Rapid Prototyping of a Transfer-based Hebrew-to-English Machine Translation System

Alon Lavie, Erik Peterson, Katharina Probst
Language Technologies Institute
Carnegie Mellon University
email: alavie@cs.cmu.edu

Shuly Wintner, Yaniv Eytani
Department of Computer Science
University of Haifa
email: shuly@cs.haifa.ac.il

Abstract

We describe the rapid development of a preliminary Hebrew-to-English Machine Translation system under a transfer-based framework specifically designed for rapid MT prototyping for languages with limited linguistic resources. The task is particularly challenging due to two main reasons: the high lexical and morphological ambiguity of Hebrew and the dearth of available resources for the language. Existing, publicly available resources were adapted in novel ways to support the MT task. The methodology behind the system combines two separate modules: a transfer engine which produces a lattice of possible translation segments, and a decoder which searches and selects the most likely translation according to an English language model. We demonstrate that a small manually crafted set of transfer rules suffices to produce legible translations. Performance results are evaluated using state of the art measures and are shown to be encouraging.

1 Introduction

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. In this paper we describe the rapid development of a preliminary Hebrew-to-English Machine Translation system under a transfer-based framework specifically designed for rapid MT prototyping for languages with limited linguistic resources. The system was developed 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.

The methodology behind the system combines two separate modules: a transfer engine which produces a lattice of possible translation segments, and a decoder which searches for and selects the most likely translation according to an English language model. This general framework has been under development by the MT group at Carnegie Mellon under the AVENUE project (Probst et al., 2002), and was previously used for rapid prototyping of an MT system for Hindi-to-English translation (Lavie et al., 2003). For the current Hebrew-to-English system, we manually developed a small set of transfer rules which reflect the most common local syntactic differences between Hebrew and English. This small set of rules turns out to be already sufficient for producing some legible translations of newspaper texts. Performance results are evaluated using state of the art measures and are shown to be encouraging. We also applied an automatic transfer-rule learning approach (Carbonell et al., 2002) to learning a Hebrew-to-English transfer grammar, and include some preliminary performance results when using the acquired grammar.

In the next section we provide some linguistic background about the Hebrew language, with an explicit focus on its challenging sources of ambiguity. Section 3 describes the structure of the MT system with an emphasis on the specific resources required for its application to the Hebrew-to-English language pair and how these resources were acquired and adapted. Section 4 provides some translation examples and describes an evaluation of the system. We conclude with directions for future research.

2 The Hebrew Language

Israeli Hebrew (also known as Modern Hebrew, henceforth *Hebrew*) is one of the two official languages of the State of Israel. It is spoken natively by about half of the population and fluently by virtually all the (over six million) residents of the country. 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,[1] 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* "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. For example, the word *NIQIWN* "cleaning" can occur also as *NQIWN*. 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 (Wintner, 2004). The collection of corpora for Hebrew is still in early stages (Wintner and Yona, 2003) 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. Good morphological analyzers are proprietary and publicly available ones are limited (Wintner, 2004). 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, as we report in the next section.

3 System Design and Architecture

We make use of a new framework (Probst et al., 2002) for rapid prototyping of MT systems for languages with limited amounts of electronically available linguistic resources and corpora, which includes a declara-

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

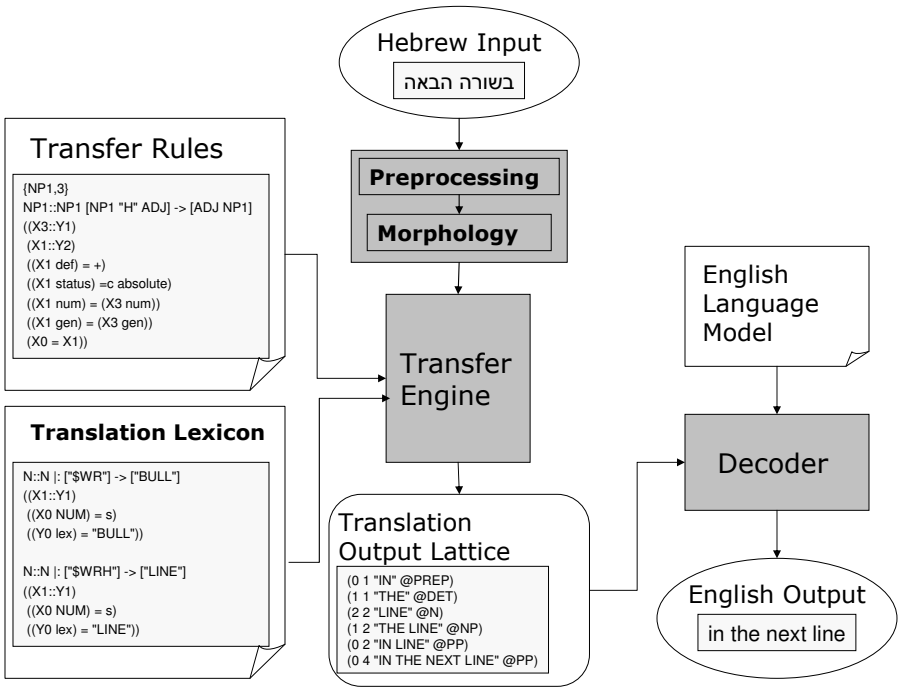

Figure 1: Architecture of the Hebrew-to-English Transfer-based MT System

tive formalism for symbolic transfer grammars. A grammar consists of a collection of transfer rules, which 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 at runtime, and produces all possible word and phrase-level translations according to the grammar. This framework was specifically designed to support advanced research on 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. While the framework itself is still research work in progress, it is sufficiently well developed for serious experimentation.

The system described in this paper is the result of a two-month-long effort to apply this MT prototyping framework to the development of a Hebrew-to-English MT system. 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. This comprehensive collection of output segments is stored in an output lattice data-structure. The lexical transfer rules used by the transfer engine are derived from a *bilingual dictionary*, 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 *language model* of English to search and select a combination of sequential translation segments that together represent the most likely translation of the entire input sentence. A schematic diagram of the system architecture can be seen in Figure 1. We now describe each of the components in more detail.

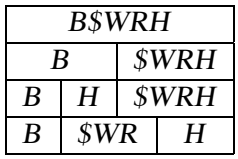

Figure 2: Lattice Representation of a set of Analyses for the Hebrew Word *B$WRH*

3.1 Hebrew Input Pre-processing

Our system is currently designed to process Hebrew input represented in Microsoft Windows Encoding. While this is the most common encoding in use for electronic Hebrew documents (such as online versions of newspapers), other encodings for Hebrew (such as UTF-8) are also quite common. The morphological analyzer we use (see next sub-section) was designed, however, to expect 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.

3.2 Morphological Analysis

We use a publicly available morphological analyzer (Segal, 1999) which 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. The two main drawbacks of this analyzer are that its lexical coverage is limited and that it is not tolerant to variations in the vowel spelling orthography. The analyzer is available to us as a "black-box" component, which does not permit us to easily extend its lexical coverage or to incorporate spelling variants into the analyzer's lexicon. We have not addressed either of these problems so far. 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.

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. Figure 2 graphically depicts the lattice representation of the various analyses, and Figure 3 shows the feature-structure representation of the same analyses.

```
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))
```

Figure 3: Feature-Structure Representation of a set of Analyses for the Hebrew Word *B$WRH*

While the analyzer of Segal (Segal, 1999) comes with a disambiguation option, its 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.

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.

3.3 Word Translation Lexicon

The bilingual word translation lexicon was constructed based on the Dahan dictionary (Dahan, 1997), 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 primarily some inflected forms not covered by the morphological analyzer and a few 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. This representation was already converted to a romanized form for a previous project, but the romanized form was not consistent with the vowel-enhanced spelling variant that we use in our system. It is possible to devise an algorithm for converting dotted script words to the standard undotted representation, but since the standard is not consistently adhered to, we would still be left with the problem of having to match words written differently in different resources. We have so far addressed the issue by developing scripts for automatically mapping the word forms based on common templates. 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.

Since the transfer system currently does not use a morphological generator on the target (English) side, we use a dictionary enhancement process to generate the various morphological forms of English words, with appropriate constraints on when these should be used. The enhancement process works as follows: for each English noun in the lexicon, we create an additional entry with the Hebrew word unchanged, and the English word in the plural form. We also add a *constraint* to the lexical rule indicating that the number is plural. A similar strategy is applied to verbs: we add entries (with constraints) for past tense, future tense, present tense (3rd person singular) and infinitive forms. The result is a set of lexical entries associating Hebrew base forms to English inflected forms. At runtime, the features associated with the analyzed Hebrew word form are unified with the constraints in the lexical rules, so that the appropriate form is selected, while the other forms fail to unify. For example, assume we are processing a plural noun in Hebrew. The lexeme (base form) of the noun, identified by the morphological analyzer, will be used to extract candidate lexical transfer rules from the lexicon. Only the entry that contains a plural form in English, however, will pass unification and succeed.

3.4 The Transfer Engine and Transfer Grammar

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. In the first stage, parsing is performed based solely on the source-language side of the transfer rules. The implemented parsing algorithm is for the most part a standard bottom-up Chart Parser. 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 transfer engine can be found in (Peterson, 2002).

The transfer engine was designed to support both manually-developed structural transfer grammars and grammars that can be automatically acquired from bilingual data. We used the currently available training algorithms for automatically learning a transfer grammar (Lavie et al., 2003) and applied them to a word-aligned Hebrew-translated version of the elicitation corpus. Some preliminary performance results when using this automatically acquired grammar are reported in Section 4.

```
{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)
 )
```

Figure 4: NP Transfer Rules for Nouns Modified by Adjectives from Hebrew to English

As part of our two-month effort so far, we developed a preliminary small manual transfer grammar, which reflects the most common local syntactic differences between Hebrew and English. The current grammar 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. Grammar development took about two days of manual labor by a native bilingual speaker who is also a member of the system development team, and is thus well familiar with the underlying formalism and its capabilities. Figure 4 contains 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.

3.5 Decoding

In the final stage, a 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 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 probability of the target language string given the source language string. The decoder is designed to use a probability model that calculates this probability as a product of two factors: a translation model for the translation units and a language model for the target language. The translation model is normally based on a probabilistic version of the translation lexicon, in which probabilities are associated with the different possible translations for each source-language word. Such a probability model can be acquired automatically using an reasonable-size parallel corpus in order to train a word-to-word probability model based on automatic word alignment algorithms. This was currently not possible for Hebrew-to-English, since we do not have a suitable parallel corpus at our disposal. The decoder for the Hebrew-to-English system therefore uses the English language model as the sole source of information for selecting among alternative translation segments. We use an English trigram language model trained on 160 million words.

The decoding search algorithm considers all possible sequences in the lattice and calculates the language

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

Figure 5: Select Translated Sentences from the Development Data

System	BLEU	NIST	Precision	Recall
No Grammar	0.0606 [0.0599,0.0612]	3.4176 [3.4080,3.4272]	0.3830	0.4153
Learned Grammar	0.0775 [0.0768,0.0782]	3.5397 [3.5296,3.5498]	0.3938	0.4219
Manual Grammar	0.1013 [0.1004,0.1021]	3.7850 [3.7733,3.7966]	0.4085	0.4241

Table 1: System Performance Results with the Various Grammars

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.

4 Results and Evaluation

The current system was developed over the course of a two month period and was targeted for translation of newspaper texts. 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 quality of the results achieved so far, and in order to comparatively assess the performance of our manual transfer grammar and our automatically acquired grammar, we tested the system on an unseen test set of 62 sentences from *HaAretz*. Three versions of the system were tested on the same data set: a version using our manual transfer grammar; a version using our current preliminary automatically-learned 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 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 (Papineni et al., 2002) and NIST (Doddington, 2002) 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 (Efron and Tibshirani, 1986) 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. The learned grammar system scores fall between the scores for the other two systems, indicating that some appropriate structural transfer rules are in fact acquired. All results are statistically highly significant.

5 Conclusions and Future Work

Our current Hebrew-to-English system was developed over the course of a two-month period with a total labor-effort equivalent to about four person-months of development. Unique problems of the Hebrew language, particularly the inherent high-levels of ambiguity in morphology and orthography, were addressed. The bilingual dictionary and the morphological analyzer that were available to us were not very high in quality and had serious limitations. These were adapted in novel ways to support the MT task. The underlying transfer-based framework which we applied proved to be sufficient and appropriate for fast adaptation to Hebrew as a new source language. This provides some encouraging validation to the general suitability of this framework to rapid MT prototyping for languages with limited linguistic resources.

The results of our rapid-development effort exceeded our expectations. Evaluation results on an unseen test-set, and an examination of actual translations of development data, indicate that the current system is already effective enough to produce comprehensible translations in many cases. This was accomplished with only a small manually-developed transfer grammar that covers the structural transfer of common noun-phrases and a small number of other common structures. We believe that it is the combination of this very simple transfer grammar with the selectional disambiguation power provided by the English target language model that together result in surprisingly effective translation capabilities.

We plan to continue the development of the Hebrew-to-English system over the next year. Significant further work will be required to improve the coverage and quality of the word translation lexicon and the morphological analyzer. Several advanced issues that were not addressed in the current system will be investigated. We plan to significantly enhance the sources of information used by the decoder in disambiguating and selecting among translation segments. These include: (1) a language model for the source language, trained from a monolingual Hebrew corpus, which can be used to help disambiguate among the various morphological readings of input words; (2) scores for the transfer rules, based on a scoring model currently under development; and (3) a probabilistic version of the translation lexicon, which we hope to train once we collect some amount of parallel Hebrew-English data. We expect dramatic further improvements in translation quality once these issues are properly addressed.

Acknowledgments

The research reported in this paper was made possible by support from the Caesarea Rothschild Institute at Haifa University and was funded in part by NSF grant number IIS-0121631. We wish to thank the Hebrew Processing Knowledge Center at the Technion for providing the morphological analyzer used in this work.

References

Carbonell, Jaime, Katharina Probst, Erik Peterson, Christian Monson, Alon Lavie, Ralf Brown, and Lori Levin. 2002. Automatic Rule Learning for Resource-Limited MT. In *Proceedings of the 5th Biennial Conference of the Association for Machine Translation in the Americas (AMTA-02)*, October.

Dahan, Hiya. 1997. *Hebrew–English English–Hebrew Dictionary*. Academon, Jerusalem.

Doddington, George. 2002. Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurrence Statistics. In *Proceedings of the Second Conference on Human Language Technology (HLT-2002)*.

Efron, Bradley and Robert Tibshirani. 1986. Bootstrap Methods for Standard Errors, Confidence Intervals and Other Measures of Statistical Accuracy. *Statistical Science*, 1:54–77.

Lavie, Alon, Stephan Vogel, Lori Levin, Erik Peterson, Katharina Probst, Ariadna Font Llitjos, Rachel Reynolds, Jaime Carbonell, and Richard Cohen. 2003. Experiments with a Hindi-to-English Transfer-based MT System under a Miserly Data Scenario. *Transactions on Asian Language Information Processing (TALIP)*, 2(2), June.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318, Philadelphia, PA, July.

Peterson, Erik. 2002. Adapting a Transfer Engine for Rapid Machine Translation Development. Master's thesis, Georgetown University.

Probst, Katharina, Lori Levin, Erik Peterson, Alon Lavie, and Jaime Carbonell. 2002. MT for Resource-Poor Languages Using Elicitation-Based Learning of Syntactic Transfer Rules. *Machine Translation*, 17(4).

Segal, Erel. 1999. Hebrew Morphological Analyzer for Hebrew Undotted Texts. Master's thesis, Technion, Israel Institute of Technology, Haifa, October. In Hebrew.

Wintner, Shuly. 2004. Hebrew Computational Linguistics: Past and Future. *Artificial Intelligence Review*, 21(2):113–138.

Wintner, Shuly and Shlomo Yona. 2003. Resources for Processing Hebrew. In *Proceedings of the MT-Summit IX workshop on Machine Translation for Semitic Languages*, New Orleans, September.