Experiments with a Hindi-to-English Transfer-based MT System under a Miserly Data Scenario

Alon Lavie, Stephan Vogel, Lori Levin, Erik Peterson, Katharina Probst, Ariadna Font Llitjós, Rachel Reynolds, Jaime Carbonell
Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, 15213
and
Richard Cohen
University Center for International Studies, University of Pittsburgh, Pittsburgh, PA, 15260

We describe an experiment designed to evaluate the capabilities of our trainable transfer-based (XFER) machine translation approach, as applied to the task of Hindi-to-English translation, and trained under an extremely limited data scenario. We compare the performance of the XFER approach with two corpus-based approaches – Statistical MT (SMT) and Example-based MT (EBMT) – under the limited data scenario. The results indicate that the XFER system significantly outperforms both EBMT and SMT in this scenario. Results also indicate that automatically learned transfer rules are effective in improving translation performance, compared with a baseline word-to-word translation version of the system. XFER system performance with a limited number of manually written transfer rules is, however, still better than the current automatically inferred rules. Furthermore, a "multi-engine" version of our system that combined the output of the XFER and SMT systems and optimizes translation selection outperformed both individual systems.

Categories and Subject Descriptors: I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Machine Translation*

General Terms: Evaluation, Hindi, Machine Translation

Additional Key Words and Phrases: Example-based Machine Translation, Limited Data Resources, Machine Learning, Multi-Engine Machine Translation, Statistical Translation, Transfer Rules

1. INTRODUCTION

Corpus-based Machine Translation (MT) approaches such as Statistical Machine Translation (SMT) [Brown et al. 1990; Brown et al. 1993; Vogel and Tribble 2002; Yamada and Knight 2001; Papineni et al. 1998; Och and Ney] and Example-based Machine Translation (EBMT) [Brown 1997; Sato and Nagao 1990] have received much attention in recent years, and have significantly improved the state-of-the-art of Machine Translation for a number of different language pairs. These approaches are attractive because they are fully automated, and require orders of magnitude

less human labor than traditional rule-based MT approaches. However, to achieve reasonable levels of translation performance, the corpus-based methods require very large volumes of sentence-aligned parallel text for the two languages – on the order of magnitude of a million words or more. Such resources are only currently available for only a small number of language pairs. While the amount of online resources for many languages will undoubtedly grow over time, many of the languages spoken by smaller ethnic groups and populations in the world will not have such resources within the forseeable future. Corpus-based MT approaches will therefore not be effective for such languages for some time to come.

Our MT research group at Carnegie Mellon, under DARPA and NSF funding, has been working on a new MT approach that is specifically designed to enable rapid development of MT for languages with limited amounts of online resources. Our approach assumes the availability of a small number of bi-lingual speakers of the two languages, but these need not be linguistic experts. The bi-lingual speakers create a comparatively *small* corpus of word aligned phrases and sentences (on the order of magnitude of a few thousand sentence pairs) using a specially designed elicitation tool. From this data, the learning module of our system automatically infers hierarchical syntactic transfer rules, which encode how constituent structures in the source language transfer to the target language. The collection of transfer rules is then used in our run-time system to translate previously unseen source language text into the target language. We refer to this system as the "Trainable Transfer-based MT System", or in short the XFER system.

The DARPA-sponsored "Surprise Language Exercise" (SLE) of June 2003 provided us with a golden opportunity to test out the capabilities of our approach. The Hindi-to-English system that we developed in the course of this exercise was the first large-scale open-domain test for our system. Our goal was to compare the performance of our XFER system with the corpus-based SMT and EBMT approaches developed both within our group at Carnegie Mellon, and by our colleagues elsewhere. Common training and testing data were used for the purpose of this cross-system comparison. The data was collected throughout the SLE during the month of June 2003. As data resources became available, it became clear that the Hindi-to-English was in fact not a "limited-data" situation. By the end of the SLE, over 1.5 million words of parallel Hindi-English text had been collected, which was sufficient for development of basic-quality SMT and EBMT systems. In a common evaluation conducted at the end of the SLE, the SMT systems that participated in the evaluation outperformed our XFER system, as measured by the NIST automatic MT evaluation metric [Doddington 2003]. Our system received a NIST score of 5.47, as compared with the best SMT system, which received a score of 7.61.

Our intention, however, was to test our XFER system under a far more limited data scenario than the one that had developed by the end of the SLE ([Nirenburg 1998; Sherematyeva and Nirenburg 2000; Jones and Havrilla 1998]). We therefore designed an "artificial" extremely limited data scenario, where we limited the amount of available training data to about 50 thousand words of word-aligned parallel text that we had collected during the SLE. We then designed a controlled experiment in order to compare our XFER system with our in-house SMT and

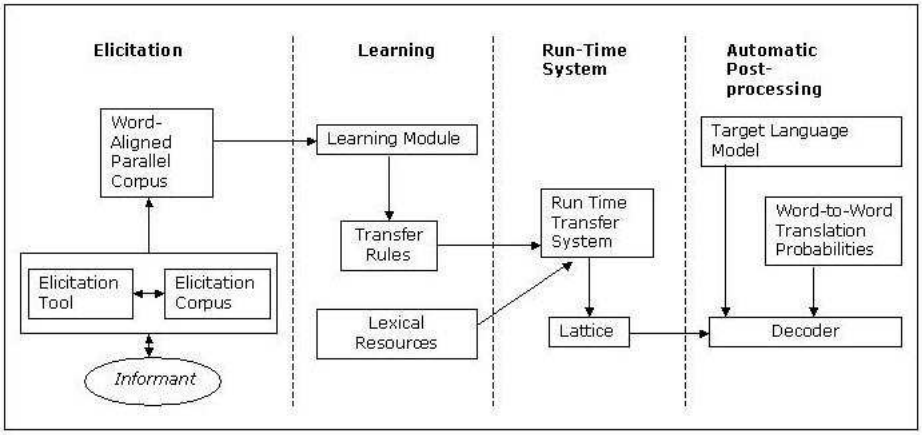

Fig. 1. Architecture of the XFER MT System and its Major Components

EBMT systems under this limited data scenario. The design, execution and results of this experiment are the focus of this paper. The results of the experiment indicate that under these extremely limited training data conditions, when tested on unseen data, the XFER system significantly outperforms both EBMT and SMT. Several different versions of the XFER system were tested. Results indicated that automatically learned transfer rules are effective in improving translation performance, compared with a baseline word-to-word translation version of our system. System performance with a limited number of manually written transfer rules was, however, still better than the current automatically inferred rules. Furthermore, a "multi-engine" version of our system that combined the output of the XFER and SMT systems and optimizes translation selection outperformed both individual systems.

The remainder of this paper is organized as follows. Section 2 presents an overview of the XFER system and its components. Section 3 describes the elicited data collection for Hindi-English that we conducted during the SLE, which provided the bulk of training data for our limited data experiment. Section 4 describes the specific resources and components that were incorporated into our Hindi-to-English XFER system. Section 5 then describes the controlled experiment for comparing the XFER , EBMT and SMT systems under the limited data scenario, and the results of this experiment. Finally, Section 6 describes our conclusions and future research directions.

2. TRAINABLE TRANSFER-BASED MT OVERVIEW

The fundamental principles behind the design of our XFER approach for MT are that it is possible to automatically learn syntactic transfer rules from limited amounts of word-aligned data, that such data can be elicited from non-expert bilingual speakers of the pair of languages, and that the rules learned are useful for machine translation between the two languages. We assume that one of

the two languages involved is a "major" language (such as English or Spanish) for which significant amounts of linguistic resources and knowledge are available.

The XFER system consists of four main sub-systems: elicitation of a word aligned parallel corpus; automatic learning of transfer rules; the run time transfer system; and a statistical decoder for selection of a final translation output from a large lattice of alternative translation fragments produced by the transfer system. Figure 1 shows how the four sub-systems are used in a configuration in which the translation is from a limited-resource source language into a major target language, such as English.

2.1 Elicitation of Word-Aligned Parallel Data

The purpose of the elicitation sub-system is to collect a high quality, word aligned parallel corpus. A specially designed user interface was developed to allow bilingual speakers to easily translate sentences from a corpus of the *major language* (i.e. English) into their native language (i.e. Hindi), and to graphically annotate the word alignments between the two sentences. Figure 2 contains a snap-shot of the elicitation tool, as used in the translation and alignment of an English sentence into Hindi. The informant must be bilingual and literate in the language of elicitation and the language being elicited, but does not need to have knowledge of linguistics or computational linguistics.

The word-aligned elicited corpus is the primary source of data from which transfer rules are inferred by our system. In order to support effective rule learning, we designed a "controlled" English elicitation corpus. The design of this corpus was based on elicitation principles from field linguistics, and the variety of phrases and sentences attempts to cover a wide variety of linguistic phenomena that the minor language may or may not possess. The elicitation process is organized along "minimal pairs", which allows us to identify whether the minor languages possesses specific linguistic phenomena (such as gender, number, agreement, etc.). The sentences in the corpus are ordered in groups corresponding to constituent types of increasing levels of complexity. The ordering supports the goal of learning compositional syntactic transfer rules. For example, simple noun phrases are elicited before prepositional phrases and simple sentences, so that during rule learning, the system can detect cases where transfer rules for NPs can serve as components within higher-level transfer rules for PPs and sentence structures. The current controlled elicitation corpus contains about 2000 sentences. It is by design very limited in vocabulary. A more detailed description of the controlled elicitation corpus, the elicitation process and the interface tool used for elicitation can be found in [Probst et al. 2001], [Probst and Levin 2002].

2.2 Automatic Transfer Rule Learning

The rule learning system takes the elicited, word-aligned data as input. Based on this information, it then infers syntactic transfer rules. The learning system also learns the composition of transfer rules. In the compositionality learning stage, the learning system identifies cases where transfer rules for "lower-level" constituents (such as NPs) can serve as components within "higher-level" transfer rules (such as PPs and sentence structures). This process generalizes the applicability of the learned transfer rules and captures the compositional makeup of syntactic corre-

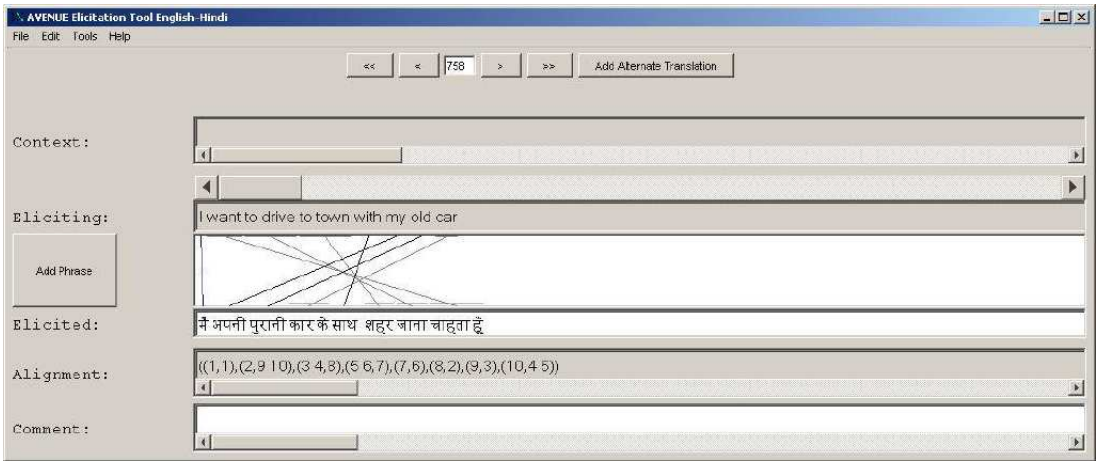

Fig. 2. The Elicitation Interface as Used to Translate and Align an English Sentence into Hindi

spondences between the two languages. The output of the rule learning system is a set of transfer rules that then serve as a transfer grammar in the run-time system. The transfer rules are comprehensive in the sense that they include all information that is necessary for parsing, transfer, and generation. In this regard, they differ from 'traditional' transfer rules that exclude parsing and generation information. Despite this difference, we will refer to them as transfer rules.

The design of the transfer rule formalism itself was guided by the consideration that the rules must be simple enough to be learned by an automatic process, but also powerful enough to allow manually-crafted rule additions and changes to improve the automatically learned rules.

The following list summarizes the components of a transfer rule. In general, the x-side of a transfer rules refers to the source language (SL), whereas the y-side refers to the target language (TL).

—**Type information:** This identifies the type of the transfer rule and in most cases corresponds to a syntactic constituent type. Sentence rules are of type `S`, noun phrase rules of type `NP`, etc. The formalism also allows for SL and TL type information to be different.

—**Part-of speech/constituent information:** For both SL and TL, we list a linear sequence of components that constitute an instance of the rule type. These can be viewed as the 'right-hand sides' of context-free grammar rules for both source and target language grammars. The elements of the list can be lexical categories, lexical items, and/or phrasal categories.

—**Alignments:** Explicit annotations in the rule describe how the set of source language components in the rule align and transfer to the set of target language components. Zero alignments and many-to-many alignments are allowed.

—**X-side constraints:** The x-side constraints provide information about features and their values in the source language sentence. These constraints are used at run-time to determine whether a transfer rule applies to a given input sentence.

```
;; PASSIVE SIMPLE PRESENT
VP::VP : [V V Aux] -> [Aux being V]
(
  (X1::Y3)
  ((x1 form) = part)
  ((x1 aspect) = perf)
  ((x2 form) = part)
  ((x2 aspect) = imperf)
  ((x2 lexwx) = 'jAnA')
  ((x3 lexwx) = 'honA')
  ((x3 tense) = pres)
  ((x0 tense) = (x3 tense))
  (x0 = x1)
  ((y1 lex) = be)
  ((y1 tense) = pres)
  ((y3 form) = part)
)
```

Fig. 3. A transfer rule for present tense verb sequences in the passive voice

—**Y-side constraints:** The y-side constraints are similar in concept to the x-side constraints, but they pertain to the target language. At run-time, y-side constraints serve to guide and constrain the generation of the target language sentence.

—**XY-constraints:** The xy-constraints provide information about which feature values transfer from the source into the target language. Specific TL words can obtain feature values from the source language sentence.

For illustration purposes, Figure 3 shows an example of a transfer rule for translating the verbal elements of a present tense sentence in the passive voice. This rule would be used in translating the verb sequence *bheje jAte hãi* ("are being sent," literally, *sent going present-tense-auxiliary*) in a sentence such as *Ab-tak patr .dAk se bheje jAte hãi.* ("Letters still are being sent by mail", literally *up-to-now letters mail by sent going present-tense-auxiliary*). The `x`-side constraints in Figure 3 show that the Hindi verb sequence consists of a perfective participle (`x1`), the passive auxiliary (*jAnA*, "go") inflected as an imperfective participle (`x2`), and an auxiliary verb in the present tense (`x3`). The `y`-side constraints show that the English verb sequence starts with the auxiliary verb *be* in the present tense (`y1`). The second element in the English verb sequence is *being*, whose form is invariant in this context. The English verb sequence ends with a verb in past participial form (`y3`). The alignment (`X1::Y3`) shows that the first element of the Hindi verb sequence corresponds to the last verb of the English verb sequence.

Rules such as the one shown in Figure 3 can be written by hand or learned automatically from elicited data. Learning from elicited data proceeds in three stages: the first phase, Seed Generation, produces initial 'guesses' at transfer rules. The rules that result from Seed Generation are 'flat' in that they specify a sequence of parts of speech, and do not contain any non-terminal or phrasal nodes. The second phase, Compositionality Learning, adds structure using previously learned rules. For instance, it learns that sequences such as `Det N P` and `Det Adj N P` can be re-written more generally as `NP P` as an expansion of PP in Hindi. This

generalization process can be done automatically based on the flat version of the rule, and a set of previously learned transfer rules for NPs.

The first two stages of rule learning result in a collection of structural transfer rules that are context-free – they do not contain any unification constraints that limit their applicability. Each of the rules is associated with a collection of elicited examples from which the rule was created. The rules can thus be augmented with a collection of unification constraints, based on specific features that are extracted from the elicited examples. The constraints can then limit the applicability of the rules, so that a rule may "succeed" only for inputs that satisfy the same unification constraints as the phrases from which the rule was learned. A constraint relaxation technique known as "Seeded Version Space Learning" attempts to increase the generality of the rules by identifying unification constraints that can be relaxed without introducing translation errors. Detailed descriptions of the rule learning process can be found in [Probst et al. 2003].

2.3 The Runtime Transfer System

At run time, the translation module translates a source language sentence into a target language sentence. The output of the run-time system is a lattice of translation alternatives. The alternatives arise from syntactic ambiguity, lexical ambiguity, multiple synonymous choices for lexical items in the dictionary, and multiple competing hypotheses from the rule learner.

The runtime translation system incorporates the three main processes involved in transfer-based MT: parsing of the SL input, transfer of the parsed constituents of the SL to their corresponding structured constituents on the TL side, and generation of the TL output. All three of these processes are performed based on the transfer grammar – the comprehensive set of transfer rules that are loaded into the runtime system. In the first stage, parsing is performed based solely on the "x" side of the transfer rules. The implemented parsing algorithm is for the most part a standard bottom-up Chart Parser, such as described in [Allen 1995]. A chart is populated with all constituent structures that were created in the course of parsing the SL input with the source-side portion of the transfer grammar. Transfer and generation are performed in an integrated second stage. A dual TL chart is constructed by applying transfer and generation operations on each and every constituent entry in the SL parse chart. The transfer rules associated with each entry in the SL chart are used in order to determine the corresponding constituent structure on the TL side. At the word level, lexical transfer rules are accessed in order to seed the individual lexical choices for the TL word-level entries in the TL chart. Finally, the set of generated TL output strings that corresponds to the collection of all TL chart entries is collected into a TL lattice, which is then passed on for decoding. A more detailed description of the runtime transfer-based translation sub-system can be found in [Peterson 2002].

2.4 Target Language Decoding

In the final stage, a statistical decoder is used in order to select a single target language translation output from a lattice that represents the complete set of translation units that were created for all substrings of the input sentence. The translation units in the lattice are organized according the positional start and end indices of

the input fragment to which they correspond. The lattice typically contains translation units of various sizes for different contiguous fragments of input. These translation units often overlap. The lattice also includes multiple word-to-word (or word-to-phrase) translations, reflecting the ambiguity in selection of individual word translations.

The task of the statistical decoder is to select a linear sequence of adjoining but non-overlapping translation units that maximizes the probability of $p(e|f)$, where $f = f_1...f_J$ is the source sentence and $e = e_1...e_I$ is the sequence of target language words. According to Bayes decision rule we have to search for

$$\hat{e} = \underset{e}{\operatorname{argmax}}\, p(e|f) = \underset{e}{\operatorname{argmax}}\, p(f|e)p(e) \quad . \tag{1}$$

The language model $p(e)$ describes how well-formed a target language sentence e is. We use a standard trigram model:

$$p(e) = \prod_{1}^{I} p(e_i|e_{i-2}e_{i-1}) \quad . \tag{2}$$

The translation probability $p(f|e)$ for the entire target sentence is the product of the translation probabilities of the individual translation units. We use the so-called "IBM-1" alignment model [Brown et al. 1993] to train a statistical lexicon. Phrase-to-phrase translation probabilities are then calculated using this lexicon:

$$p(\tilde{f}|\tilde{e}) = \prod_{j} \sum_{i} p(f_j|e_i) \quad , \tag{3}$$

where the product runs over all words in the source phrase and sum over all words in the target phrase. For any possible sequence s of non-overlapping translation units which fully cover the source sentence, the total translation model probability is then:

$$p(f|e) \;=\; \prod_{s} p(\tilde{f}|\tilde{e}) = \prod_{s} \prod_{j} \sum_{i} p(f_j^s|e_i^s) \quad . \tag{4}$$

The search algorithm considers all possible sequences s in the lattice and calculates the product of the language model probability and the translation model probability for the resulting sequence of target words. It then selects the sequence which has the highest overall probability.

As part of the decoding search, the decoder can also perform a limited amount of re-ordering of translation units in the lattice, when such reordering results in a better fit to the target language model. Reordering is performed by skipping over several words in the source sentence, i.e. leaving a gap, translating the word or phrase further towards the end of the sentence, and filling in the gap afterwards. The word re-orderings are scored using a Gaussian probability distribution, i.e. longer movements are less likely than shorter ones, with mean zero and a variance optimized on a development test set. To keep decoding time limited, we use a beam search, i.e. partial translation hypotheses which are low scoring compared to the best scoring hypothesis up to that point are pruned from further consideration.

3. ELICITED DATA COLLECTION

The data for the limited data scenario consisted entirely of phrases and sentences that were translated and aligned by Hindi speakers using our elicitation tool. Two very different corpora were used for elicitation: our typological elicitation corpus and a set of phrases from the Brown Corpus ([Fra]) that we extracted from the Penn Treebank ([Tre]).

The typological elicitation corpus covers basic sentence and noun phrase types, moving from simpler to more complex sentences as a linguistic field worker would do. We use it to insure that at least one example of each basic phenomenon (tense, agreement, case marking, pronouns, possessive noun phrases with various types of possessors, etc.) is encountered. However, the elicitation corpus has the shortcomings that we would encounter with any artificial corpus. The vocabulary is limited; the distribution and frequency of phrases does not reflect what would occur in naturally occurring text; and it does not cover everything that occurs in a natural corpus.

We would like to have the advantages of a natural corpus, but natural corpora also have shortcomings. In order to contain enough examples to fill paradigms of basic phenomena the corpus must be large and in order to contain sufficient examples of sparse phenomena, it must be very large. Furthermore, we would like to maintain the convenience of a compositionally ordered corpus, with smaller phrases building up into larger ones.

As a compromise, we used the Penn TreeBank to extract phrases from the Brown Corpus. The phrases were extracted from the parsed trees so that they could be sorted according to their daughter nodes (noun phrases containing only nouns, noun phrases containing determiners and nouns, etc.) In this way, we obtained a naturally occurring corpus that was also ordered compositionally.

The 864 phrases and sentences from the typological elicitation corpus were translated into Hindi by three Hindi speakers working together. After the first 150 sentences we checked the vocabulary, spelling, and word alignments. The Hindi speakers were then instructed on how to align case markers and auxiliary verbs in the way that is best for our rule learning system, and completed the translation and alignment of the corpus in less than 20 hours (resulting in a total of about 60 hours of human labor).

The extraction of phrases from the Brown Corpus resulted in tens of thousands of noun phrases and prepositional phrases, some containing embedded sentences. The phrases were first sorted by their complexity, determined by the depth of the parse-tree of the extracted phrase. They were then divided into files of about 200 phrases per file. The files were distributed to fifteen Hindi speakers for translation and alignment. After each Hindi speaker had translated about two hundred phrases (one file), the spelling and grammar were checked. Some non-native speakers were then eliminated from the pool of translators. Because the Brown Corpus contains some obscure vocabulary (e.g., names of chemical compounds) and because some noun phrases and prepositional phrases were not understandable out of context, the Hindi speakers were instructed to skip any phrases that they couldn't translate instantly. Only a portion of the files of extracted data were translated by reliable informants. The final resulting collection consisted of 85 files, adding up to a total

Description	Morpher	XFER
past participle	(tam = *yA*)	(aspect = perf) (form = part)
present participle	(tam = *wA*)	(aspect = imperf) (form = part)
infinitive	(tam = *nA*)	(form = inf)
future	(tam = *future*)	(tense = fut)
subjunctive	(tam = *subj*)	(tense = subj)
root	(tam = *0*)	(form = root)

Table I. Tense, Aspect, and Mood Features for Morpher and XFER

of 17,589 translated and word-aligned phrases.

We estimated the total amount of human effort required in collecting, translating and aligning the elicited phrases based on a sample. The estimated time spent on translating and aligning a file (of 200 phrases) was about 8 hours. Translation took about 75% of the time, and alignment about 25%. We estimate the total time spent on all 85 files to be about 700 hours of human labor.

Our approach requires elicited data to be translated from English into the "minor" language (Hindi in this case), even though our trained XFER system performs translation in the opposite direction. This has both advantages and disadvantages. The main advantage was our ability to rely on a extensive resources available for English, such as tree-banks. The main disadvantage was that typing in Hindi was not very natural even for the native speakers of the language, resulting in some level of typing errors. This, however, did not pose a major problem because the extracted rules are mostly generalized to the part-of-speech level. Furthermore, since the runtime translation direction is from Hindi to English, rules that include incorrect Hindi spelling will not match during translation, but will not cause incorrect translation.

4. HINDI-TO-ENGLISH TRANSFER MT SYSTEM

4.1 Morphology and Grammars

4.1.1 *Morphology.* The morphology module used by the runtime XFER system was the IIIT Morpher [Mor]. Given a fully inflected word in Hindi, Morpher outputs the root and other features such as gender, number, and tense. To integrate the IIIT Morpher with our system, we installed it as a server. The IIIT Morpher uses a romanized character encoding for Hindi known as *Roman-WX*. Since the rest of our system was designed to process Hindi in UTF-8 Unicode encoding, we implemented an input-output "wrapper" interface with the IIIT Morpher, that converted the input and output Hindi encoding as needed.

Figure 4 shows a sample of the morphology output, for the word *raha* (*continue, stay, keep*). The lefthand side of the figure shows the raw output of the morphology system. The righthand side shows our transformation of the Morpher output into our grammar formalism and feature system. Table I shows the set of features that Morpher uses for tense, aspect, and mood and the corresponding features that we mapped them into.

4.1.2 *Manually written grammar.* A complete transfer grammar cannot be written in one month (as in the Surprise Language Exercise), but partial manually developed grammars can be developed and then used in conjunction with automat-

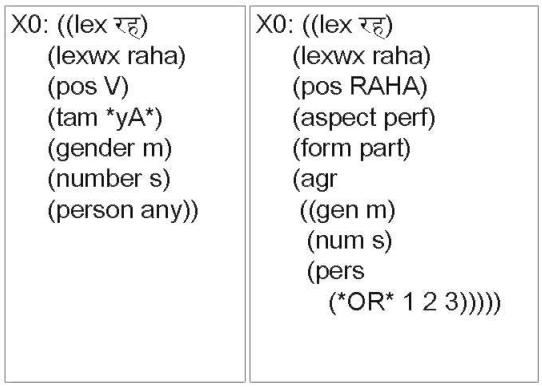

Fig. 4. Sample morphology output

ically learned rules, lexicon entries and even other MT engines in a multi-engine system. During the SLE, we experimented with both hand written grammars and automatically learned grammars. While the main focus of our research is on developing automatic learning of transfer rules, the manually developed transfer rule grammar can serve as an excellent point of comparison in translation evaluations. Furthermore, as pointed out above, the manual and automatically learned grammars can in fact be complimentary and combined together.

Our grammar of manually written rules has 70 transfer rules. The grammar includes a rather large verb paradigm, with 58 verb sequence rules, ten recursive noun phrase rules and two prepositional phrase rules.

The verb sequences that are covered by the rules cover the following tense, aspect, and mood categories: simple present, simple past, subjunctive, future, present perfect, past perfect, future perfect, progressive, past progressive, and future progressive. Each tense/aspect/mood can be combined with one of the "light" verbs *jAnA* (*go*, to indicate completion), *lenA* (*take*, to indicate an action done for one's own benefit) or *denA*, (*give* to indicate an action done for another's benefit). Active and passive voice are also covered for all tense/aspect/moods.

The noun phrase rules include pronouns, nouns, compound nouns, adjectives and determiners. For the prepositions, the PP rules invert the order of the postposition and the governed NP and move it at after the next NP. The rules shown in Figure 5 can flip arbitrarily long left branching Hindi NPs into right branching English NPs as shown in Figure 6.

4.1.3 *Automatically learned grammar.* In addition to the manually written grammar, we applied our rule-learning module to the corpus of collected NP and PP phrases, and acquired a grammar of automatically inferred transfer rules. The rules were learned as described briefly in Section 2 and in greater detail in [Probst et al. 2003].

The learned grammar consists of a total of 327 rules, which are exclusively NP and PP rules, as inferred from the Penn Treebank elicited data. In a second round of experiments, we assigned probabilities to the rules based on the frequency of the rule (i.e. how many training examples produce a certain rule). We then pruned

```
{NP,12}
NP::NP : [PP NP1] -> [NP1 PP]
((X1::Y2)
 (X2::Y1))

{NP,13}
NP::NP : [NP1] -> [NP1]
((X1::Y1))

{PP,12}
PP::PP : [NP Postp] -> [Prep NP]
((X1::Y2)
 (X2::Y1))
```

Fig. 5. Recursive NP Rules

```
Hindi NP with left recursion
(jIvana (life) ke (of)  eka (one) aXyAya (chapter)):
[np [pp [np [nbar [n jIvana]]] [p ke]] [nbar [adj eka] [n aXyAya]]]

English NP with right recursion
(one chapter of life):
[np [nbar [adj one] [n chapter]] [pp [p of] [np [nbar [n life]]]]]
```

Fig. 6. Transfer of a Hindi recursive NP into English

```
NP::NP [ADJ N] -> [ADJ N]
((X1::Y1) (X2::Y2)
((X1 NUM) = (Y2 NUM))
((X2 CASE) = (X1 CASE))
((X2 GEN) = (X1 GEN))
((X2 NUM) = (X1 NUM)))

PP::PP [NP POSTP] -> [PREP NP]
((X2::Y1)
 (X1::Y2))

PP::PP [N CONJ NUM N N N POSTP] -> [PREP N CONJ NUM N N N]
((X7::Y1) (X1::Y2) (X2::Y3) (X3::Y4) (X4::Y5) (X5::Y6) (X6::Y7))
```

Fig. 7. Some Automatically Learned Rules

rules with low probability, resulting in a grammar of a mere 16 rules. The rationale behind pruning rules is that low-probability rules will produce spurious translations most of the time, as they fire in contexts where they should not actually apply. In our experience, rule pruning has very little effect on the translation performance, but great impact on the efficiency of the system.

Figure 7 shows some rules that were automatically inferred from the training data. Note that the second rule contains a non-terminal symbol (NP) that was learned by the compositionality module.

4.2 Lexical Resources

The transfer engine uses a grammar and a lexicon for translation. The lexicon contains entries from a variety of sources. The most obvious source for lexical translation pairs is the elicited corpus itself. The translations pairs can simply be read off from the alignments that were manually provided by Hindi speakers. Because the alignments did not need to be 1-to-1, the resulting lexical translation pairs can have strings of more than one word one either the Hindi or English side or both.

Another source for lexical entries is the is English-Hindi dictionary provided by the Linguistic Data Consortium (LDC). The LDC dictionary contains many (as many as 25) English translations for each Hindi word. Since some of the English translations are not frequent or are not frequent translations of the Hindi word, two local Hindi experts "cleaned up" a portion of this lexicon, by editing the list of English translations provided for the Hindi words, and leaving only those that were "best bets" for being reliable, all-purpose translations of the Hindi word. The full LDC lexicon was first sorted by Hindi word frequency (estimated from Hindi monolingual text) and the cleanup was performed on the most frequent 12% of the Hindi words in the lexicon. The "clean" portion of the LDC lexicon was then used for the limited-data experiment. This consisted of 2725 Hindi words, which corresponded to about 10,000 translation pairs. This effort took about 3 days of manual labor.

The entries in the LDC dictionary are in root forms (both in English and Hindi). In order to be able to produce inflected English forms, such as plural nouns, we 'enhanced' the dictionary with such forms. The enhancement works as follows: for each noun in the dictionary (where the part-of-speech tags are indicated in the original dictionary and cross-checked in the British National Corpus [Leech 1992]), we create an additional entry with the Hindi word unchanged, and the English word in the plural form. In addition to the changed English word, we also add a *constraint* to the lexical rule indicating that the number is plural. A similar strategy was applied to verbs: we added entries (with constraints) for past tense, past participle, gerund (and continuous), as well as future tense entry. The result is a set of lexical entries associating Hindi root forms to English inflected forms.

How and why do these additional entries work? The transfer engine first runs each Hindi input word through the morphological analyzer. The Morpher returns the root form of the word, along with a set of morphological features. The root form is then matched against the lexicon, and the set of lexical transfer rules that match the Hindi root are extracted. The morphological features of the input word are then unified with the feature constraints that appear in each of the candidate lexical transfer rules, pruning out the rules that have inconsistent features. For example, assume we are processing a plural noun in Hindi. The root form of the noun will be used to extract candidate lexical transfer rules from the lexicon, but only the entry that contains a plural form in English will pass unification and succeed.

Since we did not have immediate access to an English morphology module, the word inflections were primarily based on spelling rules. Those rules indicate, for instance, when to reduplicate a consonant, when to add 'es' instead of a simple 's'

for plural, etc. For irregular verb inflections, we consulted a list of English irregular verbs. With these enhancements, the part of our dictionary derived from the LDC Hindi-English dictionary contains a total of 23,612 translation pairs.

To create an additional resource for high-quality translation pairs, we used monolingual Hindi text to extract the 500 most frequent bigrams. These bigrams were then translated into English by an expert in about 2 days. Some judement was applied in selecting bigrams that could be translated reliably out of context.

Finally, our lexicon contains a number of manually written rules.

—72 manually written phrase transfer rules: a bilingual speaker manually entered English translation equivalents for 72 common Hindi phrases that were observed as frequently occurring in development-test data early on during the SLE.

—105 manually written postposition rules: Hindi uses a collection of *post-position* words, that generally correspond with English prepositions. A bilingual speaker manually entered the English translation equivalents of 105 common postposition phrases (combinations of nouns and the following postpositions), as observed in development-test data.

—48 manually written time expression rules: a bilingual speaker manually entered the English translation equivalents of 48 common time expressions, as observed in development-test data.

4.3 Runtime Configuration

The transfer engine is run in a mode where it outputs all possible (partial) translations. The result of a transfer instance on a given input sentence is a lattice where each entry indicates which Hindi words the partial translation spans, and what the potential English translation is. The decoder then rescores these entries and finds the best path through the lattice (for more details, see Section 2.4).

Production of the lattice takes place in three passes. Although all possible (partial) translations are output, we have to treat with care the interaction between the morphology, the grammar rules, and the lexicon. The first pass matches the Hindi sentence against the lexicon in their full form, before applying morphology. This is designed especially to allow phrasal lexical entries to fire, where any lexical entry with more than one Hindi word is considered a phrasal entry. In this phase, no grammar rules are applied.

Another pass runs each Hindi input word through the morphology module to obtain the root forms along with any inflectional features of the word. These root forms are then fed into the grammar rules. Note that this phase takes advantage of the enhancements in the lexicon, as described in Section 4.2.

Finally, the original Hindi words (in their inflected forms) are matched against the dictionary, producing word-to-word translations, without applying grammar rules.

These three passes exhaust the possibilities of matching lexical entries and applying grammar rules in order to maximize the size of the resulting lattice. A bigger lattice is generally preferable, as its entries are rescored and low-probability entries are not chosen by the language model for the final translation.

5. THE LIMITED DATA SCENARIO EXPERIMENT

5.1 Training and Testing Data

In order to compare the effectiveness of the XFER system with data-driven MT approaches (SMT and EBMT) under a scenario of truly limited data resources, we artificially crafted a collection of training data for Hindi-to-English translation, that intentionally contained extremely limited amounts of parallel text. The training data was extracted from the larger pool of resources that had been collected throughout the SLE by our own group as well as the other participating groups in the SLE. The limited data consisted of the following resources:

—**Elicited Data Corpus:** 17,589 word-aligned phrases and sentences from the elicited data collection described in section 3. This includes both our translated and aligned *controlled* elicitation corpus, and also the translated and aligned *uncontrolled* corpus of noun phrases and prepositional phrases extracted from the Penn Treebank.

—**Small Hindi-to-English Lexicon**: 23,612 "clean" translation pairs from the LDC dictionary (as described in Section 4.2).

—**Small Amount of Manually Acquired Resources**: (As described in Section 4.2) — 500 most common Hindi bigrams, 72 manually written phrase transfer rules, 105 manually written postposition rules, and 48 manually written time expression rules.

The limited data setup includes no additional parallel Hindi-English text. The total amount of bilingual training data was estimated to amount to about 50,000 words.

A small, previously unseen, Hindi text was selected as a test-set for this experiment. The test-set chosen was a section of the data collected at Johns Hopkins University during the later stages of the SLE, using a web-based interface [JHU]. The section chosen consists of 258 sentences, for which four English reference translations are available. An example test sentence can be seen in Figure 8.

5.2 Experimental Testing Configuration

The experiment was designed to evaluate and compare several different configurations of the XFER system, and the two corpus-based systems (SMT and EBMT), all trained on the same limited-data resources, and tested on the same test set.

The transfer engine was run in a setting that finds all possible translations that are consistent with the transfer rules. The transfer engine produces a complete lattice of all possible partial translations. The lattice is then input into a statistical decoder. Decoding is performed as described in Section 2.4. We used an English language model that was trained on 70 million English words. As an additional functionality, the decoder can perform a limited reordering of arcs during decoding. At any point in the decoding, arcs that have a start position index of up to *four* words ahead are considered, and allowed to be "moved" to the current position in the output if this improves the overall score of the resulting output string according to the English language model.

The following systems were evaluated in the experiment:

(1) The following versions of the XFER system:

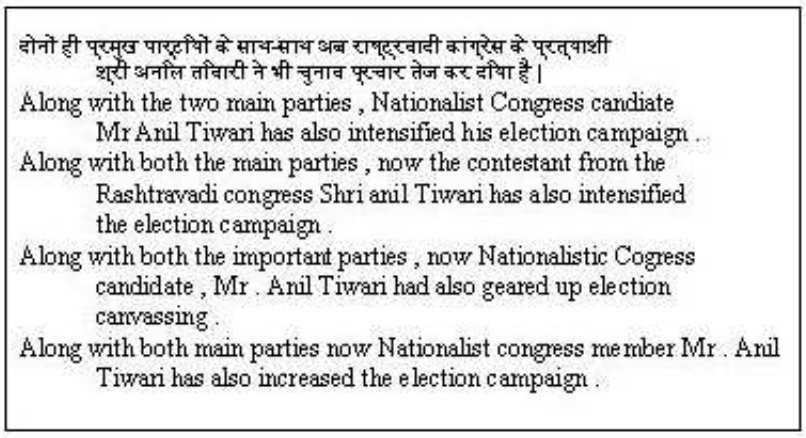

Fig. 8. Example of a Test Sentence and Reference Translations

(a) XFER **with No Grammar:** XFER with no syntactic transfer rules (i.e. only phrase-to-phrase matches and word-to-word lexical transfer rules, with and without morphology).
(b) XFER **with Learned Grammar:** XFER with automatically learned syntactic transfer rules, as described in section 4.1.3.
(c) XFER **with Manual Grammar:** XFER with the manually developed syntactic transfer rules, as described in section 4.1.2.

(2) **SMT:** The CMU Statistical MT (SMT) system [Vogel et al. 2003], trained on the limited-data parallel text resources.
(3) **EBMT:** The CMU Example-based MT (EBMT) system [Brown 1997], trained on the limited-data parallel text resources.
(4) **MEMT:** A "multi-engine" version that combines the lattices produced by the SMT system and the XFER system with manual grammar. The decoder then selects an output from the joint lattice.

5.3 Experimental Results

Performance of the systems was measured using the NIST scoring metric [Doddington 2003], as well as the BLEU score [Papineni et al. 2002]. In order to validate the statistical significance of the differences in NIST and BLEU scores, we applied a commonly used sampling technique over the test set: we randomly draw 258 sentences independently from the set of 258 test sentences (thus sentences can appear zero, once, or more in the newly drawn set). We then calculate scores for all systems on the randomly drawn set (rather than the original set). This process was repeated 10,000 times. Median scores and 95% confidence intervals were calculated based on the set of scores.

Table II compares the results of the different systems (with the decoder's best reordering window), along with the 95% confidence intervals. Figures 9 and 10 show the effect of different systems with different reordering windows in the decoder. For

System	BLEU	NIST
EBMT	0.058	4.22
SMT	0.102 (+/- 0.016)	4.70 (+/- 0.20)
XFER no grammar	0.109 (+/- 0.015)	5.29 (+/- 0.19)
XFER learned grammar	0.112 (+/- 0.016)	5.32 (+/- 0.19)
XFER manual grammar	0.135 (+/- 0.018)	5.59 (+/- 0.20)
MEMT (XFER +SMT)	0.136 (+/- 0.018)	5.65 (+/- 0.21)

Table II. System Performance Results for the Various Translation Approaches

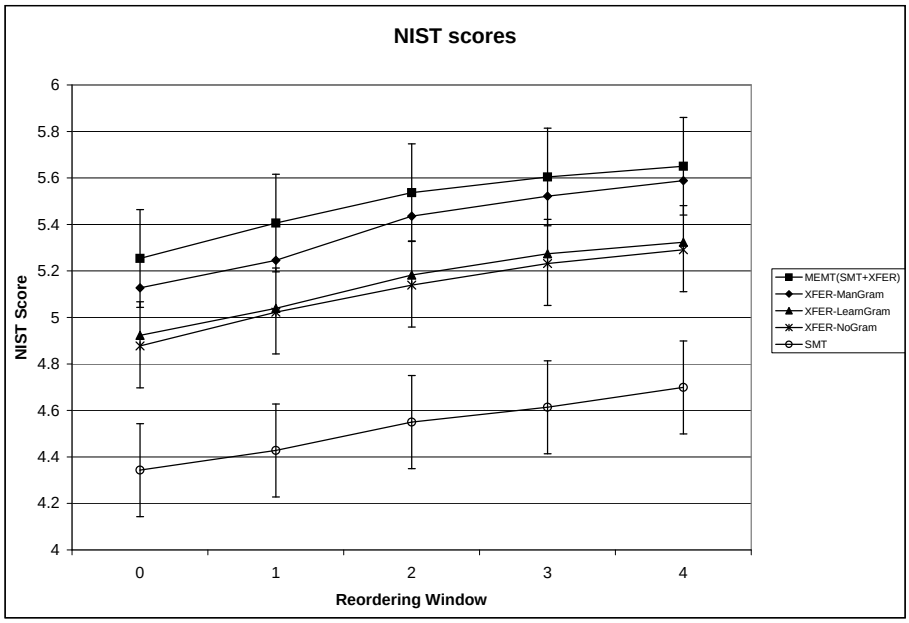

Fig. 9. Results by NIST score

clarity, confidence intervals are graphically shown only for the NIST scores (not for BLEU), and only for SMT, XFER with no grammar, and MEMT.

5.4 Discussion of Results

The results of the experiment clearly show that under the specific miserly data training scenario that we constructed, the XFER system, with all its variants, significantly outperformed the SMT system. While the scenario of this experiment was clearly and intentionally more favorable towards our XFER approach, we see these results as a clear validation of the utility and effectiveness of our transfer approach in other scenarios where only very limited amounts of parallel text and other online resources are available. In earlier experiments during the SLE, we observed that SMT outperformed the XFER system when much larger amounts of parallel text data were used for system training. This indicates that there exists a data "cross-over point" between the performance of the two systems: given more and more data, SMT will outperform the current XFER system. Part of future work

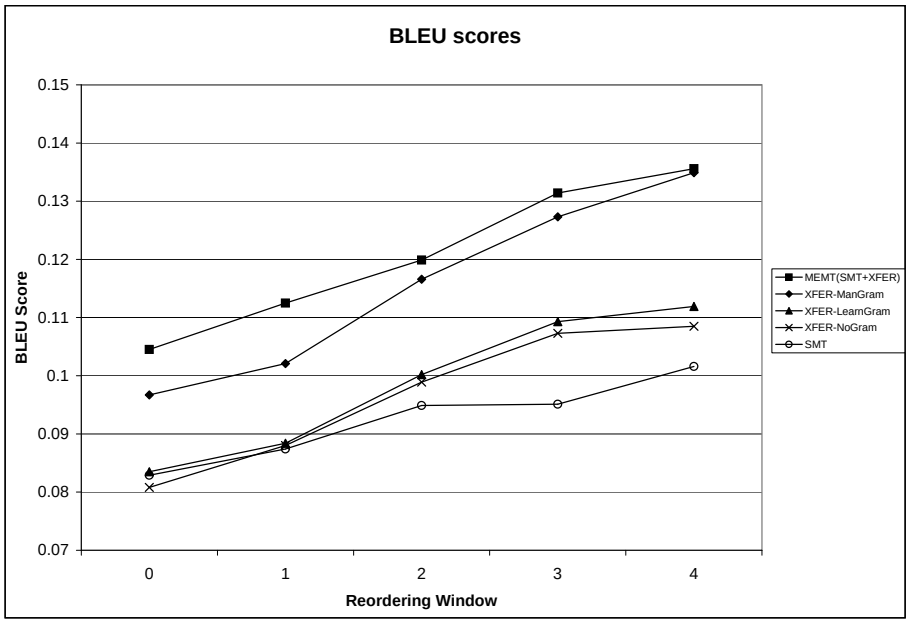

Fig. 10. Results by BLEU score

will be to first determine this cross-over point and then to attempt to push this point further toward scenarios where more data is given, thus making the XFER system applicable to a wider variety of conditions.

The use of morphology within the XFER system was also a significant factor in the gap in performance between the XFER system and the SMT system in the experiment. Token word coverage of the test-set without morphology is about 70%, whereas with morphology, token coverage increases to around 79%. We acknowledge that the availablility of a high-coverage morphological analyzer for Hindi worked to our favor, and a morphological analyzer of such quality may not be available for many other languages. Our XFER approach, however, can function even with partial morphological information, with some consequences on the effectiveness and generality of the rules learned.

The results of the comparison between the various versions of the XFER system also show interesting trends, although the statistical significance of some of the differences is not very high. XFER with the manually developed transfer rule grammar clearly outperformed (with high statistical significance) XFER with no grammar and XFER with automatically learned grammar. XFER with automatically learned grammar is slightly better than XFER with no grammar, but the difference is statistically not very significant. We take these results to be highly encouraging, since both the manually written and automatically learned grammars were very limited in this experiment. The automatically learned rules only covered NPs and PPs, whereas the manually developed grammar mostly covers verb constructions. While our main objective is to infer rules that perform comparably to hand-written rules, it is encouraging that the hand-written grammar rules result in

a big performance boost over the no-grammar system, indicating that there is much room for improvement. If the learning algorithms are improved, the performance of the overall system can also be improved significantly.

The significant effects of decoder reordering are also quite interesting. On one hand, we believe this indicates that various more sophisticated rules could be learned, and that such rules could better order the English output, thus reducing the need for re-ordering by the decoder. On the other hand, the results indicate that some of the "burden" of reordering can remain within the decoder, thus possibly compensating for weaknesses in rule learning.

Finally, we were pleased to see that the consistently best performing system was our multi-engine configuration, where we combined the translation hypotheses of the SMT and XFER systems together into a common lattice and applied the decoder to select a final translation. The MEMT configuration outperformed the best purely XFER system with reasonable statistical confidence. Obtaining a multi-engine combination scheme that consistently outperforms all the individual MT engines has been notoriously difficult in past research. While the results we obtained here are for a unique data scenario, we hope that the framework applied here for multi-engine integration will prove to be effective for a variety of other scenarios as well. The inherent differences between the XFER and SMT approaches should hopefully make them complementary in a broad range of data scenarios.

6. CONCLUSIONS AND FUTURE WORK

The DARPA-sponsored SLE allowed us to develop and test an open-domain large-scale Hindi-to-English version of our XFER system. This experience was extremely helpful for enhancing the basic capabilities of our system. The lattice-based decoding was added to our system at the very end of the month-long SLE, and proved to be very effective in boosting the overall performance of our XFER system.

The experiments we conducted under the extremely limited Hindi data resources scenario were very insightful. The results of our experiments indicate that our XFER system in its current state outperforms SMT and EBMT when the amount of available parallel training text is extremely small. The XFER system with manually developed transfer-rules outperformed the version of the system with automatically learned rules. This is partly due to the fact that we only attempted to learn rules for NPs and VPs in this experiment. We see the current results as an indication that there is significant room for improving automatic rule learning. In particular, the learning of unification constraints in our current system requires significant further research.

In summary, we feel that we have made significant steps towards the development of a statistically grounded transfer-based MT system with: (1) rules that are scored based on a well-founded probability model; and (2) strong and effective decoding that incorporates the most advanced techniques used in SMT decoding. Our work complements recent work by other groups on improving translation performance by incorporating models of syntax into traditional corpus-driven MT methods. The focus of our approach, however, is from the "opposite end of the spectrum": we enhance the performance of a syntactically motivated rule-based approach to MT, using strong statistical methods. We find our approach particularly suitable for

languages with very limited data resources.

Acknowledgements

This research was funded in part by the DARPA TIDES program and by NSF grant number IIS-0121-631. We would like to thank our team of Hindi-English bilingual speakers in Pittsburgh and in India that conducted the data collection for the research work reported in this paper.

REFERENCES

The Brown Corpus. http://www.hit.uib.no/icame/brown/bcm.html.

The Johns Hopkins University Hindi translation webpage. http://nlp.cs.jhu.edu/ hindi.

Morphology module from IIIT. http://www.iiit.net/ltrc/morph/index.htm.

The Penn Treebank. http://www.cis.upenn.edu/~treebank/home.html.

Allen, J. 1995. *Natural Language Understanding*, Second Edition ed. Benjamin Cummings.

Brown, P., Cocke, J., Della Pietra, V., Della Pietra, S., Jelinek, F., Lafferty, J., Mercer, R., and Roossin, P. 1990. A statistical approach to Machine Translation. *Computational Linguistics 16,* 2, 79–85.

Brown, P., Della Pietra, V., Della Pietra, S., and Mercer, R. 1993. The mathematics of statistical Machine Translation: Parameter estimation. *Computational Linguistics 19,* 2, 263–311.

Brown, R. 1997. Automated dictionary extraction for knowledge-free example-based translation. In *International Conference on Theoretical and Methodological Issues in Machine Translation*. 111–118.

Doddington, G. 2003. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of Human Language Technology*.

Jones, D. and Havrilla, R. 1998. Twisted pair grammar: Support for rapid development of machine translation for low density languages. In *Proceedings of the Third Conference of the Association for Machine Translation in the Americas (AMTA-98)*.

Leech, G. 1992. 100 million words of english: The British National Corpus. *Language Research 28,* 1, 1–13.

Nirenburg, S. 1998. Project Boas: A linguist in the box as a multi-purpose language. In *Proceedings of the First International Conference on Language Resources and Evaluation (LREC-98)*.

Och, F. J. and Ney, H. Discriminative training and maximum entropy models for statistical machine translation.

Papineni, K., Roukos, S., and Ward, T. 1998. Maximum likelihood and discriminative training of direct translation models. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP-98)*. 189–192.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics (ACL)*. Philadelphia, PA, 311–318.

Peterson, E. 2002. Adapting a transfer engine for rapid machine translation development. M.S. thesis, Georgetown University.

Probst, K., Brown, R., Carbonell, J., Lavie, A., Levin, L., and Peterson, E. 2001. Design and implementation of controlled elicitation for machine translation of low-density languages. In *Workshop MT2010 at Machine Translation Summit VIII*.

Probst, K. and Levin, L. 2002. Challenges in automated elicitation of a controlled bilingual corpus. In *Theoretical and Methodological Issues in Machine Translation 2002 (TMI-02)*.

Probst, K., Levin, L., Peterson, E., Lavie, A., and Carbonell, J. 2003. Mt for resource-poor languages using elicitation-based learning of syntactic transfer rules. *Machine Translation*. to appear.

Sato, S. and Nagao, M. 1990. Towards memory-based translation. In *COLING-90*. 247–252.

Sherematyeva, S. and Nirenburg, S. 2000. Towards a unversal tool for NLP resource acquisition. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC-00)*.

Vogel, S. and Tribble, A. 2002. Improving statistical machine translation for a speech-to-speech translation task. In *Proceedings of the 7th International Conferenc on Spoken Language Processing (ICSLP-02)*.

Vogel, S., Zhang, Y., Tribble, A., Huang, F., Venugopal, A., Zhao, B., and Waibel, A. 2003. The cmu statistical translation system. In *MT Summit IX*.

Yamada, K. and Knight, K. 2001. A syntax-based statistical translation model. In *Proceedings of the 39th Anniversary Meeting of the Association for Computational Linguistics (ACL-01)*.