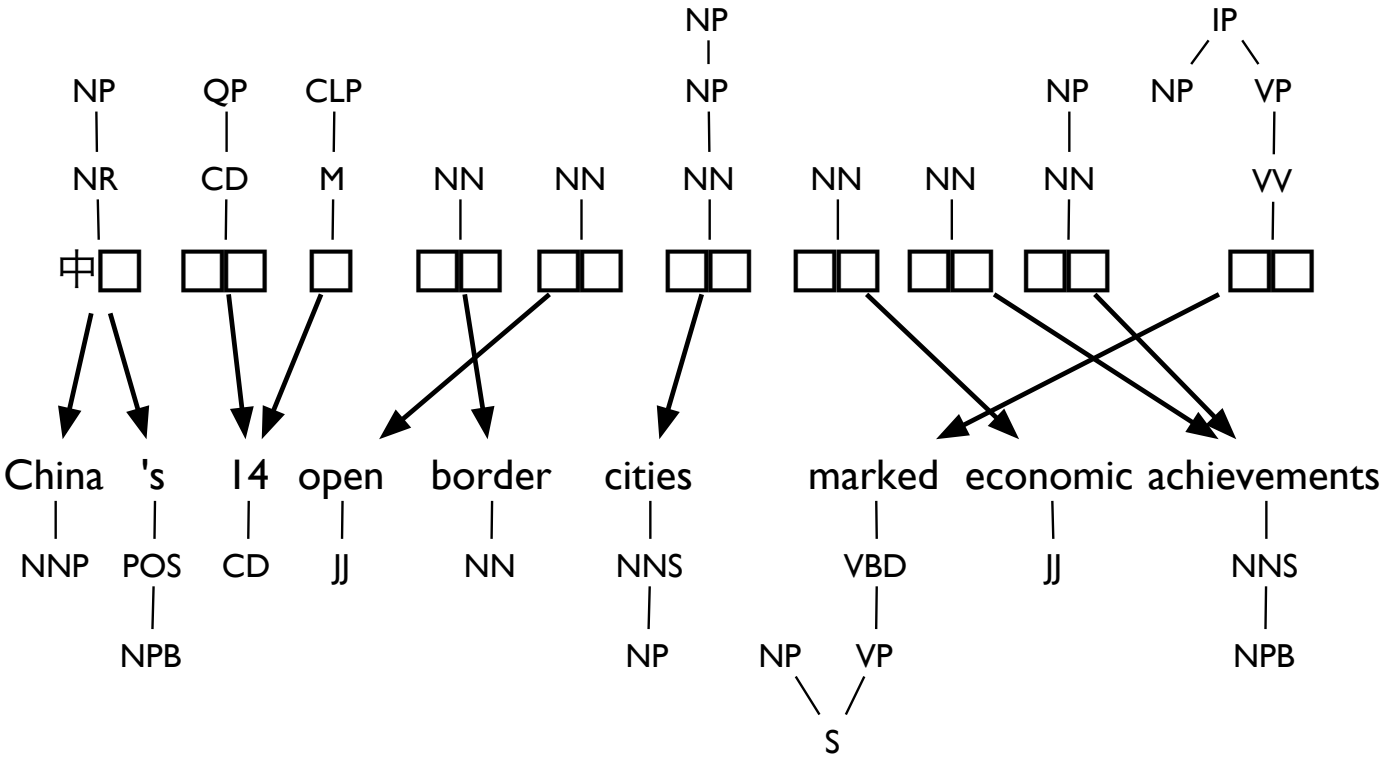
Parse trees have head and argument information



Parse trees have head and argument information



Each word gets a tree: elementary tree



Models over alignments of elementary trees

- Trained using SRI LM Toolkit using 60K aligned parse trees: 1300 elementary tree templates each for Chinese/English
- Unigram model over alignments: $\prod_i P(f_i, t_{f_i}, e_i, t_{e_i})$
- Conditional model: $\prod_i P(e_i, t_{e_i} \mid f_i, t_{f_i}) \times P(f_{i+1}, t_{f_{i+1}} \mid f_i, t_{f_i})$
- IBM Model 1 on aligned elementary trees (Kenji)

Results

Method	BLEU[%]
Baseline	31.6
AT Boundary Parser Prob	31.7
Tree-to-string without machete (covers only 273/993 for dev, 237/878 for test) Tree-to-string with machete	31.7 32.0
Model 1 on elementary trees Unigram model over aligned elementary trees Conditional bigram model over aligned elementary trees	31.6 31.7 31.9

Cross-Lingual Constituent Alignment Features

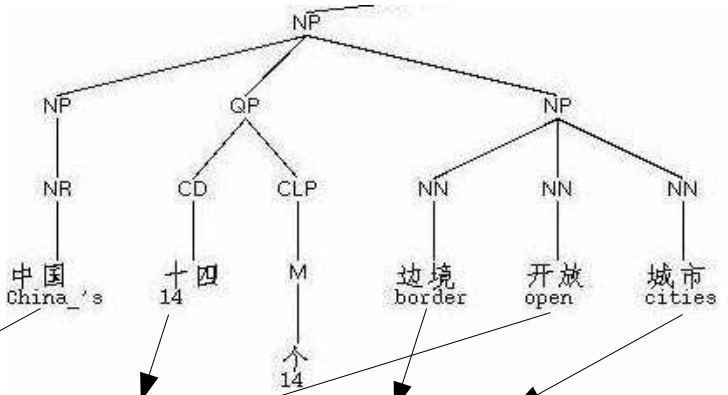
Motivation: parse trees are a potentially rich source of information – can we use simple metrics to compare Chinese and English trees?

Idea: penalize sentences in which syntactically *related* Chinese words are translated to syntactically *unrelated* English words

---

Implementation 1: Tree to String Penalty

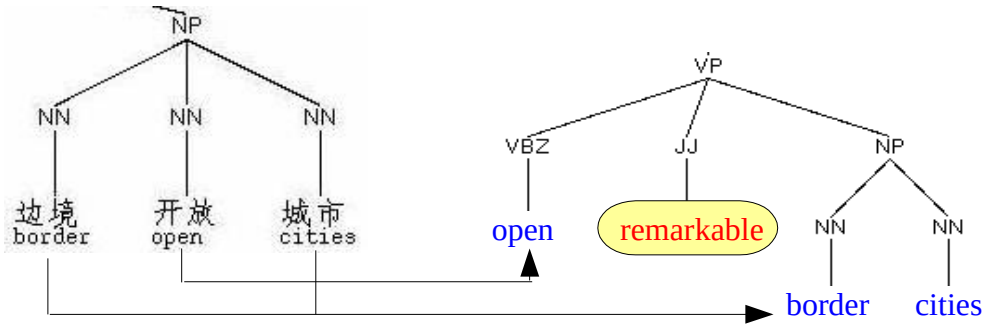
- Chinese parse tree
- English translation
- MT alignment information



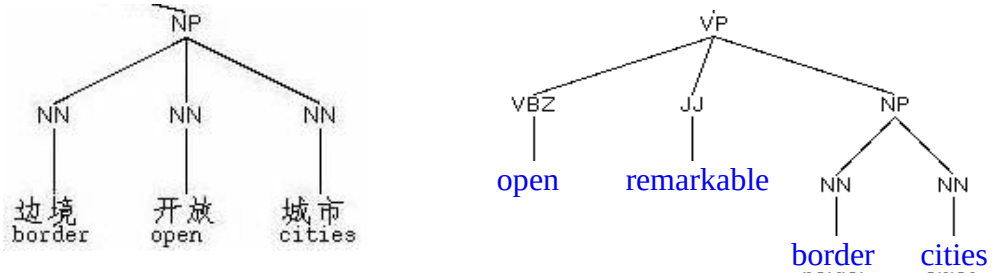
China's marked 14 open border cities economic achievements

Implementation 2: Tree to Tree Penalty

- Chinese parse tree
- English parse tree
- English translation
- MT alignment information



Cross-Lingual Constituent Alignment Features



Idea: learn relationships between Chinese and English parse trees

---

Implementation 3: Subtree Root Label Translation Probability

- Learn from parsed parallel training data (~50,000 sentences).

  e.g., p (label(english)=VP I label(chinese)=NP) = 0.019

- Multiply probabilities across Chinese constituents

- Additional features..

p (number of words in English subtree | number of words in Chinese subtree, *label*(CHINESE)=NP)

p (number of nodes in English subtree | number of nodes in Chinese subtree, *label*(CHINESE)=NP)

Cross-Lingual Constituent Alignment Features

Did we learn anything useful from the training data?

English Constituents

Chinese Constituents	CC	CONJP	RB	TO	NN
CC COORDINATING CONJUNCTION	.845	.023	.02	.026	0.008
	NP	VP	PP	JJ	NN
VP VERB PHRASE	.327	.327	.101	.036	.033
	NP	VP	S	PP	SBAR
IP INFLECTIONAL PHRASE	.48	.172	.20	.047	.026

Results

	BLEU[%]
Baseline	31.6
Tree to String Penalty	31.6
Tree to Tree Penalty	31.1
Root Label Transformation Probability	31.2
Number of Leafs in Source vs Target Prob.	31.7

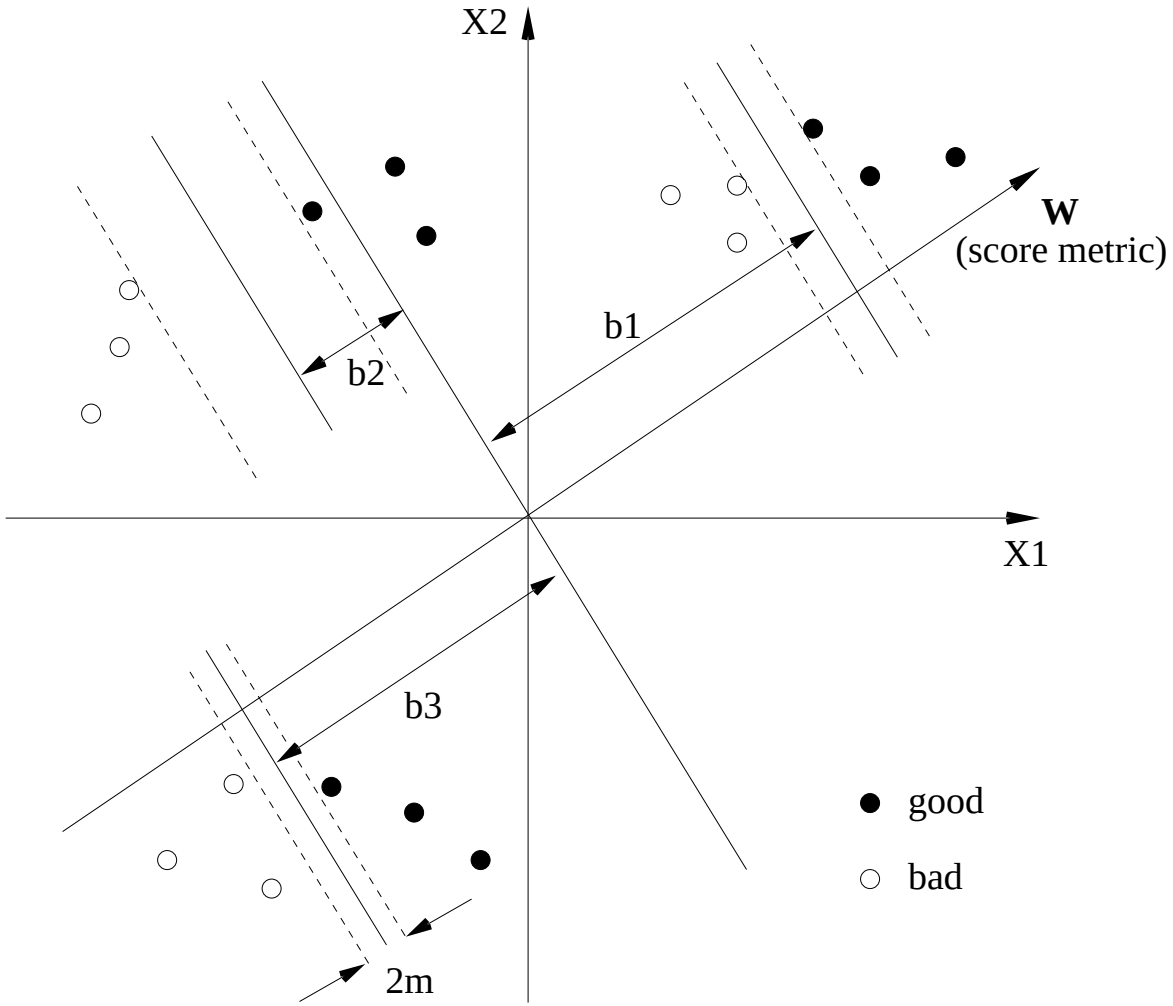
Outline

- Approach/Motivation (Franz)
- Syntactic Framework and Preprocessing (Dan)
- Features
  - Implicit Syntax (David)
  - Shallow Syntax (Alex, Zhen)
  - Deep Syntax (Kenji, Katherine)
  - Tricky Syntax (Anoop, Viren)
- Rescoring Techniques
  - Perceptron Learning (Libin)
  - Minimum Bayes Risk (Shankar)
- Feature Combination and Evaluation (Sanjeev)
- Student Proposals (Libin, Viren)
- Conclusion (Franz)

Reranking with Linear Classifiers

- Assumption: The *weight vector* of the separating hyperplane and the *score metrics* are in the same direction.
  distance to the hyperplane = quality of a translation
- Large dimensional space. Allowing the use of various features.
- Inseparable in a space with 12 baseline features
  - Do not use all the translations in the training.
  - λ-*trick*: a new dimension for each sample (Herbrich 02)
  - It still doesn't work. Each Chinese sentence requires a different *bias*.

Inseparable across different Chinese sentences



Variants of Perceptrons for Reranking

- **Alg 1**: To use pairwise translations as samples.
  - $class(T_{ij}, T_{ik}) \in \{-1, +1\}$, for two translations T_{ij} and T_{ik} of sentence i.
  - T_{ij} and T_{ik} in top and bottom $p\%$ of the n-best list.
  - Fast implementation.
  - It violates the *i.i.d.* assumption.
- **Alg 2**: Multi-Bias Perceptron
  - A weight vector $\mathbf{w}$ for all the English translations
  - A unique bias b_i for the translations of Chinese sentence i

The Multi-Bias Perceptron Algorithm with Margin

- **Input**: translations $(\mathbf{x}_{ij}, y_{ij})_{i=1..s,\ j=1..n}$, where $\mathbf{x}_{ij}$ in top or bottom $p\%$. **Output**: $\mathbf{w}$. **Internal Variable**: $b_1, b_2, ..., b_s$.
- **Updating**: on iteration $t+1$, *if* $y_{ij}(\mathbf{w}^t \cdot \mathbf{x}_{ij}) \leq \tau$, *then* $\mathbf{w}^{t+1} = \mathbf{w}^t + \eta y_{ij}\mathbf{x}_{ij},\ b_i^{t+1} = b_i^t + \eta y_{ij} R^2$.
- **Convergence**: $t \leq 2(s+1)((\frac{R}{m})^2 + \frac{\tau}{\eta m^2})$, if the training data is separable with margin m by some $\tilde{\mathbf{w}}, \tilde{b}_1, ..., \tilde{b}_s$, where $||\tilde{\mathbf{w}}|| = 1,\ |\tilde{b}_i| \leq R$.
- **Margin** and **margin based bound of expected risk**

Reranking results

- Multi-Bias Perceptron with the 30 best features.
  - BLEU [%] on test is 31.6 (Baseline BLEU [%] 30.1)
  - Discriminative learning does not fit our data set very well
- Using fragments of syntactic structures as features
  - Trained the dev set, tested on the test set.
  - The small number of the Chinese sentences in dev set limits the use of many useful features.
  - Preliminary results are promissing.

Using syntactic fragments as features

- **Experiment 1**: (Pair of) POS sequences in aligned templates
  - 30K features totally, avg. 31 active features each sample.
  - BLEU [%] on dev: 34.2, BLEU [%] on test: 30.9
- **Experiment 2**: Fragments in parse trees of the translations
  - 65K features totally, avg. 100 active features each sample.
  - BLEU [%] on dev: 30.3, BLEU [%] on test: 30.5
- To train on 100,000 Chinese sentences with 10 good and 10 bad translations, and employ more useful fragments of syntactic structures as features ... (Stay tuned for the proposal)

Outline

- Approach/Motivation
- Syntactic Framework and Preprocessing
- Features
  - Implicit Syntax
  - Shallow Syntax
  - Deep Syntax
  - Tricky Syntax
- Rescoring Techniques
  - Perceptron learning
  - Minimum Bayes Risk
- Feature Combination and Evaluation
- Conclusion

Minimum Bayes-Risk Decoders for Statistical Machine Translation Incorporating Syntactic Structure via Loss Functions

Aug 20, 2003

Shankar Kumar and Bill Byrne

Center for Language and Speech Processing
The Johns Hopkins University.

Minimum Bayes-Risk (MBR) decoding : An overview

MBR decoders have been shown to optimize performance in speech and language processing tasks using application dependent loss functions

MBR Decoding in ASR

- ASR can be viewed as a classification problem: $\delta(A) = W$.
- MAP decoding: $\hat{W} = \text{argmax}_{W \in \mathcal{W}} P(W|A)$
- Performance measure in ASR is usually Word Error Rate
  - Loss Function $L(W, W')$ is string-edit distance between W' and W
- MBR Speech Recognizer
  - Evaluate the expected loss of each hypothesis

$$E(W') = \sum_{W \in \mathcal{W}} l(W, W') P(W|A)$$

  - Consensus Decoding : Select the hypothesis which is most similar to other hypotheses

$$\delta_{MBR}(A) = \underset{W' \in \mathcal{W}}{\text{argmin}} \sum_{W \in \mathcal{W}} l(W, W') P(W|A)$$

Minimum Bayes-Risk (MBR) Decoders for SMT

- Goal: Design MBR decoders to optimize translation quality under various loss functions that measure translation performance
- A Discriminative Rescoring Approach
- Setup for MBR Rescoring in SMT
  - A Baseline Translation/Language Model to give the scores $P((e,a)|f)$
  - A set $\mathcal{E}$ of most likely translations of f. A hypothesis :(e',a',t')
  - A parse-tree t_f for the source sentence f
  - A Loss function $L((e,a,t),(e',a',t');t_f)$ that measures the quality of e' wrt e using information from word sequences (e,e'), alignments (a,a') and their parse-trees (t,t') and t_f.
- Decoder for Statistical MT

$$(\hat{e},\hat{a},\hat{t}) = \underset{\{e',a',t'\}\in\mathcal{E}}{\operatorname{argmin}} \sum_{\{e,a\}\in\mathcal{E}} L((e,a,t),(e',a',t');t_f)P((e,a)|f)$$

A hierarchy of loss functions for SMT

Tier 0: Loss Function without Parse Trees or Word Alignments

- Characteristics
  - $L((e, a, t), (e', a', t'))$ simplifies to $L(e, e')$
  - Examples: sentence-level BLEU score (smoothed), WER (Levenshtein distance), PER
  - Loss Function depends only on the word strings
- Examples

Produced: last year , the japanese commodity exports eight billion us dollars , representing a growth 10.9 % , the import of primary products one billion us dollars , up 4.4 % .
MBR-BLEU: last year , the japanese commodity exports eight billion us dollars , up 10.9 % than the previous year , the import of primary products one billion us dollars , up 4.4 % .

Produced: akayev spoke highly of the talks with premier li peng .

MBR-PER: akayev spoke highly of the prime minister li peng and talks .

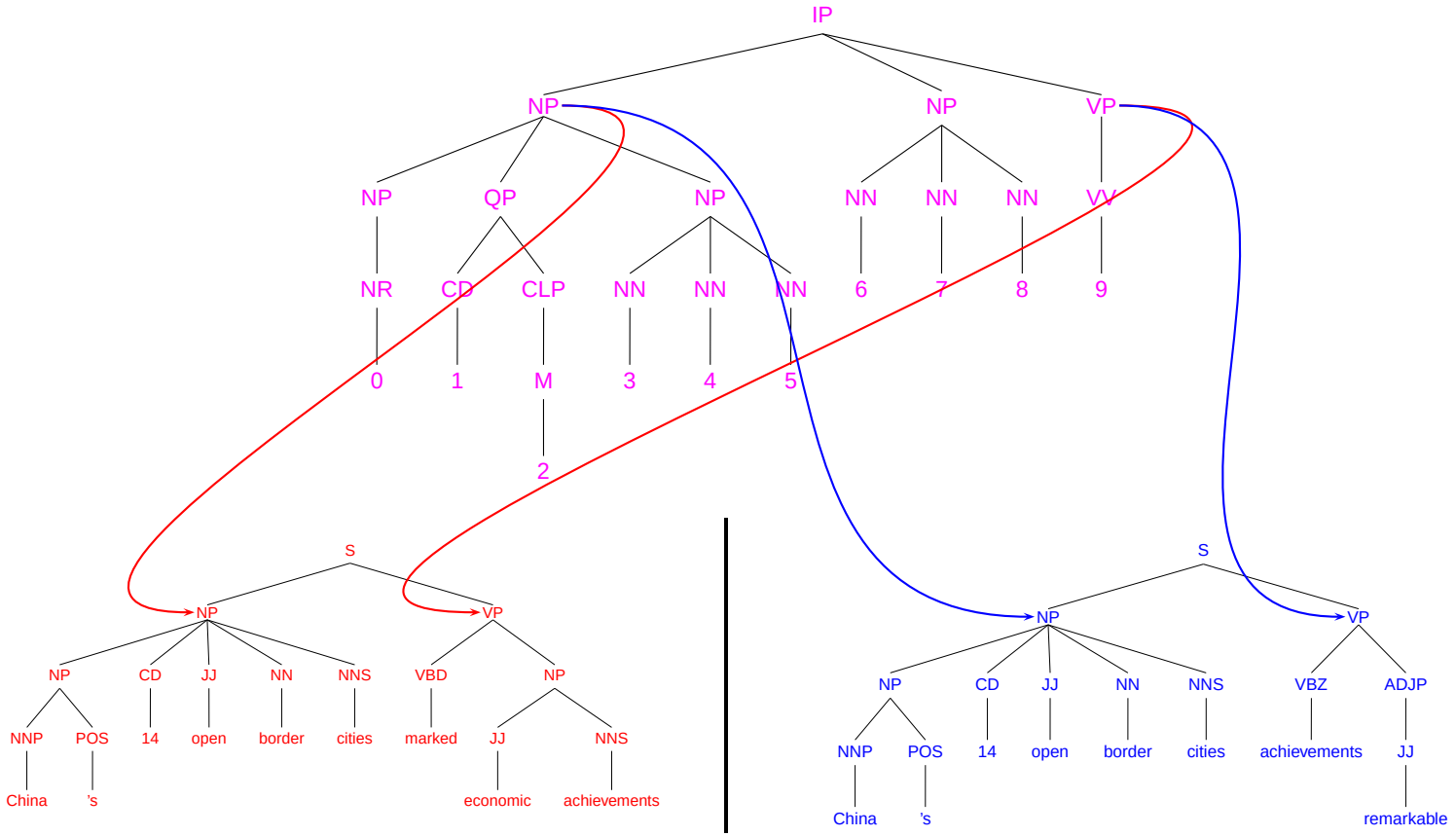
Tier 1: Loss Function with Target Language Parse Trees

- Characteristics
  - $L((e, a, t), (e', a', t'))$ simplifies to $L((e, t), (e', t'))$
  - *Tree Kernel* (Collins '02) computes the # of common subtrees between two parse-trees $h(t) \cdot h(t') = \sum_{i=1}^{d} h_i(t) h_i(t')$
    - The metric measures "structural" similarity between parses of any two translations
- In the output generated by the MBR decoder, only 14 were not parseable. In comparison, there were 29 sentences in the output of the baseline system that were not parseable.
- MBR decoder under this loss function improves the "grammaticality" of hypotheses.
- Example Output shows that translations become more fluent!

  Produced: this year , a leading role still prominent .

  MBR Hyp: this is a very important role .

Tier 2: Cross-Lingual Loss Function with Source & Target Language Parse Trees and Word-to-Word alignments



Node-to-Node alignments between English and Chinese Trees obtained using MT alignments (Viren)

Tier 2: Cross-Lingual Loss Function with Source & Target Language Parse Trees and Word-to-Word alignments

Joint work with Viren Jain

- Form of Loss Function:
  $L((e, a, t), (e', a', t'); t_f)$
- Node $n \in t_f$ maps (via word alignments) to nodes a_n in t and a'_n in t'
- t_{a_n} and $t'_{a'_n}$ are sub-trees in $t(t')$ rooted at nodes $a_n(a'_n)$
- Computing the loss function:
  $L((e, a, t), (e', a', t'); t_f) = \sum_{n \in t_f} L(t_{a_n}, t'_{a'_n})$
- 0/1 Loss function between sub-trees: $L(t, t')$
- Example Output shows better fluency and adequacy!
  Produced: at present , there is no organization claimed responsibility for the attack .
  MBR Hyp: at present , no organization claimed responsibility for the attack .

Performance of MBR Decoders

Test Set: 878 sentences, 1000-best lists

- Baseline from ISI

	Performance Metrics				
	BLEU (%)	mWER(%)	mPER (%)	mParseKernel	mBiTree Loss
Baseline	31.6	62.4	39.3	1002.2	30.42
MBR Decoder					
BLEU	31.9	62.5	39.2	1113.1	-
WER	31.8	61.8	38.8	1016.4	-
PER	31.7	62.2	38.5	835.5	-
Parse-Kernel	29.9	68.5	43.2	4478.1	-
BiTree Loss	31.1	61.6	39.2	840.5	30.19

- Improving the best system (as of 4^{th} week) trained using MAX-BLEU training with syntactic feature functions

	Performance Metrics			
	BLEU	mWER	mPER	NIST
Best System	32.9	61.7	38.3	9.40
MBR-BLEU	33.2	61.7	38.3	9.65

Conclusions and Future Work

- Discussion
  - MBR Decoding allows translation process to be tuned for specific loss functions
  - MBR gives further performance improvements on top of the best system trained under Max-BLEU Criterion
  - Development of Loss functions that make use of syntactic features from English and Chinese parse trees and word-to-word alignments
  - MBR-decoder is based on a sentence-level loss function
    - this highlights need for a sentence-level evaluation criterion
- Future Work
  - Design of Loss functions based on Tree-to-tree alignments
  - Constrain search space of MBR decoder to contain syntactically plausible translations
  - Using word graphs for MBR decoding

Are We There Yet?

- Approach/Motivation (Franz)
- Syntactic Framework and Preprocessing (Dan)
- Features
  - Implicit Syntax (**David**)
  - Shallow Syntax (**Alex**, **Zhen**)
  - Deep Syntax (Kenji, **Katherine**)
  - Tricky Syntax (Anoop, **Viren**)
- Rescoring Techniques
  - Perceptron learning (**Libin**)
  - Minimum Bayes Risk (**Shankar**)
- ➔ **Feature Combination and Evaluation** (Sanjeev)
- Proposals for Post-Workshop Research (**Viren** and **Libin**)
- Conclusion (Franz)

The Results So Far

- Many interesting ways to test the MT output for "grammaticality" were investigated this summer
  - More than 440 individual features
  - Each integrated individually with the baseline system
    - Trained on the NIST-Eval 2001 (**dev**) set to maximize BLEU
    - Evaluated on NIST-Eval 2002 (**test**) using BLEU
- This is arguably adequate for combining a few new features at a time with the baseline system
  - Results already presented for many feature functions
- Great infrastructure for rapid empirical investigation of ideas for improvement
  - Permitted parallel experiments by many individuals with an easy integration framework

For Those Who Like Numbers

Implicit Syntax

Feature Name	BLEU(%)
dev-ec-at	32.5
model1-ec	32.3
sentATRightContProb	32.0
sentATLeftMonotoneProb	31.9
missingWordFF	31.9
ATR4	31.9
ATR3	31.9
ATR0.2-4	31.9
wordpop2-idf	31.8
missingWordFF-0.01	31.8
missingWordFF-0.005	31.8
wordpop	31.7
wordpen_total	31.7
WordAlignDiagonal	31.6

Shallow Syntax

Feature Name	BLEU(%)
Scores_ProjectedTagLM	31.8
lchunkseq-ec	31.7
ProjectedTagLM_NULLlex_n…	31.7
chunkTagWRB_best	31.7
chunkTagVBP	31.7
chunkNotClosedPhrase	31.7
DecompTagFert10	31.7
chunkClosedPhrase	31.6
chunkPP	31.5
chunkseq-ec	31.4
lexchunk4	31.4
chunkTagNN	31.4
MapTagFert2	31.3
ChunkTagFert2	30.4

For Those Who Love Numbers

Deep Syntax

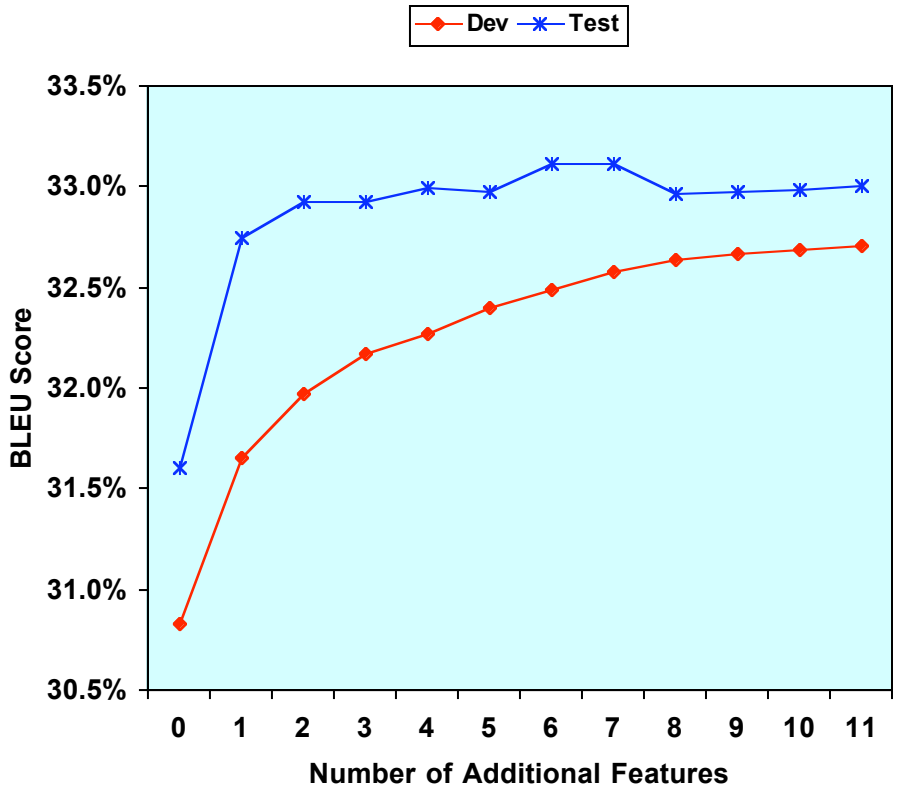
Feature Name	BLEU(%)
t2s-24-vit	31.7
t2s-24-tot	31.7
DepLMscore.CDN	31.7
VPalign	31.6
verbarg	31.6
rightBranch	31.6
ParseProb6	31.6
dependProject	31.6
treeDepth	31.5
DepLMratio2.or.failed	31.5
NPalign	31.4
diffVP	31.4
dependProjectTransBoth	31.4
parserScoreDivLM	31.0

Tricky Syntax

Feature Name	BLEU(%)
ualnLexAlignEtreesLM1	31.8
ualnLexAlignEtreesLM3	31.7
machete_labelmap	31.7
lexAlignEtreesLM1	31.7
unlexAlignEtreesLM3	31.6
Model1MacheteEC	31.6
Model1MacheteCE	31.6
ElmTreeModel1EC	31.6
ElmTreeModel1CE	31.6
ualnUnLexAlignEtreesLM3	31.5
lexAlignEtreesLM3	31.5
unlexAlignEtreesLM1	31.4
SentATParserOverlapProb	31.4
Tree2StrMachete	30.8

Integrating Multiple Features

- Use a greedy approach
  - Try each new feature on top of a state-of-the-art baseline system
  - Choose the one that improves BLEU the most on the dev set
  - Rebuild a new improved baseline
  - Iterate with remaining features … to point of diminishing returns
- After training models to maximize BLEU, one could also search using a maximum BLEU criterion
  - MBR improves BLEU on nearly-the-best system by another 0.3%.



Features that Improve BLEU

- Model-1 probability $P(e,f)$
  - Higher probability is better
- A verb heads the sentence
  - True is better
- Number of singular nouns used
  - More is worse
- "Fixed points" in multiple refs
  - Using fixed points is good
- Number of possessives used
  - More is better: China vs China's
- Number of times "that" is used
  - More is better (same for "a")
- Number of present-tense verbs that aren't 3rd-person-singular
  - More is better
- Difference in tree-depth of aligned constituent sub-trees
  - More is worse
- Matching constituents around coordinating conjunctions
  - More is better
- Number of Chinese content-words that go untranslated
  - More is worse
- Left monotonicity of alignment templates
  - Consistence in monotonicity is good; do as seen in training

Whole ≥ Sum of Parts?

- Many features with small improvements do not add up to a huge improvement
  - But they address different deficiencies of the MT output
  - Ought to be complementary
  - Need better machine learning techniques for feature combination?
- Many features which ought to improve readability of the output do not show an improvement in BLEU
  - The BLEU score of the "best" hypothesis among the 1000-best list is 105% of the BLEU score of the average human.
    - Even with BLEUn9r3!!!
  - Need to look into evaluation of a feature more closely, identify criteria other than BLEU which may warrant including a feature in re-ranking

Syntax for Statistical
Machine Translation

Presentation Outline

- Approach/Motivation (Franz)
- Syntactic Framework and Preprocessing (Dan)
- Features
  - Implicit Syntax (David)
  - Shallow Syntax (Alex, Zhen)
  - Deep Syntax (Kenji, Katherine)
  - Tricky Syntax (Anoop, Viren)
- Rescoring Techniques
  - Perceptron Learning (Libin)
  - Minimum Bayes Risk (Shankar)
- Feature Combination and Evaluation (Sanjeev)
- Student Proposals (Viren, Libin)
- Conclusions (Franz)

Syntax for Statistical
Machine Translation

Conclusions

Summary

- Starting point: best existing Chinese-English MT system as baseline
  - Phrase-based statistical machine translation
  - Alignment template system
  - Feature-function combination approach
- Goal: improve syntactic well-formedness of output by reranking of n-best lists

Summary

- 450(!) syntactic feature functions
  - Implicit syntax
  - Shallow syntax
  - Deep syntax
  - Tricky syntax
- Typical improvements:
  - Implicit > Shallow ~ Tricky > Deep
- End-to-End BLEU score improvement
  - From 31.6 % to 33.2 % (ok!)
  - (Unfortunately) main contribution from IBM Model 1 (… semantics??? :-)

Summary

- Created Resources
  - N-best lists
    - 16384-best translations for 5780 dev, 878 test, 929 blind test
  - Parses of first 1000 best sentences
  - POS tags, Chunk parses
  - Training sentence parses: 1.3 million English sentences, 50,000 Chinese sentences
  - 450 feature function files
  - …
  - We work on making resources available…
- Useful also for: language modeling research, machine learning research

Lessons Learned

- Exploited/experienced the limits/weaknesses of BLEU score (and maximum BLEU training)
  - Needed: better automatic sentence level metrics
  - Research proposal: Viren
- Current maximum BLEU training does not scale up to hundreds of features
  - Needed: refined discriminative training techniques
  - Research proposal: Libin

Lessons Learned

- Off-the-shelf parsers for noisy MT output are problematic
- We would like:
  - Parsers that don't hallucinate structures that are not there
  - Syntactic analysis tools that indicate **confidence** in the analysis
  - Research area: Bilingual syntactic analysis

Lessons Learned

- Very productive work environment:
  - Combining feature functions
  - Working on n-best lists
- Advantages:
  - Different people can work independently on the same system
  - Arbitrary dependencies can be exploited
    - E.g. no constraint to left-to-right dependencies

WS 2003

Syntax for Statistical
Machine Translation

WS 2003
Syntax for Statistical
Machine Translation

WS 2003
Syntax for Statistical
Machine Translation

WS 2003
Syntax for Statistical
Machine Translation

WS 2003
Syntax for Statistical
Machine Translation

WS 2003
Syntax for Statistical
Machine Translation