Stat-XFER: A General Search-based Syntax-driven Framework for Machine Translation

Alon Lavie

Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213 USA
alavie@cs.cmu.edu.edu

Abstract. The CMU Statistical Transfer Framework (Stat-XFER) is a general framework for developing search-based syntax-driven machine translation (MT) systems. The framework consists of an underlying syntax-based transfer formalism along with a collection of software components designed to facilitate the development of a broad range of MT research systems. The main components are a general language-independent run-time transfer engine and decoder, along with several different tools for creating the various underlying language-pair-specific resources that are required for building a specific MT system for any given language pair. We describe the general framework, its unique properties and features, and its application to the construction of MT research prototype systems for a diverse collection of language pairs.

1 Introduction

The field of Machine Translation (MT) has dramatically shifted in the course of the past decade. Modern state-of-the-art approaches to MT rely on machine learning methods of increasing complexity and sophistication in order to automatically acquire their underlying translation models from available data resources. Phrase-based Statistical MT (PB-SMT) [1–3] has become the predominant approach in recent years. In PB-SMT, simple statistical modeling methods are used to acquire likely phrase-to-phrase translation equivalents from large volumes of sentence-parallel text corpora. In the absence of large sentence-parallel data, the statistical estimation methods break down, and the approach becomes ineffective. Vast sentence-parallel corpora exist only for a limited number of language pairs (primarily pairs of European languages, Chinese, Japanese and Arabic), severely limiting the applicability of this approach. 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 foreseeable future. Corpus-based MT approaches will therefore not be effective for such languages for some time to come.

Furthermore, even for language pairs with large amounts of sentence-parallel data such as Chinese-English, the phrase-based models are often too simple and naive for capturing many complex divergences between the languages. There has been increasing recognition in the MT research community in recent years that high-quality fully-automatic MT will require learning translation models that can capture advanced syntax and semantic representations and how they correspond across languages. Automatically acquired syntax-based models for MT have started to receive increasing attention in the last few years [4, 3, 5].

Over the past six years, the AVENUE MT research group at Carnegie Mellon, under DARPA and NSF funding, has been developing a new MT framework that is designed to address many of the above challenges. The framework is inspired by many of the ideas of modern statistical MT. Most prominently, it is founded on the basic notion of search-based "decoding". The framework consists of an underlying syntax-based transfer formalism, a general, language-independent translation engine, and a collection of software components designed to facilitate the acquisition of the underlying language resources required for development of an MT system for any specific language pair. These resource acquisition tools target different scenarios, ranging from low-resource to high-resource availability, and support the development of a broad range of MT research systems. The framework has been designed to be able to handle large-scale broad-coverage lexical resources and transfer grammars. The acquisition of these resources can be done in diverse and creative ways, effectively combining automatic acquisition from data with human knowledge. We refer to this framework using the name "*statistical transfer*", or in short, Stat-XFER.

The Stat-XFER framework was originally developed to support rapid MT prototype development for translation between low-resource source languages (such as Hebrew) and high resource target languages (such as English). Over the past year, the Stat-XFER framework has been greatly extended to also support effective automatic acquisition of translation resources from vast parallel corpora. The focus of this paper, however, is mostly on scenarios involving low-resource source languages. We describe the general framework, its unique properties and features, and its application to the construction of MT research prototype systems for a diverse collection of language pairs. We use our Hebrew-to-English MT prototype system developed under the Stat-XFER framework to highlight many of the important aspects of the system.

2 The Stat-XFER Framework

The Stat-XFER framework uses a declarative formalism for symbolic transfer grammars. A grammar consists of a collection of *synchronous context-free* rules, which can be augmented by unification-style feature constraints. These transfer rules specify how phrase structures in a source-language correspond and transfer to phrase structures in a target language, and the constraints under which these rules should apply. The framework also includes a fully-implemented transfer engine that applies the transfer grammar to a source-language input sentence

```
{NP1,2}
;;SL: $MLH ADWMH
;;TL: A RED DRESS
;;Score:2
NP1::NP1 [NP1 ADJ] -> [ADJ NP1]
(
 (X2::Y1)
 (X1::Y2)
 ((X1 def) = -)
 ((X1 status) =c absolute)
 ((X1 num) = (X2 num))
 ((X1 gen) = (X2 gen))
 (X0 = X1)
)
```

```
{NP1,3}
;;SL: H $MLWT H ADWMWT
;;TL: THE RED DRESSES
;;Score:4
NP1::NP1 [NP1 "H" ADJ] -> [ADJ NP1]
(
 (X3::Y1)
 (X1::Y2)
 ((X1 def) = +)
 ((X1 status) =c absolute)
 ((X1 num) = (X3 num))
 ((X1 gen) = (X3 gen))
 (X0 = X1)
 )
```

Fig. 1. NP Transfer Rules for Nouns Modified by Adjectives from Hebrew to English

at runtime, and produces collections of scored word and phrase-level translations according to the grammar. Scores are based on a log-linear combination of several features, and a beam-search controls the underlying parsing and transfer process. The framework was designed to support research on a variety of methods for automatically acquiring transfer grammars from limited amounts of elicited word-aligned data. The framework also supports manual development of transfer grammars by experts familiar with the two languages.

The Stat-XFER framework has been applied to building research prototype MT systems for quite a number of language pairs over the past five years. The most developed prototype systems to date are our Hebrew-to-English and Chinese-to-English systems. The Hebrew system is described in detail in later sections of this paper. The Chinese system has been under development for the past year, and is being used as one of several engines for Chinese-to-English translation within the IBM-led Rosetta team as part of the DARPA/GALE program. Other integrated Stat-XFER prototypes include a Hindi-to-English system developed under the DARPA/TIDES "Surprise Language Exercise" in June-2003 [6] [7], and preliminary systems for German-to-English, Dutch-to-English and French-to-English. We have also been applying the approach to several native languages in North and South America, starting with a Mapudungun-to-Spanish system[1]. A prototype system for Inupiaq-to-English[2] is in initial stages of development. We are currently also collaborating with research groups in Brazil and in Turkey on developing MT prototypes for Portuguese-to-English and Turkish-to-English.

2.1 The Transfer Formalism

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

[1] Mapudungun is a native language of southern Chile.
[2] Inupiaq is a native language of northern Alaska.

to improve the automatically learned rules. To illustrate the rule formalism, we show two transfer rules for structurally transferring nouns modified by adjectives from Hebrew to English, depicted in Figure 1.

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.
- **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.

2.2 Runtime System Architecture

To describe the runtime archirecture of the Stat-XFER framework, we use our integrated Hebrew-to-English prototype for illustrative purposes. The core components, consisting of the *transfer engine* and the *decoder*, however, are language independent. The system consists of the following main components: a Hebrew input sentence is pre-processed, and then sent to a *morphological analyzer*, which produces all possible analyses for each input word, represented in the form of a lattice of possible input word lexemes and their morphological features. The input lattice is then passed on to the *transfer engine*, which applies a collection of lexical and structural *transfer rules* in order to parse, transfer and generate English translations for all possible word and phrase segments of the input. Each possible translation segment is scored by a combination of various features. The collection of translation segments is stored in an output lattice data-structure.

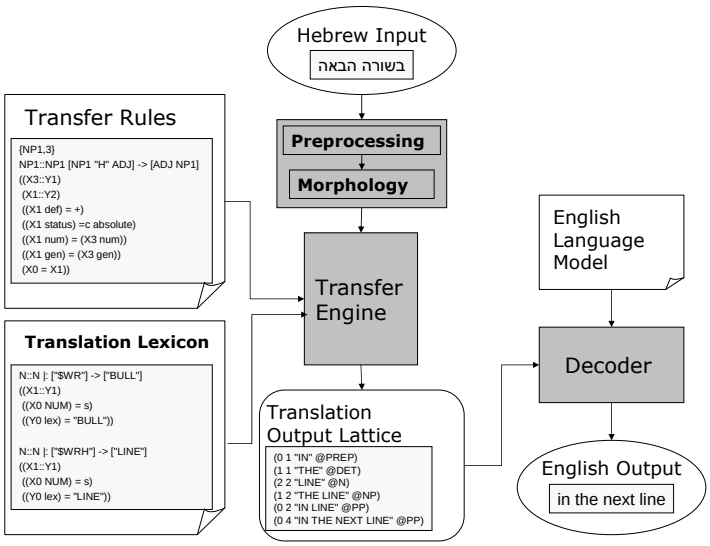

Fig. 2. Architecture of the Hebrew-to-English Transfer-based MT System

The transfer engine uses a beam-search to control the number of possible translation segments explored. The lexical transfer rules used by the transfer engine are derived from a *bilingual lexicon*, while the higher-level structural transfer rules come from either a manually-developed or automatically-acquired transfer grammar. In the final stage, the English lattice is fed into a *decoder* which uses a log-linear combination of several features to search and select a combination of sequential translation segments that together represent the best scoring translation of the entire input sentence. A schematic diagram of the system architecture can be seen in Figure 2.

2.3 The Transfer Engine

The transfer engine is the module responsible for applying the comprehensive set of lexical and structural transfer rules, specified by the translation lexicon and the transfer grammar (respectively), to the source-language (SL) input lattice, producing a comprehensive collection of target-language (TL) output segments. The output of the transfer engine is a lattice of alternative translation segments. The alternatives arise from syntactic ambiguity, lexical ambiguity, and multiple synonymous choices for lexical items in the translation lexicon.

The transfer engine 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 transfer engine at runtime. Parsing, transfer and generation are fully integrated into an interleaved bottom-up "parse-and-transfer" algorithm, which is essentially an extended Chart Parser. Parsing is performed based solely on the source-language side of the transfer rules. 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. A parallel TL chart is populated in lock-step, containing the translations created by transfering the source-side constituents as specified in the transfer rules. The bottom-up process is initialized by populating the TL chart with the lexical translations of all source words, based on all available lexical transfer rules. TL lexical generation, driven by a TL morphology engine, can also be applied at this initial stage. The TL chart maintains "stacks" of scored translation options for all substrings of the SL input. As parsing progresses, whenever a new source-side constituent is created, the transfer "instructions" of the completed rule are applied, thus creating the possible translations that correspond to the SL constituent. The set of translations is then added to the appropriate "stack" within the TL chart. Feature constraints contained within the rules are also applied in an integral interleaved fashion. "X-side" constraints are applied whenever a source-side constituent is completed. "X-Y" constraints and "Y-side" constraints are applied when performing transfer. Constraints do not generate additional translation alternatives. They can block rules from applying, or "weed out" possible translations created by any rule application. 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. The transfer engine was designed to support both manually-developed structural transfer grammars and grammars that can be automatically acquired from bilingual data. A more detailed description of the transfer engine can be found in [8].

2.4 Decoding

In the final stage, a monotonic decoder is used in order to create complete translation hypotheses from the lattice created during the transfer stage. The translation units in the lattice are organized according to 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 decoder is to select a linear sequence of adjoining but non-overlapping translation units that maximizes the overall score of the target language string given the source language string. The decoder uses a log-linear scoring model that combines scores from several different features. The current set of features include a *language model* of English, a score derived from the

rule probabilities, two lexical probability scores (for "target given source" and "source given target"), a measure that reflects the number of translation fragments being combined and a feature that reflects the source-to-target relative sentence length. For language modeling, we use the Suffix Array Toolkit (SALM) developed at CMU [9]. The framework also supports using the SRI language modeling toolkit. The decoder is monotonic in the sense that it cannot reorder any translation units from the lattice.

3 The Hebrew-to-English Stat-XFER System

Machine translation of Hebrew is challenging due to two main reasons: the high lexical and morphological ambiguity of Hebrew and its orthography, and the paucity of available resources for the language. We developed a first, fully functional, version of the Hebrew-to-English Stat-XFER system [10] over the course of a two-month period with a total labor-effort equivalent to about four person-months of development. To the best of our knowledge, our system is the first broad-domain machine translation system for Hebrew. We used existing, publicly available resources which we adapted in novel ways for the MT task, and directly addressed the major issues of lexical, morphological and orthographical ambiguity.

3.1 The Hebrew Language

Modern Israeli Hebrew, henceforth *Hebrew*, exhibits clear Semitic behavior. In particular, its lexicon, word formation and inflectional morphology are typically Semitic. The major word formation machinery is root-and-pattern, where roots are sequences of three (typically) or more consonants and patterns are sequences of vowels and, sometimes, also consonants, with "slots" into which the root's consonants are inserted. Inflectional morphology is highly productive and consists mostly of suffixes, but also prefixes and circumfixes.

The Hebrew script,[3] not unlike the Arabic one, attaches several short particles to the word which immediately follows them. These include, *inter alia*, the definite article *H* ("the"), prepositions such as *B* ("in"), *K* ("as"), *L* ("to") and *M* ("from"), subordinating conjunctions such as *$* ("that") and *K$* ("when"), relativizers such as *$* ("that") and the coordinating conjunction *W* ("and"). The script is rather ambiguous as the prefix particles can often also be parts of the stem. Thus, a form such as *MHGR* can be read as a lexeme "immigrant", as *M-HGR* "from Hagar" or even as *M-H-GR* "from the foreigner". Note that there is no deterministic way to tell whether the first *m* of the form is part of the pattern, the root or a prefixing particle (the preposition *M* ("from")).

An added complexity arises from the fact that there exist two main standards for the Hebrew script: one in which vocalization diacritics, known as *niqqud*

[3] To facilitate readability we use a transliteration of Hebrew using ASCII characters in this paper.

"dots", decorate the words, and another in which the dots are omitted, but where other characters represent some, but not all of the vowels. Most of the modern printed and electronic texts in Hebrew use the "undotted" script. While a standard convention for this script officially exists, it is not strictly adhered to, even by the major newspapers and in government publications. Thus, the same word can be written in more than one way, sometimes even within the same document. This fact adds significantly to the degree of ambiguity, and requires creative solutions for practical Hebrew language processing applications.

The challenge involved in constructing an MT system for Hebrew is amplified by the poverty of existing resources [11]. The collection of corpora for Hebrew is still in early stages [12] and all existing significant corpora are monolingual. Hence the use of aligned bilingual corpora for MT purposes is currently not a viable option. There is no available large Hebrew language model which could help in disambiguation. No publicly available bilingual dictionaries currently exist, and no grammar is available from which transfer rules can be extracted. Still, we made full use of existing resources which we adapted and augmented to fit our needs.

3.2 Hebrew Input Pre-processing

Our system is currently designed to process Hebrew input represented in UTF-8, but can also handle Microsoft Windows encoding. The morphological analyzer we use (see next sub-section) was designed, however, to produce Hebrew in a romanized (ASCII) representation. We adopted this romanized form for all internal processing within our system, including the encoding of Hebrew in the lexicon and in the transfer rules. The same romanized transliteration is used for Hebrew throughout this paper. The main task of our pre-processing module is therefore to map the encoding of the Hebrew input to its romanized equivalent. This should allow us to easily support other encodings of Hebrew input in the future. The pre-processing also includes simple treatment of punctuation and special characters.

3.3 Morphological Analysis

We use a publicly available morphological analyzer which is distributed through the Knowledge Center for Processing Hebrew. It is based on the morphological grammar of [13], but is re-implemented in Java so that it is faster and more portable [14]. The analyzer produces all the possible analyses of each input word. Analyses include the lexeme and a list of morpho-syntactic features such as number, gender, person, tense, etc. The analyzer also identifies prefix particles which are attached to the word. Our experiments with development data indicate that, at least for newspaper texts, the overall coverage of the analyzer is in fact quite reasonable. The texts we have used so far do not exhibit large amounts of vowel spelling variation, but we have not quantified the magnitude of the problem very precisely.

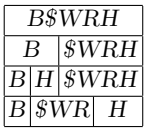

Fig. 3. Lattice Representation of a set of Analyses for the Hebrew Word *B$WRH*

While the set of possible analyses for each input word comes directly from the analyzer, we developed a novel representation for this set to support its efficient processing through our translation system. The main issue addressed is that the analyzer may split an input word into a sequence of several output lexemes, by separating prefix and suffix lexemes. Moreover, different analyses of the same input word may result in a different number of output lexemes. We deal with this issue by converting our set of word analyses into a lattice that represents the various sequences of possible lexemes for the word. Each of the lexemes is associated with a feature structure which encodes the relevant morpho-syntactic features that were returned by the analyzer.

As an example, consider the word form *B$WRH*, which can be analyzed in at least four ways: the noun *B$WRH* ("gospel"); the noun *$WRH* ("line"), prefixed by the preposition *B* ("in"); the same noun, prefixed by the same preposition and a hidden definite article (merged with the preposition); and the noun *$WR* ("bull"), with the preposition *B* as a prefix and an attached pronominal possessive clitic, *H* ("her"), as a suffix. Such a form would yield four different sequences of lexeme tokens which will all be stored in the lattice. To overcome the limited lexicon, and in particular the lack of proper nouns, we also consider each word form in the input as an unknown word and add it to the lattice with no features. This facilitates support of proper nouns through the translation dictionary. Figure 3 graphically depicts the lattice representation of the various analyses, and Figure 4 shows the feature-structure representation of the same analyses.

While two modules for morphological disambiguation of the output of the analyzer are currently being developed [15, 16], their reliability is limited. We prefer to store all the possible analyses of the input in the lattice rather than disambiguate, since our transfer engine can cope with a high degree of ambiguity, and information accumulated in the translation process can assist in ambiguity resolution later on, during the decoding stage. A ranking of the different analyses of each word could, however, be very useful. For example, the Hebrew word form *AT* can be either the (highly frequent) definite accusative marker, the (less frequent) second person feminine personal pronoun or the (extremely rare) noun "spade". We currently give all these readings the same weight, although we intend to rank them in the future.

```
Y0: ((SPANSTART 0)
     (SPANEND 4)
     (LEX B$WRH)
     (POS N)
     (GEN F)
     (NUM S)
     (STATUS ABSOLUTE))

Y1: ((SPANSTART 0)
     (SPANEND 2)
     (LEX B)
     (POS PREP))

Y2: ((SPANSTART 1)
     (SPANEND 3)
     (LEX $WR)
     (POS N)
     (GEN M)
     (NUM S)
     (STATUS ABSOLUTE))

Y3: ((SPANSTART 3)
     (SPANEND 4)
     (LEX $LH)
     (POS POSS))

Y4: ((SPANSTART 0)
     (SPANEND 1)
     (LEX B)
     (POS PREP))

Y5: ((SPANSTART 1)
     (SPANEND 2)
     (LEX H)
     (POS DET))

Y6: ((SPANSTART 2)
     (SPANEND 4)
     (LEX $WRH)
     (POS N)
     (GEN F)
     (NUM S)
     (STATUS ABSOLUTE))

Y7: ((SPANSTART 0)
     (SPANEND 4)
     (LEX B$WRH)
     (POS LEX))
```

Fig. 4. Feature-Structure Representation of a set of Analyses for the Hebrew Word *B$WRH*

3.4 Word Translation Lexicon

The bilingual word translation lexicon was constructed based on the Dahan dictionary [17], whose main benefit is that we were able to obtain it in a machine readable form. This is a relatively low-quality, low-coverage dictionary. To extend its coverage, we use both the Hebrew-English section of the dictionary and the inverse of the English-Hebrew section. The combined lexicon was enhanced with a small manual lexicon of about 100 entries, containing some inflected forms not covered by the morphological analyzer and common multi-word phrases, whose translations are non-compositional.

Significant work was required to ensure spelling variant compatibility between the lexicon and the other resources in our system. The original Dahan dictionary uses the dotted Hebrew spelling representation. We developed scripts for automatically mapping the original forms in the dictionary into romanized forms consistent with the undotted spelling representation. These handle most, but not all of the mismatches. Due to the low quality of the dictionary, a fair number of entries require some manual editing. This primarily involves removing incorrect or awkward translations, and adding common missing translations. Due to the very rapid system development time, most of the editing done so far was based on a small set of development sentences. Undoubtedly, the dictionary is one of the main bottlenecks of our system and a better dictionary will improve the results significantly. The final resulting translation lexicon is automatically converted into the lexical transfer rule format expected by our transfer engine. A small number of lexical rules (currently 20), which require a richer set of unification feature constraints, are appended after this conversion. The translation lexicon contains only *lexeme base forms*. At runtime, morphological analysis (for Hebrew) produces the lexemes for each input word. Morphological generation (for English) is responsible for producing the various surface forms for each

target-side lexeme, and transfer rule constraints create translation segments that are grammatically consistent from these surface forms.

3.5 The Hebrew-English Transfer Grammar

The Hebrew-to-English transfer grammar developed so far was initially developed manually in about two days by a bilingual speaker who is also a member of the system development team, and is thus well familiar with the underlying formalism and its capabilities. It was later revised and extended by a linguist working for about a month. The current grammar is very small and reflects the most common local syntactic differences between Hebrew and English. It contains a total of 36 rules, including 21 noun-phrase (NP) rules, one prepositional-phrase (PP) rule, 6 verb complexes and verb-phrase (VP) rules, and 8 higher-phrase and sentence-level rules for common Hebrew constructions. As we demonstrate in Section 4, this small set of transfer rules is already sufficient for producing reasonably legible translations in many cases. Figure 1 depicts an example of transfer rules for structurally transferring nouns modified by adjectives from Hebrew to English. The rules enforce number and gender agreement between the noun and the adjective. They also account for the different word order exhibited by the two languages, and the special location of the definite article in Hebrew noun phrases.

4 Results and Evaluation

The current system is targeted for translation of newspaper texts. it was developed with minimal amounts of manual labor (beyond the work that went into the existing resources used). In total, we estimate the amount if labor spent directly on the MT system to be about four to six months of human labor. Most of this time was devoted to the construction of the bilingual lexicon and stabilizing the front-end Hebrew processing in the system (Morphology and input representation issues). Once the system was reasonably stable, we devoted about two weeks of time to improving the system based on a small development set of data. For development we used a set of 113 sentences from the Hebrew daily *HaAretz*. Average sentence length was approximately 15 words. Development consisted primarily of fixing incorrect mappings before and after morphological processing and modifications to the bilingual lexicon. The small transfer grammar was also developed during this period. Given the limited resources and the limited development time, we find the results to be highly encouraging. For many of the development input sentences, translations are reasonably comprehensible. Figure 5 contains a few select translation examples from the development data.

To quantitatively evaluate the results achieved so far we tested the system on a set of 62 unseen sentences from *HaAretz*. Two versions of the system were tested on the same data set: a version using our manual transfer grammar and a version with no transfer grammar at all, which amounts to a word-to-word translation version of the system. Results were evaluated using several automatic metrics for

maxwell anurpung comes from ghana for israel four years ago and since worked in cleaning in hotels in eilat

a few weeks ago announced if management club hotel that for him to leave israel according to the government instructions and immigration police

in a letter in broken english which spread among the foreign workers thanks to them hotel for their hard work and announced that will purchase for hm flight tickets for their countries from their money

Fig. 5. Select Translated Sentences from the Development Data

Table 1. System Performance Results with and without the Transfer Grammar

System	**BLEU**		**NIST**		**Precision**	**Recall**
No Grammar	0.0606	[0.0599,0.0612]	3.4176	[3.4080,3.4272]	0.3830	0.4153
Manual Grammar	0.1013	[0.1004,0.1021]	3.7850	[3.7733,3.7966]	0.4085	0.4241

MT evaluation, which compare the translations with human-produced reference translations for the test sentences. For this test set, two reference translations were obtained. We use the BLEU [18] and NIST [19] automatic metrics for MT evaluation. We also include aggregate unigram-precision and unigram-recall as additional reported measures. The results can be seen in Table 1. To assess statistical significance of the differences in performance between the three versions of the system, we apply a commonly used bootstrapping technique [20] to estimate the variability over the test set and establish confidence intervals for each reported performance score. As expected, the manual grammar system outperforms the no-grammar system according to all the metrics.

5 Conclusions and Future Work

The focus of this article has been on the functional aspects of the Stat-XFER framework and on the implementational details of our Stat-XFER Hebrew-to-English prototype MT system. The critical issues of how to generally acquire both translation lexicons and transfer grammars were not addressed in this paper. Our group has been working extensively on developing acquistion methods under a variety of scenarios. The main approach we have been developing targets low-resource languages for which little or no sentence-parallel data is available. Our methodology under such scenarios is based on *elicitation*. We assume 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, a transfer-rule learning module can automatically infer

hierarchical syntactic transfer rules. The collection of transfer rules can then be used in our run-time system to translate previously unseen source language text into the target language. Details about this approach are described in [6] and [21].

Over the past year, we have been extensively developing an acquisition approach for language pairs for which large amounts of sentence-parallel data are available. We are currently applying these new methods for large-scale resource acquistion for our Chinese-to-English Stat-XFER system. This new approach is based on extracting translation resources from parallel sentences that are annotated with their parse structures. We use a relatively small manually word-aligned corpus for the purpose of extracting high-quality transfer rules. We use broad, automatically word-aligned, parallel corpora for extracting broad-coverage translation lexicons. The application of these methods to building a large-scale Chinese-to-English Stat-XFER system is still in progress. Preliminary results are encouraging and indicate that the system is capable of producing translations that are more grammatical and fluent than current phrase-based approaches.

Acknowledgments

This research was supported in part by NSF grants IIS-0121631 (AVENUE) and IIS-0534217 (LETRAS), and by the DARPA TIDES and GALE programs. The Hebrew Stat-XFER MT project is a joint collaboration between the CMU AVENUE group and Dr. Shuly Wintner's CL research group at the University of Haifa. It is supported by the Israel Science Foundation (grant No. 137/06); by the Caesarea Rothschild Institute for Interdisciplinary Application of Computer Science at the University of Haifa; and by NSF's Office of International Science and Education. The author thanks all members of the AVENUE MT research group at CMU for their dedicated research work described in this paper.

References

1. Koehn, P., Och, F.J., Marcu, D.: Statistical Phrase-based Translation. In: Proceedings of HLT-NAACL 2003, Edmonton, Alberta, Canada, Association for Computational Linguistics (2003) 127–133
2. Venugopal, A., Vogel, S., Waibel, A.: Effective Phrase Translation Extraction from Alignment Models. In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL-03), Sapporo, Japan (2003) 319–326
3. Chiang, D.: A Hierarchical Phrase-based Model for Statistical Machine Translation. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL-05), Ann Arbor, Michigan (2005) 263–270
4. Imamura, K., Okuma, H., Watanabe, T., Sumita, E.: Example-based Machine Translation Based on Syntactic Transfer with Statistical Models. In: Proceedings of COLING-2004, Geneva, Switzerland (2004) 99–105

5. Galley, M., Graehl, J., Knight, K., Marcu, D., DeNeefe, S., Wang, W., Thayer, I.: Scalable Inference and Training of Context-Rich Syntactic Translation Models. In: Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Sydney, Australia (2006) 961–968
6. Lavie, A., Vogel, S., Levin, L., Peterson, E., Probst, K., Llitjos, A.F., Reynolds, R., Carbonell, J., Cohen, R.: Experiments with a Hindi-to-English Transfer-based MT System under a Miserly Data Scenario. Transactions on Asian Language Information Processing (TALIP) **2** (2003)
7. Lavie, A., Probst, K., Peterson, E., Vogel, S., Levin, L., Font-Llitjos, A., Carbonell, J.: A Trainable Transfer-based Machine Translation Approach for Languages with Limited Resources. In: Proceedings of Workshop of the European Association for Machine Translation (EAMT-2004), Valletta, Malta (2004)
8. Peterson, E.: Adapting a Transfer Engine for Rapid Machine Translation Development. Master's thesis, Georgetown University (2002)
9. Zhang, Y., Vogel, S.: Suffix Array and its Applications in Empirical Natural Language Processing. Technical Report CMU-LTI-06-010, Language Technologies Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA (2006)
10. Lavie, A., Wintner, S., Eytani, Y., Peterson, E., Probst, K.: Rapid Prototyping of a Transfer-based Hebrew-to-English Machine Translation System. In: Proceedings of the 10th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-2004), Baltimore, MD (2004) 1–10
11. Wintner, S.: Hebrew Computational Linguistics: Past and Future. Artificial Intelligence Review **21** (2004) 113–138
12. Wintner, S., Yona, S.: Resources for Processing Hebrew. In: Proceedings of the MT-Summit IX workshop on Machine Translation for Semitic Languages, New Orleans (2003)
13. Yona, S., Wintner, S.: A Finite-State Morphological Grammar of Hebrew. Natural Language Engineering (2007) To appear.
14. Wintner, S.: Finite-state Technology as a Programming Environment. In Gelbukh, A., ed.: Proceedings of the Conference on Computational Linguistics and Intelligent Text Processing (CICLing-2007). Volume 4394 of Lecture Notes in Computer Science., Berlin and Heidelberg, Springer (2007) 97–106
15. Adler, M., Elhadad, M.: An Unsupervised Morpheme-Based HMM for Hebrew Morphological Disambiguation. In: Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Sydney, Australia, Association for Computational Linguistics (2006) 665–672
16. Shacham, D.: Morphological Disambiguation of Hebrew. Master's thesis, University of Haifa (2007)
17. Dahan, H.: Hebrew–English English–Hebrew Dictionary. Academon, Jerusalem (1997)
18. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a Method for Automatic Evaluation of Machine Translation. In: Proceedings of 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, PA (2002) 311–318
19. Doddington, G.: Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurrence Statistics. In: Proceedings of the Second Conference on Human Language Technology (HLT-2002). (2002)

20. Efron, B., Tibshirani, R.: Bootstrap Methods for Standard Errors, Confidence Intervals and Other Measures of Statistical Accuracy. Statistical Science **1** (1986) 54–77
21. Probst, K., Levin, L., Peterson, E., Lavie, A., Carbonell, J.: MT for Resource-Poor Languages Using Elicitation-Based Learning of Syntactic Transfer Rules. Machine Translation **17** (2002)