A Weighted Finite State Transducer Implementation of the Alignment Template Model for Statistical Machine Translation

May 29, 2003

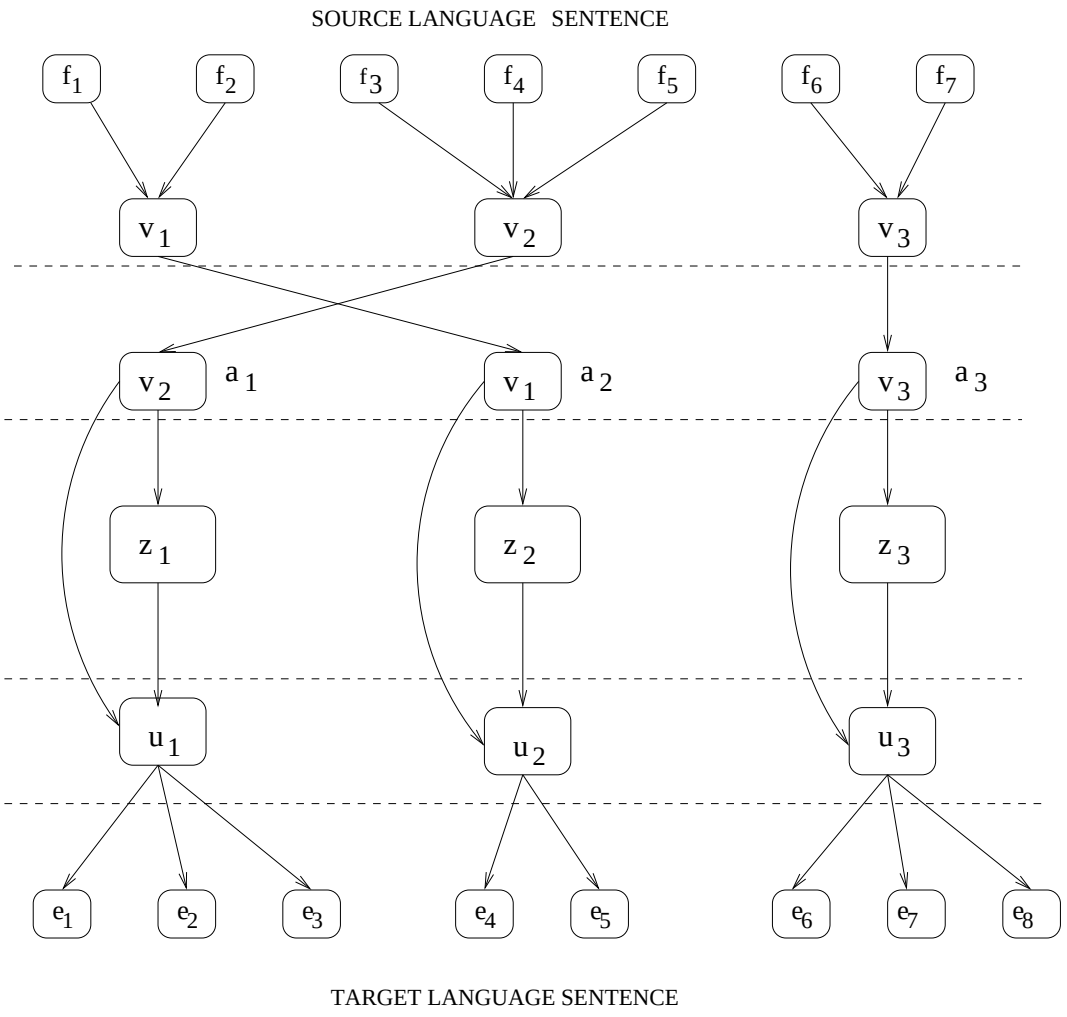
Shankar Kumar and Bill Byrne

Center for Language and Speech Processing
Department of Electrical and Computer Engineering
The Johns Hopkins University, Baltimore, MD, U.S.A.

Motivation and Goals

- Alignment Template Translation Model (ATTM) (Och, Tillmann and Ney '99) has emerged as a promising model for Statistical MT
- Goal Reformulate the ATTM so that bitext-word alignment and translation can be implemented using Weighted Finite State Transducer operations
- Advantages
  - FSM formulation reduces cost of implementation
  - FSM implementation makes it unnecessary to develop a specialized decoder
    - This decoder can even generate alignment/translation lattices and N-best lists
  - FSM architecture provides support for generating bitext word alignments and alignment lattices
    - These will form the basis for ATTM parameter estimation procedures

Alignment Template Translation Model Architecture



Component Models

Segmentation Model

$P(v_1^K, K|f_1^J)$

Phrase Permutation Model

$P(a_1^K|v_1^K, K, f_1^J)$

Template Sequence Model

$P(z_1^K|a_1^K, v_1^K, K, f_1^J)$

Phrasal Translation Model

$P(u_1^K|z_1^K, a_1^K, v_1^K, K, f_1^J)$

Target Language Model

$P(e_1^I)$

ATTM Joint Distribution

$P(e_1^I, u_1^K, z_1^K, a_1^K, v_1^K, K, f_1^J)$

Source Segmentation Model

- The WFST-ATTM assumes that sentence to be translated is first segmented into a sequence of phrases in the word order of the source language
- Segmentation Model A joint distribution over phrase segmentations:

$$P(v_1^K, K|f_1^J) = P(v_1^K|K, f_1^J)P(K|f_1^J)$$

  – Distribution of # of phrases:

$$P(K|f_1^J) = \frac{\binom{J-1}{K-1}}{2^{J-1}}$$

  – Phrase unigram model:

$$P(v_1^K|K, f_1^J) = \begin{cases} \frac{1}{Z_K}\prod_{k=1}^{K} p_u(v_k) & v_1^K = f_1^J \\ 0 & otherwise \end{cases}$$

- Computing optimal segmentation

$$\{\hat{v}_1^{\hat{K}}, \hat{K}\} = \text{argmax}_{v_1^K, K}\, P(v_1^K|K, f_1^J)P(K|f_1^J)$$

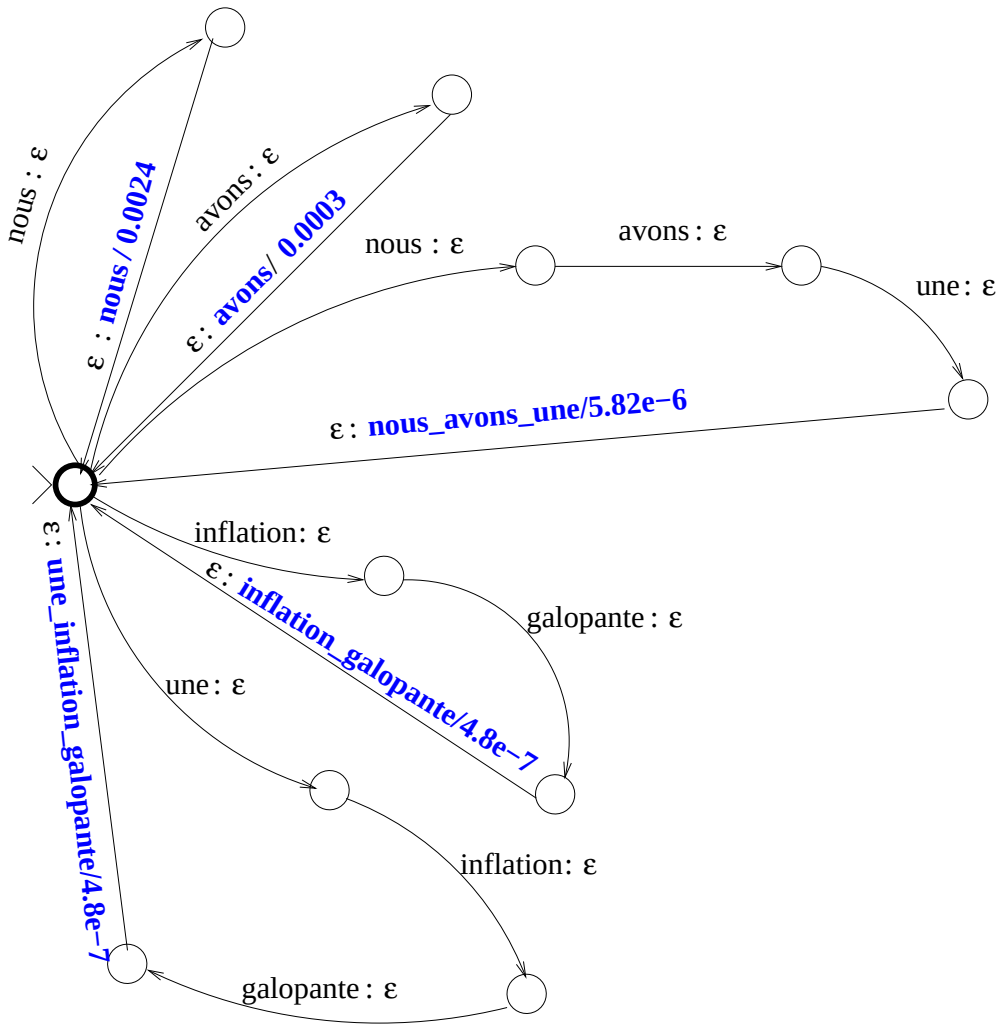
Segmentation Transducer Ω

French Sentence S:

- nous avons une inflation galopante

$S \circ \Omega$:

- nous avons une_inflation_galopante
- nous_avons_une inflation_galopante



Phrase Permutation Model

Reorder the source-language phrase sequence $v_1, v_2, ..., v_K$ into target-language word order $v_{a_1}, v_{a_2}, ..., v_{a_K}$.

1st-order Markov model on permutation indices $a_1, a_2, ..., a_K$

$$P(a_1^K | v_1^K, K, f_1^J) = P(a_1) \prod_{k=2}^{K} P(a_k | a_{k-1}, v_1^K).$$

The Markov-Chain probabilities are assigned so as to penalize reorderings that diverge from original word order (Och '02)

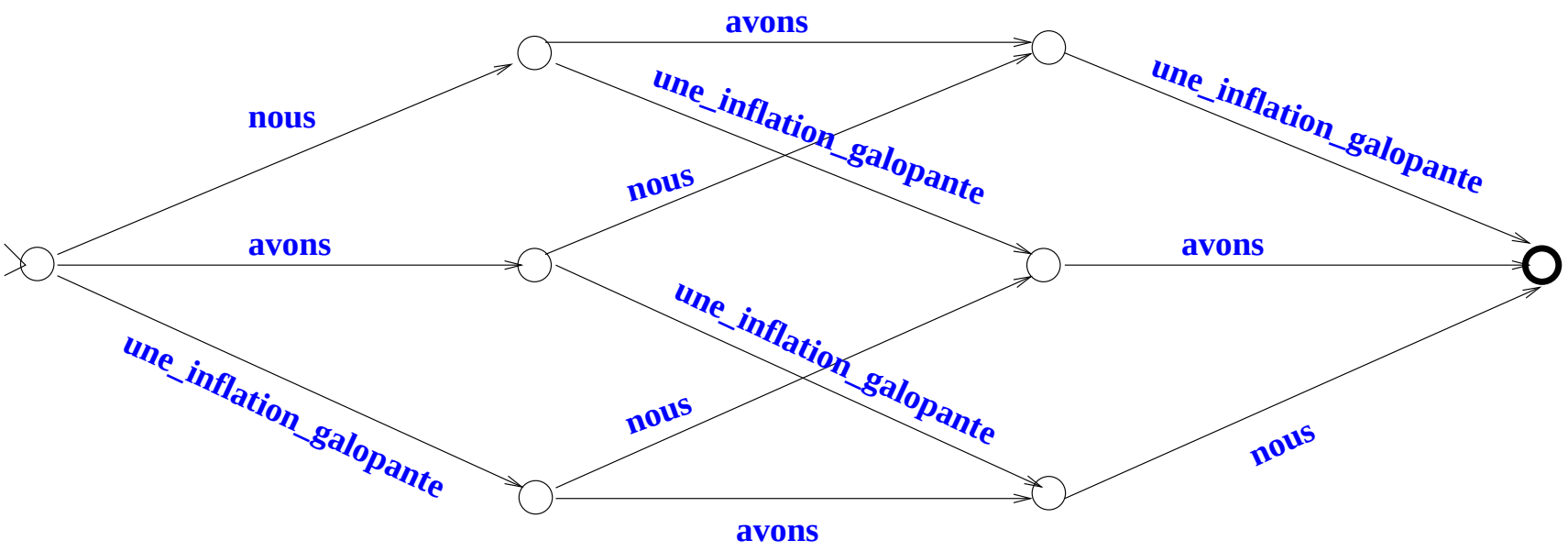
Weighted FSTs to implement the Phrase Permutation Model

- Permutation Acceptor Π_V to generate all permutations of source-language phrase sequence
- Phrase Permutation Model Acceptor H to assign probabilities to any permutation of the source-language phrase sequence

Permutation Acceptor Π_V

- Generates all re-orderings of source-language phrase sequence
- Construction similar to Permutation Acceptor (Knight and Al-Onaizan '98)
- Pruning required while construction (details in paper)

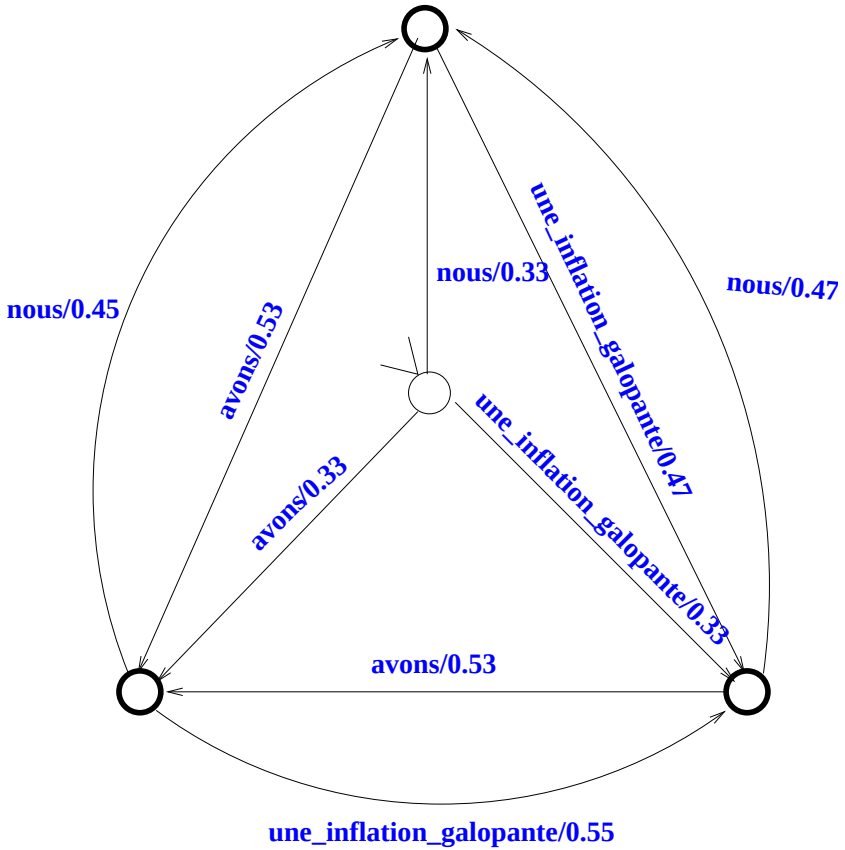
Source Phrase Sequence: nous avons une_inflation_galopante



Phrase-Permutation Model Acceptor H

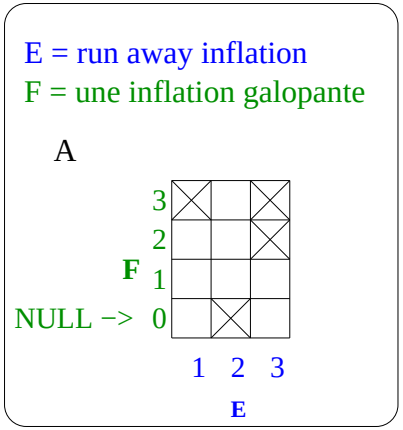
- Acceptor assigns permutation model probabilities to any permutation of the source-language phrase sequence
- The 1st-order Markov-model results in a simple structure for this FSA

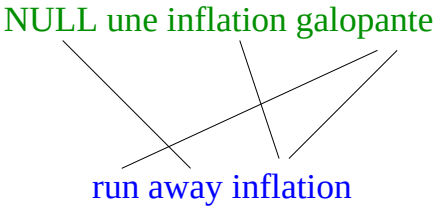
Source Phrase Sequence: nous avons une_inflation_galopante



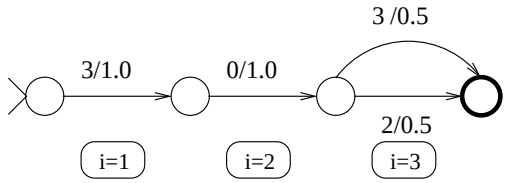
WFST Implementation of Alignment Templates

- Each word in the source & target language is assigned to a unique class
- Alignment Template $z = (E_1^M, F_0^N, A)$ specifies word alignments between class sequences E_1^M and F_0^N through a 0/1 valued matrix A.
  - For simplicity, the presentation is in terms of words and not classes

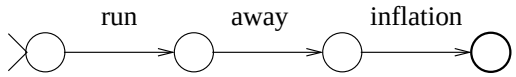




WFST Implementation



O



C

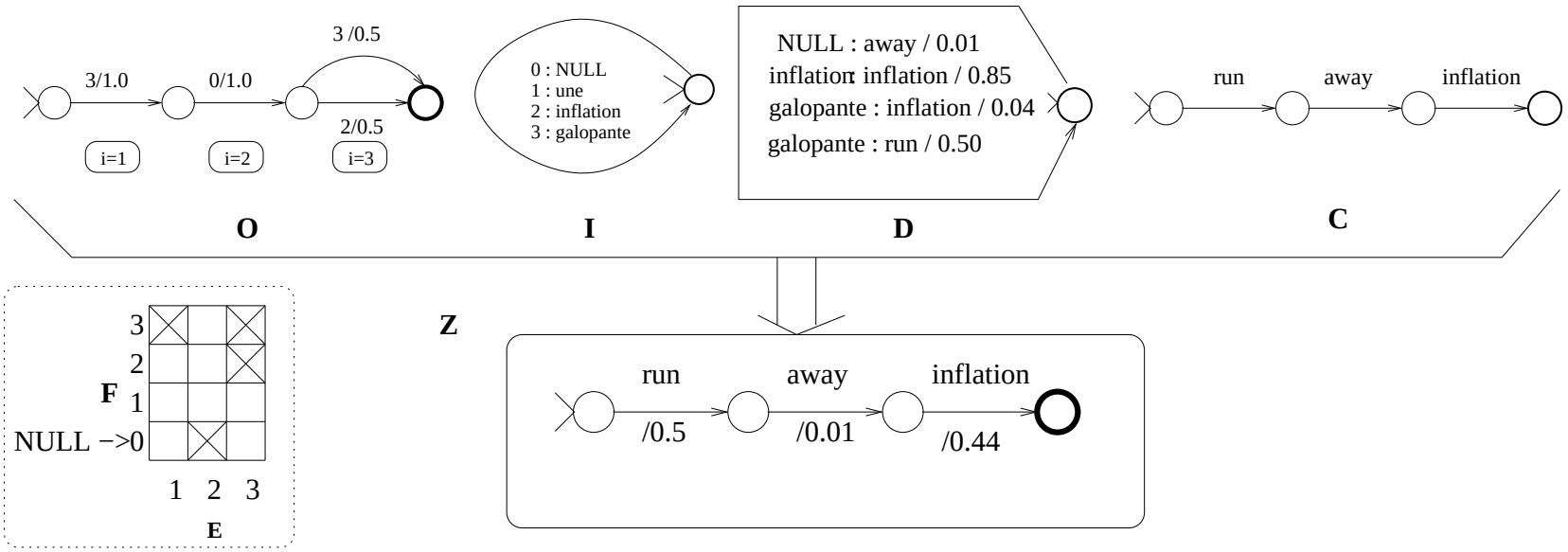
Map source phrase into one or more alignment templates via WFST Y

Phrase-to-Phrase Translation Model

Map an alignment template (and its associated source phrase) to a consistent target phrase

$$P(u|v,z) \quad u = e_1^M, v = f_0^N$$

- Build an Acceptor Z that assigns scores $P(u|v,z)$ for a given z and v



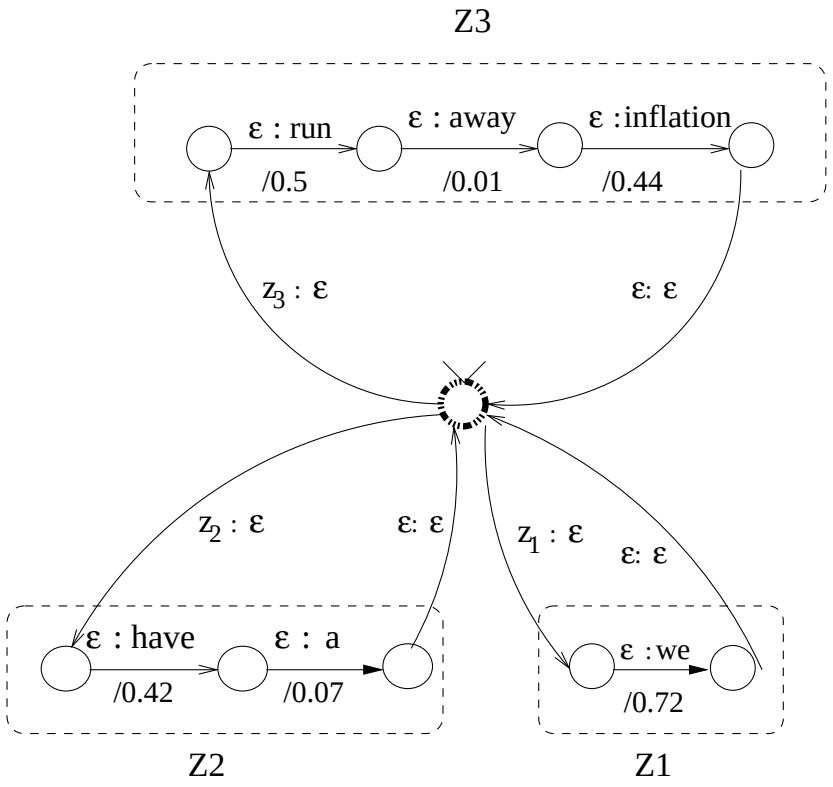
- $P(u|z = (E_1^M, F_0^N, A), v) = \prod_{i=1}^{M} \sum_{j=0}^{N} P(\phi_i = j|A) P(e_i|f_j)$
- $Z = \text{Determinize}_{+,X}(\text{Project-output}(O \circ I \circ D \circ C))$

Phrasal Translation Model

- Realize the alignment template library as a single transducer
- Restrict the transducer to templates consistent with phrases in the source-language phrase sequence v_1^K

A sample template library

z_1	nous $\Rightarrow$ we
z_3	avons $\Rightarrow$ have_a
z_2	une_inflation_galopante $\Rightarrow$ run_away_inflation



- $P(u_1^K|z_1^K, a_1^K, v_1^K, K, f_1^J) = \prod_{k=1}^{K} P(u_k|z_k, v_{a_k})$

An Overview of WFSTs for ATTM

Given a source sentence f_1^J we first segment it into K-length phrase sequence $\hat{v}_1^K$

- Permutation Acceptor $\Pi_{\hat{V}}$ allows permutations of phrase sequence
- Permutation Model Acceptor H that assigns permutation probabilities to any re-ordering of source-phrase sequence
- WFST Y that maps source phrases to templates
- WFST W that maps templates to target-language phrase sequences

Build a transducer that maps the source language phrase sequence to target language phrase sequences

$$X = \Pi_{\hat{V}} \circ H \circ Y \circ W$$

Bitext Word Alignment and Translation Via WFSTs

Alignment of a sentence pair f_1^J,e_1^I

$$\{\hat{u}_1^{\hat{K}}, \hat{z}_1^{\hat{K}}, \hat{a}_1^{\hat{K}}, \hat{v}_1^{\hat{K}}, \hat{K}\} = \underset{u_1^K, z_1^K, a_1^K, v_1^K, K}{\operatorname{argmax}} P(u_1^K, z_1^K, a_1^K, v_1^K, K | e_1^I, f_1^J)$$

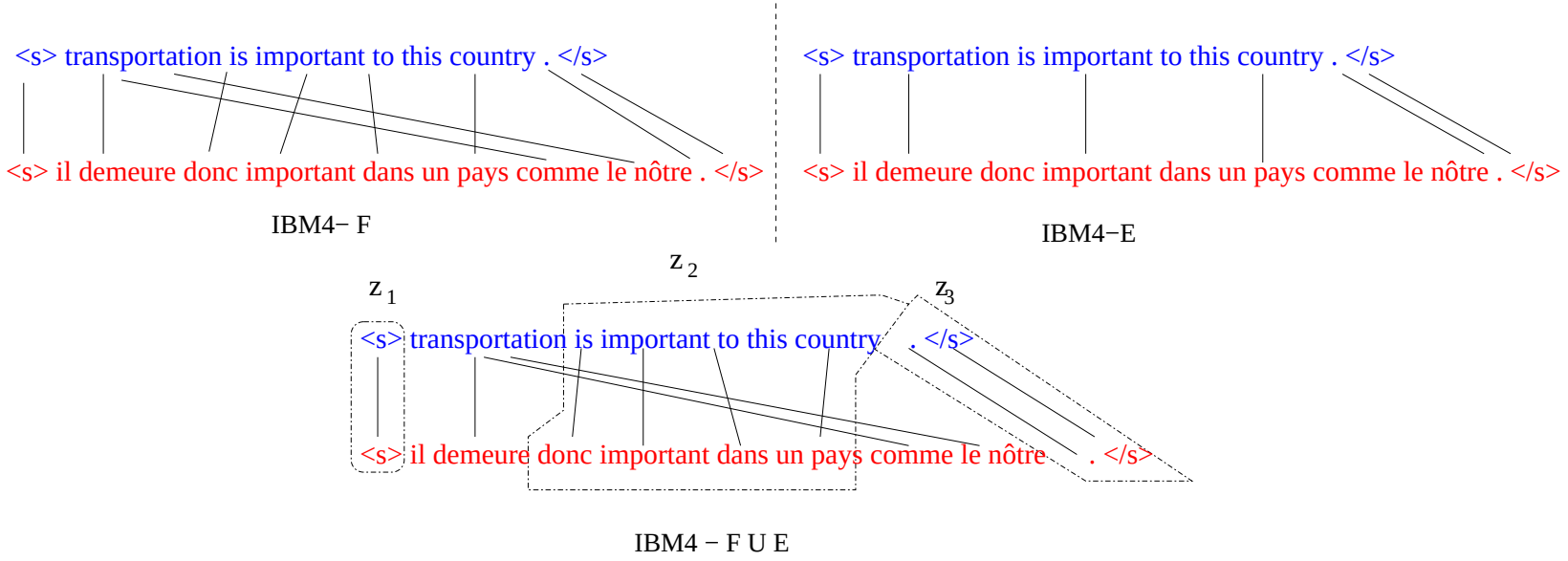
Translation of a source sentence f_1^J

$$\{\hat{e}_1^{\hat{I}}, \hat{u}_1^{\hat{K}}, \hat{z}_1^{\hat{K}}, \hat{a}_1^{\hat{K}}, \hat{v}_1^{\hat{K}}, \hat{K}\} = \underset{e_1^I, u_1^K, z_1^K, a_1^K, v_1^K, K}{\operatorname{argmax}} P(e_1^I, u_1^K, z_1^K, a_1^K, v_1^K, K | f_1^J)$$

- Stage 1: Segment source sentence f_1^J into phrase sequence v_1^K
- Stage 2: Alignment Via Finite State Composition
  - Build Acceptor T for the Target sentence e_1^I
  - Obtain the alignment lattice $\mathcal{B} = X \circ T$
- Stage 2: Translation Via Finite State Composition
  - Compile a n-gram language model as a weighted acceptor G
  - Obtain the translation lattice $\mathcal{T} = X \circ G$

Bitext Word Alignments Under ATTM

Generate alignment template library from IBM-4 alignments (Och '02)



- The model does not guarantee alignment between every sentence pair in bitext and most sentence pairs are assigned zero probability
- *Solution*: Add "dummy" templates to the template library
  - These templates allow alignment of any source phrase to any target word, deletion of source phrases and insertion of target words
  - The probability of "dummy" templates is fixed to a value that discourages their use except when regular templates are unavailable

Alignment and Translation Performance

Experiment Setup (Och and Ney '00)

- Task: French to English Canadian Hansards
- Training Bitext: $50,000$ sentence pairs
- Test Set: 500 unseen sentence pairs

Alignment Performance

Model		Alignment Metrics (%)		
		Precision	Recall	AER
IBM-4 F		88.9	89.8	10.8
IBM-4 E		89.2	89.4	10.7
IBM-4 $F \cup E$		84.3	93.8	12.3
ATTM	- Include "Dummy" Links	64.2	63.8	36.2
	- Exclude "Dummy" Links	94.5	55.8	27.3

Translation Performance

Model	BLEU	NIST	WER (%)
IBM-4	0.1711	5.0823	67.5
ATTM	0.1941	5.3337	64.7

Oracle-Best BLEU scores on N-best Lists

N	Oracle-best BLEU
1	0.1941
10	0.2264
100	0.2550
400	0.2657
1000	0.2735

Conclusions and Future Work

- Salient features of ATTM WFST Modeling Framework
  - Simple translation process compared to DP/A^* decoders
    - Generating lattices/N-best lists requires no additional effort
  - A novel approach to generate bitext word alignments and alignment lattices
    - Facilitate development of ATTM parameter estimation procedures
  - Modular implementation of component distributions
- Future Work
  - Iterative parameter estimation for ATTM
  - New Strategies for Template Selection to improve coverage on test set
  - Decoupling of segmentation and alignment/translation may lead to search errors
  - Refining the component distributions