A Log-linear Block Transliteration Model based on Bi-Stream HMMs

Bing Zhao

Joint work with

Nguyen Bach, Ian Lane, and Stephan Vogel

Language Technologies Institute

Carnegie Mellon University

April 2007

OOV-words in Machine-Translation

- **Machine Translation systems are closed vocabulary**
  - Translation hypotheses cannot be generated for any source word that did not appear in training corpora
- **Rejecting OOV words will drastically degrade the quality & usability of translation**
  - OOV words often major components of semantic content

    i.e. Named-Entities (Person/Place names)

To generate semantically equivalent translations OOV words must also be accurately translated

- **Improve not only translation usability but also effectiveness of multilingual applications**

Transliteration for Machine Translation

- **In large-vocabulary SMT systems OOV-words are typically person or place names**

  → these words can be accurately translated via transliteration

Source Language	English Transliteration
German: K**onstantino**polis	Constantino**ple**
Arabic: دمشق **(Dmk)**	Damascus
Spanish: **Adelaid**a	Adelaid**e**

Transliteration of place-names for different language pairs

- **Difficulty of transliteration dependent on language pair**
  - **Arabic → English**
    - Vowels must be hypothesized
    - Ambiguity arises due to multiple possible transliterations

i.e: **خفاجي** → **xfAjy** → Mahasin / Muhasan / Mahsan

Arabic Script — Romanized — English Transliteration

Machine Transliteration: Previous Works

- **Rule-based approaches**
  - Rule-set either manually defined or automatically generated

  → Only appropriate for close language-pairs

  (*poor performance for Arabic→English transliteration*)

- **Statistical approaches**
  - Finite state transducers (Knight & Graehl 1997, Stalls & Knight, 1998)
  - Model combination (Al-Onaizan 2002, Huang, 2005)

  → Specific approach typically limited to target language pair

- **Transliteration as Statistical-Machine-Translation**
  - Highly portable framework
    - Only require transliteration examples (i.e. from Bilingual dictionary)
  - Able to generate high quality transliterations
    - Outperforms rule-based approaches language pairs with high ambiguity

Transliteration-specific SMT

- **Define phonetic and position-dependent letter classes**
  - Broad phonetic classes consistent across languages

    i.e. transliterate: consonant → consonant, vowel → vowel
  - <u>**Propose Bi-Stream HMM framework**</u> to estimate both letter and letter-class
- **Constrain fertility**
  - Typically, number of letters similar across language-pair
  - Constrained fertility for Arabic → English
- **Force monotonicity**
  - Phonetic reordering does not occur in transliteration
- **Perform transliteration via "*transliteration-blocks*"**
  - Improve handling of context during transliteration
  - <u>**Propose "*block-level*" transliteration framework**</u>

Multiple features combined via Log-linear model

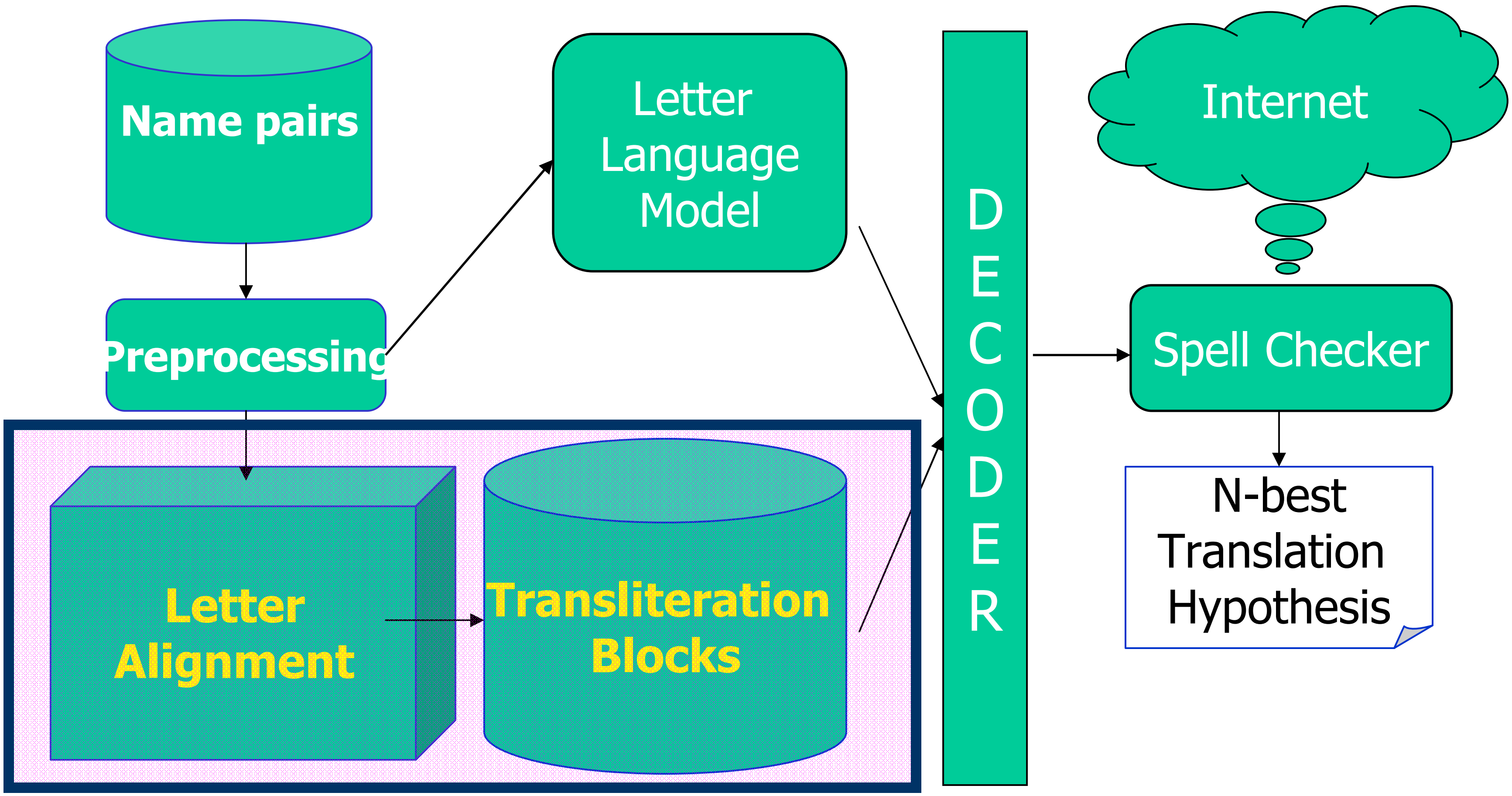
Transliteration-specific SMT

Proposed Framework

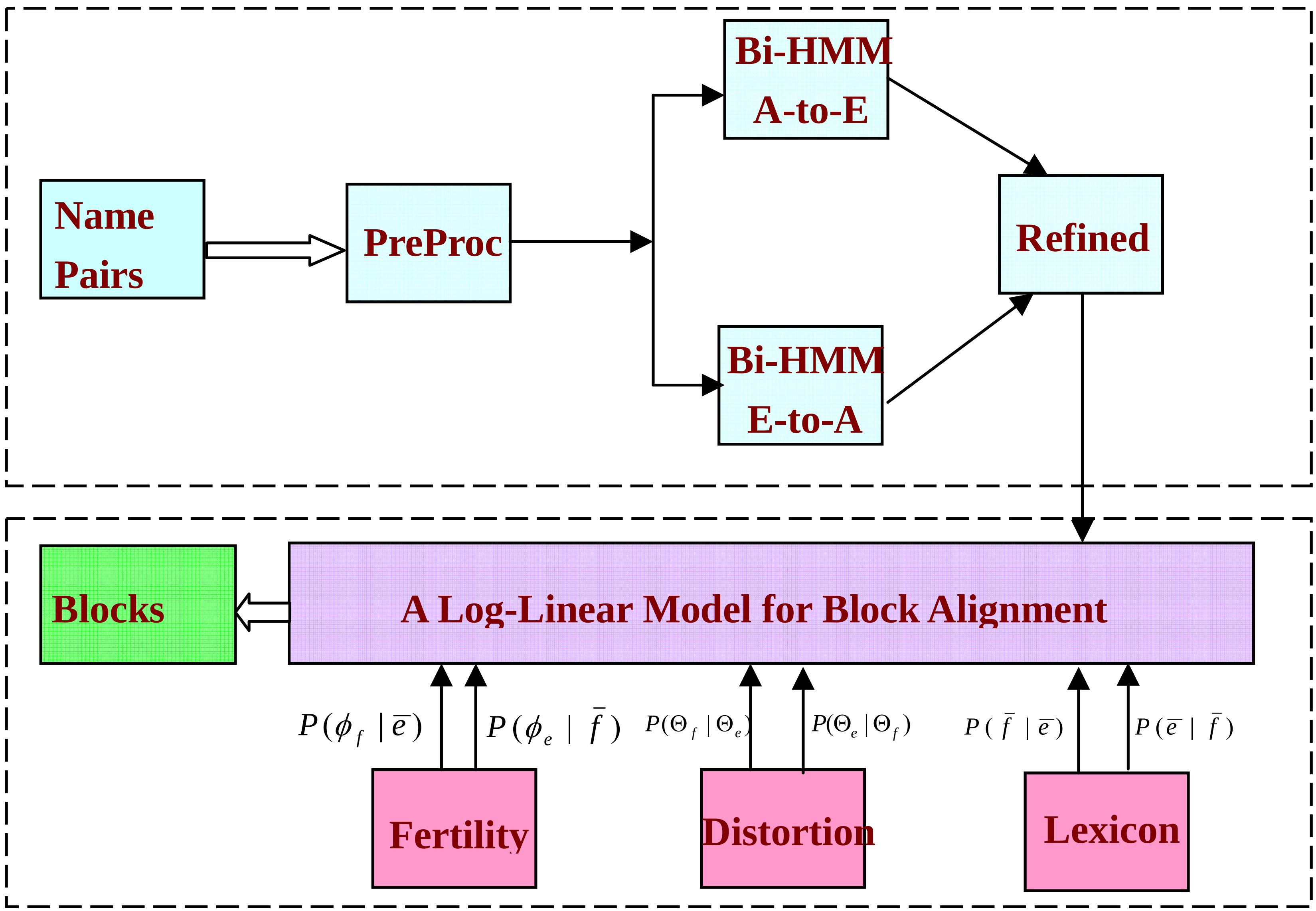
Outline

- **Transliteration as Translation (T.a.T)**
- **Models for Block Transliteration**
  - IBM-Model-4
  - Bi-Stream HMM
  - Bi-Stream HMM combined with a Log-linear model
- **Transliteration of Unseen Named-Entities**
  - Special setups for transliterations
  - Configurations of SMT decoder
  - Spelling checker
- **Conclusions and Discussions**

System Architecture



Alignment for Transliteration



Letter-classes in Bi-stream HMM (I)

- **English Pronunciation is structured**
  - *CVC*: Consonant-Vowel-Consonant
- **Defining Non-Overlapping Letter classes**
  - Vowels: a e i o u ….
  - Consonants: k j l ….
  - Ambi-class: can be both vowel and consonant, e.g "y"
  - Unknown: letters without linguistic clues
    - numbers like 'III'
    - punctuations like '-'
    - typos in the names
  - Additional position markers: initial & final

From HMM to Bi-Stream HMM (II)

- **Monotone nature in letter alignment**
  - From left to right letter-level alignment
- **Bi-Stream HMM**
  - Enriched with letter classes
  - Generating letter sequence
  - Generating letter-class sequence
- **Configure Transition Probability**
  - Configured for strict monotone alignment

From HMM to Bi-Stream HMM (III)

$$\Pr(f_1^J \mid e_1^I) = \sum_{a_1^J} \prod_{j=1}^{J} p(f_j \mid e_{a_j}) p(a_j \mid a_{j-1})$$

Name-Pair

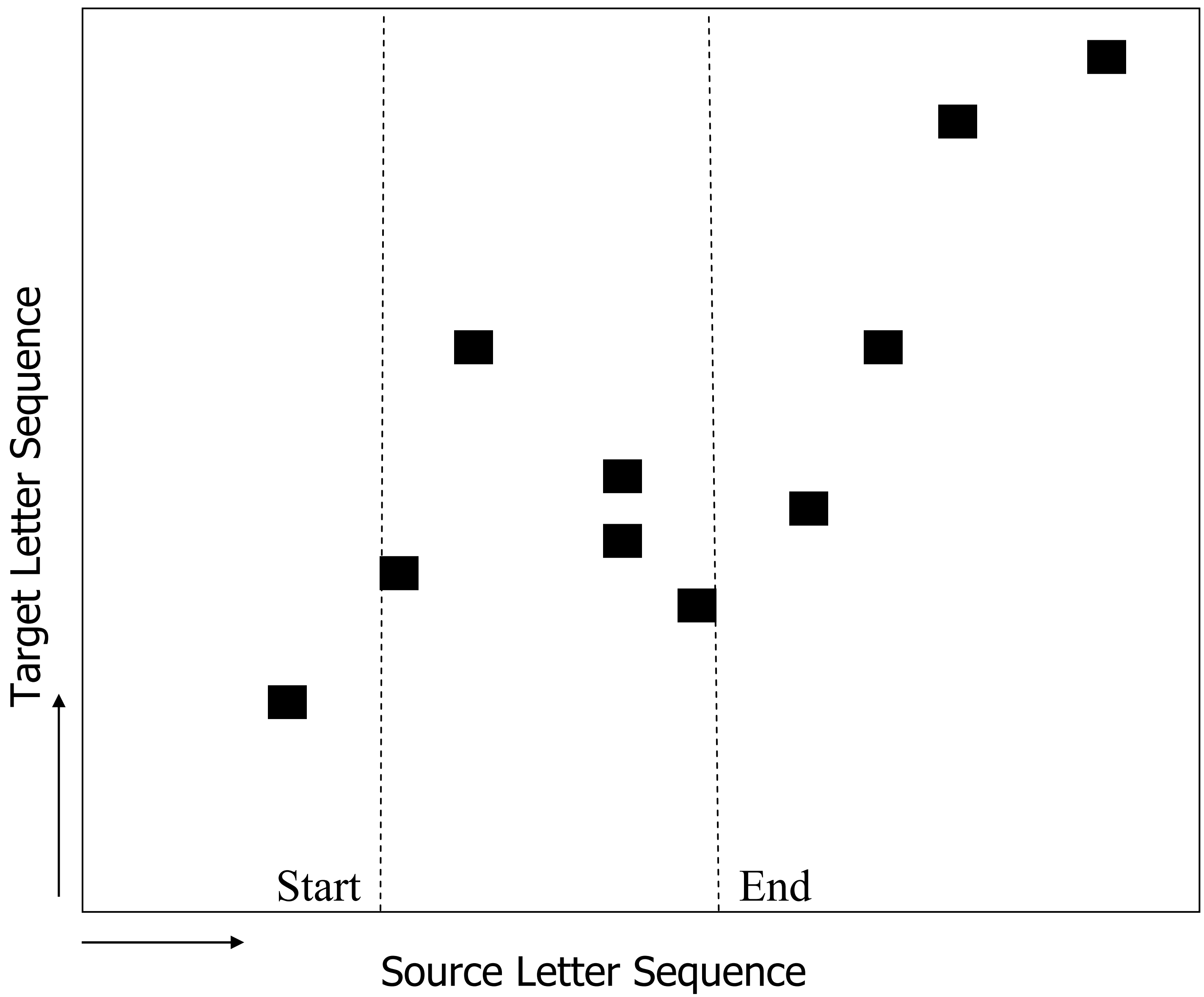
Letter-transliteration

State-Transition

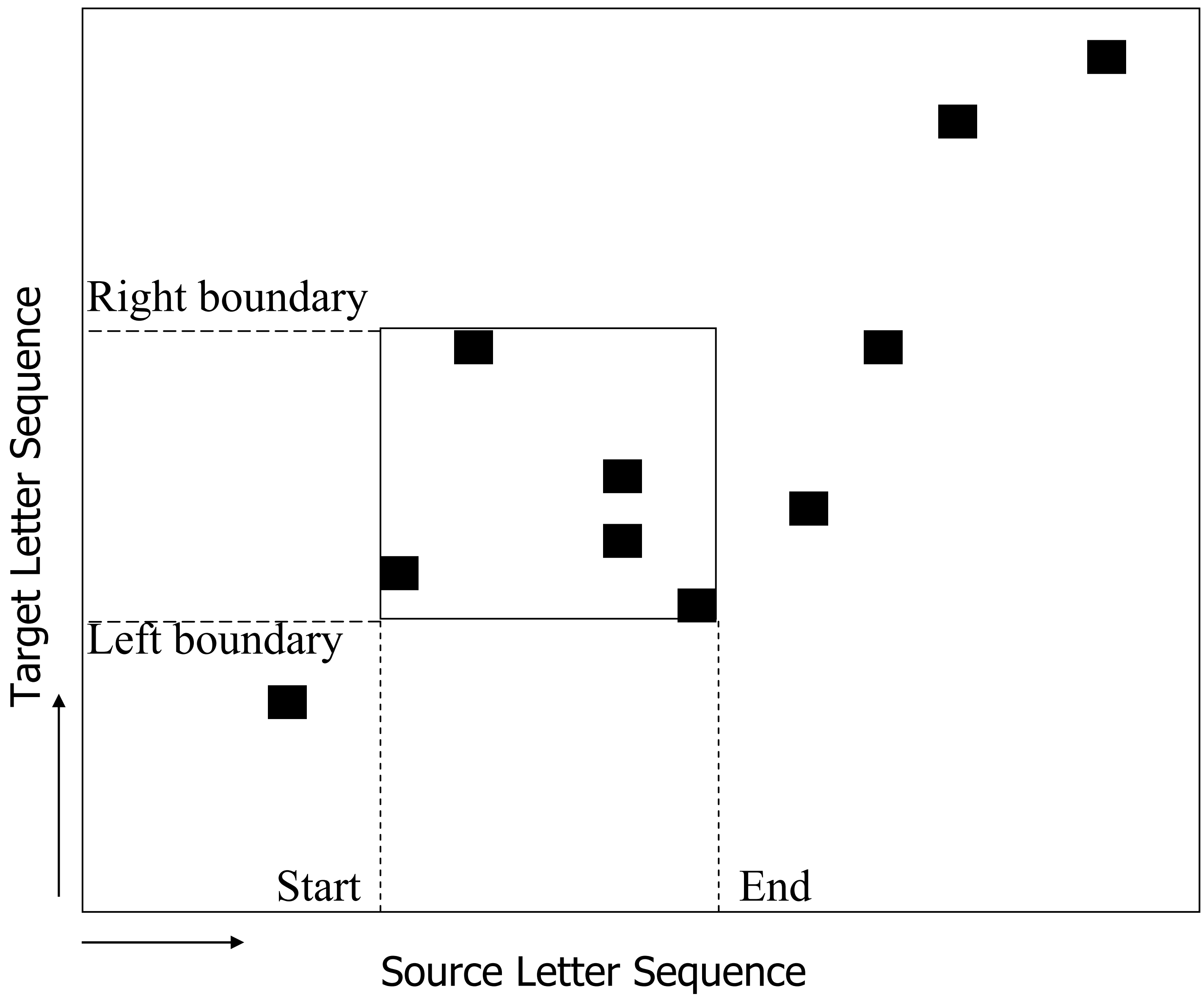
$$\Pr(f_1^J, F_1^J \mid e_1^I, E_1^I) = \sum_{a_1^J} \prod_{j=1}^{J} p(f_j \mid e_{a_j}) p(F_j \mid E_{a_j}) p(a_j \mid a_{j-1})$$

$$a_j - a_{j-1} \geq 0$$

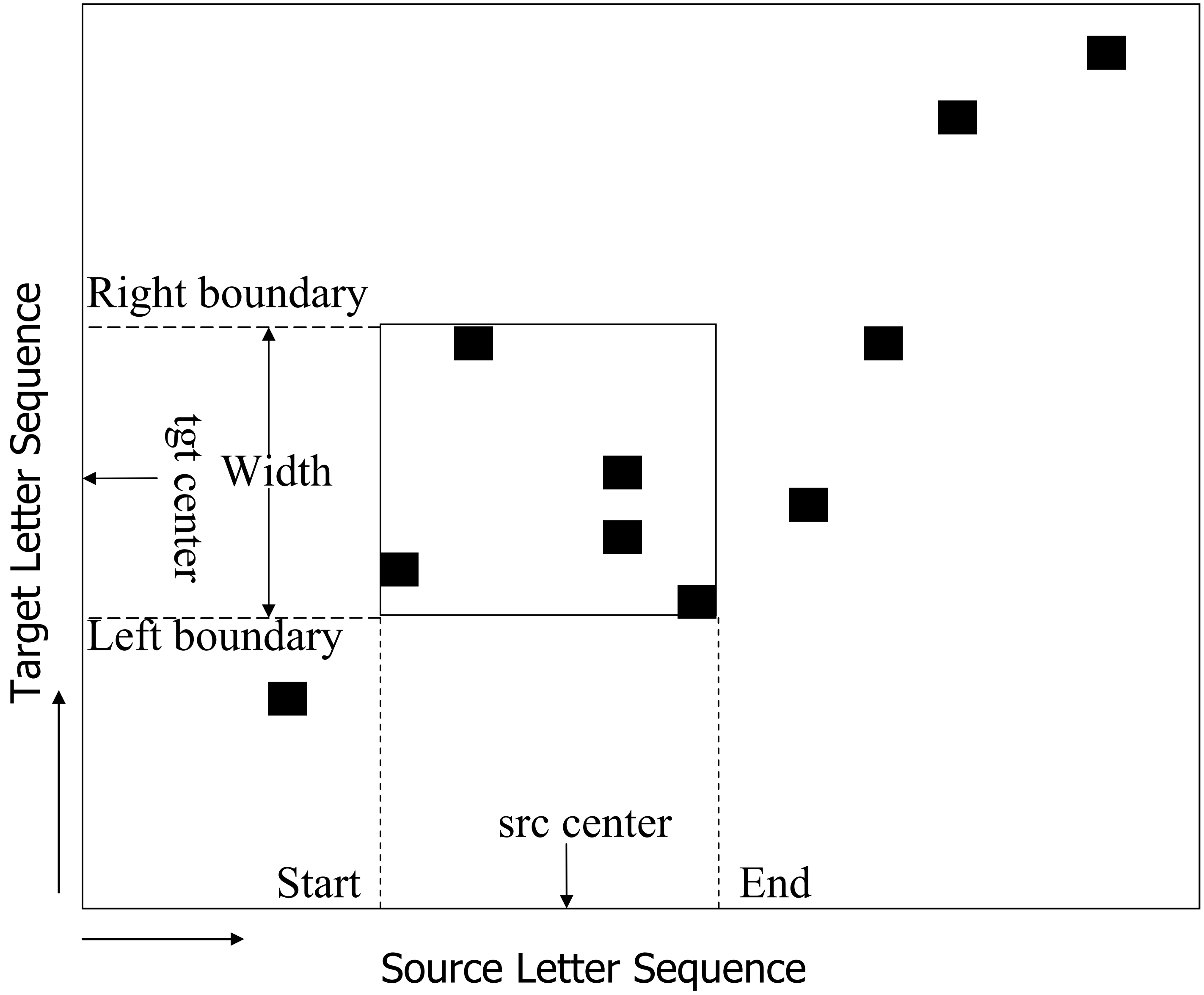
Block Extraction from Letter Alignment



Block Extraction from Letter Alignment

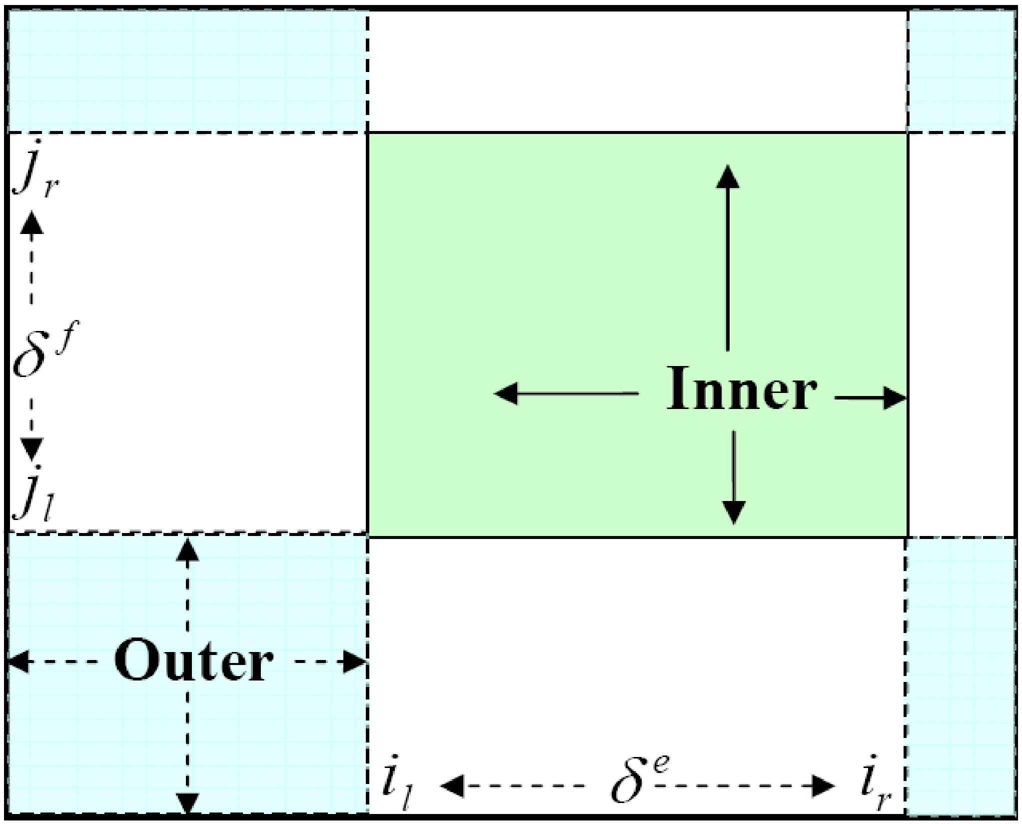


Block Extraction from Letter Alignment



Feature Functions by a Block (1)

- Two main non-overlapping parts: inner & outer
- Both parts should be explained well



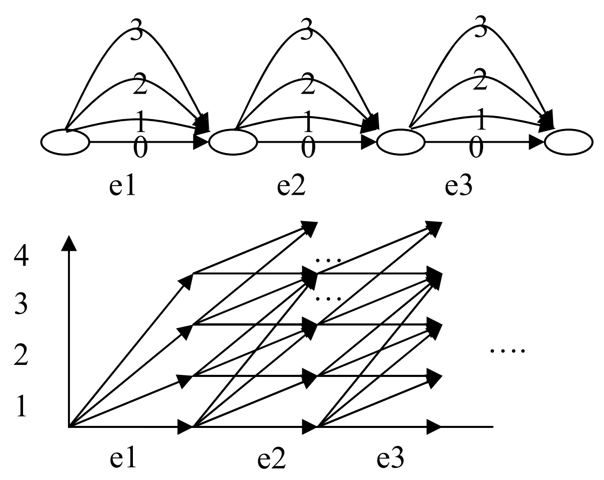
Feature Functions by a Block (2)

- **Length relevance**
  - Letter level fertility probability
  - A dynamic programming
- **Letter n-gram lexicon scores**
  - IBM-1 letter-to-letter transliteration prob.
  - IBM Model-1 style score for named-entity pair
- **Distortions of the letter n-gram centers [inner only]**
  - Letter n-gram pairs are assumed along the diagonal
  - Gaussian distribution for the centers' positions

Feature functions are computed for both **Inner** and **Outer** parts, and in both directions

Length Relevance Score

- **Motivations**
  - Name-pairs usually have similar lengths in characters;
  - A letter is transliterated into less than 4 letters.
- **Length Relevance Score**
  - How many letters we want to generate in the target name;
  - Letter fertilities in both direction.
- **Dynamic Programming**
  - Compute length relevance



Letter N-gram Lexicon Score

- **Motivations**
  - Letter to letter transliteration probabilities
  - Letter to letter mapping is captured by lexicons
- **Transliteration Prob.**
  - Compute statistics from letter alignment
  - Learn lexicons in both directions
- **Name-Pair Transliteration score**
  - Compute IBM Model-1 style scores:

$$\Pr(\vec{e} \mid \vec{f}) = (\frac{1}{I})^J \prod_j \sum_i \Pr(f_j \mid e_i)$$

$$\Pr(\vec{f} \mid \vec{e}) = (\frac{1}{J})^I \prod_i \sum_j \Pr(e_i \mid f_j)$$

Distortions of the letter n-gram centers

- **Motivations**
  - Monotone alignment nature for name-pairs
  - Aligned n-gram pairs are mostly located along the diagonal

- **Position relevance for ngram-pairs**
  - The center of the block should be along the diagonal
  - Define the centers for source and target letter-ngrams:

$$\odot_{e_i^{i+k}}(f_{j'}) = \frac{1}{|E|} \cdot \frac{\sum_{i'=i}^{(i+k)} i' \cdot P(f_{j'}|e_{i'})}{\sum_{i'=i}^{(i+k)} P(f_{j'}|e_{i'})}$$

$$\odot_{f_j^{j+l}} = \frac{1}{|F|} \sum_{j'=j}^{j'=j+l} \frac{j'}{l+1}$$

- **Gaussian Distribution**

$$\begin{aligned} P(\odot_{f_j^{j+l}} | \odot_{e_i^{i+k}}) &\approx P(\odot_{f_j^{j+l}} - \odot_{e_i^{i+k}}) \\ &= N((\odot_{f_j^{j+l}} - \odot_{e_i^{i+k}}); \mu, \sigma) \end{aligned}$$

Learning a log-linear model

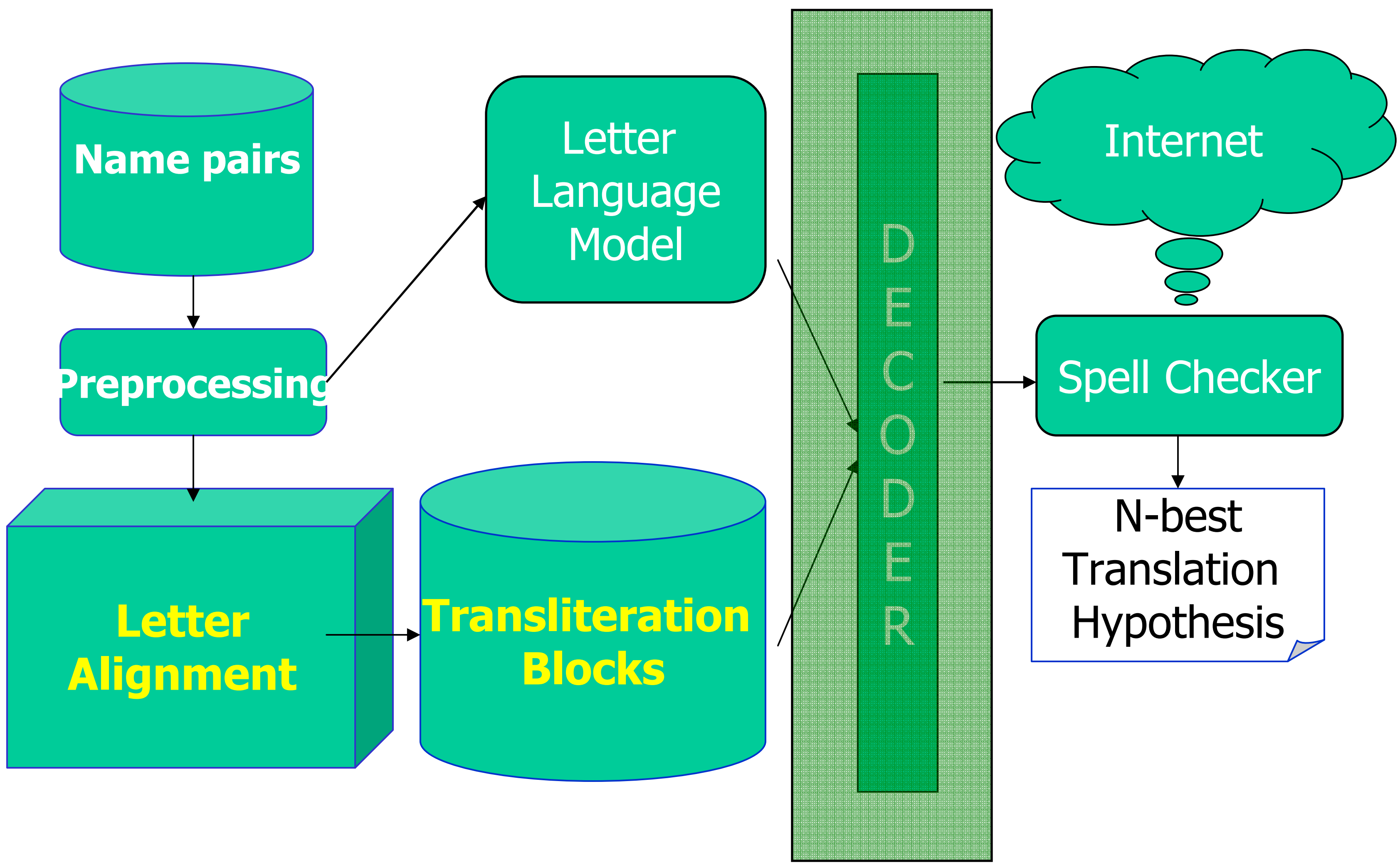
- **Gold standard blocks from human labeled data**
- **Log-linear model to combine feature functions:**

$$Pr(X|\mathbf{e},\mathbf{f})=\frac{\exp(\sum_{m=1}^{M}\lambda_m\phi_m(X,\mathbf{e},\mathbf{f}))}{\sum_{\{X'\}}\exp(\sum_{m=1}^{M}\lambda_m\phi_m(X',\mathbf{e},\mathbf{f}))},$$

- **Model parameters:** $\{\lambda_m\}$
  - Weights for particular feature functions

- **Learning algorithm:**
  - Improved Iterative Scaling
  - Simplex downhill

System Architecture

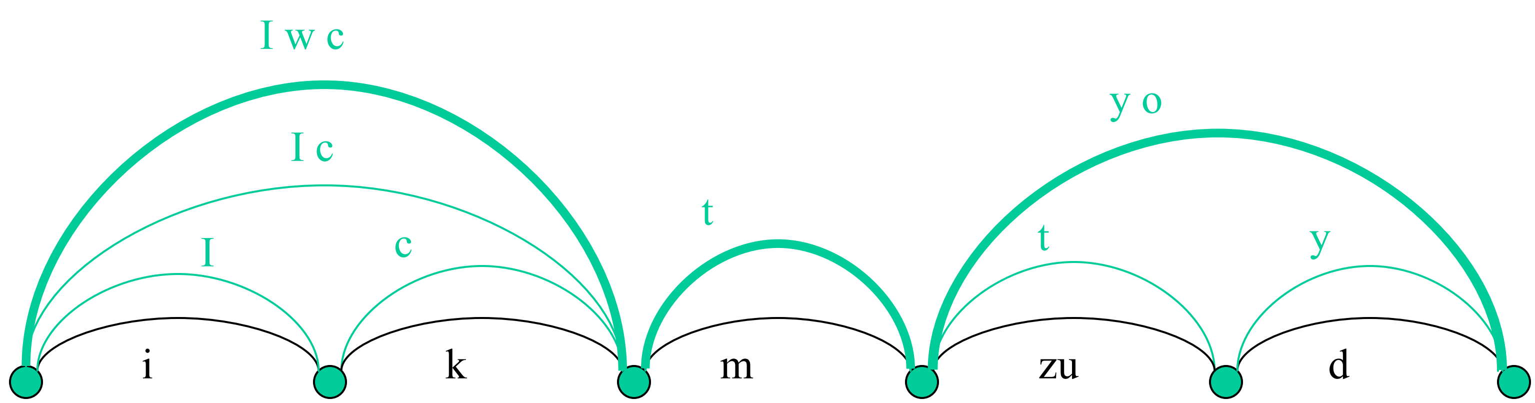


Decoding Transliteration Lattice

Source: ***i k m zu d***

Target: ***I w c t y o***

- **Search in corpus for Transliteration Blocks**
- **Insert edges into the lattice**



Experiments

Experiments

- **Training and Test data sets**
- **Evaluation metric**
- **Comparisons across systems**
  - Three systems
  - Applying a spelling checker
  - Simple Comparison with Google Translations
  - Some examples for MT output

Training and Test Data

Corpus	Size	Type
LDC2005G01-NGA	74K	Bilingual geographic names
LDC2005G021	11K	Bilingual person names
LDC2004L02	6K	Bulkwalter Arabic Morph

- 91K name-pairs training dataset
- 100 name-pairs development dataset
- 540 *unique* name-pairs as the held-out dataset
- 97 *unique* name-pairs from MT03 NIST-SMT eval.

Additional Test Data (II)

- **Blind test set: Arabic-English Tides 2003**
  - ***286*** unique tokens were left **un-translated**
  - Among them: **97** un-translated unique person, location names

Arabic	BAMA	Reference
غرابو	grAbw	Grabo
قشطة	qXTp	Qishta
فايتساخر	fAytsAxr	Weizsacker
والدهماني	wAldHmAny	al-Dahmani
زيلويغر	zylwygyr	Zellweger
ثاكسين	vAksyn	Thaksin

Experimental Setup (I)

- **System-1 (Baseline)**
  - IBM Model-4 in both directions
  - Refined letter alignment
  - Blocks are extracted according to heuristics
- **System-2 (L-Block)**
  - IBM Model-4 in both directions
  - Refined letter alignment
  - Blocks are extracted according to a log-linear model
- **System-3 (LCBE)**
  - Bi-stream HMM in both directions
  - Refined letter alignment
  - Blocks are extracted according to a log-linear model
- **Evaluation method:**
  - Edit-Distance between hyp against possibly multiple references

    Src = "mHmd" Ref = Muhammad / Mohammed

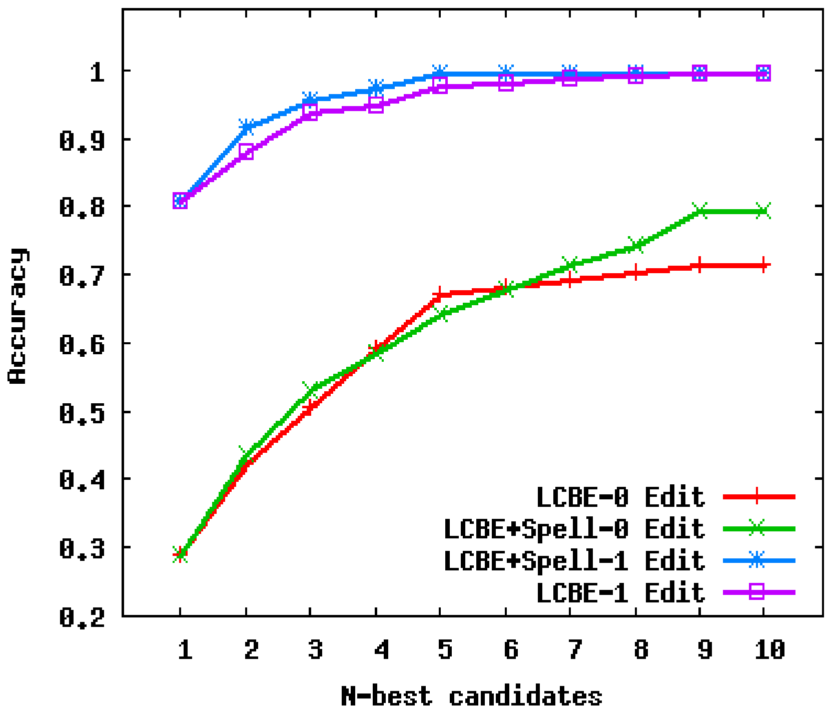
    Acceptable translation if edit distance = 1
    Perfect match if edit distance = 0

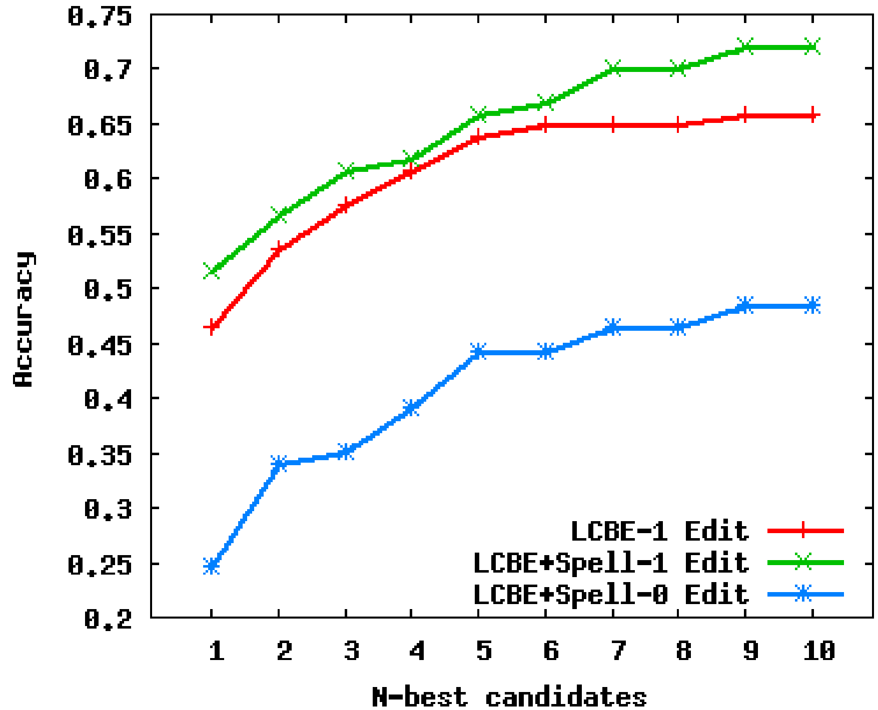
Experiments for the unseen MT03

System	Accuracy
Baseline	39.2%
L-Block	41.3%
LCBE	46.4%
LCBE+Spell	52%

- **Log-linear Block extraction: +2.1%**
- **Bi-stream HMM with letter-classes: +5.1%**
- **Spelling checker: +3.6%**

Experiments for Held-out and Test data





- **Held-out set 540 uniq names**
  - Perfect/Exact match
  - Edit-distance of 1
- **Unseen set (MT03) 97 uniq names:**
  - Perfect/Exact match
  - Edit-distance of 1

Comparing with Google v.s. T.a.T

- **The Arabic-English Google Web Translation (Google)**
- **Accuracy 45% (as in June 20, 06) for the 1-best hypothesis while our system archives 52%**

Source	Reference	T.a.T	Google
سومای	*Sumaye*	*Sumaye*	*Somai*
هازوميتسو	Hazumitsu	Hazumitsu	Hazoumitso
يلاه	*Yalahow*	*Ylahu*	*Elaho*
نكباخت	Nikbakht	Nkbakht	Nkbacht
ميكوياس	Mikulas	Mikulas	Mikoias
كومارا تونج	Kumaratunga	Kumaratunga	Kumaratung
همدان	*Hamdan*	*Hamdan*	*Hamedan*
لمازانداران	Mazandaran	Mazandaran	Mazandaran
ويكرمسينغه	Wickremasinghe	Wikramsinghe	The Ekermsingh

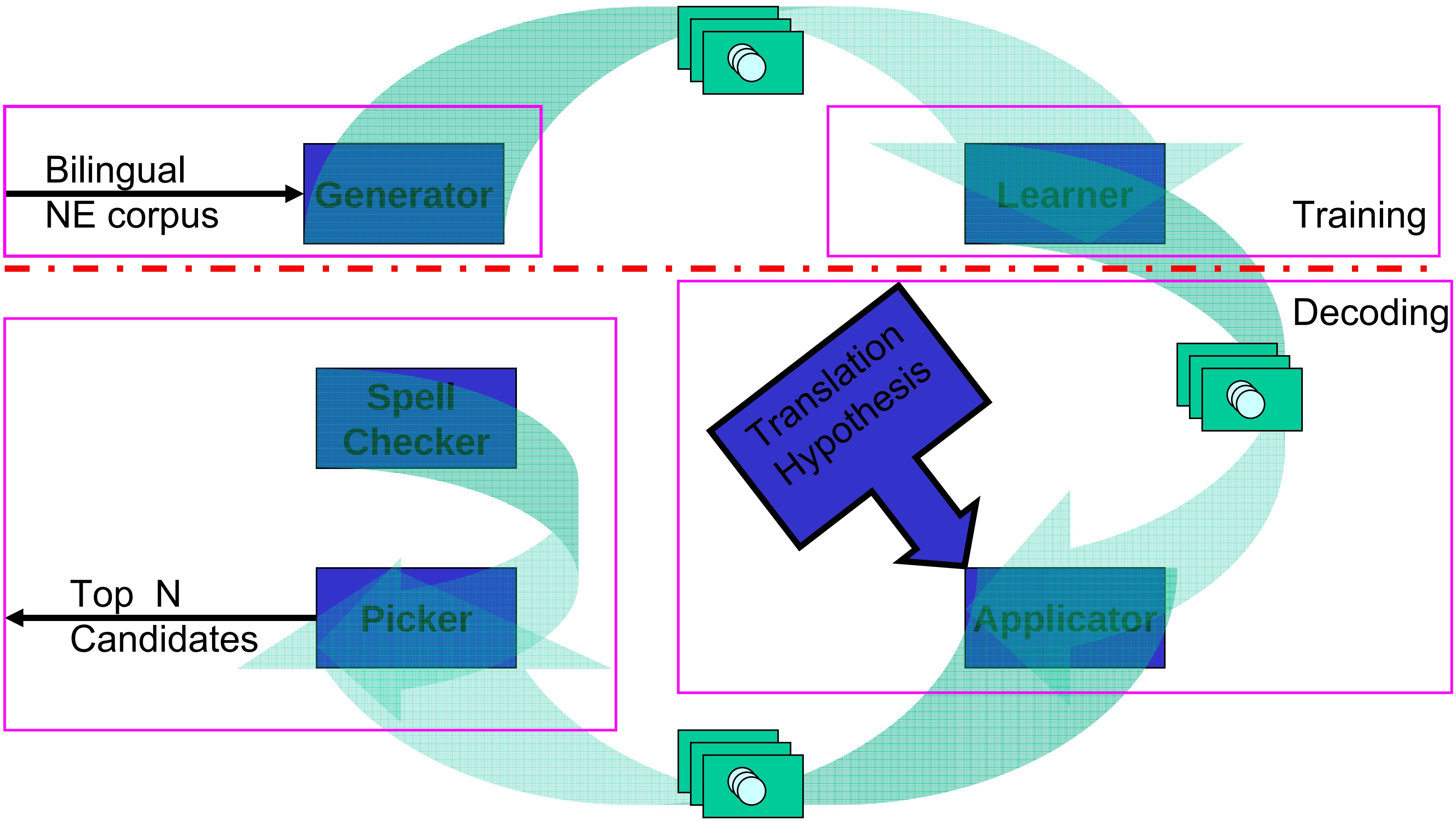
Conclusion & Future Work

- **A transliteration system using available SMT sys**
- **The result is comparable with the state-of-the-art systems**
  - Significantly better than Rule based system (52% v.s. 14%)
  - Log-linear model, Bi-stream HMM, and Spelling checker
- **Future extensions**
  - System re-configurations for other language pairs
  - New features for transliterations
  - Models for letter alignment for transliteration
  - Algorithms for extracting letter n-gram pairs for transliteration

Thanks!

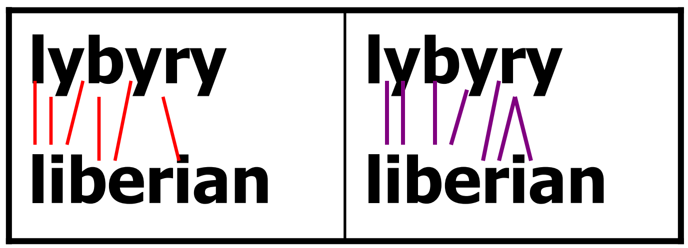
Questions?

Rule-based Architecture Overview



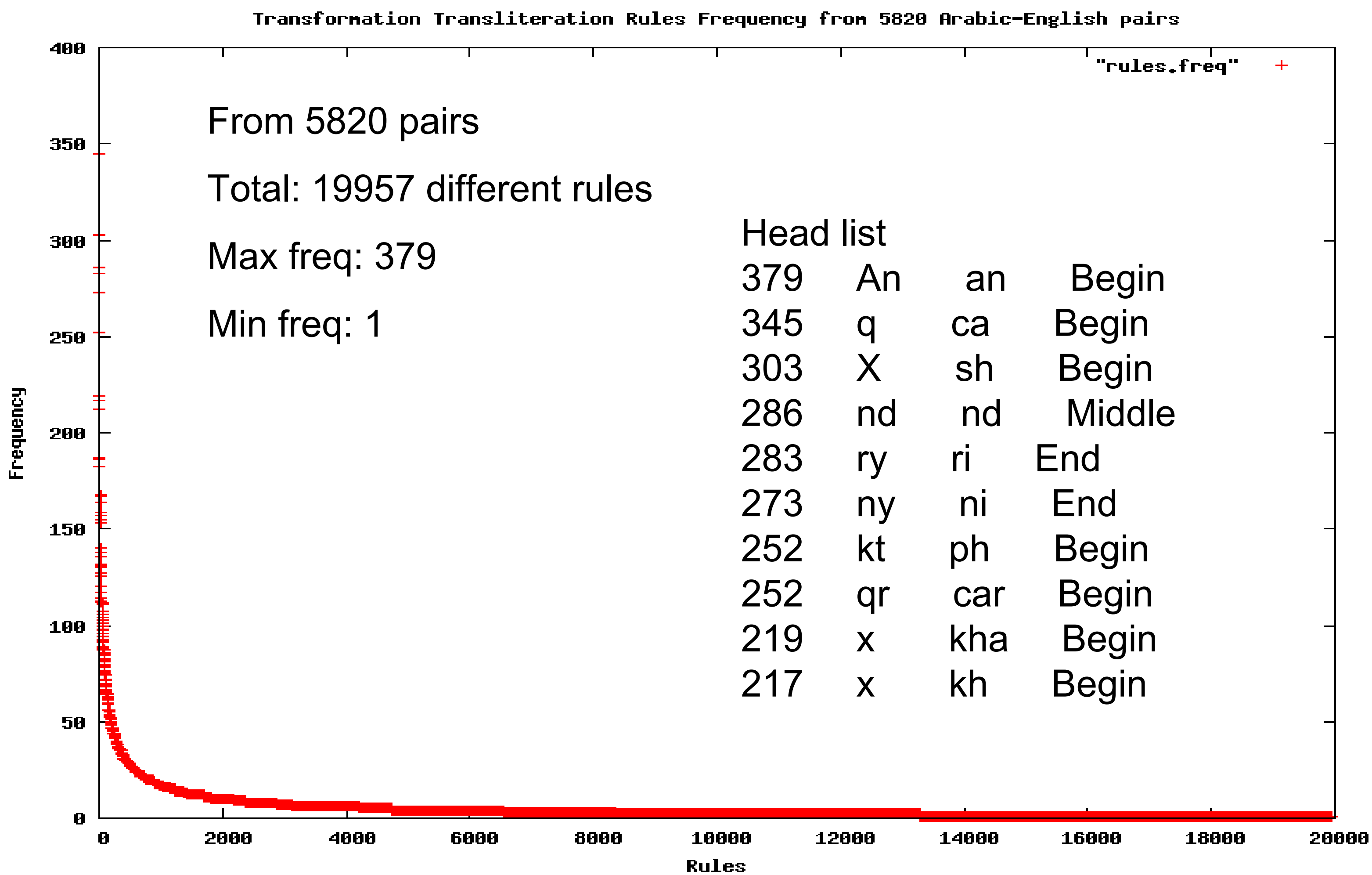
Rule-based Architecture Overview

- **Training - Generator:**
  - Given “lybyry” & “liberian” how <u>many possible rules</u>?
  - A: Alignment by calculating edit distance



  - Use all optimal paths to extract rules according to alignment paths
  - Distinguish rules for begin, middle, and end
  - Use consonants to anchor rule

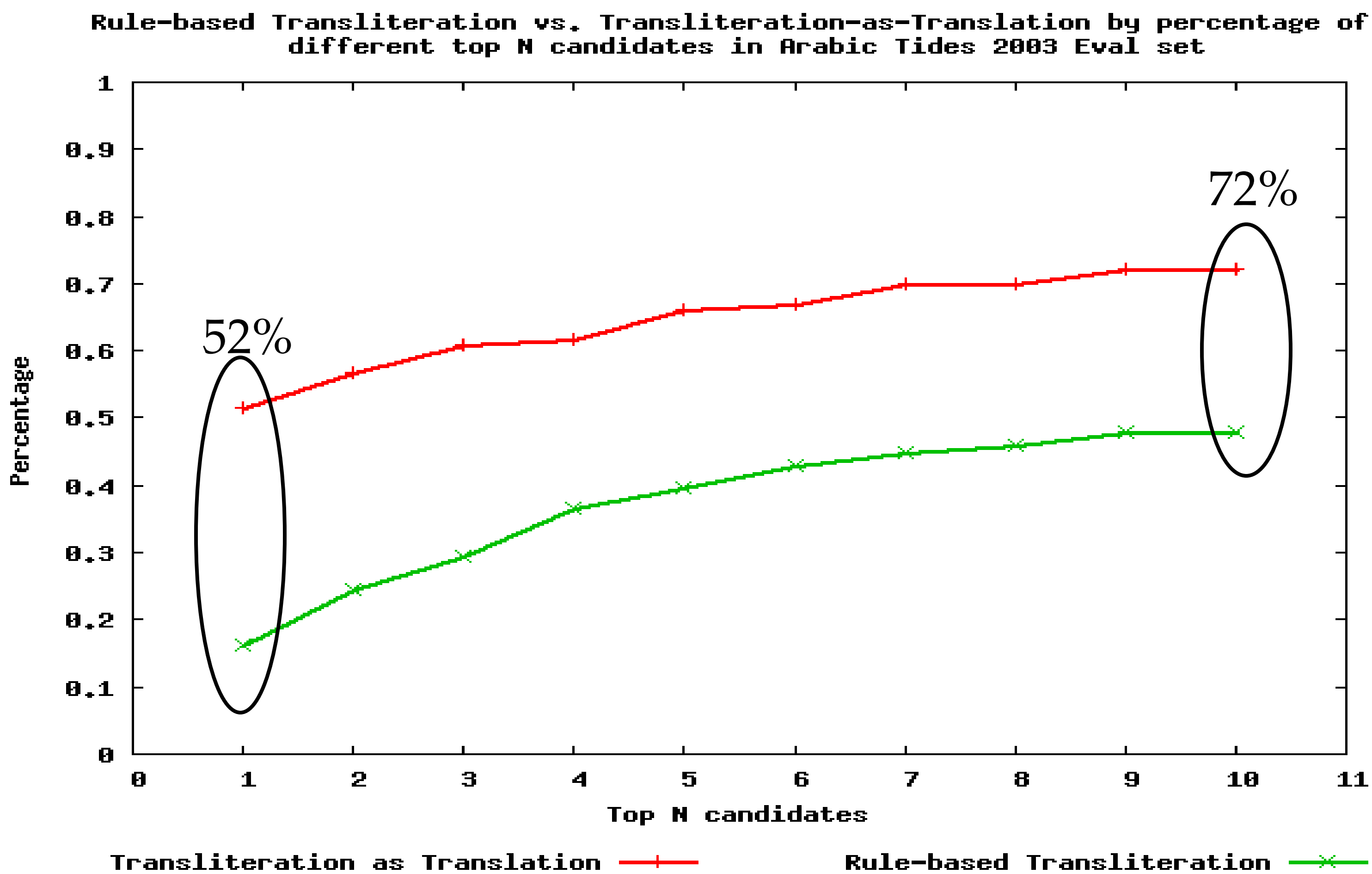
Rule-based Architecture Overview



Rule-based Architecture Overview

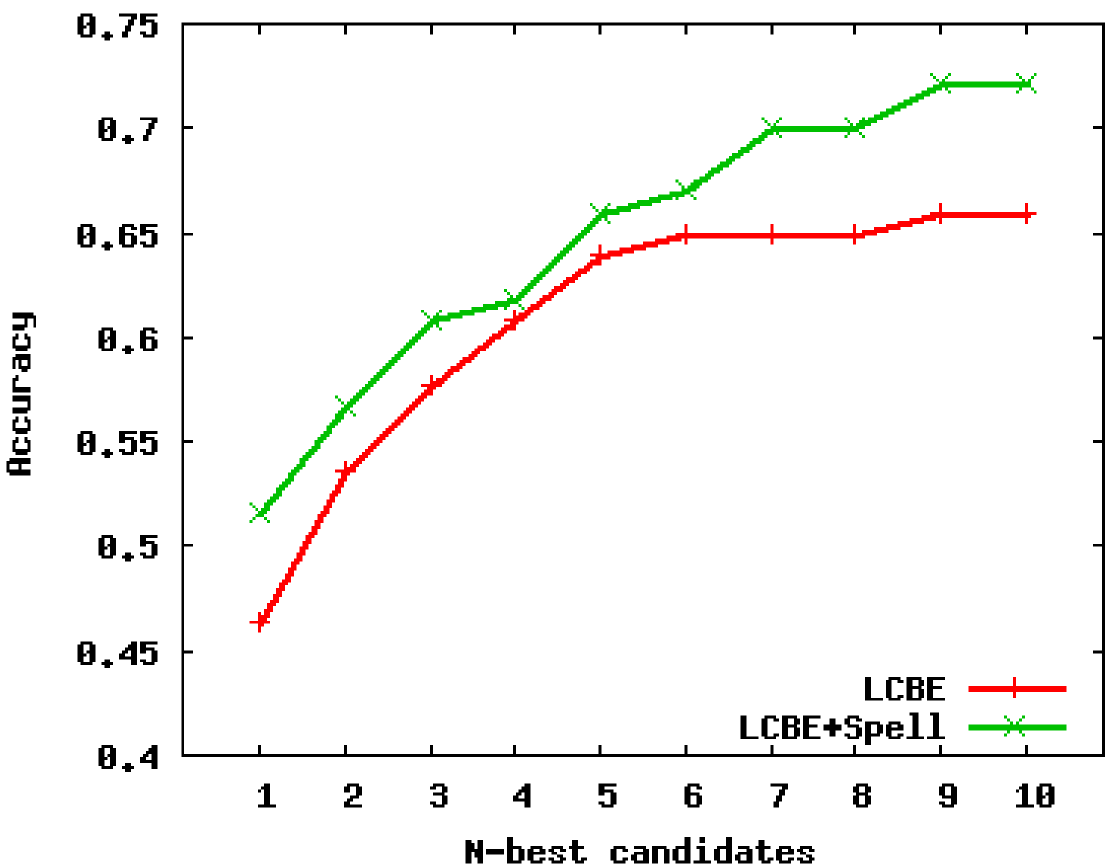
- **Training - Learner:**
  - How to know which rule is good or bad?
  - For each rule, apply it to the held-out data & use reduction of character errors as figure of merit
- **Decoding - Applicator:**
  - Application order: Begin -> End -> Middle
  - Confidence threshold: filter out unreliable rules
  - Application strategy: for each source word, find all possible rules, and apply them in order

Evaluation (Rule-based vs. T.a.T)



- **Significantly outperform rule-base**

Applying a spelling checker



Spelling Checker effectively improved the accuracy significantly

Incorporating T.a.T to SMT

Arabic text source sentence

كولمبو 4 يناير / شينخوا/ حذر رئيس الوزراء السريلانكى رانيل ويكرمسينغه
الرئيسة تشاندريكا كوماراتونجا من مغبة تدمير عملية السلام التى ترعاها
النرويج

SMT *hypothesis*

- **in colombo 4 january 1997 , the xinhua / warned by the prime minister {UNK السريلانكى رانيل ويكرمسينغه } chairperson {UNK تشاندريكا كوماراتونج } cautioned the destruction of the peace process sponsored by norway**

SMT with *T.a.T*

- **in colombo 4 january 1997 , the xinhua / warned by the prime minister** Srilankan Ranil Wikramsinghe **charperson** Chandrika Kumaratunga **cautioned the destruction of the peace process sponsored by norway**

Reference translation

- **Colombo 04/01 (Xinhua)** Sri Lankan **Prime Minister** Ranil Wickremasinghe **warned the country's President** Chandrika Kumaratunga **of the consequences of destroying the peace process sponsored by the Norwegians**