Novel Probabilistic Finite-State Transducers for Cognate and Transliteration Modeling

Charles Schafer
Department of Computer Science
Center for Language and Speech Processing
Johns Hopkins University
Baltimore, MD 21218 USA
cschafer@cs.jhu.edu

Abstract

We present and empirically compare a range of novel probabilistic finite-state transducer (PFST) models targeted at two major natural language string transduction tasks, transliteration selection and cognate translation selection. Evaluation is performed on 10 distinct language pair data sets, and in each case novel models consistently and substantially outperform a well-established standard reference algorithm.

1 Introduction

This paper presents and empirically compares a range of novel probabilistic finite-state transducer (PFST) models targeted at two major natural language string transduction tasks. Four distinctly original transducer models are introduced: *symbol networks*, *interlingua transducers*, the *joint distribution/conditional operations tranducer*, and *acquired alphabetic identity* transducers. These transduction models, some variations on previously published models, and also a standard reference algorithm, the probabilistic memoryless transducer introduced by Ristad and Yianilos (1997), are applied to two important problems in NLP: **transliteration selection** and **cognate translation selection**.

Evaluation is performed on 10 distinct language pair data sets, and in each case novel models consistently and substantially outperform a well-established standard reference algorithm.

This work is distinguished by the variety of tasks and languages addressed. Transliteration selection is evaluated on 2 language pairs (Inuktitut-English and Arabic-English). Cognate selection is evaluated on a total of 8 distinct language pairs drawn from the Romance, Slavic, Turkic, Germanic, and North Indian language groups. The contribution of this work lies not in attempting to model in detail the complexities of a particular generative process for a particular task. Rather, the goal was to investigate which *general* structures for transduction are effective over a reasonable variety of natural language monotonic string transduction tasks. The model we later denote the "Conditional Distribution/Unconditional Insertions" transducer (**CDUI**) consistently figures among the most successful across tasks and language testbeds. The *acquired alphabetic identity* **(AI)** model also performed extremely well, achieving the highest performance of any model on 6 out of the total of 10 experiments across all tasks and language.

Transliteration Selection: Examples	
Name from English Corpus	**Source Language Rendering**
	Arabic
Jensen	JEEM NOON SEEN NOON
Virginia	FEH REH JEEM YEH NOON YEH ALEF
William	WAW LAM YEH ALEF MEEM
	Inuktitut
Chretien	kurittian
Chartrand	saaturaan

Table 1: Instances of transliteration drawn from Arabic and Inuktitut.

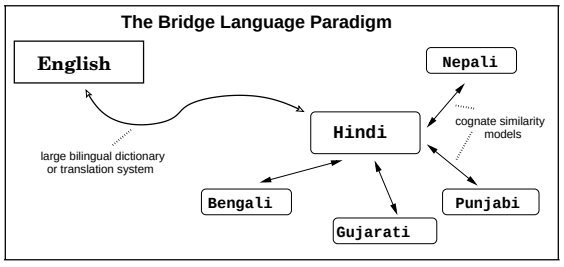

Figure 1: Intra-language-family cognate selection as a part of the translation bridge to English.

Transliteration selection is an important subtask in machine translation. This paper focuses on orthographic transductions for transliteration modeling (as opposed to phonemic ones). The transliteration task this paper uses for relative comparison of transduction models is that of selecting the correct English word corresponding to a back-transliterated

Proceedings of the 7th Conference of the Association for Machine Translation in the Americas, pages 203-212,
Cambridge, August 2006.

Cognate Selection: Examples	
Language Pair	**Cognate Examples**
Spanish-Italian	homogenizar omogeneizzare
Polish-Serbian	befsztyk biftek
German-Dutch	gefestigt gevestigd

Table 2: Examples of cognates for 3 language pairs.

foreign form,[1] for two interesting cases: Arabic and Inuktitut. Arabic presents difficulties because of its alphabet and convention of not writing short vowels. Inuktitut's orthography, when rendered in Roman characters, uses only a subset of the Roman alphabet (including, for example, only 3 vowels) and its transliteration process is not standardized. Table 1 shows typical instances of transliteration for these languages. Table 4 lists additional Inuktitut-English examples indicating the degree of variability encountered in attested Inuktitut transliterations of English.

Cognate translation selection is a useful component for translation within language families or as part of a two-stage bridge-language approach to handling lower-resource languages (Mann and Yarowsky, 2001; Schafer and Yarowsky, 2002). For a known Turkish or Hindi word present in an English bilingual dictionary, if we can pick out sets of likely cognates from an Uzbek or a Punjabi vocabulary, these can be reranked using corpus contextual information and other similarity measures, such that we can hypothesize an English translation link to an Uzbek word using Turkish as an intermediary, or bridge. This concept is illustrated in Figure 1, and Table 2 lists some instances of cognate pairs from our data. Improving measures of string similarity – i.e., better transducers or ensembles of transducers – thus improves the translation lexicon building process for low-resource languages.

In the tasks and language pair testbeds we address, novel models proposed herein outperform an existing baseline.

2 Transducer Models for Cross-Language String Similarity

This section will motivate and present several models for weighted string similarity, beginning with an established one from the literature and moving on to the novel models introduced herein. Each model has an associated graphical figure illustrating, within the limited space available, the structure of a weighted finite-state transducer implementing it. Given the impossibility of displaying realistic alphabet sizes, small example alphabets are used; given, in some cases, the large number of states and transitions involved, representative or evocative graphical shorthands have been employed. Second, each nontrivial model is described both textually and via an intuitive mathematical formalism.

2.1 A Baseline: Memoryless Transducer

Ristad and Yianilos (1997) proposed 3 variants on memoryless probabilistic string transducers for the task of English word pronunciation modeling. Their core model, which we reproduce here, was a single-state transducer having a self-loop transition: an emission function on the transition encodes a joint distribution over individual symbol substitutions, deletions and insertions. Their second variant, also a memoryless transducer, had a radically reduced tied parameter set, consisting of 4 parameters: one each for any insertion, any deletion, any substition of and identical symbol, and any non-identity substition. Their third technique involved using finite mixtures of the first two models.

As mentioned, the core Ristad and Yianilos model (subsequently denoted **R&Y**) can be thought of as a machine of single state having a self-loop transition. On this transition is a probability distribution over output 2-tuples, which can be of the form (a, ϵ) [insertion in language 1], (ϵ, b) [insertion in language 2] , or (a, b) [substitution]. More formally, given

a language 1 alphabet Σ_1 and language 2 alphabet Σ_2,

a language 1 string α and language 2 string β,

for $a \in \{\epsilon \bigcup \Sigma_1\}$, $b \in \{\epsilon \bigcup \Sigma_2\}$,

we want to learn a joint probability distribution $P(a, b)$ over individual insertions[2] and substitutions (defining $p(\epsilon, \epsilon) = 0$).[3] Further, we assess the similarity of string pair (α, β) as the sum over the probabilities of all sequences of these operations generating (α, β):

[1] That is, a foreign rendition of a natively English-language proper name, or proper name that at least is commonly rendered in English corpora unmodified from its original form, as other European language proper names generally are, modulo possible diacritic-stripping.

[2] When discussing non-directional models in this paper, we use the term "insertions" generically to refer to what in edit distance nomenclature are either called "insertions" or "deletions", since an insertion in string 1 is a deletion in string 2.

[3] In discussing these probability models, we distinguish, notationally, between a probability distribution $P(X)$ which is a function of values X, and specific values $P(X)$ may take, such as $p(x)$, $x \in X$.

$\sum_z \prod_{z_i=1}^{z_m} p(a_{z_i}, b_{z_i})$
which is the sum over all $z > 0$ operation sequences producing α, β. Note that there is a natural bound on m, the length of operation sequences: these sequences cannot be longer than $|\alpha| + |\beta|$, which would be the length of an operation sequence generating the string pair using only the insertion operations. Figure 2 is a graphical illustration of this model.

As Ristad and Yianilos note, this transducer specifies a probability distribution over string pairs given a particular edit sequence length. They discuss options for interpreting transducer-assigned string pair probabilities as string similarities:
(1) using the Viterbi algorithm and scoring string pairs by the log probability of the most likely edit sequence, and
(2) using the log probability of the string pair under the model (summing over all possible edit sequences).

In all cases (and for all transduction models) in this work, we calculate similarity using (2), the sum over all edit sequences. For some models (for example the **SN** model discussed in Section 2.4 and model **JDCO** discussed in Section 2.6) it is clear from the transducer structure that a single edit operation sequence may not solely dominate the probability assigned a string pair.

Finally, the sections describing each transduction model will each make note of the number of parameters for the model in question, as a function of the language 1 and language 2 alphabet sizes. The number of parameters for the **R&Y** model is
$|\Sigma_1| + |\Sigma_2| + |\Sigma_1| * |\Sigma_2|$

We trained this and the subsequently described transducer models via the Expectation-Maximization algorithm (EM) using the Dyna modeling language and DynaMITE parameter optimization toolkit (Eisner et al, 2004), further discussed in Section 3.

2.2 Adding State to the Baseline Model

We enhanced the memoryless string transducer discussed in the previous section by adding state.

There are multiple ways to do this. We chose the following and instantiated it as a fully-connected two-state model without tied parameters (we continue to use the standard nomenclature developed in Section 2.1):

$P(a, b, s_j|s_i)$ where s_i and s_j are elements of the state set S, which in this case is of size 2. Notice that this model reduces to **R&Y** when the state set

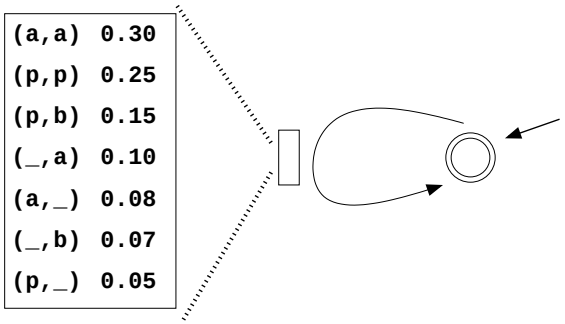

Figure 2: *Baseline Memoryless Transducer* **(R&Y)** with single distribution over all emission operations. We illustrate the model by a single-state transducer having a self-loop transition with associated emission function; the emission function is a probability distribution over pairs $a \in \{\epsilon \bigcup \Sigma_1\}$, $b \in \{\epsilon \bigcup \Sigma_2\}$, not including the pair (ϵ, ϵ). *Variants on this simple running example will be used to illustrate the other proposed transducer models. "_" is used in the figures in place of ϵ.*

is of size 1.

Figure shows the structure of the transducer described, subsequently denoted **2STEF**. This two-state model outperforms the baseline **R&Y** transducer across all of the task evaluations in Section 3.

The number of parameters for this model is
$4 * (|\Sigma_1| + |\Sigma_2| + |\Sigma_1| * |\Sigma_2|)$

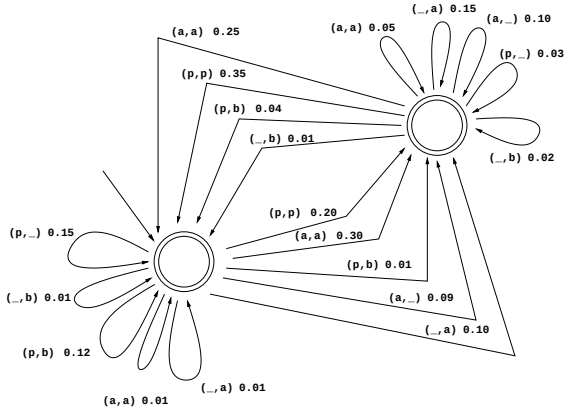

Figure 3: *2-State Transducer with Transition Emission Function* **(2STEF)**. Two-state transducer having individual input/output pairs tied to state-to-state transitions. The hidden states and transition-tied outputs allow limited modeling of sequence structure.

2.3 Interlingua Transducers

This class of transducers models joint distributions $P(\alpha, \beta, \gamma)$, where there are a language 1 string α

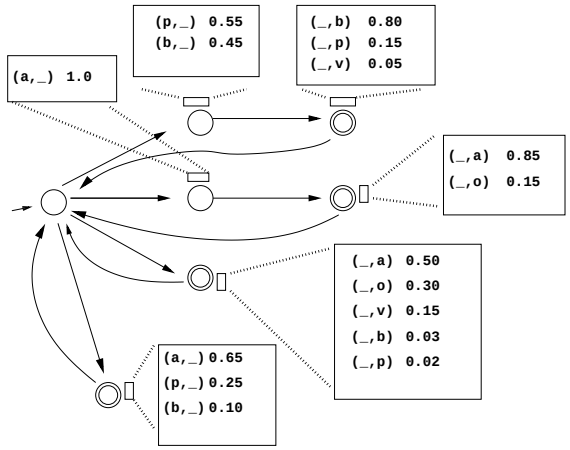

Figure 4: *Unigram Interlingua Transducer* **(UIT)**, having partial decoupling of input and output. The transducer pictured corresponds to a unigram interlingua model using only 2 interlingua symbols to model substitution, as opposed to the full 25, to reduce the visual complexity of the drawing.

and language 2 string β as before, but where there is also an unobserved interlingua string γ which is hypothesized to account for the observed string α and β. γ serves much the same function as interlingual concept states in machine translation for reducing model dimensionality.

A convenient way to work with these models is to factor $P(\alpha, \beta, \gamma)$ as

$$P(\alpha|\beta,\gamma)P(\beta|\gamma)P(\gamma)$$

and then assume α to be conditionally independent of β given γ, simplifying the model structure to:

$$P(\alpha|\gamma)P(\beta|\gamma)P(\gamma)$$

For language 1 and language 2 alphabets Σ_1 and Σ_2 respectively, and for interlingua alphabets Σ_0, for alphabet symbols $a \in \{\epsilon \bigcup \Sigma_1\}$, $b \in \{\epsilon \bigcup \Sigma_2\}$, $c \in \Sigma_0$: we can model symbol generation for languages 1 and 2 as

$$P(a|c)P(b|c)P(c)$$

leaving the "interlingua language model" $P(\gamma)$ to be specified as desired.[4]

The interlingua models in this work simply set the interlingua model $P(c)$ to a unigram model, which does not incorporate any memory of the interlingua sequence.

We investigated two variants of the unigram interlingua model: model **UIT** is exactly as described above, whereas **UIT2** is a variant which allows atomic generation of two-character language 1 ($a_i a_{i+1}$) or language 2 ($b_j b_{j+1}$) sequences from a single interlingua symbol c_k. Figure 4 illustrates the **UIT** model. As implemented, **UIT** was allowed an alphabet of 25 distinct interlingua symbols used to model "substitution," as well as 2 additional interlingua symbols, one for exclusively language 1 insertion and one for exclusively language 2 insertion.

The number of parameters for the **UIT** model is (language 1 insertions + language 2 insertions + "substitutions" + probabilities of performing each operation)
$1 + |\Sigma_1| + 1 + |\Sigma_2| + 25 + 25 * (|\Sigma_1| + |\Sigma_2|)$
$= 27 + 26 * (|\Sigma_1| + |\Sigma_2|)$

2.4 Symbol Networks

We define a *Symbol Network* (henceforth, **SN**) transducer as follows: Symbol network transducers have one state per alphabet symbol per language. For every $a \in \Sigma_1$ there is a state $S_{(a,\epsilon)}$ and for every $b \in \Sigma_2$ there is a state $S_{(\epsilon,b)}$. These states emit the symbol by which they are denoted with probability 1. On each state there are outgoing probabilistic transitions to all states $S_{(a,\epsilon)}$ and $S_{(\epsilon,b)}$. These transition probabilities make up the set of parameters to be trained.

Thus we construct $P(a, b)$ as a kind of history-based model, a bigram model $P(a_i, b_i|a_{i-1}, b_{i-1})$ where every allowed pair (a, b) must have the form either (a, ϵ) or (ϵ, b). An interesting property of this transducer structure is that some symbol states learn to allocate most of their outgoing transition probability to symbols internal to the language, whereas other symbols more strongly prefer transitions cross-language.

A representation of a *Symbol Network* transducer, scaled down for display, is pictured in Figure 5.

The **SN** transducer yields relatively low accuracy (below **R&Y**) on the transliteration and cognate selection tasks.

The number of parameters for this model, including the outgoing transitions from its start state, is
$(|\Sigma_1| + |\Sigma_2|) + (|\Sigma_1| + |\Sigma_2|)^2$

2.5 A Conditional Model

In their study of back-transliteration for English words rendered in Japanese, Knight and Graehl (1997) formulate a conditional transducer which generates Japanese sounds from English sounds. The allowed operations in that transducer are restricted to substitutions, where each English sound

[4]Note: the interlingua symbols c are never deleted.

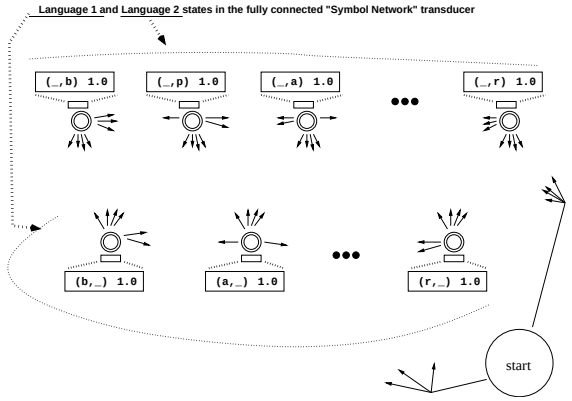

Figure 5: Illustration of fully-connected *Symbol Network* **(SN)** string transducer. (The only exception to full-connectedness is that there are no incoming arcs to start state). The transducer pictured has one state per alphabet symbol per language. For every $a \in \Sigma_1$ there is a state $S_{(a,\epsilon)}$ and for every $b \in \Sigma_2$ there is a state $S_{(\epsilon,b)}$. These states emit the symbol by which they are denoted with probability 1. On each state there are outgoing probabilistic transitions to all states $S_{(a,\epsilon)}$ and $S_{(\epsilon,b)}$.

is strictly required to generate one or more Japanese sounds. This was based on an observation that Japanese sound sequences were never shorter than their corresponding English sequence.

Stalls and Knight (1998) addressed back-transliteration for English words rendered in Arabic. They describe a similar conditional transducer generating Arabic orthographic symbols from English sounds. Given the Arabic convention of not writing short vowels, Stalls and Knight modified the conditional transducer used for the English-Japanese work, additionally allowing English sounds the option of generating nothing (ϵ).

In this section we consider a model of string generation in which the underlying probability distribution is conditional (that is, one language string is generated conditionally from the other): keeping with the notation developed earlier, we model $P(\beta|\alpha)$. We incorporate additional flexibility beyond the Stalls and Knight model, allowing both insertion and deletion as well as substitution.

We term this proposal the "Conditional Distribution/Unconditional Insertions" transducer, or **CDUI** in mnemonic form. Given a string α, we probabilistically choose from operations I (insert in β), D (delete from α), or S (substitute a symbol b given current symbol a).[5]

[5]Note that this conditional model is deficient, since it allocates probability for deleting any symbol at each position in string α even though the identity of the symbol at each position in α is always known.

Figure 6 illustrates the structure of this generative process.

The number of parameters for this model is the same as for **R&Y**:
$|\Sigma_1| + |\Sigma_2| + |\Sigma_1| * |\Sigma_2|$

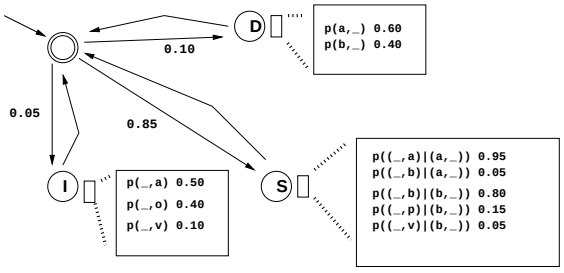

Figure 6: *"Conditional Distribution/Unconditional Insertions" Transducer* **(CDUI)**. Operation probabilities correspond to the probabilistic transitions from the start state to the respective insertion, deletion and substitution states I, D and S.

2.6 A Joint Model Utilizing Conditional Probabilities

The "generative story" of this model is easily understood by refering to Figure 7. We start by adding a special start symbol **/s/** to each string in the pair: for example, considering the Spanish-Italian cognate pair *delegación - delegazione*, we get **/s/***delegación* - **/s/***delegazione*. Starting at (**/s/**,**/s/**), we probabilistically choose an operation from a probability distribution $P(\omega)$, where ω is an operation in $\{$D,I,S_1,$S_2\}$. The operations are defined as follows:
D (insert in a_i in α given a_{i-1})
I (insert b_j in β given b_{j-1})
S_1 (substitute b given a, cross-language)
S_2 (substitute a given b, cross-language)

Thus we weave our way back and forth across the string pair generating symbols, picking operations from a single joint distribution but always conditioning symbols using a conditional distribution. This transduction paradigm is denoted "Joint Distribution/Conditional Operations" **(JDCO)** in subsequent graphs and figures. In the cognate selection task, it consistently achieves high performance across the diverse set of 8 language pairs in our testbed. Table 7 and Table 8 show performance on the cognate task across these 8 languages.

The number of parameters for this model is

$|\Sigma_1|^2 + |\Sigma_2|^2 + 2 * |\Sigma_1| * |\Sigma_2|$

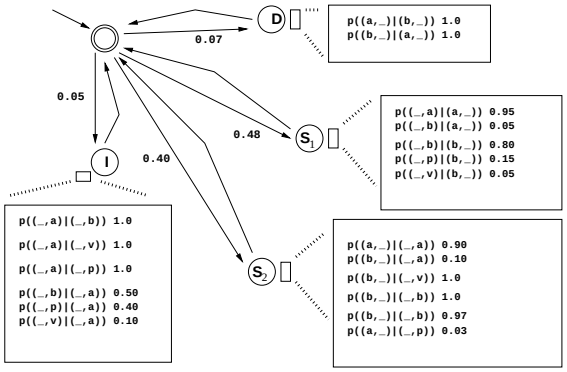

Figure 7: *"Joint Distribution/Conditional Operations" Transducer* **(JDCO)**.

2.7 Incorporating Cross-Alphabetic Identities

We present a final model which is simple but highly effective across tasks. This is a variation on the **R&Y** memoryless transducer. The distinction of this particular reformulation is that there is a single model probability for all "identical" substitutions.

Training for this model is performed in 2 stages. First, we train **R&Y**. We then extract the top corresponding a for each b and vice versa, creating a set of "identical" pairs
$\{(a_{ident1}, b_{ident1}), (a_{ident2}, b_{ident2}), ..\}$.
In the second phase, substitution pairs in this set share a single model probability. All substitutions not in this set are treated as in **R&Y**: each individual symbol-to-symbol substitution has its own parameter. The model is then retrained, with, of course, the set of "identities" fixed.

The resulting transducer is at or near the top performance in both transliteration and cognate selection. We can draw the conclusion that, when there is a near one-to-one symbol correspondence across languages, this model is hard to beat, whereas the mapping cardinality of orthographic symbols and phonetic strings is not amenable to such representational simplification. This transduction paradigm is denoted **AI** in subsequent tables and figures. Finally, we note here that an interesting avenue for future work would be to incorporate the alphabetic identity idea into more complex models of string transduction, since it seems to be highly effective even when employed in a simple, single-state model.

The number of parameters for this model is
$|\Sigma_1| + |\Sigma_2| - K + 1 + |\Sigma_1| * |\Sigma_2|$,
where $K < (|\Sigma_1| + |\Sigma_2|)$
is the number of learned alphabetic identities in the model.

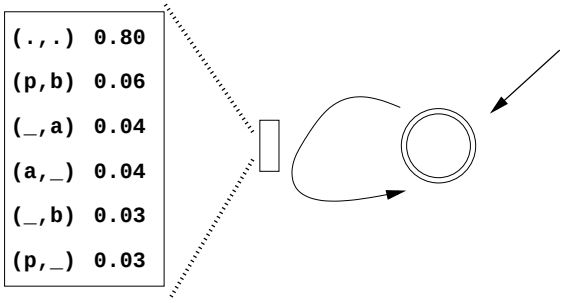

Figure 8: *"Alphabetic Identity" Transducer* **(AI)**. (. , .) represents the "identical" substitution.

3 Tasks, Training & Evaluation

Expectation-Maximization (EM) training was performed for the learning of probabilistic FST weights. For all the models discussed above, we used the Dyna modeling language and DynaMITE parameter optimization toolkit (Eisner et al, 2004), to perform EM training. The following discussion gives details for the target tasks of transliteration selection and cognate translation selection, first explaining training and evaluation methodologies and then presenting results. The focus in this work is on *comparative* evaluation of the transduction paradigms presented. In particular, the results in the cognate task are meant exclusively to compare model performance across a wide variety of language pairs under a controlled experimental setting.

3.1 Transliteration Selection for Machine Translation

We investigated the problem of back-transliteration for 2 language pairs. In this task a word or name from an English corpus has been rendered in Arabic or Inuktitut orthography. Under our task formulation, we are given an Arabic or Inuktitut word and the goal is to select, from a list of English words, which English word is being rendered in the foreign language. This corresponds to the real-world problem of using transliteration modeling for bitext word alignment.

The test condition for both Arabic and Inuktitut is that we are given the correct English word plus 99 other randomly selected English words, and from this list of 100 possibilities we wish to rank the correct word as highly as possible. The choice of 100

in this case is meant to yield a disambiguation set of size on the order of a long sentence. If we can solve this problem effectively, then the solution can contribute to word alignment (and thus translation) performance.

Arabic Minimally Supervised Training: For many language pairs, it may be difficult to find a list of named entity translations (named entities are commonly the subjects of the transliteration processes we wish to model). We took an approach to gathering English-Arabic string pairs for training which required little human intervention, and yet yielded successful performance figures. We first aligned roughly 63K Arabic-English sentence pairs of translated news text, running IBM Model 4 alignment in both directions using the Giza++ toolkit (Och and Ney, 2000). Second, we extracted English-Arabic translations having high translation probability in both directions. Finally, we intersected the English words of these translation pairs with (1) an English gazetteer of world city/country names, and (2) the most common 1000 each of surnames, male first names, and female first names from the 1990 United States Census (United States Census Bureau). There was no selection criterion based on string resemblance between Arabic and English. A total of 652 pairs were extracted in this way; a subsequent by-hand procedure verified that 73% of city/country names and 70% of person names were correct, although we trained on the noisy data regardless. 8 randomly selected train/test splits, training size 200 and test size 90, were generated, and the resulting averaged performance numbers are given in Table 5. Results as high as 92% exact-match accuracy (for transducer **AI**) were observed. The nature of the English-Arabic string transformations can be seen listed in Table 3 (Arabic characters are represented via their Unicode names).

Inuktitut Minimally Supervised Training: We used a slightly different method to extract training pairs for English-Inuktitut. Taking the Nunavut Hansards corpus of parallel English and Inuktitut sentences, training string pairs were acquired from the bitext in the following manner. Whenever single instances of corresponding honorifics were found in a sentence pair – these included the correspondences (Ms , mis); (Mrs , missa/missis); (Mr , mista/mistu) – the immediately following capitalized English words (up to 2) were extracted and the same number of Inuktitut words were extracted to be used as training pairs. Thus, given the appearance in aligned sentences of "Mr. Quirke" and "mista kuak", the training pair (Quirke,kuak) would be extracted. Common distractions such as "Mr Speaker" were filtered out. In order to focus on the native English name (back-transliteration) problem the English extractions were required to have appeared in a large, news-corpus-derived English wordlist. This procedure resulted in a conservative, high-quality list of 434 unique name pairs. To motivate this problem, note that although in this corpus English and Inuktitut are both written in Roman characters, English names are significantly transformed when rendered in Inuktitut text. The following is a (partial) list of the corpus-attested variations found in the Inuktitut corpus for "Williams", "Campbell", and "McLean".

Inuktitut Transliteration Examples	
Name from English Corpus	**Inuktitut Renderings**
Williams	uialims uilialums uiliammas viliams
Campbell	kaampu kaampul kamvul kaamvul
McLean	makalain maklainn makliin makkalain

Table 4: Examples of Inuktitut-English transliteration. Notable characteristics include both the degree of the string transformations and the extreme lack of standardization for this process.

Results for this task are in Table 6; the results are for four randomly generated splits of training size 200 with 220 test words per split. Model **AI** achieved around 78% performance on this task, as did model **CDUI**.

Model Comparison for Arabic Transliteration					
	% Having Correct at <= Rank				
Model	1	2	5	10	20
AI	92.1	94.1	95.6	97.2	97.8
CDUI	89.5	92.9	94.6	96.0	97.1
2STEF	86.8	90.8	94.9	96.0	97.4
R&Y	84.4	89.0	92.7	95.2	97.0
UIT	83.4	88.9	93.0	95.3	97.1
JDCO	79.6	84.9	91.4	94.8	96.5
SN	77.0	83.3	89.1	92.4	95.4
UIT2	76.7	85.2	91.5	94.5	96.3

Table 5: Model comparison table for **Arabic** *transliteration selection* task.

3.2 Cognate Selection

Identifying cognates – in this context, we use the term to mean words of related meaning in related languages, which share a surface resemblace – is potentially of great use for translation, as noted in Schafer and Yarowsky (2002). We wished to ex-

Examples of Arabic-English Transliteration	
Name from English Corpus	**Arabic Rendering**
Piedade	BEH YEH YEH DAL ALEF DAL YEH
Bolivia	BEH WAW LAM YEH FEH YEH ALEF
Luxembourg	LAM KAF SEEN MEEM BEH WAW REH GHAIN
Zanzibar	ZAIN NOON JEEM YEH BEH ALEF REH

Table 3: Examples of Arabic-English transliteration.

Model Comparison for Inuktitut Transliteration					
	% Having Correct at $<=$ Rank				
Model	1	2	5	10	20
AI	78.7	85.0	89.6	91.2	94.1
CDUI	77.6	84.5	89.9	91.8	93.9
UIT2	70.1	79.9	87.6	92.1	95.0
2STEF	69.8	79.9	88.8	92.5	94.8
UIT	68.6	76.2	85.1	90.0	94.4
R&Y	68.0	75.8	85.4	91.3	94.6
SN	67.5	77.6	87.6	93.3	96.9
JDCO	64.5	75.7	85.5	91.2	94.6

Table 6: Model comparison table for **Inuktitut** *transliteration selection* task.

plore the effectiveness of a diverse set of string similarity measures on this task. In addition, the problem of cognate selection is one in which we can acquire training sets across enough languages to compare model performance on the language variation dimension while holding the task constant.

Although there is not a large, readily available list of cognate words for a large number of language pairs, we can make do. We generated training data for this problem by taking numerous English-X dictionaries, where X is for example Polish, Serbian, Nepali, Hindi, Turkish, Uzbek, etc. For related languages such as Polish and Serbian (both members of the Slavic family), we examine the intersection of Polish-Serbian word pairs having low Levenshtein distance[6] with the Polish-Serbian word pairs specified by a Polish-English/English-Serbian dictionary join. The pairs in this intersection have high probability of being cognate. For example, we examined a small subset of Spanish-Italian pairs acquired in this way, referring to a print Spanish-Italian dictionary (Vox Diccionario Esencial Italiano-Espanol) and estimated 90% of these pairs to be cognate. These string pairs are "presumed true" cognates, and are taken for training data.

Next, we trained all of the transduction models described in this paper, using the string pairs just described.

The test setting for these experiments was created by running the trained **R&Y** transducer on the full held-out language 1 vocabulary against the full language 2 vocabulary, scoring all pairs. 100-best lists were generated, restricted to those which had a presumed true Polish-Serbian cognate translation pair from the Polish-English English-Serbian dictionary join, and all models were run on these 100-best lists for 500 test words in each "language 1" language.[7] Although this process may somewhat inflate the absolute performance numbers relative to unrestricted search, it serves the purpose of creating a viable testbed for direct and efficient PFST model comparison.

Relative performance for the full set of models averaged across all 8 language pairs is shown in Table 7: these models rank the top-100 candidate list for each test word and performance is listed at several ranks for each model. Table 8 shows performance per model for each language pair at rank 2. Both tables show performance on a training size of 400 pairs per language. Figure 9 provides learning curves for representative transduction models averaged over 3 language pairs. At training size 200, each language's performance is computed as an average of 4 randomly generated training sets.

There are several interesting observations regarding the learning curve. With the exception of **SN**, the models do not radically improve as the initial training size of 200 is increased 16-fold. Of the high-performing models for cognate selection, **JDCO** continues improving through all training sizes. **AI** peaks quickly, and the remaining models display remarkably flat learning profiles. This is good to know, as it suggests we can get most of the

[6] Levenshtein distance $<=$ 3, with vowel insertions and vowel-vowel nonidentity substitutions weighted 0.5 instead of 1.

[7] This cognate selection task is directional: we are ranking 100 Serbian potential cognates for each of the 500 Polish test words

gains to be had from these models even when the amount of available training data is small.

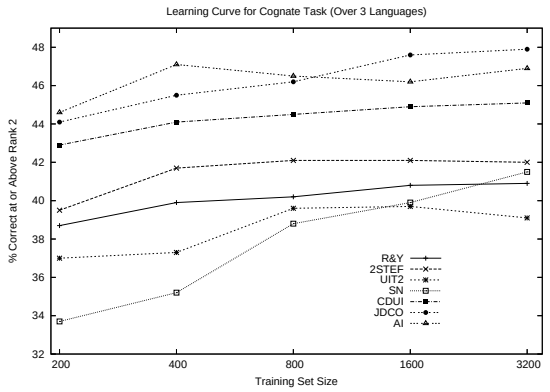

Figure 9: Learning curve for representative models, showing aggregate performance values over Spanish-Italian, German-Dutch, and Hindi-Nepali cognate testbeds. There are 1500 total test examples for each data point (500 per language).

Model Comparison for Cognate Selection						
	% Having Correct at $<=$ Rank					
Model	1	2	5	10	20	50
JDCO	32.9	40.3	48.2	54.9	64.2	80.5
AI	32.7	40.1	49.1	57.5	67.1	82.6
CDUI	31.5	38.4	47.5	54.6	64.6	81.0
2STEF	28.2	35.1	44.2	52.2	62.1	80.6
R&Y	27.1	33.3	42.8	50.5	60.4	79.4
UIT2	26.3	32.1	40.9	48.3	60.2	80.2
UIT	26.2	32.7	42.5	50.2	60.1	79.1
SN	23.3	29.1	38.6	48.0	58.5	77.7

Table 7: Model comparison table for *cognate selection* testbed. Results are averaged over 8 language pairs(enumerated in more detail in Table 8). Training size is 400 examples for each language; the evaluation set consists of test 500 items for each language (for 4000 test items total, across all languages).

3.3 Model Performance Across Tasks

The results in transliteration selection and cognate selection raise the question: what is responsible for marked differences in model efficacy across tasks? Most notably, model **JDCO** performs at or near the top on the cognate tasks but at or near the bottom for the transliteration tasks. We suggest a probable explanation by considering how the tasks differ most. In both Arabic and Inuktitut transliteration, source and target strings generally display more surface differences are to be found in the cognate transduction problem instances we looked at. Further, the transduction process into English for both Arabic and Inuktitut transliteration displays substantial variability, as evidenced by the examples shown in Tables 3 and 4. The conclusion we draw is that the **JDCO** model is less well suited to transduction problems exhibiting these characteristics.

4 Model Tolerance to Noise

In practical experimental situations, training data for string transduction applications is often acquired automatically, through some minimally supervised means. It is important in such cases to have an expectation as to which models perform well in the presence of high amounts of training data noise.

We decided to test a diverse set of transduction models under an extremely harsh noise tolerance condition, one which corresponds to our conception of an "outlier" in terms of difficulty. For the experiments referenced in this section, 200 presumed-correct training pairs are used, and then noise of 100 or 200 randomly selected incorrect pairs is introduced into the training set, such that the total training pairs are $>= 33\%$ or $>= 50\%$ incorrect, respectively. We then measured performance degradation in these cases. Figure 10 graphically depicts the different noise robustness characteristics of a representative set of models. Noise robustness was measured on the cognate task, for correct-at-rank-1 evaluation. Results shown are aggregated over two language pair experiments: 4 folds of Spanish-Italian and 4 folds of German-Dutch. Training set size is 200 presumed correct pairs (subject to the general expectations as to training data quality, that is, roughly 90% correct) + some number (100 or 200 depending on the experiment) of incorrect pairs. Shown in the bar chart is the fraction of default performance (performance at training size 200) which is attained when 100 or 200 (respectively) randomly generated incorrect pairs are added to the training set. The various models show differing robustness to this high-noise condition. This knowledge is useful in informing practical decisions regarding which models to employ when training data is of poor quality, or when there is no readily available means by which to measure training data quality.

5 Conclusions

This paper has presented and empirically contrasted several novel probabilistic finite-state transducer models on the tasks of transliteration selection and cognate translation selection for intra-family machine translation. A number of the proposed models were shown to have strengths in various test conditions over 10 distinct language pair data sets, and in each case several of the novel models consistently and substantially outperform a well-established standard reference algorithm.

Model Comparison for Cognate Selection (Detailed by Language)									
Model	**Polish Serbian**	**Czech Serbian**	**Spanish Italian**	**Turkish Uzbek**	**German Dutch**	**German Swedish**	**Hindi Punjabi**	**Hindi Nepali**	**Avg.**
R&Y	27.6	37.2	48.6	27.8	40.4	33.8	20.4	30.8	33.3
2STEF	30.2	38.6	49.2	28.6	43.6	35.6	22.8	32.2	35.1
UIT	24.8	32.0	47.4	28.0	41.6	35.8	21.0	31.0	32.7
UIT2	28.4	36.4	45.6	26.0	37.8	34.6	19.4	28.4	32.1
SN	24.2	32.0	45.4	20.8	34.8	31.0	19.0	25.4	29.1
CDUI	32.0	43.4	51.2	36.2	**50.0**	44.8	18.4	31.2	38.4
JDCO	**35.4**	42.8	53.6	**38.6**	48.2	**48.6**	20.4	34.8	**40.3**
AI	34.2	**43.8**	**55.6**	35.8	47.2	39.4	**25.8**	**38.6**	40.1

Table 8: Cognate selection performance results, broken down by language pair. Training size is 400 string pairs per language. Test set size is 500 per language. Performance displayed is % correct at or above rank 2.

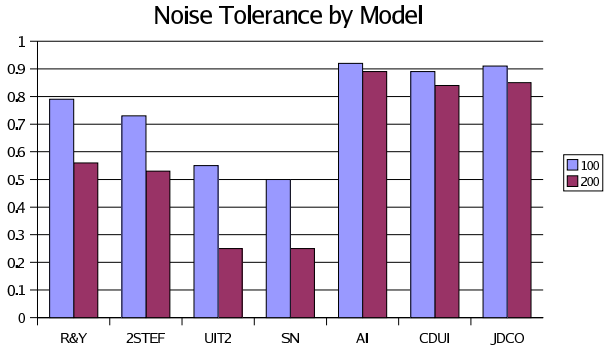

Figure 10: Model tolerance to noise. Measured on cognate task, for correct-at-rank-1 evaluation. Results averaged over 4 folds of Spanish-Italian and 4 folds of German-Dutch experiments. Training set size is 200 presumed correct pairs + some number (100 or 200) of incorrect pairs. Shown in the bar chart is fraction of the default performance at training size 200 which is attained when 100 or 200 (respectively) randomly generated incorrect pairs are added to the training set. The various models show differing robustness to this high-noise condition.

Our main goal in this work was to explore a variety of transduction modeling choices that would allow us to improve performance of transliteration and cognate selection components in end-to-end applications, across languages, and without resort to language-specific measures. The controlled model comparison experiments discussed herein have allowed us to identify high-performing models that can be employed for these tasks in subsequent lexicon induction experiments. In addition, the investigation of model noise tolerance, and resulting insights into which models can best be trusted to perform well in the presence of large amounts of noise, is of great use as we apply transducer models for string scoring in minimally supervised learning situations where clean training data is hard to find.

References

J. Eisner, E. Goldlust, and N. A. Smith. 2004. Dyna: A declarative language for implementing dynamic programs. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004)*, Companion Volume, pages 218-221.

K. Knight and J. Graehl. 1997. Machine transliteration. In *Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pp. 128-135.

Linguistic Data Consortium. 1995. COMLEX English Pronouncing Lexicon (PronLex). http://www.ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC97L20.

G. Mann and D. Yarowsky. 2001. Multipath translation lexicon induction via bridge languages. In *Proceedings of NAACL-2001*, pp. 151–158.

F. J. Och and H. Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 440–447.

E. S. Ristad and P. N. Yianilos. 1997. Learning string edit distance. In *Machine Learning: Proceedings of the Fourteenth International Conference*, pp. 287–295.

C. Schafer and D. Yarowsky. 2002. Inducing translation lexicons via diverse similarity measures and bridge languages. In *Proceedings the Sixth Conference on Natural Language Learning*, pp. 146–152.

B. Stalls and K. Knight. 1998. Translating names and technical terms in arabic text. In *Proceedings of the COLING/ACL Workshop on Computational Approaches to Semitic Languages*.

United States Census Bureau. Frequently Occurring First Names and Surnames From the 1990 Census. http://www.census.gov/genealogy/names/.

Vox Diccionario Esencial Italiano-Espanol, Espanol-Italiano, Spain, October 1993.