Johns Hopkins University - Cambridge University Chinese→English and Arabic→English 2005 NIST MT Evaluation Systems

Shankar Kumar, Yonggang Deng, Bill Byrne

Cambridge University Engineering Department

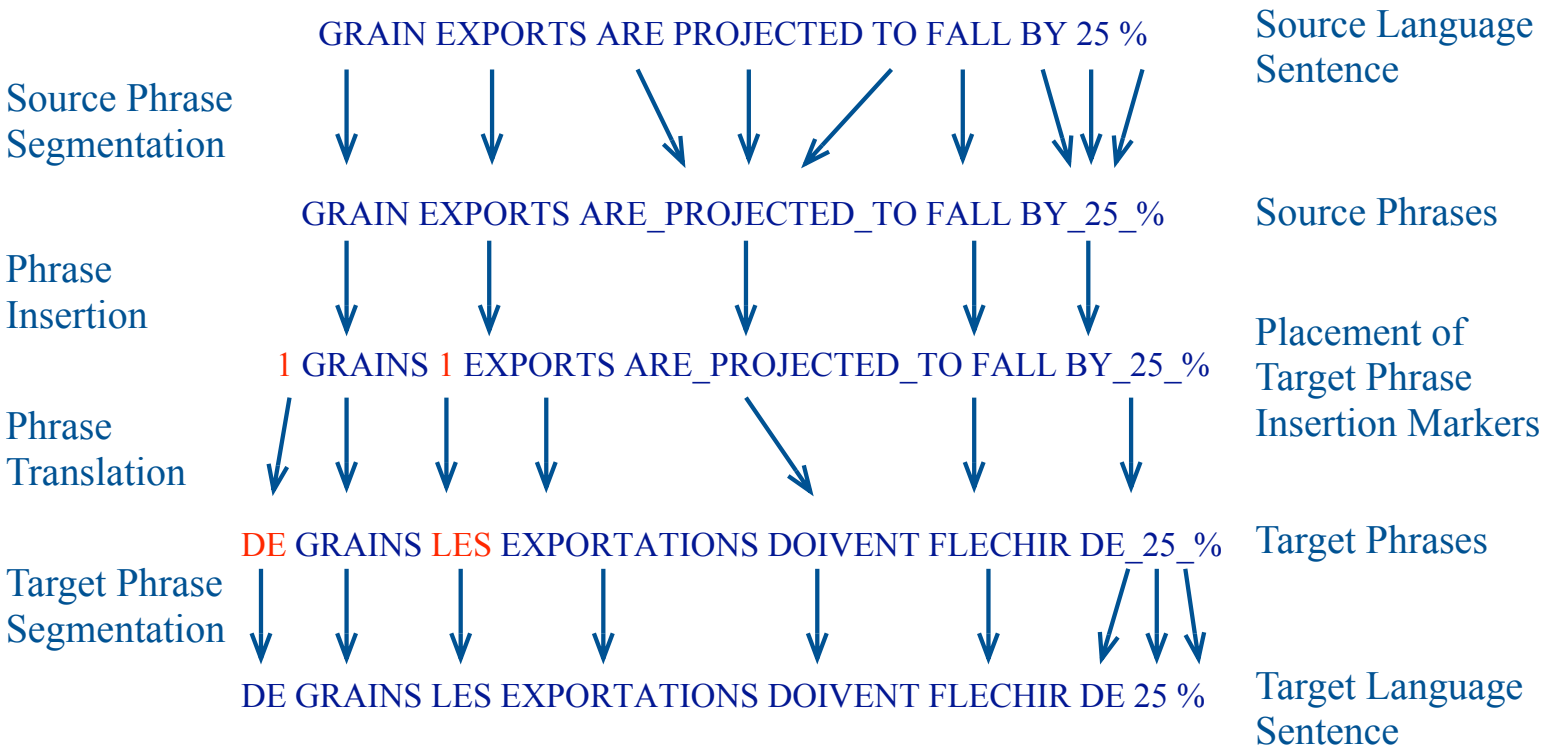
The Johns Hopkins University
Center for Language and Speech Processing

20 June 2005

Overview

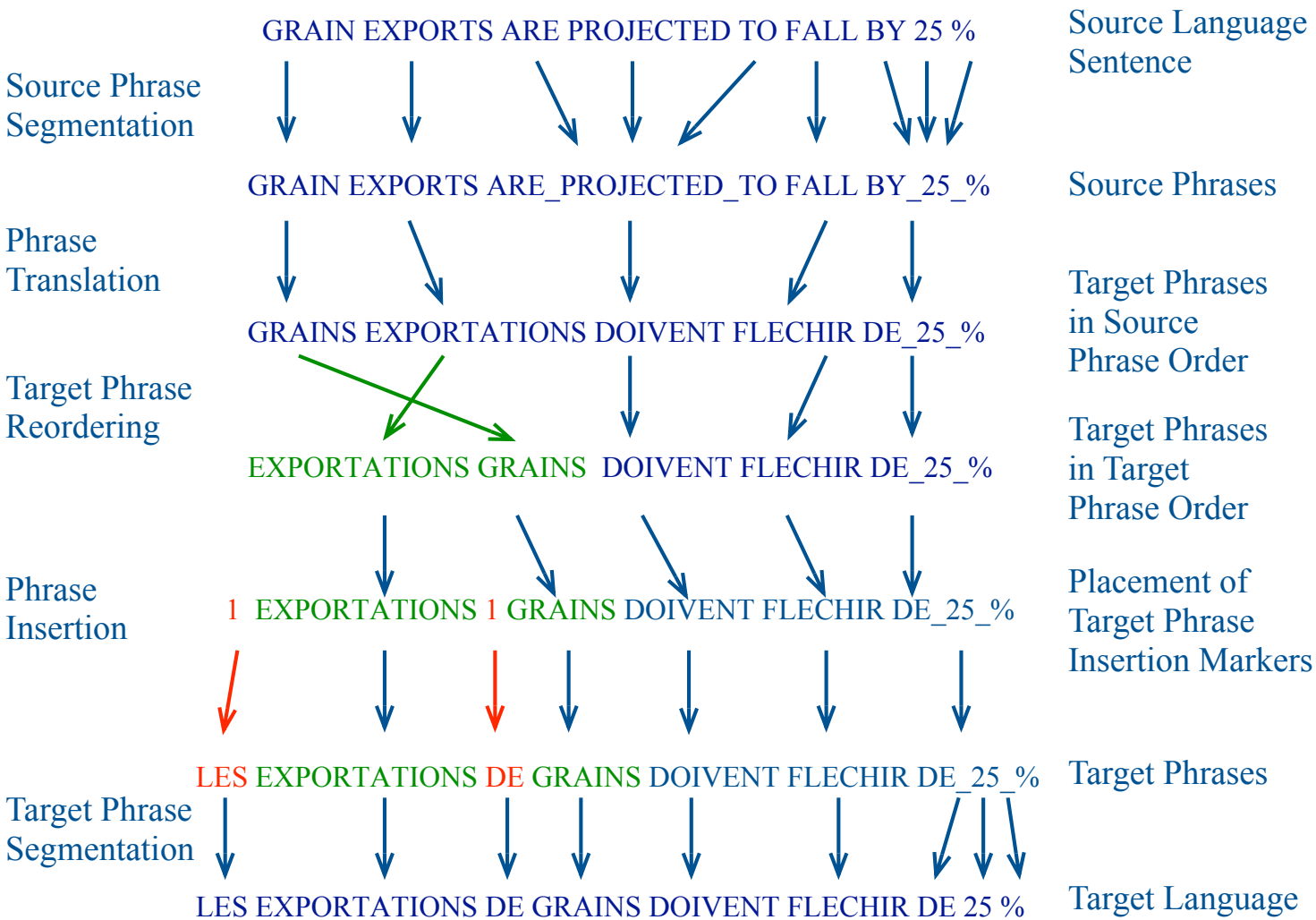
- Single architecture for Arabic→English and Chinese→English MT
- Based on 2004 Evaluation System
  - Bitext Chunking by Divisive Clustering
    - better use of bitext, yields improved parameter estimation
  - Translation Template Model (TTM)
    - Phrase-based SMT with Weighted Finite State Transducer implementation
    - Generative Source-Channel translation model
- New this year:
  - TTM Phrase Reordering – Shankar Kumar
  - MTTK – Yonggang Deng
    - Bitext word alignment
    - Phrase-pair induction from bitext
  - Minimum Error Training of TTM component weights – Shankar Kumar

TTM (2004) – Translation with Monotone Phrase Order

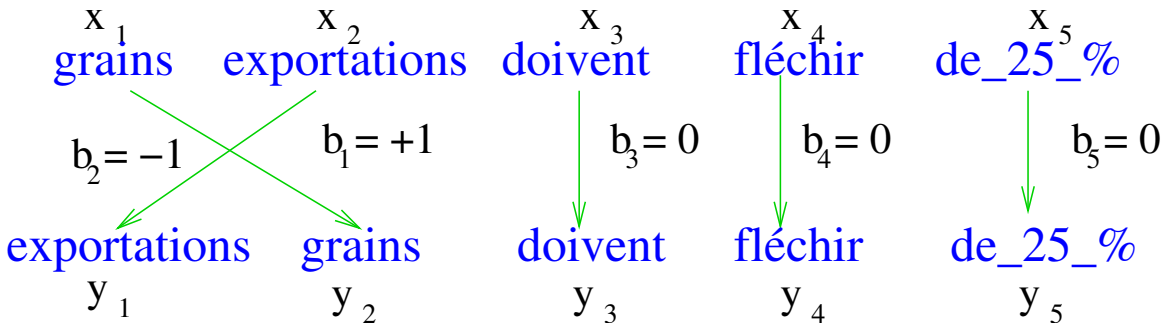


- Transformations via stochastic models implemented as WFSTs
- Target phrases remain in source phrase order
- Word movement takes place within phrase translation
  - even so, within long phrases (5 or more words) movement can be extensive

TTM (2005) – Translation with Moving Target Phrase Order

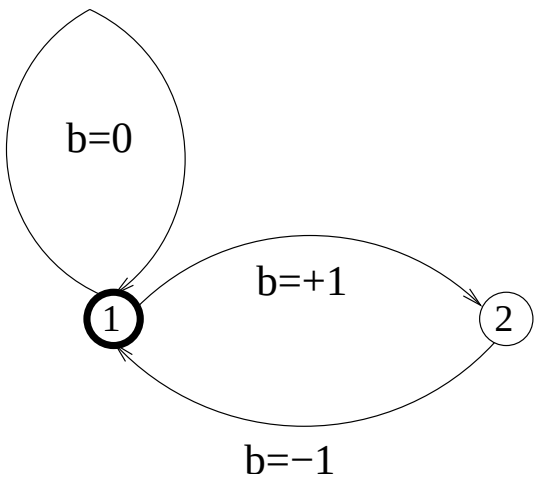


Phrase Swapping by WFSTs



Associate a jump sequence b_1^K with each sequence y_1^K

$$P(y_1^K|x_1^K, u_1^K, K, e_1^I) = P(b_1^K|x_1^K, u_1^K, K, e_1^I) = \prod_{k=1}^{K} P(b_k|x_k, u_k)$$



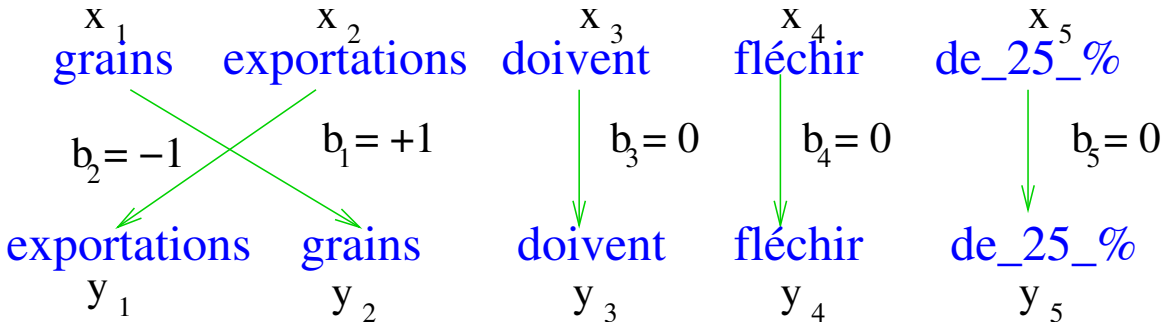
Motivated by Tillmann (HLT'04)

MJ-1 : maximum jump of 1

$b \in \{0, +1, -1\}$

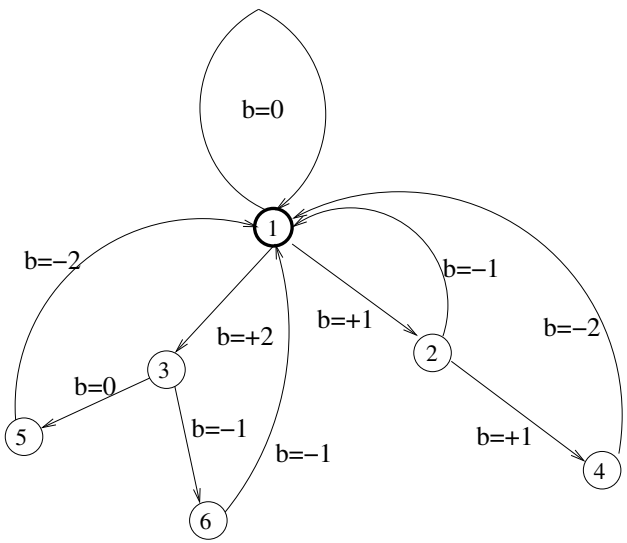
Parameterized / not degenerate

Phrase Swapping by WFSTs



Associate a jump sequence b_1^K with each sequence y_1^K

$P(y_1^K|x_1^K, u_1^K, K, e_1^I) = P(b_1^K|x_1^K, u_1^K, K, e_1^I) = \prod_{k=1}^K P(b_k|x_k, u_k)$



Motivated by Tillmann (HLT'04)

MJ-2 : maximum jump of 2

$b \in \{0, +1, -1, +2, -2\}$

Parameterized / not degenerate

Incorporating Reordering in Translation Under the TTM

Local Phrase Reordering Model

- Proper probabilistic model over reordered phrases
  - fits within the entire source-channel model of phrase translation
  - not degenerate
- Reordering and phrase insertion allows fairly-far word movement
- Possible to realize with WFSTs both in alignment and translation
- Reordering is done prior to insertion of Target Phrases

Reordering is an added FSM composition step in the translation pipeline

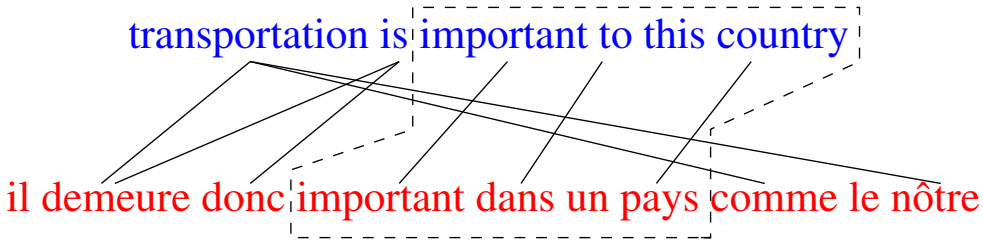
Embedded reestimation of reordering model parameters

- Phrase-pair dependent reordering probability : $P(b_k|x_k, u_k)$
- Estimated via Viterbi approximation to EM
- Exact estimation: bitext alignment is done under the translation model
  - implemented via FSM operations very similar to translation

Word Alignments and Phrase Translation

Translation via TTM incorporates a Phrase Pair Inventory (PPI)

- Viterbi PPI : Extracted from word-aligned bitext to cover test set phrases



- Add {important to this country, important dan un pays} to the PPI

Approach needs good quality word alignments : IBM Model-4

- Model-4 / GIZA++ alignments are difficult to beat, esp. with large bitexts
  - Model-4 is complex enough to benefit from large training sets
- Model-4 complexity can be a limitation
  - Exact EM is difficult - typically use hill climbing for parameter estimation
  - Parameter estimation is difficult to parallelize
  - Hard to compute statistics under Model-4, other than from alignments

Goal: Develop an HMM-based alternative of equal quality to Model-4

MTTK – HMM-Based Word and Phrase Alignment

- HMM architecture motivated by Model-4
  - Embedded Baum Welch reestimation and incremental build
- Alignment performance (AER) equals that of Model-4, so far

Bitext	English Words	Model	C→E	E→C
FBIS	10M	M-4	37.3	45.0
		MTTK	36.1	44.8
NEWS	71M	M-4	36.1	44.5
		MTTK	36.1	44.8
NEWS+ UN01-02	96M	M-4	36.5	
		MTTK	36.2	44.8
ALL C-E	200M	MTTK	36.8	44.7

  - larger bitexts needn't reduce AER, but do improve phase coverage
- Efficient training via EM – no need to partition the bitext

MTTK : 3 days on 60 CPUS to generate 1 set of C-E models and alignments *vs.*
M-4 : 1 week on 6 CPUS to generate 3 sets of C-E models and alignments

- Parallel E-Steps reduces the size of the co-occurrence tables
  → improved memory management

Phrase Pair Induction Under MTTK Alignment Posteriors

Viterbi PPI can be limited :

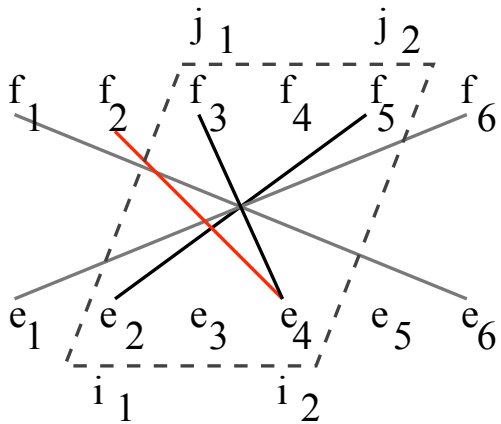
- some test set phrases will not be in the PPI, even if they're in the bitext

We can use MTTK to induce translations for any phrase found in the bitext

Suppose we have :

Bitext (e_1^l, f_1^m)

Alignment Process $a_1^m : f_j \rightarrow e_{a_j}$



What's the probability that $f_{j_1}^{j_2} \rightarrow e_{i_1}^{i_2}$?

Define

$$A(i_1, i_2; j_1, j_2) = \{a_1^m : a_j \in [i_1, i_2] \text{ iff } j \in [j_1, j_2]\}$$

If we can compute the *phrase pair posterior* $P(A(i_1, i_2; j_1, j_2)|e_1^l, f_1^m)$

we can find the most probable translation for any phrase in the bitext

- Allows for alternative PPI strategies – not limited to the 1-Best alignment
- Difficult to do with Model-4
- Here, we only improve the Viterbi PPI

PPI Induction Improves Test Set Coverage and Translation Performance

Decoding strategy:

- All phrases (up to length 5) are extracted from the test set
- The Viterbi PPI is created from the aligned bitext (all A-E and C-E)
- If a test phrase isn't in the Viterbi PPI, it is added via induction, if possible

		eval02		eval03		eval04	
V-PPI	PPI Induction	cvg	BLEU	cvg	BLEU	cvg	BLEU
		Large C→E System					
M-4	-	32.5	27.7	29.3	27.1	32.5	26.6
MTTK	-	30.6	27.9	27.5	27.0	30.6	26.4
MTTK	√	38.2	28.2	32.3	27.3	37.1	26.8
		Large A→E System					
M-4	-	26.4	38.1	28.1	40.1	28.2	39.9
MTTK	-	24.8	38.1	26.6	40.1	26.7	40.6
MTTK	√	30.7	39.3	32.9	41.6	32.5	41.9

2004 Eval System Architecture – 3-gram LM, monotone phrase order

Results:

- PPI induction improves test set coverage and translation
- PPI induction can be used to improve Model-4 itself (not shown)
- Translation with MTTK is comparable to using Model-4 alignments
  - ∼ 1 - 2 BLEU points improvement in A→E

TTM and MET

MET can be used to optimize the combination of TTM components

- Recast TTM as a log-linear model with scaling factors $\Lambda = \lambda_1^M$

$$\prod_{m=1}^{M} p_m(E, F)^{\lambda_m}$$

- λ's are applied to each WFST in the translation pipeline
- Minimum Error Training (Och 2003) –
  Maximize BLEU over a development corpus:
  - N-best lists used for training
  - Multidimensional search in M dim space by Powell's algorithm
- MET gives good improvement over a state-of-the-art baseline

Training and Translation Pipeline

1. Bitext Chunking
   - 1.1 Monotone alignment into coarse chunks of documents
   - 1.2 Divisive clustering into subsentence chunks
2. MTTK model training, F→E and E→F

⇒ Eval sets arrive ...

3. Extract foreign phrases from the eval sets
   - 3.1 extract phrases from the alignments using the 'usual' heuristics
   - 3.2 use phrase-pair induction under MTTK to augment the PPI
4. Construct component WFSMs for the TTM
5. Viterbi estimation of TTM reordering parameters over training bitext
6. Translation lattice generation with pruned 4-gram
7. Translation lattice rescoring with unpruned 4-gram ⇒ contrast system
8. Minimum Error Training
   - 8.1 transducer weights optimized for BLEU on heldout data (from Eval04)
   - 8.2 rescore lattices from Step 6
   - 8.3 regenerate N-Best lists, add MTTK IBM-1 features, repeat MET, ...
9. MET rescoring of final lattices and N-Best lists ⇒ primary system

Evaluation Systems – Performance and Resources

System Performance - BLEU

System	A→E				C→E			
	02	03	04-N	05 (c)	02	03	04-N	05 (c)
Eval 04 primary	39.4	42.1			28.5	27.4		
PPI Induction+MJ-1	43.1	45.0	45.6	41.3	30.2	28.2	28.9	26.3
PPI Induction+MJ-1+MET	45.2	48.2	49.7	43.5	31.8	30.7	31.0	28.3

- MTTK used for word alignment and phrase pair induction
- Significant improvements relative to 2004 evaluation system
- Additive gains from PPI Induction, Phrase Reordering, and MET

System Resources

	LM text (words)	Bitext (F/E words)	
A→E :	428M	123M / 132M	modified Buckwalter tokenizer
C→E :	373M	176M / 207M	LDC segmenter

Summary: Working Towards Integrated Modeling and Decoding

Modeling: Given an English Sentence e and a French sentence f, construct a joint distribution over their alignments, e.g.

$$P(e, a, f) = \underbrace{P(f|a, e)}_{\text{Translation Model}} \underbrace{P(a|e)}_{\text{Alignment Model}} \underbrace{P(e)}_{\text{Language Model}}$$

Decoding (ideally) : Given f, find a translation $\hat{e}$ and an alignment $\hat{a}$ as

$$(\hat{e}, \hat{a}) = \underset{e,a}{\text{argmax}}\ P(f|a, e)\ P(a|e)\ P(e)$$

Decoding (really) : lacks integrated modeling and decoding

- Models are trained & alignments are generated over the training set
- The models are discarded, and the alignments are kept
- PPI, etc. are extracted from the alignments and used in translation

Goal: tight integration of TTM and MMTK

- same models in alignment and translation – this is 'what works' for ASR
- needed for : MMI, clustering, context dependence, ...

New! – TTM Tutorial is Available

German→English translation based on

- Europarl corpus
- Giza++ alignments
- AT&T FSM Toolkit
  - any FSM toolkit should work ...

Tutorial steps through building and using all the transducers

Very much an alpha version, but available to anybody who's interested