



Statistical Machine Translation and Speech-to-Speech Translation

Bowen Zhou

IBM T. J. Watson Research Center

zhou@us.ibm.com

April 6th 2010
Columbia University

Outline

- Introduction to S2S: An overview of IBM MASTOR
- DARPA TRANSTAC Program: Bring S2S to real world
 - Mission and the progress
 - How S2S is evaluated?
 - Video demo: Iraqi Arabic-English S2S on Tablet PC
- SMT and S2S Technologies
 - Recap: Word alignment and phrase-based SMT
 - Multiple graph-based phrasal SMT using finite state
 - Real-time speech recognition & text-to-speech synthesis (no discussion today)
- Formal syntax-based SMT and SCFG
 - Overview of syntax-based SMT and SCFG
 - Efficiently integrating linguistic syntax information
- Case study: SMT systems used in IBM S2S
 - Live demo: Pashto-English S2S on Google Nexus One (Android)

IBM Speech-to-Speech Translator

A Real-time and Portable Solution to Mitigate Language Barriers

- Automatic Universal Translator
 - The dream of scientists for decades – most challenging research
- MASTOR (Multilingual Automatic Speech-to-Speech TranslatOR)
 - Attempting to facilitate cross-lingual oral communication for designed domains
- Challenges
 - Background noise in the field
 - Accented speech and various dialects
 - Ubiquitous ambiguity presented in speech and language, etc
 - Conversational spontaneous speech: disfluent & ungrammatical input
 - Real-time performance on low-end mobile computational platforms

Outline

- Introduction to S2S: An overview of IBM MASTOR
- DARPA TRANSTAC Program: Bring S2S to real world
 - Mission and the progress
 - Video demo: Iraqi Arabic-English S2S on Tablet PC
 - How S2S is evaluated?
- SMT and S2S Technologies
 - Real-time speech recognition & text-to-speech synthesis (no discussion today)
 - Recap: Word alignment and phrase-based SMT
 - Multiple graph-based phrasal SMT using finite state
- Formal syntax-based SMT and SCFG
 - Overview of syntax-based SMT and SCFT
 - Efficiently integrating linguistic syntax information
 - Effective learning of SCFG rules
- Case study: SMT systems used in IBM S2S
 - Demo: Pashto-English S2S on Smart Phones

DARPA TRASTAC

- Spoken language communication & translation system for Tactical Use
- Missions:
 - Demonstrate capabilities to rapidly develop and field two-way translation systems
 - Enable speakers of different languages to spontaneously communicate with one another in real-world tactical situations.
- Program started in 2005/2006, as a continuation of DARPA Babylon/CAST:
 - Phase I (05/06), II (06/07), III (07/08): focused on Iraqi-English
- Phase IV (08/09): added colloquial Afghanistan languages to the portfolio
 - Dari-English
 - Pashto-English
- Prototypes for both Dari & Pashto were built within the 6 months of 2009
 - Demo later in this talk

How S2S is evaluated?

- S2S/SMT: there is often no ground truth in speech translation
- Evaluations led by NIST and MITRE using multiple dimensional matrices
 - Offline: component evaluation
 - ASR WER
 - Translation accuracy (BLEU, TER, METOR, and human judge)
 - TTS (human judge and WER)
 - Low-level concept transfer odds
 - Live: simulated real world scenarios between monolingual users
 - Task completion rate: accuracy and speed
 - High-level concept transfer rate
 - Number of attempts per success
 - Time to retrieve a concept
 - Post-live-session anonymous user feedback/questionnaires
 - Both English/foreign users provide scaled feedback on satisfaction
 - Performance, usability, eyes-free, mobility, form factors etc
 - Commentary on overall performance

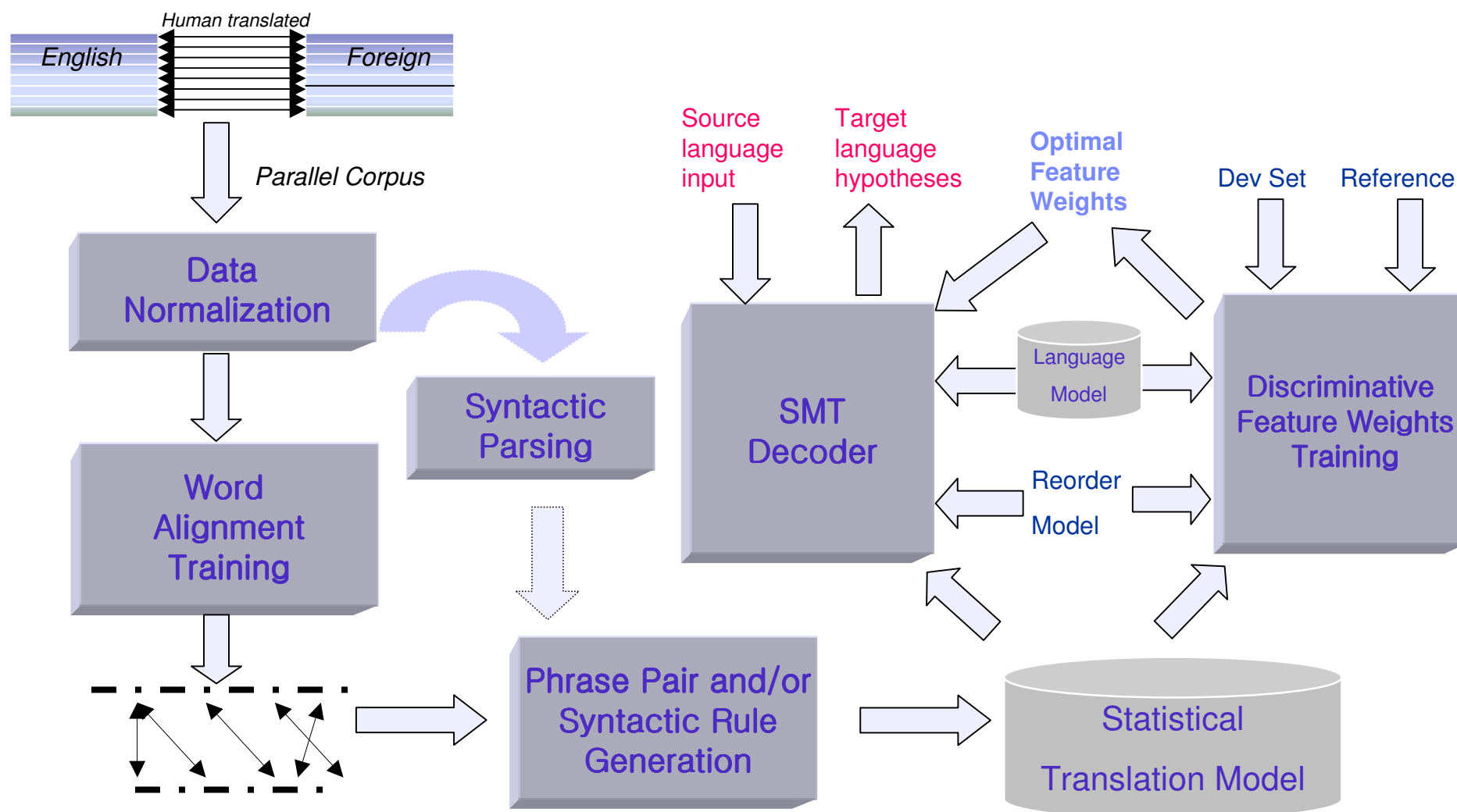


Iraqi Arabic-English Video Demo

Outline

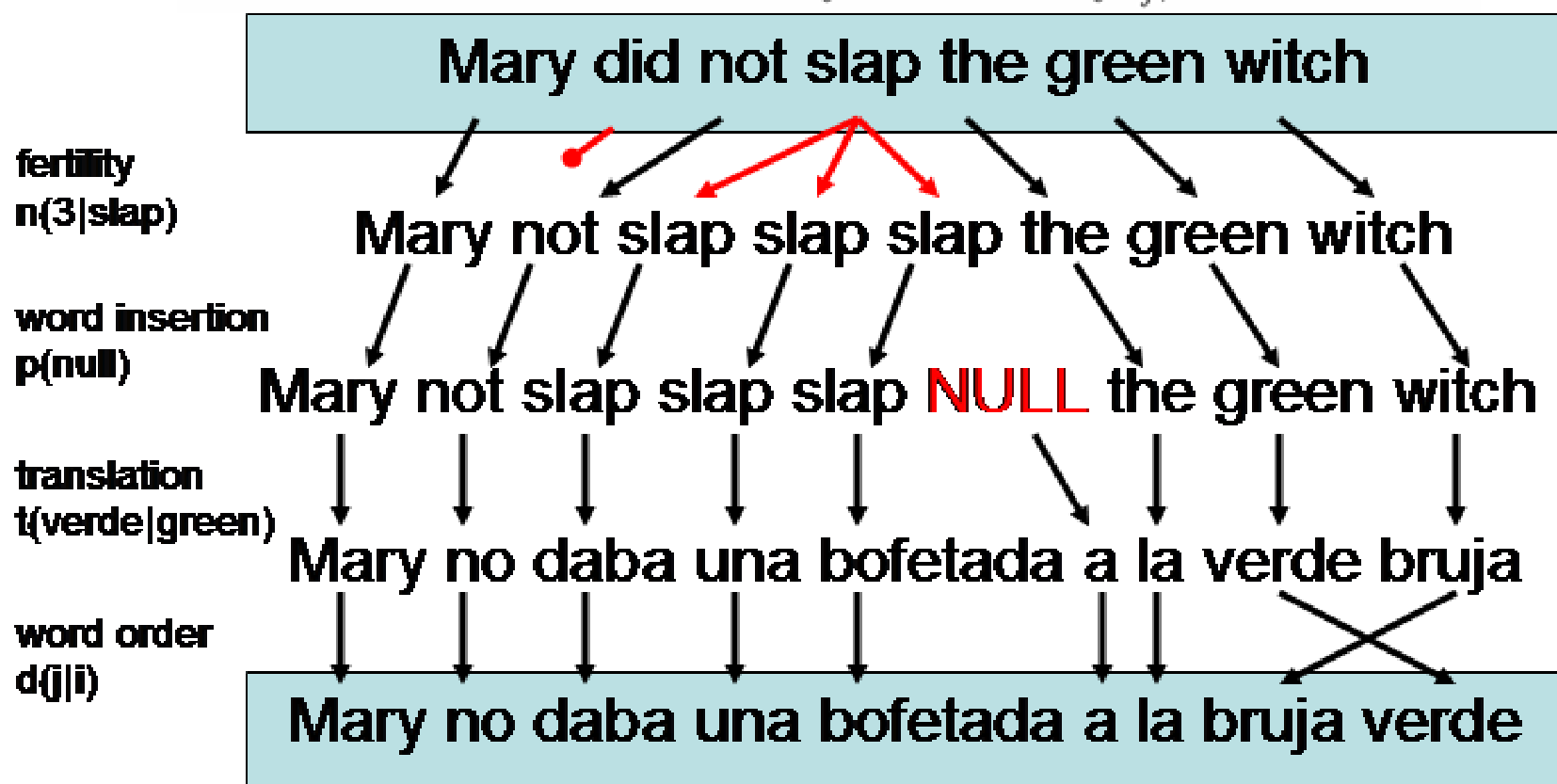
- Introduction to S2S: An overview of IBM MASTOR
- DARPA TRANSTAC Program: Bring S2S to real world
 - Mission and the progress
 - How S2S is evaluated?
 - Video demo: Iraqi Arabic-English S2S on Tablet PC
- SMT and S2S Technologies
 - Recap: Word alignment and phrase-based SMT
 - Multiple graph-based phrasal SMT using finite state
 - Real-time speech recognition & text-to-speech synthesis (no discussion today)
- Formal syntax-based SMT and SCFG
 - Overview of syntax-based SMT and SCFG
 - Efficiently integrating linguistic syntax information
- Case study: SMT systems used in IBM S2S
 - Live demo: Pashto-English S2S on Google Nexus One (Android)

A Typical Pipeline of SMT

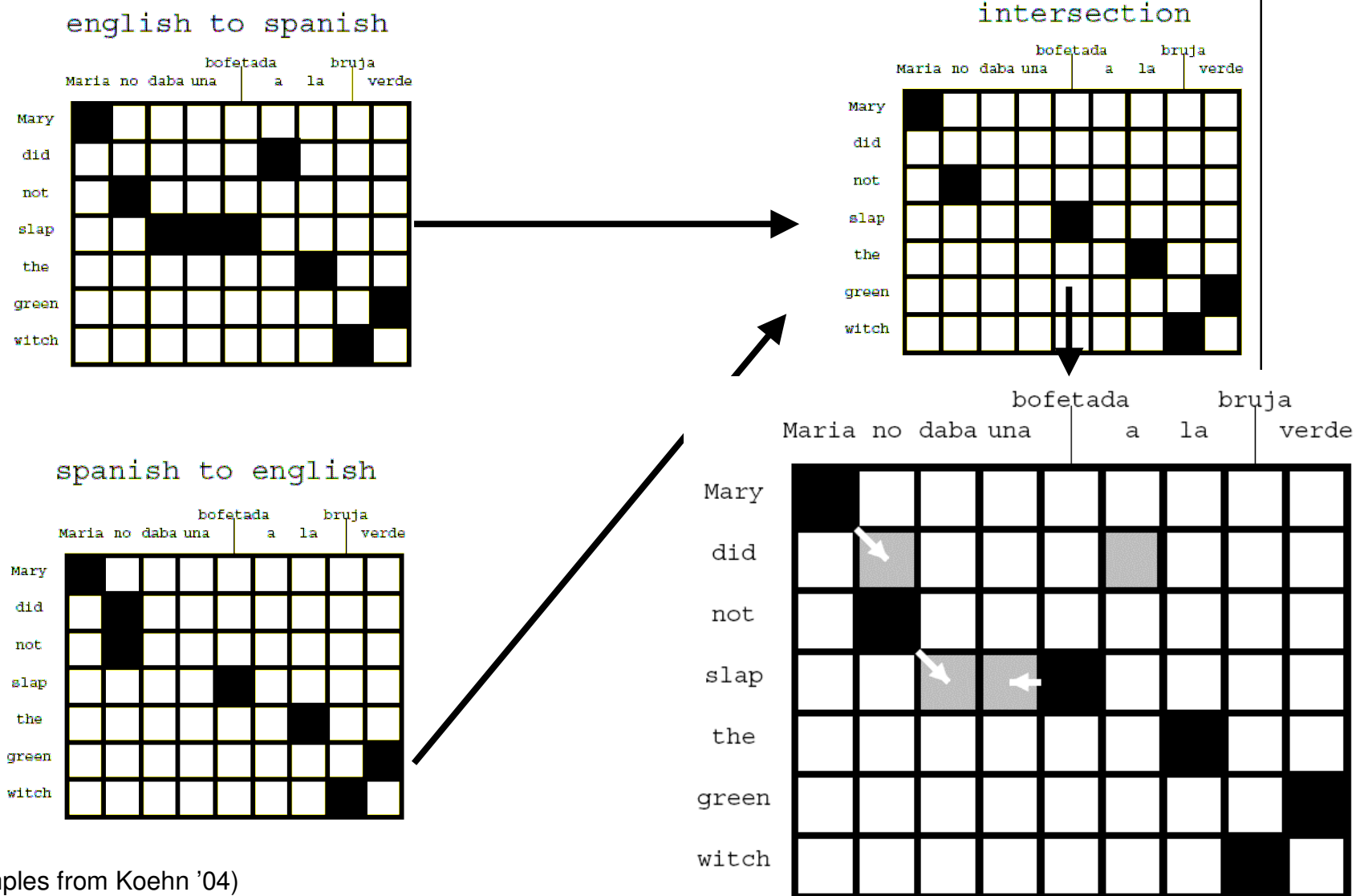


How word alignment is learnt: IBM Model 4 & EM (Brown'93)

$$P(a, f | e) = \prod_{i=0}^l n(\phi_i | e_i) p^{\phi_0} \prod_{j=1}^m t(f_j | e_{a_j}) \prod_{j:a_j \neq 0}^m d(j | a_j, l, m)$$

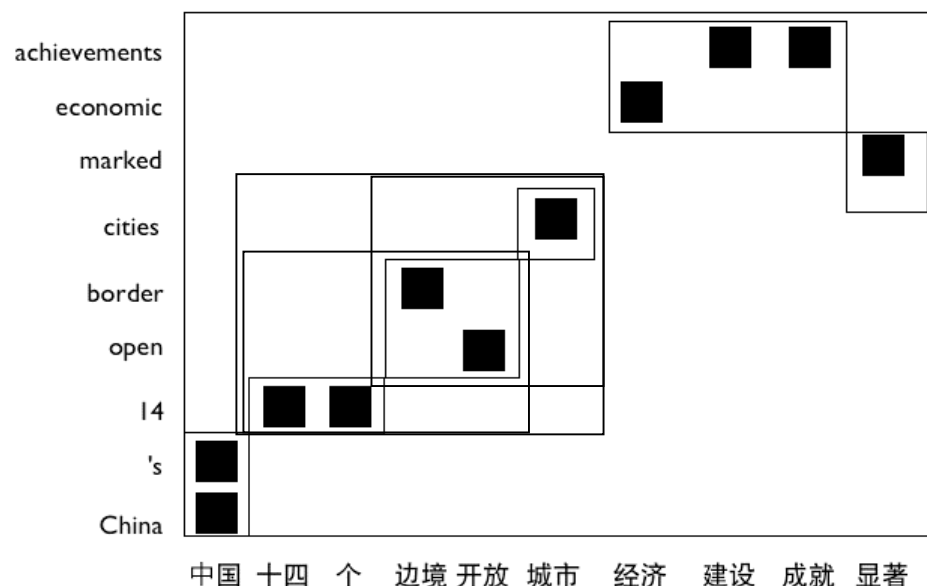


Alignment Symmetry & Refinement



(Examples from Koehn '04)

Phrase Translation: Putting More Context into Consideration



经济 || economy || 0.31
 经济 || economic || 0.63
 中国大陆 || chinese mainland || 0.25
 中国大陆 || mainland china || 0.75
 开放 || open || 1.00
 边境 开放 || border open || 1.00

- Enumerate all phrase pairs w.r.t. word alignments boundary [Och et al, '99]
- A phrase is just a n-gram, not necessarily in linguistic sense
 - Every rectangle box in the above picture is a phrase pair
- Estimate phrase to phrase translation table by relative frequency
- Others (lexicalized distortion models, word-to-word translation model, etc) can also be estimated from alignment
- Simple yet most widely-used SMT techniques

Decoding: Phrase-based Statistical Machine Translation

- Phrase-based: state-of-the-art MT performance for many tasks

$$\begin{aligned}\hat{e} &= \arg \max_{e_1^E} \Pr(e_1^E \mid f_1^J) \\ &= \arg \max_{K, \bar{e}_1^K} \Pr(\bar{e}_1^K \mid \bar{f}_1^K)\end{aligned}$$

- Log-linear model combination: language model, length bonus etc.
- Decoding: Stack (A^*) beam search (Och'04, Koehn'04) is commonly used
 - Moses: widely used open source toolkit
 - Many other implementations around the world
- Alternatively, the decoding can be done by WFST techniques
 - No consideration of recursion or hierarchical structures in languages, phrase-based SMT is essentially a finite state problem!

Formulate Phrase-based SMT in WFST

- Pros of applying Weighted Finite State Transducer (WFST):
 - Mature algorithms for operations and optimization (Mohri'02): compact models
 - Incorporate multiple sources, multi-level & heterogeneous statistical knowledge
 - Better integration with uncertainties; Suitable for S2S translation

- Early studies: Knight'98, Bangalore'01

- General framework of using WFST for translation

$$\hat{e} = \text{best} - \text{path} (s = I \circ M_1 \circ M_2 \circ \dots \circ M_m)$$

- A WFST Implementation for Phrase-based SMT (Kumar'05)

$$S = I \circ U \circ P \circ Y \circ T \circ L$$

- WFST for constrained Phrase-based translation (Zhou'05)

$$S = I \circ M \circ (N \circ G \circ T)^{-1} \circ L$$

- In above cases, decoding is performed with a general purpose FSM Toolkit

Issues with Previous Approaches

- Ideal case of WFST approach:
 - Compute entire H offline: perform all composition operations ahead of time.
 - Determinization & minimization: further reduces computation
 - At translation time: only need to do *best-path* ($I \circ H$)
- In reality, very difficult to do full offline composition or optimization :
 - The nondeterministic nature of the phrase translation transducer interacts poorly with the LM;
 - H is of intractable size (even for inf. memory);
 - $I \circ H$ expensive: even w/ on-the-fly composition followed by beam-pruning search
 - Reordering is a big challenge, making search NP-hard, and H non finite-state
- In previous work, compositions have all been done online for given input
 - Slow speed (<5 words/second) (kumar'05),
 - Needs multiple GB memory at runtime

A Multiple-Graph based Approach for Phrasal SMT

$$\Pr(e_1^E | f_1^J) \Pr(e_1^E) \approx \max_{\bar{f}_1^K} \{ \\ P(K | f_1^J) P(\bar{f}_1^K | K, f_1^J) \times \\ P(\bar{e}_1^K | \bar{f}_1^K, K, f_1^J) \times \\ P(e_1^E | \bar{e}_1^K, \bar{f}_1^K, K, f_1^J) \times \\ P(e_1^E) \}$$

$$S = I \circ P \circ T \circ W \circ L$$

P: source language segmentation

T: phrasal translation

W: target language phrase-to-word

L: target language model

I: input with dynamic uncertainty (reorder, ASR, segmentation, morphology etc)

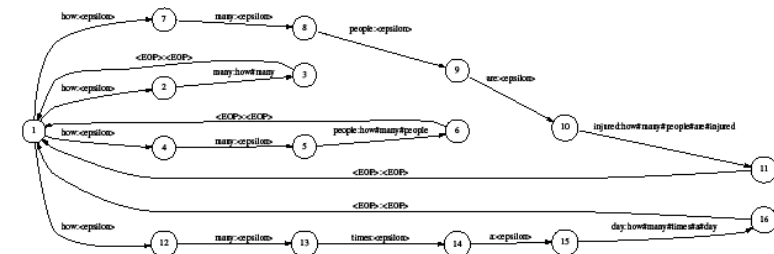
- Decompose the problem as a chain of conditional probabilities (see left)
- Each represented by WFST: models the relationships between their inputs/ outputs.
- Compose & optimize the static graph as much as possible
- Encode reordering into a separate dynamically expanded graph that can combine other uncertainty on-the-fly
- A dedicated decoder needed for efficient decoding
 - Dynamic composition of multiple graphs
 - Multi-dimensional synchronous Viterbi search

Determinize Phrase Segmentation Transducer P

- Mapping word sequences to all “acceptable” phrase sequences:

- How many people are injured

- How
 - How many
 - How many people
 - many people are
 - people are injured
 - ...

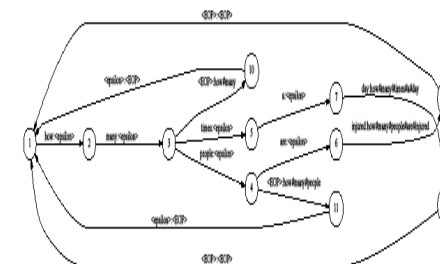


- Determinization is crucial here:

- Reduce the size of this machine,
 - Making following compositions possible

- Non-determinizability is caused by overlap between phrases,

- word sequences segmented into phrases in multiple nested ways
 - phrase identity may not be determined **until** entire sentence is observed
 - such **unbounded** delays make *P* non-determinizable



- Our Solution:

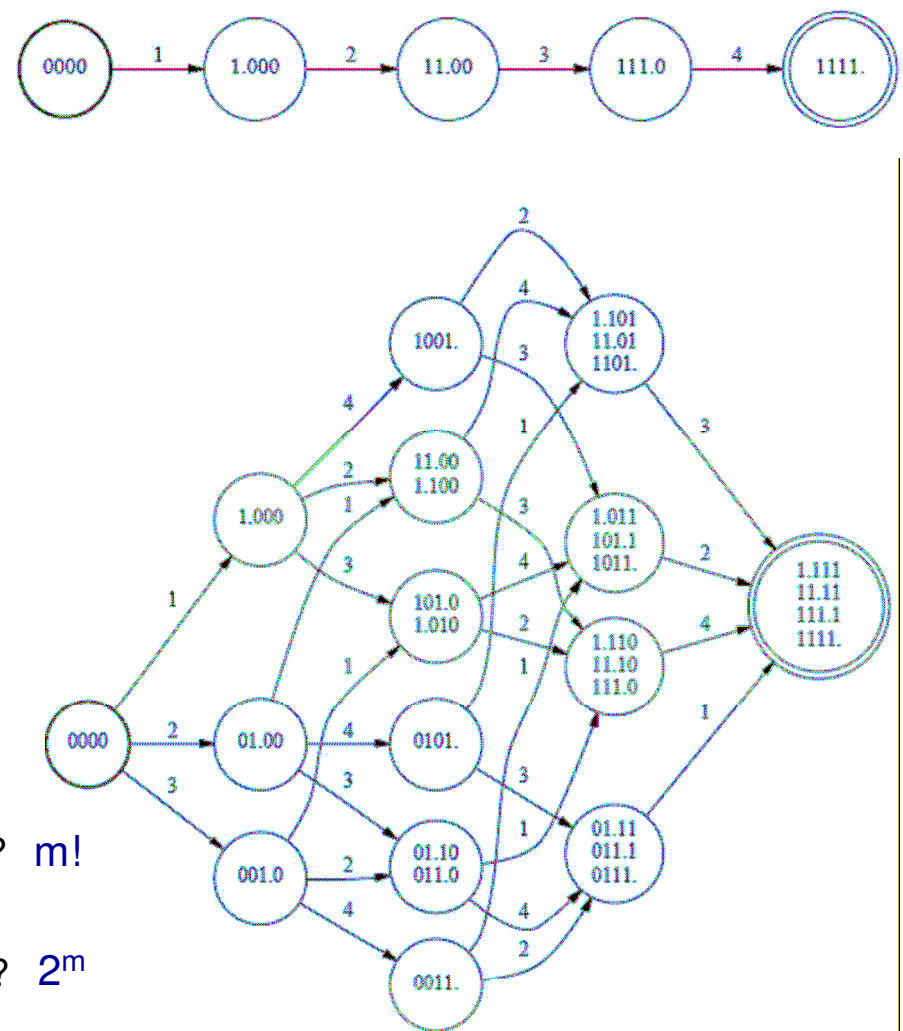
- introduce an auxiliary symbol, *EOP*,
 - Marking the end of each distinct source phrase.

Other Component Transducers and Offline Optimization

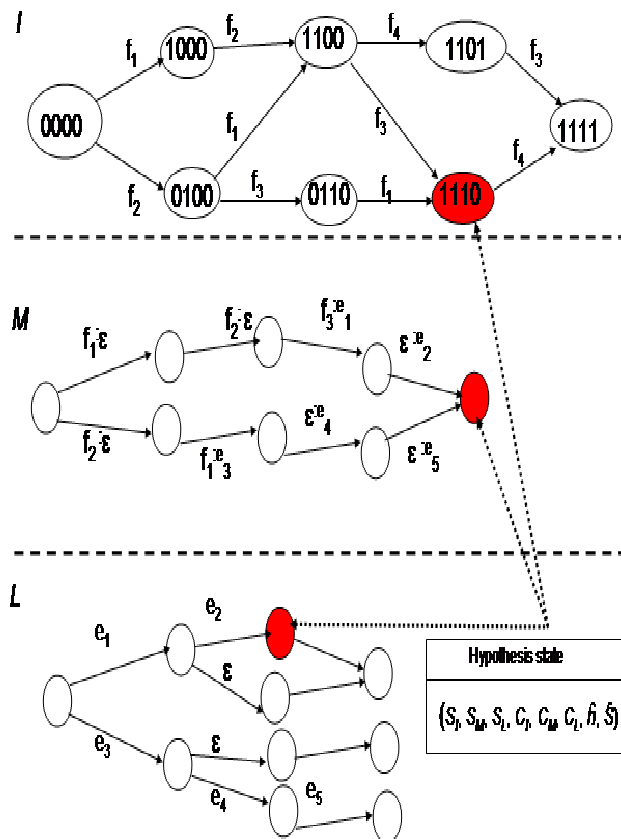
- T: maps source phrases to target phrases.
 - One-state machine: every arc corresponds to a phrase pair
 - Weights determined by log-linear of multiple models
 - phrasal translation
 - word lexicons
 - phrase penalty etc
 - One arc maps *EOP* to itself w/o cost
- W: maps target phrase to words
 - A deterministic machine
- L: Back-off N-gram target language model
 - a weighted acceptor assigns probabilities to target word sequences
 - Mostly determinized
- $H = P \circ T \circ W \circ L$, not computable offline!
- Solution: Separate H as: $H = M \circ L$
- $M = \text{Min}(\text{Min}(\text{Det}(P) \circ T) \circ W)$
 - tropical semiring for Viterbi compability
 - Further optimization w/ minimization
- M can be computed fully offline due to the determinizability of P
 - Millions of states
 - Tens of millions arcs

Lexicalized Word Reordering in Graph (Zhou et al 08)

- Reordering graph embedded in decoding
 - To incorporate ordering ambiguity
 - Bit-vector to indicate covering status
 - 000..0 indicates that no words translated
 - 111..1 indicates that all finished
 - Reordering graph (topology & weights) controlled by reordering constraints & models
 - Maximum window (4), maximum skip (2)
 - Reordering graph is determinized and minimized *on-the-fly* during decoding
 - Reordering cost is added into log-linear models
- Similar implementation can incorporate speech recognition (ASR lattice) ambiguity for S2S
- Quiz: when there is no reorder constraint
 - For a m word input, how many reorder options? $m!$
 - How many states needed in this reorder graph? 2^m



Folsom: Multiple-graph SMT (Zhou et al., 06;07;08)



- SMT built upon multi weighted finite state graphs:

- **Input graph I** : model uncertainty in inputs

- Reordering, ASR ambiguity, morphological, segmentation, and/or their combinations
 - Statically or lazily constructed

- **Translation graph M** : encode phrasal translations

- **Target graph L** : measure target acceptability

- Decoder: *Best-path* ($I \circ M \circ L$)

- Sync-Viterbi search on each layer & joint graph
 - 7-tuple search hypothesis organized as a prefix tree; merge hyp. as early as possible

- WFST perspective: can be viewed as optimized implementation of *combined* WFST operations:

- Lazy **multiple** composition
 - Lazy determinization and minimization
 - Viterbi search

- Use lexicalized reordering models (Zhou et al., 08)

Outline

- Introduction to S2S: An overview of IBM MASTOR
- DARPA TRANSTAC Program: Bring S2S to real world
 - Mission and the progress
 - How S2S is evaluated?
 - Video demo: Iraqi Arabic-English S2S on Tablet PC
- SMT and S2S Technologies
 - Recap: Word alignment and phrase-based SMT
 - Multiple graph-based phrasal SMT using finite state
 - Real-time speech recognition & text-to-speech synthesis (no discussion today)
- Formal syntax-based SMT and SCFG
 - Overview of syntax-based SMT and SCFG
 - Efficiently integrating linguistic syntax information
- Case study: SMT systems used in IBM S2S
 - Live demo: Pashto-English S2S on Google Nexus One (Android)

Putting Syntax into Translation Model: Introduction

■ Syntax analysis

- Parse the source and/or target sentence (string) into a structured representation (tree)
- trees reveal translation patterns that are more generalizable than what string can offer

■ Syntax-based translation:

- Improved performance over state-of-the-art phrase-based (Chiang, 2005; Galley et al., 2004; Liu et al., 2006)

■ One of the hottest topics in SMT/NLP fields

What's both ~~common~~ with different ~~models~~ & syntax-based SMT?

■ *Linguistic* syntax-based:

- Explicitly utilizes structures defined over linguistic theory & human annotations (e.g., Penn Treebank)
- SCFG rules (define later) derived from parallel corpus guided by parsing on at least one side of the corpus:
 - *tree-to-string, string-to-tree, tree-to-tree...*
- Examples: (Yamada and Knight, 01), (Galley et al., 04), (Huang, 07) etc

■ *Formal* syntax-based:

- Based on hierarchical structures of natural language
- No annotation needed
- Synchronous grammars extracted w/o any usage of linguistic knowledge
- A good fit for low-resource spoken language
- Examples: ITG (Wu, 97) & hierarchical models (Chiang, 07)

■ Will linguistic theory & annotations help formal syntax-based models?

SCFG: (Probabilistic) Synchronous Context-Free Grammar

- A synchronous rewriting system generating source & target side simultaneously, based on PCFG
- Each production (i.e., rule) rewrites a nonterminal into a pair of strings
 - Include both terminals & nonterminals in both languages,
 - One-to-one correspondence between nonterminal occurrences
- Explore hierarchical structure & utilize a unified nonterminal X in grammar, which is replaceable with any other X

$$X \rightarrow \langle \gamma, \alpha, \sim \rangle,$$

\sim : one-to-one correspondence indicated by co-indices on both sides.

- Examples: English-to-Chinese production rules

$$\begin{aligned} X &\rightarrow \langle X_1 \text{enjoy reading } X_2, \\ &X_1 \text{xihuan}(\text{enjoy}) \text{yuedu}(\text{reading}) X_2 \rangle \\ X &\rightarrow \langle X_1 \text{enjoy reading } X_2, \\ &X_1 \text{xihuan}(\text{enjoy}) X_2 \text{yuedu}(\text{reading}) \rangle \end{aligned}$$


Why does it help?

- Syntax-based translation:
 - Observed improved performance over state-of-the-art phrase-based (Chiang05; Galley et al.04; Liu et al. 06)
- Engagement of synchronous context-free grammars (SCFG): enhanced generative capacity through recursive replacement
- Phrase-based → syntax-based: one level higher in [Chomsky Hierarchy](#) more principled long-distance reordering
 - Regular language (pair) → Context-free language (pair)
 - Finite-state machinery (FSM) → Push-down automata
- Phrasal translation structures to handle local fluency (borrowed from phrase-based models, Och04)

Example of SCFG Learning

GER: die herausforderung besteht darin diese systeme zu den besten der welt zu machen

ENG: the challenge is to make the system the very best



German-English: Phrasal Rule Extraction

die herausforderung besteht darin diese systeme zu den besten der welt zu machen

Long distance reorderings require
jumping over untranslated text

... and back

the challenge is to make the system the very best

X → <die herausforderung, the challenge>

X → <besteht darin, is>

X → <zu machen, to make>

X → <diese systeme, the system>

X → <zu den besten der welt, the very best>

Rules have probabilities, the
decoder searches for the most
probable translation

Example of German-English Non-terminal rule extraction

die herausforderung besteht darin diese systeme zu den besten der welt zu machen

the challenge is to make the system the very best

r1: $X \rightarrow \langle \text{machen, make} \rangle$

r2: $X \rightarrow \langle \text{die herausforderung, the challenge} \rangle$

r3: $X \rightarrow \langle \text{diese systeme, the system} \rangle$

r4: $X \rightarrow \langle \text{zu den besten der welt, the very best} \rangle$

r5: $X \rightarrow \langle \text{besteht darin } X_1 \text{ zu } X_2, \text{is to } X_2 X_1 \rangle$

The reordering is captured by this rule

glue: $X \rightarrow \langle X_1 X_2, X_1 X_2 \rangle$

SCFG based SMT

- All rules paired with statistical parameters (i.e., Probabilistic SCFG); combined with other features using a log-linear framework
- Decoding:

Find the best translation using SCFG for an input f



Search for the optimal derivation on source and target sides

- Optimal derivation D : maximizes following log-linear models over all possible derivations:

$$P(D) \propto P_{LM}(e)^{\lambda_{LM}} \times \prod_i \prod_{X \rightarrow \langle \gamma, \alpha \rangle \in D} \phi_i(X \rightarrow \langle \gamma, \alpha \rangle)^{\lambda_i}$$

Common SCFG Models

■ Standard features in log-linear models

- Conditional rule probabilities in both directions:
- Lexical weights in both directions:
- Word counts $|e|$;
- Rule counts $|D|$;
- Target n-gram language model $P_{LM}(e)$;
- Glue rule penalty

$$P(\gamma|\alpha) \text{ and } P(\alpha|\gamma)$$

$$P_w(\gamma|\alpha) \text{ and } P_w(\alpha|\gamma)$$

$$X \rightarrow \langle X_1 X_2, X_1 X_2 \rangle$$

- We propose a new feature, abstraction penalty $\exp(-N_a)$, to balance the decoder's choice on rules with 0, 1 or 2 nonterminals. N_a is defined as:

$$\sum_{X \rightarrow \langle \gamma, \alpha \rangle \in D} n(\gamma)$$

Chart-parsing based Decoder for SCFG

- Objective: search for the optimal derivation tree from all possible trees covering input
- Synchronous:
 - source & target side isomorphic tree;
 - string-to-tree-to-string
- Decoder: a modified CKY parser in C++ with integrated n-gram LM scoring
- LM scoring is implemented as a Viterbi search in FSM
- Chart cells filled in a bottom-up fashion until a tree rooted from nonterminal is generated that covers the entire input
- Lazy cube pruning (Chiang07) used for decoding speed up

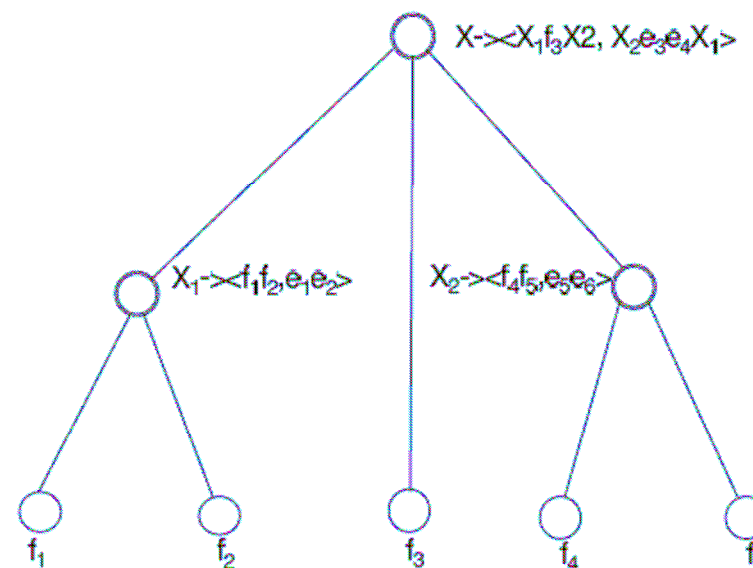


Figure 1: A chart parsing-based decoding on SCFG produces translation from the best parse: $f_1 f_2 f_3 f_4 f_5 \rightarrow e_5 e_6 e_3 e_4 e_1 e_2$.

Motivation of Prior Derivation

- Baseline uses heuristic-based estimation of $P(\gamma|\alpha)$ and $P(\alpha|\gamma)$
 - Relative counts collected from *hypothesized* rule distribution
 - *Inaccurate* estimation compared to terminal phrasal pairs
- No discrimination between parses on one side, when the other side is unknown
- If we can learn some prior distribution of rules, we rewrite:

$$P(D) \propto P_{LM}(e)^{\lambda_{LM}} \times \prod_{X \rightarrow \langle \gamma, \alpha \rangle \in D} (\prod_i \phi_i(X \rightarrow \langle \gamma, \alpha \rangle)^{\lambda_i}) L(X \rightarrow \langle \gamma, * \rangle)^{\lambda_L}$$

- $L(\cdot)$ defined over each rule production
- $Prior_derivation(D)$ = Production of $L(\cdot)$ over all rules in D ;
- Here we show PD on source side; however, it can be computed on either *source* and/or *target* side

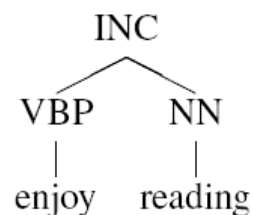
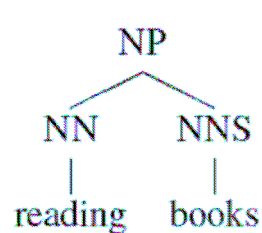
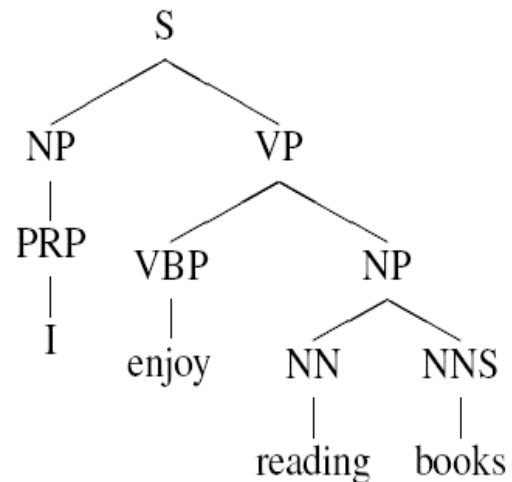
A Prior Derivation Model

- Link the source side derivation prior probability with the expected ambiguity on target side
 - A derivation is favored *if* it introduces less ambiguity on target generation
- Observation: same source side maps into different target orders, often depending on the *syntactic role* of nonterminal(s)

| | |
|--|------------------|
| $X \rightarrow \langle X_1 \text{enjoy reading } X_2,$ | |
| $X_1 \text{xihuan}(\text{enjoy}) \text{yuedu}(\text{reading}) X_2 \rangle$ | ← If X_2 is NP |
| $X \rightarrow \langle X_1 \text{enjoy reading } X_2,$ | |
| $X_1 \text{xihuan}(\text{enjoy}) X_2 \text{yuedu}(\text{reading}) \rangle$ | ← If X_2 is PP |

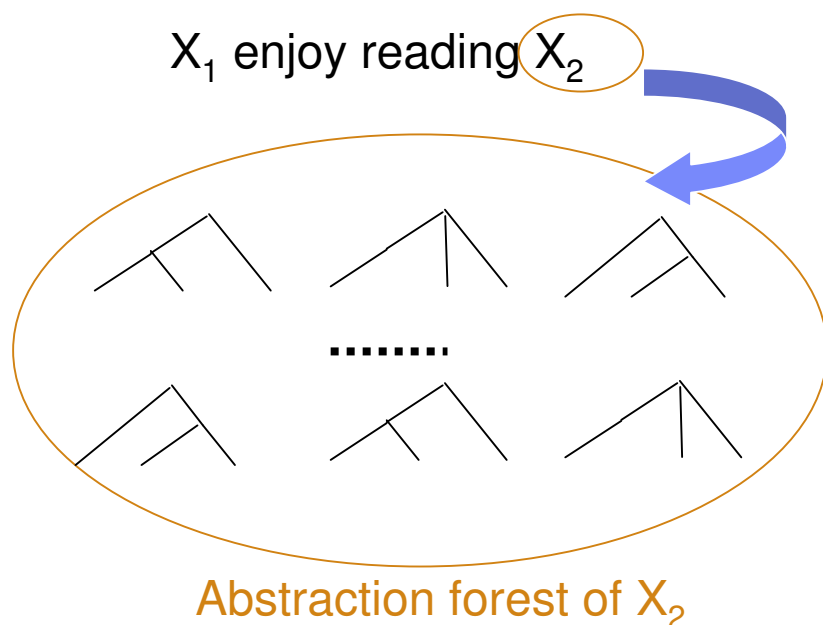
- ➔ Hypothesis: Higher variation of syntax structures the nonterminal embodies, the more translation options needed to account for various syntactic roles; estimated models are thus less reliable.
- Prefer nonterminals that cover more *syntactically homogeneous* expressions
- Now, how to quantify & model it?

Model Syntactic Variations: Definitions

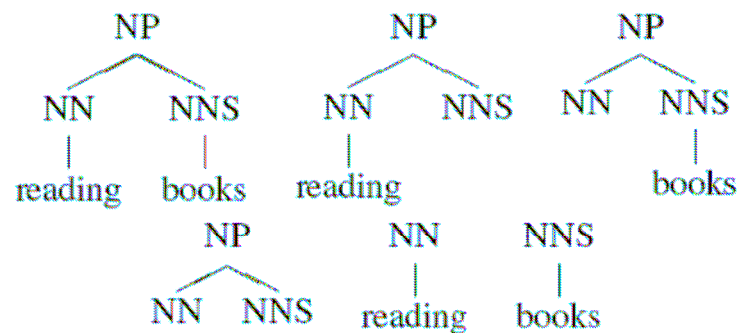
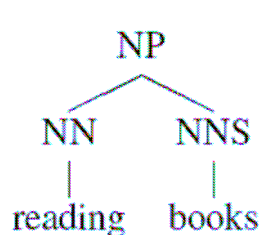


- At training: parse the English side of the parallel corpus
- *Tree fragment* of a phrase: the minimal set of internal tree whose leaves span exactly over this phrase
 - e.g., “reading books” a tree fragment rooted from NP
- Two special kinds of fragment root
 - INC: incomplete tree fragment; phrase pairs crossing constituency boundary
 - EMPTY: failed parsing

Definitions (continued)



- **Abstraction forest:** the set of tree fragments of all sub-phrases abstracted by a nonterminal
- **Subset trees:** any sub-graph that contains more than one node, with the restriction that entire rule productions must be included.



Compute Syntactic Homogeneity

- *Tree fragment similarity: naturally defined by*
 $K(T_1, T_2) = \text{number of common subset trees in } T_1 \text{ and } T_2$
- Conceptually, enumerate all possible subset trees $1, \dots, M$, and let $h(T) = (c_1, \dots, c_M)$, a vector of counts of each subset tree
 - $K(T_1, T_2) = \langle h(T_1), h(T_2) \rangle$; an inner product
 - Note: $h(T)$ will be **a vector with a huge number of dimensions**
- **Kernel methods**: an efficient way to carry out computation when original feature dimension is large or infinite
- (Collins & Duffy, 02) suggested to employ convolution kernels for tree structures

Tree Kernel Methods

$$K(T_1, T_2) = \sum_{n_1 \in N_1} \sum_{n_2 \in N_2} C(n_1, n_2) \quad (9)$$

where $C(n_1, n_2) = \sum_i I_i(n_1) I_i(n_2)$ and N_1, N_2 are the set of nodes in the tree fragment T_1 and T_2 respectively. It is noted that $C(n_1, n_2)$ can be computed recursively (Collins and Duffy, 2002):

1. $C(n_1, n_2) = 0$ if the productions at n_1 and n_2 are different;
2. $C(n_1, n_2) = 1$ if the productions at n_1 and n_2 are the same and both are pre-terminals;
3. Otherwise,

$$C(n_1, n_2) = \lambda \prod_{j=1}^{nc(n_1)} (1 + C(ch_{n_1}^j, ch_{n_2}^j)) \quad (10)$$

where $ch_{n_1}^j$ is the j th child of node n_1 , $nc(n_1)$ is the number of children at n_1 and $0 < \lambda \leq 1$ is a decay factor to discount the effects of deeper tree structures.

- Dynamic-programming based computation: worst case complexity is $O(|N_1| \times |N_2|)$
- In practice, linear time on average

- *Forest purity:*

$$Pur(X) = \frac{2}{N(N-1)} \sum_j \sum_{i < j} K'(T_i, T_j)$$

$$L(X \rightarrow \langle \gamma, * \rangle) = \left| -\log\left(\min_{X_1, X_2 \in \gamma} (Pur(X_1), Pur(X_2))^k\right) \right|$$

- Quadratic complexity:
 - Lazy pruning in training: prune forest with large N
 - Parallel computation

How does prior derivation model impact in MT

- Straightforward motivation: *some derivations preferred over others.*
- However, there are other interpretations:
 1. An analogy between prior derivation distributions to non-uniform source side segmentation in phrase-based models.

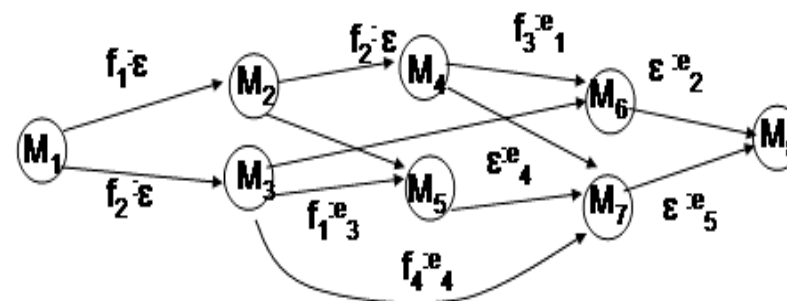
➔ However, prior derivation models influence *not only* on phrase choices, *but also* on ordering options due to the nonterminal usage
 2. Smoothing on rule translation probabilities estimated from heuristics
 - More translation options in a rule ↔ More ambiguity for this rule.
 - When a dominating translation option is overestimated, all translation options of this rule are discounted, as they are less favored by prior derivation models.

Outline

- Introduction to S2S: An overview of IBM MASTOR
- DARPA TRANSTAC Program: Bring S2S to real world
 - Mission and the progress
 - Video demo: Iraqi Arabic-English S2S on Tablet PC
 - How S2S is evaluated?
- SMT and S2S Technologies
 - Real-time speech recognition & text-to-speech synthesis (no discussion today)
 - Recap: Word alignment and phrase-based SMT
 - Multiple graph-based phrasal SMT using finite state
- Formal syntax-based SMT and SCFG
 - Overview of syntax-based SMT and SCFT
 - Efficiently integrating linguistic syntax information
 - Effective learning of SCFG rules
- Recap & case study: SMT systems used in IBM S2S
 - Demo: Pashto-English S2S on Smart Phones

Recap: Various Translation Models

عن نقاط تفتيش السيارات || on vehicle checkpoints || 0.4 0 1 0
 نقاط تفتيش السيارات || vehicle checkpoints || 0 0 1 0.0308615
 سيارات || vehicles || 0 0.00203285 0.08 0.0832386
 السيارة || vehicle || 0 0 0.285714 0.407666



Phrase-based translation model: encode context and local reorder information

Graph-based phrasal translation model: optimize translation options into a compact graph

$X \Rightarrow \langle X_1 \text{ امرار معاش ی برا } X_2, X_2 \text{ cover } X_1 \text{ living expenses} \rangle$

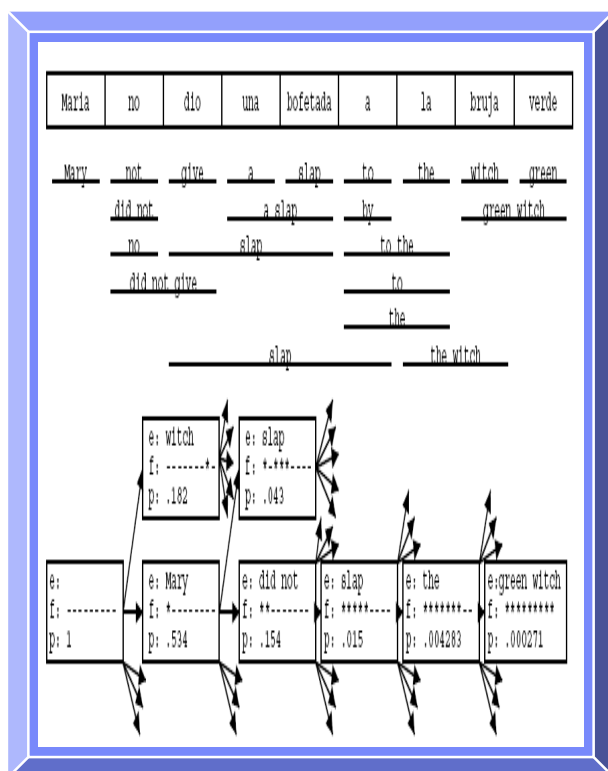
سعيديه X1 بله اسم ايشون || yes his name is X1 saaedi || 2.11e-08 0 7.5 8.824
 خوشبختانه X1 مريضى X2 بگيرم || fortunately X1 i get X2 illness || 3.1e-08 0 21. 16.7
 X1 امرار معاش X2 || X2 cover X1 living expenses || 2.0-07 0 11.5 18.5
 ميرود X1 او به سوريه || he visits syria X1 || 6.73709e-08 0 7.25473 8.77122
 فقط X1 براى X2 || to just X1 for X2 || 7.07394e-08 0 18.0796 19.9809

Statistical *synchronous context-free grammar* (SCFG), where

- Co-indexed X's are non-terminals that can be recursively instantiated
- Models language's hierarchical characteristics

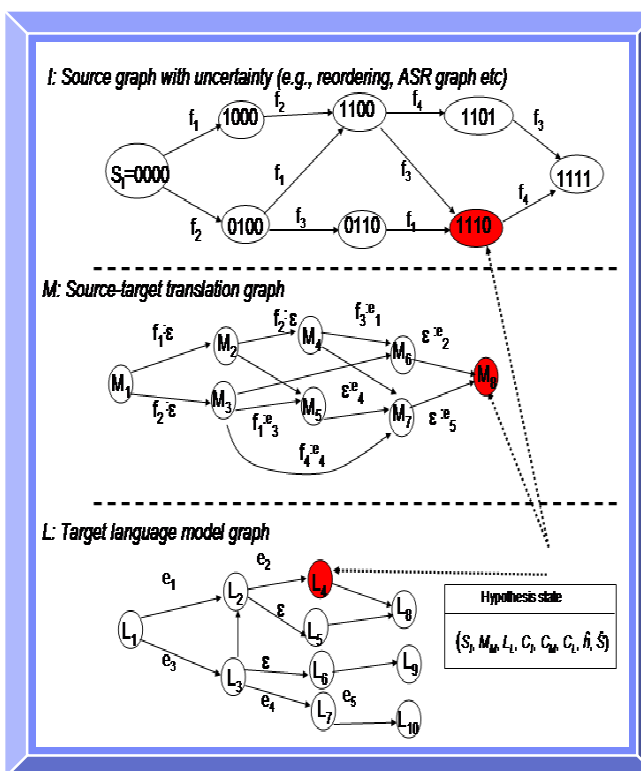
IBM S2S Decoders: Search for the best translation

Stack: Phrasal SMT



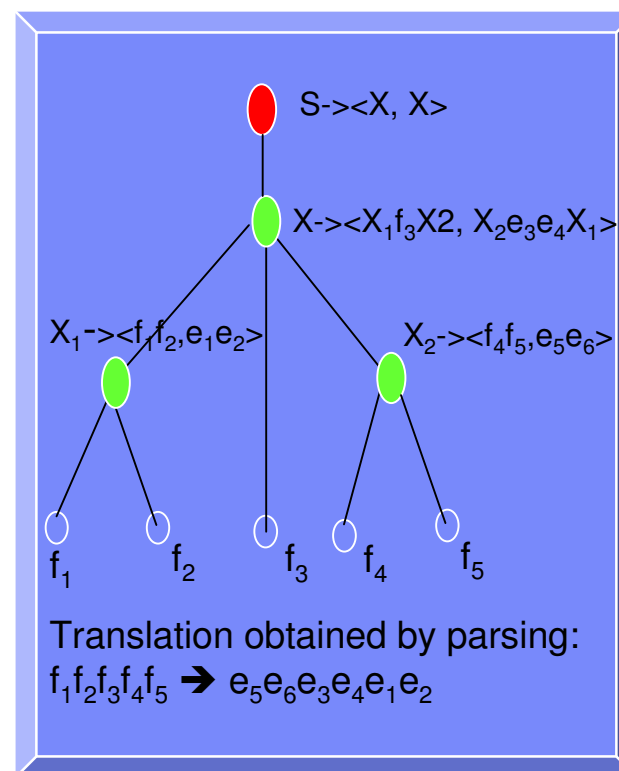
Fast decoding & efficient training

Folsom: Multi-graph SMT



Fast & memory efficient; Enable large vocabulary translation on small devices; Efficient coupling with ASR for integrated speech translation

ForSyn: Chart-based SCFG SMT



Better generalization for unseen data; more principled reordering; Better accuracy for difficult language pairs (e.g, Pashto)

Optionally, the independent best translations from different decoders can be combined to produce a better translation than any single of them

More Questions ?

Email me:
zhou@us.ibm.com