An Overview of IBM Speech-to-Speech Translation (S2S)

Bowen Zhou
IBM T. J. Watson Research Center
Yorktown Heights, New York, USA
zhou@us.ibm.com

1 December 2009

Outline

- Overview of IBM S2S
- DARPA TRANSTAC Program
  - Mission and the progress
  - Video demo: IBM Dari system
  - How S2S is evaluated?
- IBM S2S Technologies
  - Real-time speech recognition (no discussion today)
  - Text-to-speech synthesis for low-resource languages (no discussion today)
  - Statistical Machine Translation (SMT)
    - Word alignment
    - Phrase-based SMT
    - Multiple graph-based SMT using FST
    - Syntax-based SMT and SCFG
  - Recap & case study: SMT systems used in IBM S2S

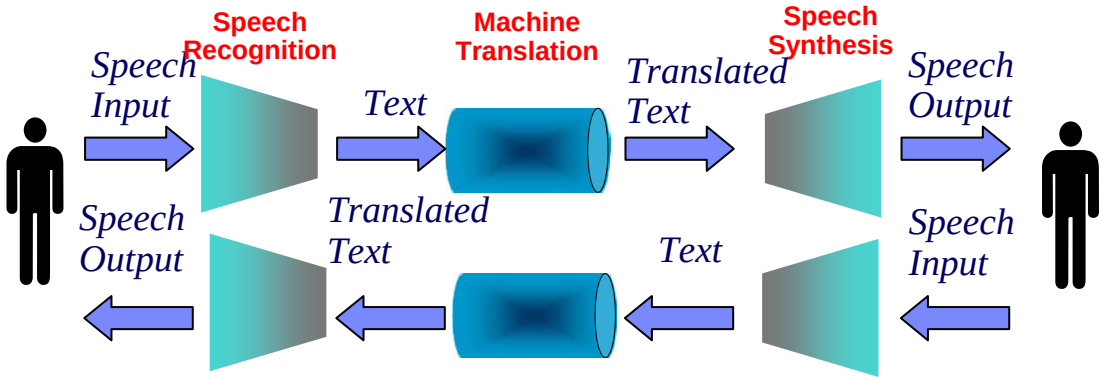
IBM Speech-to-Speech Translator

A Real-time Solution to Mitigate Language Barriers

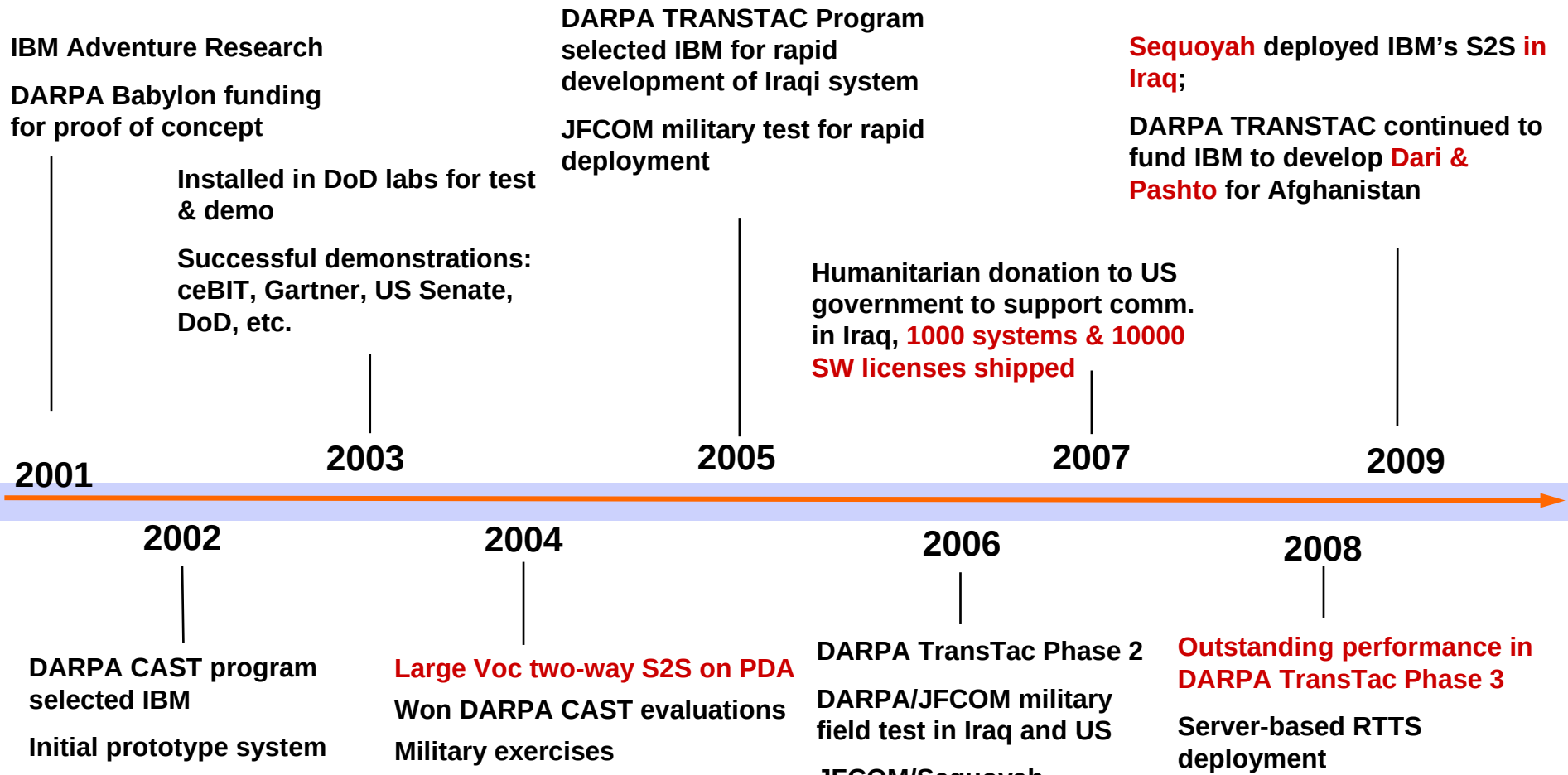
- Automatic *Universal* Translator
  - The dream of scientists for decades – most challenging research
- MASTOR (**M**ultilingual **A**utomatic **S**peech-to-Speech **T**translat**OR**)
  - Attempting to facilitate cross-lingual oral communication for *designed* domains
- Challenges
  - Background noise 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

IBM Speech-to-Speech Translator

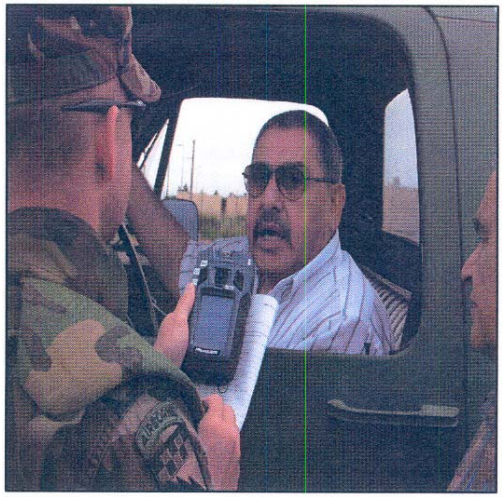
- **Audibly and visually translates between two languages**
  - 2-way translation of "*free form*" conversational speech
  - Target for instantaneous & highly accurate S2S on mobile platforms including handheld
  - Robust & speaker independent speech recognition, accommodating differences in tone and accent with online adaptation
  - Data-driven statistical syntax-based machine translation
  - Machine learning techniques ubiquitously applied, which enables rapid development for new languages & domains



A History of IBM's Statistical Speech-to-speech Translation



Handheld S2S – A mobile solution



Outline

- Overview of IBM S2S
- DARPA TRANSTAC Program
  - Mission and the progress
  - Video demo: IBM Dari system
  - How S2S is evaluated?
- IBM S2S Technologies
  - Real-time speech recognition (no discussion today)
  - Text-to-speech synthesis for low-resource languages (no discussion today)
  - Statistical Machine Translation (SMT)
    - Word alignment
    - Phrase-based SMT
    - Multiple graph-based SMT using FST
    - Syntax-based SMT and SCFG
  - Recap & case study: SMT systems used in IBM S2S

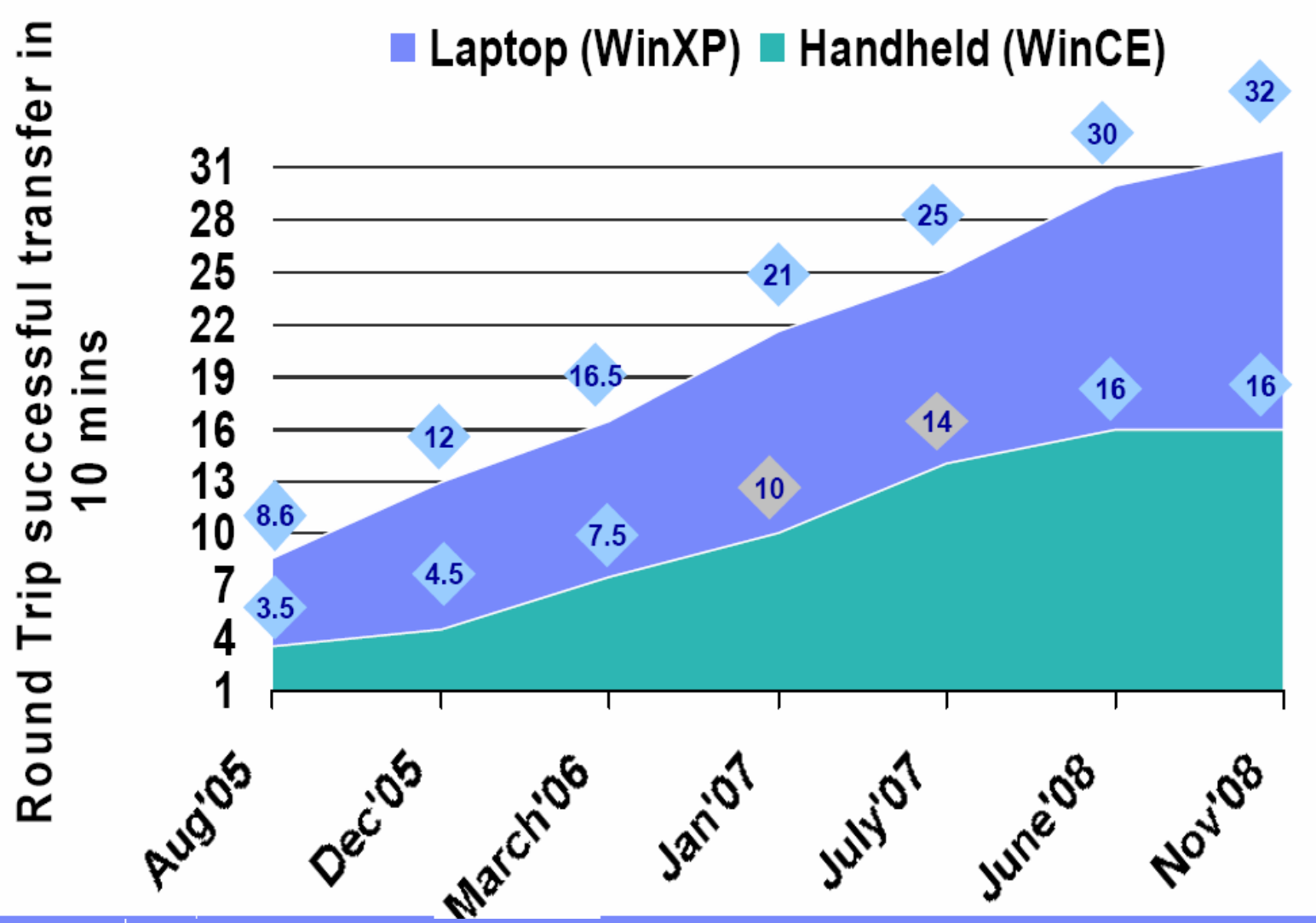
DARPA TRASTAC

- 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
  - Dari-English
  - Pashto-English
- We built prototypes for both Dari & Pashto during the past 6 months
- Dari video Demo

How S2S is evaluated?

- Evaluations led by NIST 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

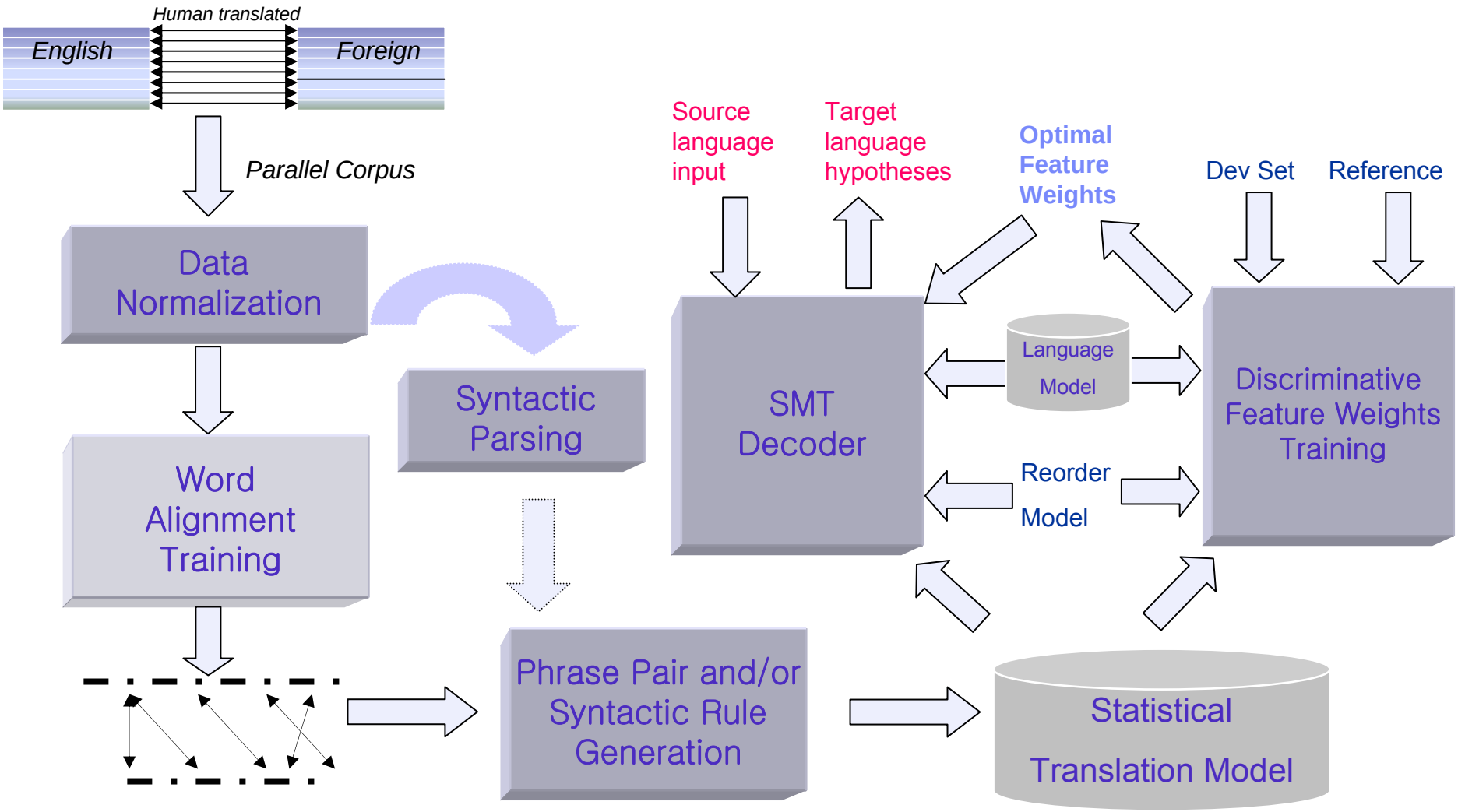
Progress over the years: IBM's high-level concept transfer rate



Outline

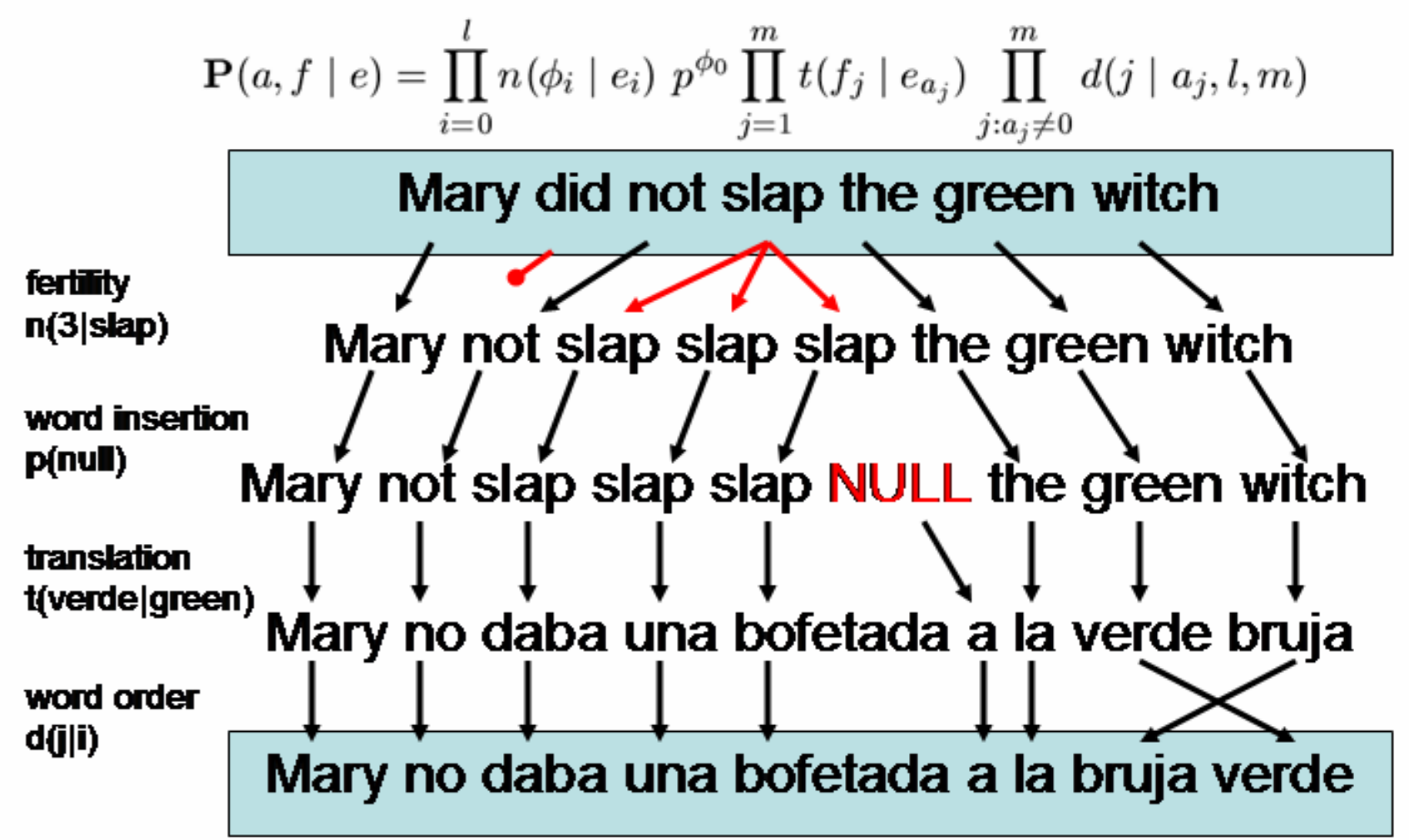
- Overview of IBM S2S
- DARPA TRANSTAC Program
  - Mission and the progress
  - Video demo: IBM Dari system
  - How S2S is evaluated?
- IBM S2S Technologies
  - Real-time speech recognition (no discussion today)
  - Text-to-speech synthesis for low-resource languages (no discussion today)
  - Statistical Machine Translation (SMT)
    - Word alignment
    - Phrase-based SMT
    - Multiple graph-based SMT using FST
    - Syntax-based SMT and SCFG
  - Recap & case study: SMT systems used in IBM S2S

A Typical Pipeline of SMT

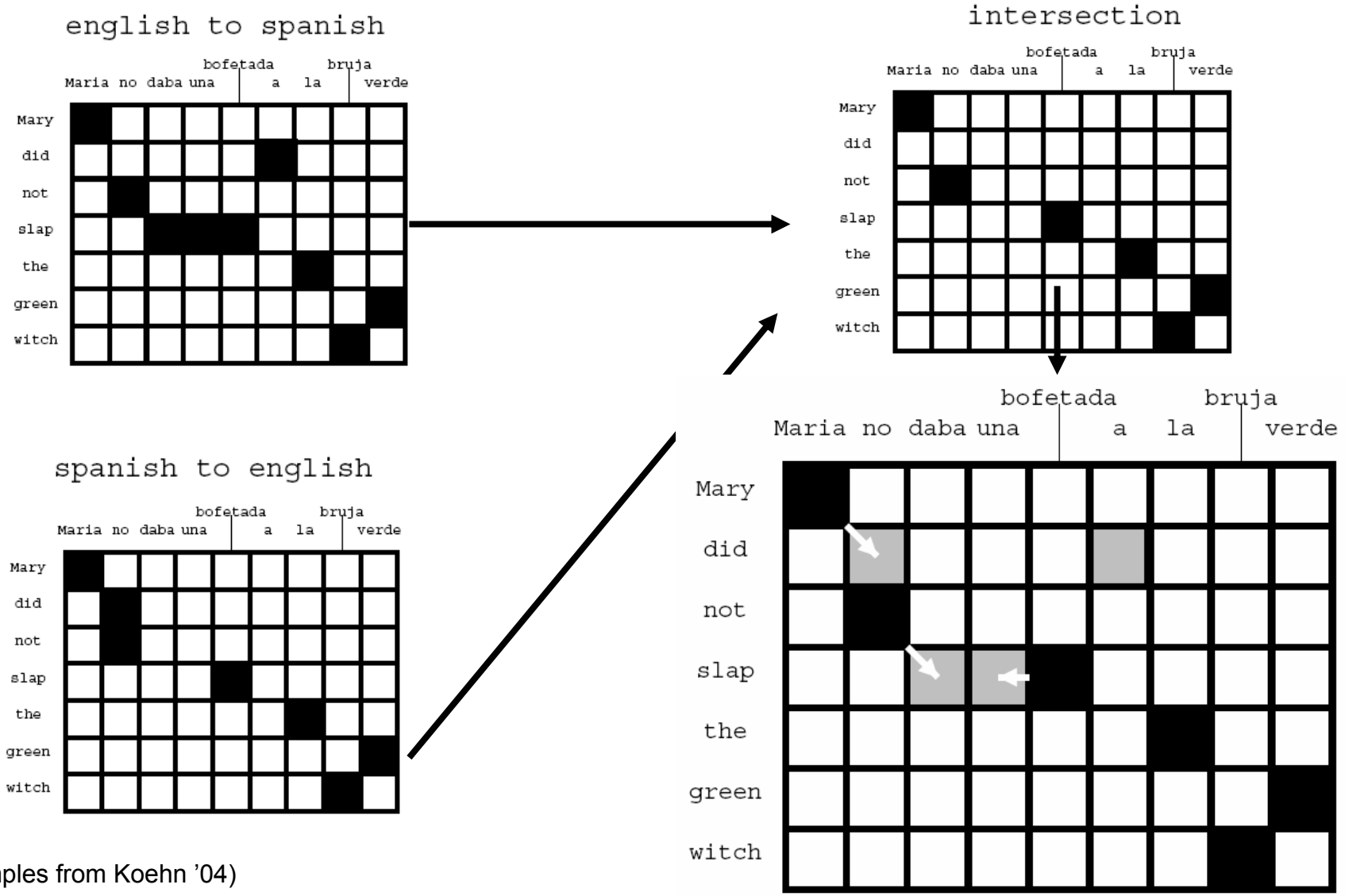


How word alignment is learnt: IBM Model 4 & EM (Brown'93)

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

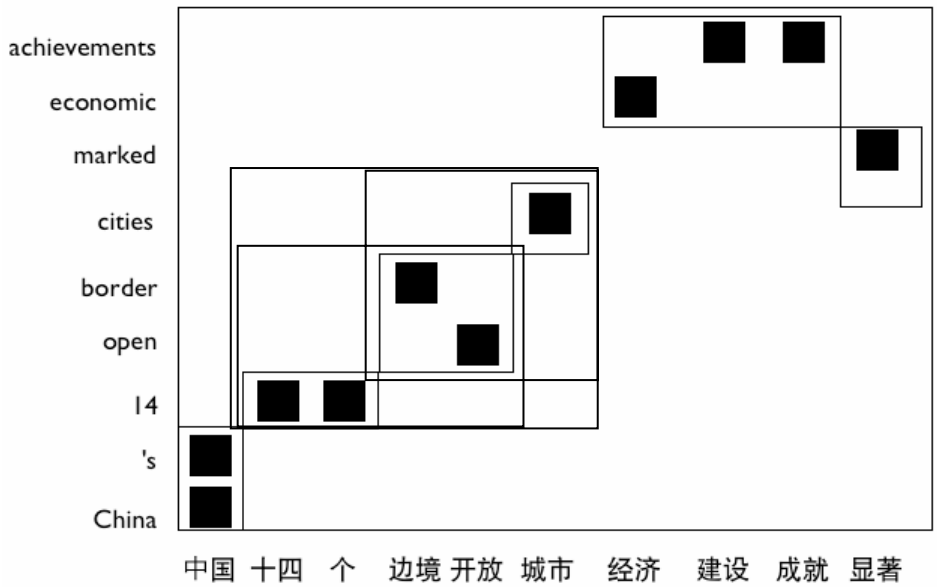


Alignment Symmetry & Refinement



(Examples from Koehn '04)

Phrase Translation: 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
- Other models, including 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

$$\hat{e} = \arg\max_{e_1^E} \Pr(e_1^E \mid f_1^J)$$

$$= \arg\max_{K, \bar{e}_1^K} \Pr(\bar{e}_1^K \mid \bar{f}_1^K)$$

- Log-linear model combination: language model, length bonus etc.
- Decoding: Stack (A*) beam search (Och'04, Koehn'04) is commonly used
- Alternatively, the decoding can be done by WFST techniques
  - Without consideration of recursion or hierarchical structures, 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
- Early studies: Knight'98, Bangalore'01
- General framework of using WFST for translation

$$\hat{e} = best - path\ (s = IoM_1 oM_2 o\ \Lambda\ oM_m)$$

  - 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 these cases, decoding is performed with 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 $best\text{-}path\ (I \circ H)$
- 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);
  - $I \circ H$ still expensive (even with on-the-fly composition followed by beam-pruning search)
  - Reordering is a big challenge, making search NP-hard, and H not finite-state
- In previous work, compositions have all been done *online* for given input
  - Slow speed (<5 wps) (kumar'05),
  - Needs hundreds of MB to several GB memory at runtime

Folsom: A Multiple-Graph based Approach for SMT

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

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

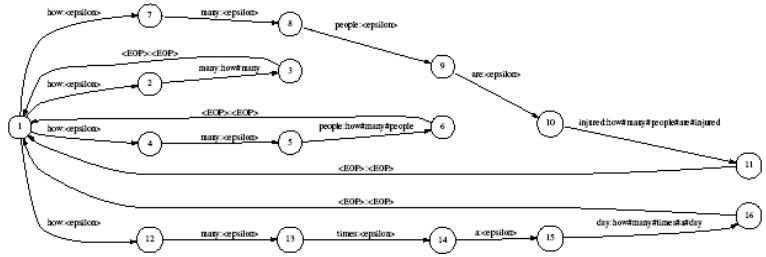
P: source language segmentation
T: phrasal translation
W: target language phrase-to-word
L: target language model

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

- Decompose the problem as a chain of conditional probabilities
- Each of which represented by WFST modeling the relationships between their inputs and 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

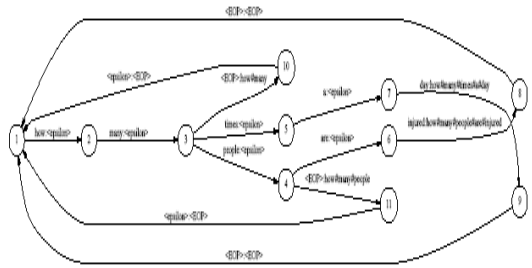
Determinize Phrase Segmentation Transducer P

- Mapping *word* sequences to all "acceptable" *phrase* sequences:
  - How many people are injured
    - How
    - How many
    - How many people
    - many people are
    - peole are injured
    - …



- Previous work (Kumar'05) is not determinizable. Determinization is crucial here:
  - Reduce the size of this machine,
  - Making following compositions possible

- Non-determinizability is caused by overlap between phrases,
  - a word sequence may be segmented into phrases in multiple ways that are nested
  - phrase identity may not be determined ***until*** the entire sentence is observed,
  - such unbounded delays make *P* non-determinizable



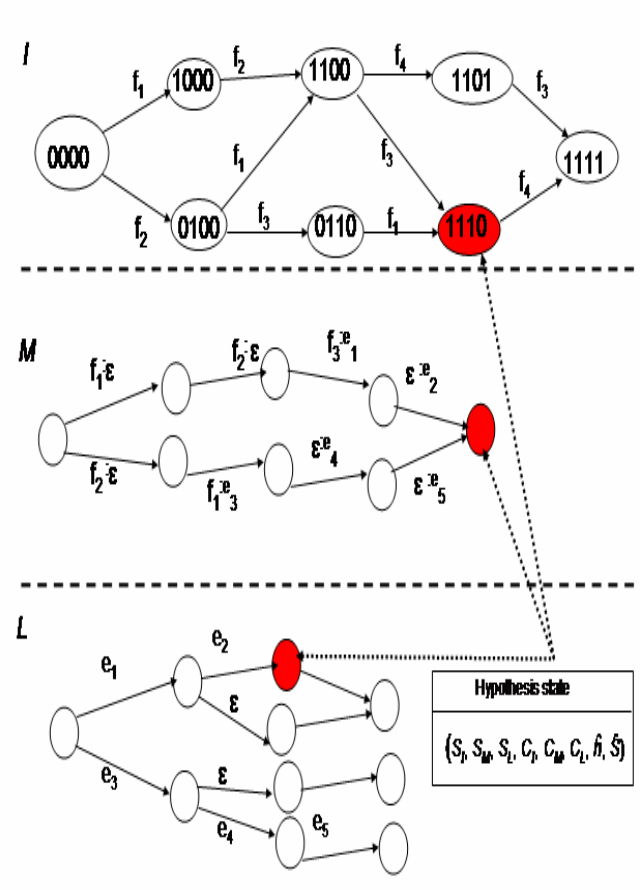
- Our Solution:
  - introduce an auxiliary symbol, *EOP*,
  - Marking the end of each distinct source phrase.

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 o T o W o L, not computable offline!
- Solution: Separate H as: H = M o L
- $M = Min(Min(Det(P) \circ T) \circ W)$
  - –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
- L (support arbitrary long span n-gram LM):
  - –3-gram: 300K states/1.7 M arcs after minimization
  - –Typically 4-gram for evaluation

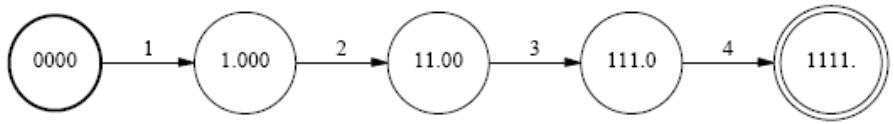
Folsom: Multiple-graph SMT (Zhou et al., 06;07;08)

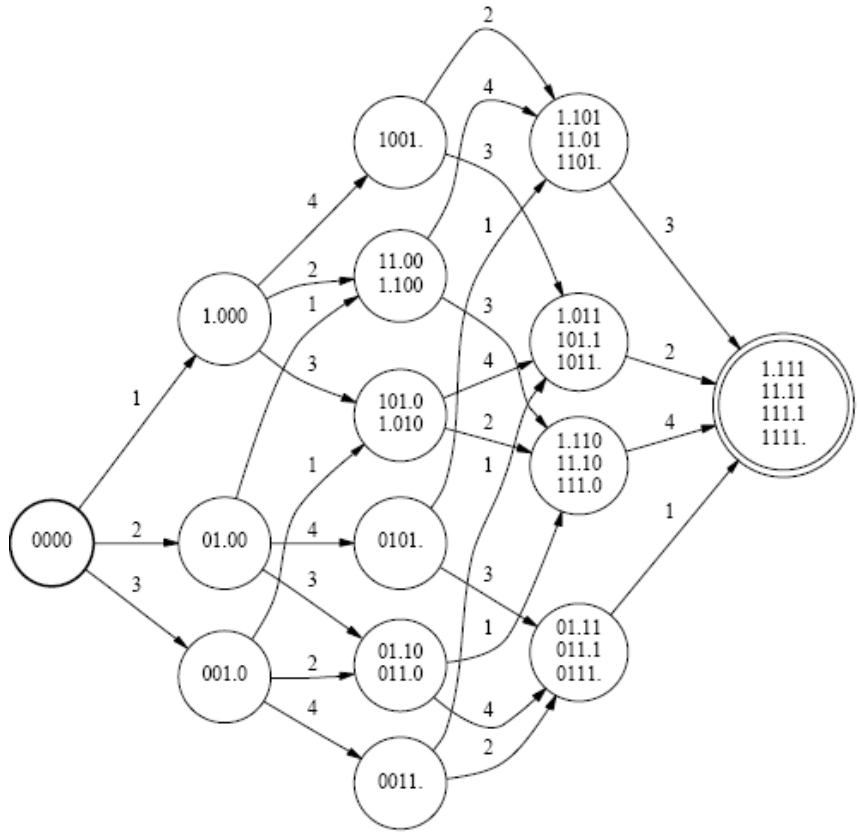


- SMT built upon multi weighted finite state graphs:
  - **Input graph *I*:** model *uncertainty* in inputs
    - Reordering, ASR ambiguity, morphological, segmentation, and/or their combinations
    - *Statically or lazily constructed*
  - **Translation graph *M*:** encode phrasal translations
  - **Target graph *L*:** measure target acceptability
- Decoder: *Best-path (I o M o 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)

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:
  - For a m word input, how many states are needed in this reorder graph, when there is no reorder constraint?





Putting Syntax into Translation Model

- 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
- *Linguistic* syntax-based:
  - Explicitly utilizes structures defined over linguistic theory & annotations (e.g., Penn Treebank)
  - SCFG rules 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
  - Synchronous grammars extracted w/o any usage of linguistic knowledge
  - Examples: ITG (Wu, 97) & hierarchical models (Chiang, 07)
  - Linguistic information can be added as soft-constraints (Zhou08)

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)

Phrase table ➔ SCFG

- A synchronous rewriting system generating source & target side simultaneously, based on CFG
- 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

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

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

- Examples: English-to-Chinese production rules

$$X \longrightarrow \langle X_1 \text{enjoy reading} X_2, X_1 \text{xihuan}(\textit{enjoy}) \text{ yuedu}(\textit{reading}) X_2 \rangle$$

$$X \longrightarrow \langle X_1 \text{enjoy reading} X_2, X_1 \text{xihuan}(\textit{enjoy}) X_2 \text{yuedu}(\textit{reading}) \rangle$$

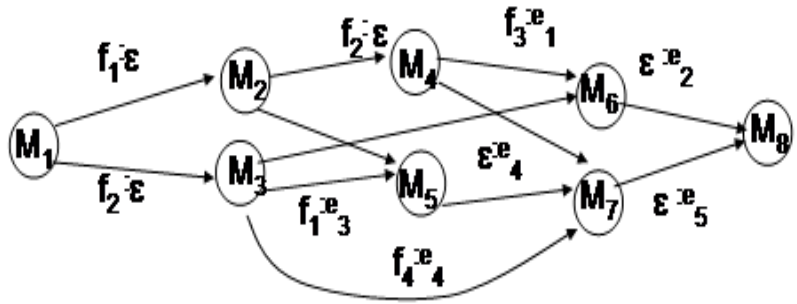
Recap: Various Translation Models

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 ➔ <X_1 برای امرار معاش X_2 , X_2 cover X_1 living expenses>

سعیدیه X1بله اسم ایشون || yes his name is X1 saaedi || 2.11e-08 0 7.5 8.824

بگیرم X2 مریضی X1خوشبختانه || fortunately X1 i get X2 illness || 3.1e-08 0 21. 16.7

برای امرار معاش X2 || X2 cover X1 living expenses || 2.0-07 0 11.5 18.5 X1

میرود X1او به سوریه || he visits syria X1 || 6.73709e-08 0 7.25473 8.77122

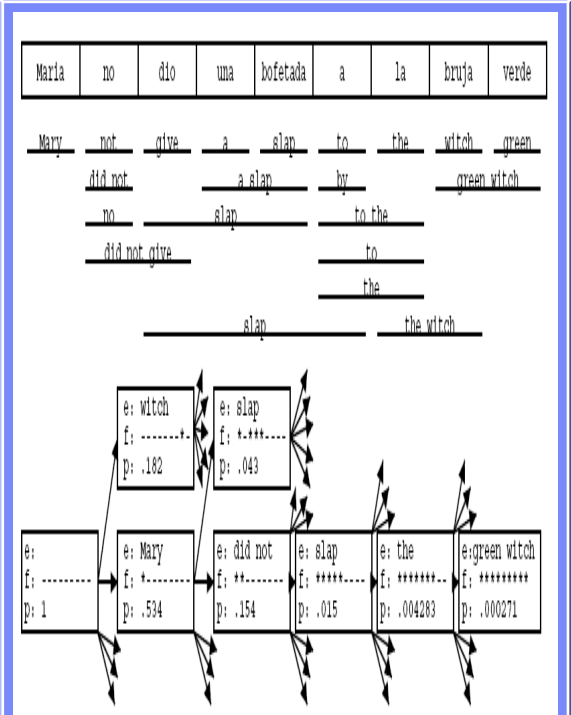
برای X1که فقط X2 || to just X1 for X2 || 7.07394e-08 0 18.0796 19.9809

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

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

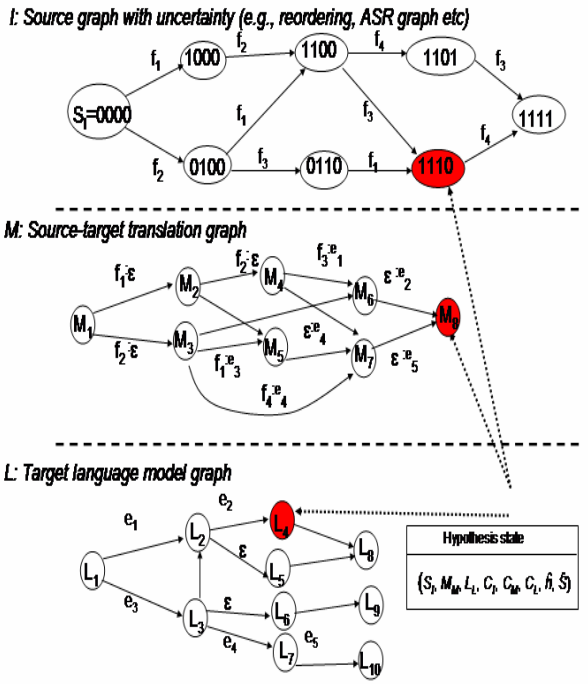
IBM S2S Decoders: Search for the best translation

Stack: Phrasal SMT



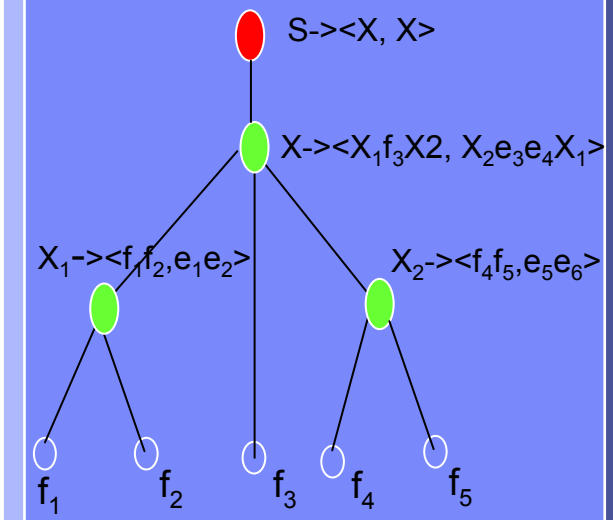
Fast decoding & efficient training

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



Translation obtained by parsing:
$f_1f_2f_3f_4f_5$ ➔ $e_5e_6e_3e_4e_1e_2$

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

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

Questions ?