Integration of Morphology into Statistical Machine Translation

by

Eric H. Davis

11-734: Advanced Machine Translation Seminar

Carnegie Mellon University

Language Technologies Institute

Professors Alone Lavie & Stephan Vogel

May 5, 2008

1. Introduction

The state-of-the-art in machine translation (MT) is statistical machine translation (SMT). Such models began as word-based models, but they have evolved to phrase-based models, as translating chunks of the input text at a time yields better results than merely translating word-by-word. Phrase-based SMT systems assume that it is possible to construct a target sentence by segmenting a source sentence into phrases, translating each phrase, and combining the translated phrases to create the target sentence. SMT follows the noisy channel model, where a target language sentence t is distorted by a channel into a foreign language sentence s (Sarikaya, et al 2007). The goal is to translate the input word sequence in the source language to the target language word sequence by maximizing the probability of the target sentence given the source sentence. This is equivalent to maximizing the product of the probability of the source sentence given the target sentence and the probability of the target sentence: P(s|t) * P(t). P(s|t) is known as the translation model, and P(t) is known as the language model of the target language. The translation model represents the correspondence between words in the source and target sequences, and the language model represents the well-formedness of the produced target sequence. The language model and translation model are modeled independently of each other (Popovic, et al 2004).

Such SMT systems use word-based n-gram models, which need a large amount of data to train, so as to estimate the probabilities of the language model and translation model. The large amount of data is necessary, so the models can see a large majority of the forms in a language and accurately estimate probabilities. Seeing all possible forms is not possible, but with enough data, reliable probability estimates are possible. Without a large amount of parallel data available, the main issue is data sparseness. SMT systems exhibit a large drop in performance if the target or source domains are not properly covered in the data (Sarikaya, et al 2007). The sparsity problem becomes even worse if the source or target language (or possibly both languages) is a morphologically-rich language such as Arabic. Morphology is the study of the way words are built from smaller meaning-bearing units. The most important morpheme is the stem or lemma, which is the root of the word. A morpheme can be an affix (prefix or suffix), and the morpheme provides additional meaning to the main concept provided by the stem (Karageorgakis, 2005). Morphologically-rich languages have many different surface forms, even though the stem of a word may be the same. This leads to rapid vocabulary growth, as various prefixes and suffixes can combine with stems in a large number of possible combinations and worse language model probability estimation because of more singletons (forms occurring just once in the data), and a lower number of occurrences over all distinct words. In addition, SMT systems rely on phrase extraction, which is highly dependent on word alignment. Morphologically-rich languages may create many-to-many mappings, as a surface form may map to multiple words; especially if prefixes or suffixes take on the meaning of separate tokens in the other language. Typical SMT systems rely on the IBM word alignment models, but such models produce poor alignment with morphologically-rich languages, as they cannot handle many-to-many mappings (Zollman, et al 2006).

A possible solution to both data sparsity caused by a rich morphological system and word misalignments is to incorporate morphological information into the SMT system. Adding the morphological information reduces data sparsity by attempting to break down words into prefixes, stems, and suffixes (Goldwater, et al 2005). The intuition is that words that appear different in terms of surface forms may have the same stem or root. It is then possible to conflate statistics and claim that these two seemingly different forms are actually the same (underlying) form. In theory, the conflated statistics should help reduce the data sparsity issue and result in better trained language and translation models. The three most common solutions are to preprocess the data so that the input language more closely resembles the output language, adapt the language model to make use of the morphological information, and post-process the output of a MT system to add on the proper inflections. It should be noted that all these schemes make use of preprocessing (in terms of deriving morphological

information), but the main difference is the change in the MT systems. The first solution preprocesses the data and does not change the underlying MT system. The second solution preprocesses the data but also modifies the language model(s). The third solution preprocesses the data, translates stems with the MT system, and then adds on proper inflections in a post-processing step.

All of the above solutions depend highly on the amount of training data. With less data, adding the morphological information helps a lot more because the sparse data issue is that much more of a problem with less data. With more data, the above solutions help less because sparse data is less of a problem the more data that is available.

This paper will explore integration of morphological information in SMT systems, translating from Arabic to English, English to Turkish, English to Czech, English to Greek, and English to Russian. The first section explores various preprocessing techniques. The second sections explores a couple modifications to the language model, and the third sections explores one post-processing technique.

2. Preprocessing

The use of preprocessing highly depends on the direction of translation. Translating from a morphologically rich language to a morphologically poor one is often the easier direction. This is because the inflectional and morphological distinctions in the highly-inflected language are often not present in the poorly-inflected language. As such, merely stemming a word to its root is often good enough to create common (underlying) word forms, conflate statistics for such word forms, and improve the translation and language models enough to produce decent translations. Translating from a morphologically-poor language to a morphologically rich language is not as simple, however. Translating in this direction is especially challenging, as there is a need to decode detailed morphological information from a language that does not encode such information at all or does so only implicitly.

2.1 Arabic to English Statistical Machine Translation

Arabic to English MT is representative of translating from a morphologically-rich language (Arabic) to a morphologically-poor one (English). The goal of preprocessing the data is to make the Arabic training data resemble English. This can be done by making use of some already existing morphological resources. A good example of such an approach is a paper by Habash and Sadat. In this work, preprocessing entails modifying the raw training data and evaluation texts, making them suitable for model training and decoding. Such modification includes different kinds of tokenization, stemming, part-of-speech tagging, and lemmatization. A word is defined as strings in Modern Standard Arabic separated by white space, meaning that prepositional particles and conjunctions were part of a word's morphology and not separate.

The major issues in Arabic are that it is a complex language with a large set of morphological features. These features are present both as concatenative affixes and templatic morphology, and there are morphological and phonological adjustments that appear in a word's orthography and interact with other orthographic variations. Certain letters are not spelled consistently, and this leads to an increase in sparsity and ambiguity (the same form can be multiple different words). Additionally, clitics must be distinguished from inflectional features such as gender, but these clitics are written attached to a word, and this further increases the ambiguity. Two important issues in preprocessing are: the need to resolve ambiguity and determine if a feature or clitic to be split off is actually present in the word. Once this is determined, the need to know the proper form of the resulting word once the clitic or feature is split off is also important. Not normalizing such split-off forms leads to increased sparsity and ambiguity, as the resulting word could be multiple different forms (both nominal and verbal for example).

Habash and Sadat outline three preprocessing techniques, which they distinguish from preprocessing schemes. A scheme is a a specification of the form of the preprocessed output, whereas a technique is how such output is created. The first technique is called REGEX, and this is the baseline technique. REGEX uses a greedy pattern-matching approach to simply split off prefixes or suffixes matching a clitic regular expression. The second technique makes use of BAMA - the Buckwalter Arabic Morphological Analyzer - which is used to obtain multiple word analyses. Disambiguation is done by selecting the first analysis returned by BAMA. The third and most advanced technique uses the MADA (Morphological Analysis and Disambiguation for Arabic) tool. This tool was developed at Columbia by Habash and Rambow in 2005, and it is an off-the-shelf resource for Arabic disambiguation. The tool chooses among the BAMA analyses using a combination of classifiers along 10 orthogonal dimensions, including part-of-speech, gender, number, and pronominal clitics. Using BAMA and MADA to preprocess involves moving features specified by a scheme out of a chose word analysis and generating the word without the split-off features. This guarantees normalization of the word.

The three preprocessing techniques fit into six preprocessing schemes. The first is ST (Simple Tokenization), which splits off punctuation and numbers from words and removes all diacritics appearing in the input. The second, third, and fourth all split off various particles and clitics. D1 splits off the class of conjunction clitics (w+ and f+), D2 splits off the class of particules (l+, k+, b+, and s+), and D3 splits off what D2 does as well as definite articles (Al+) and pronominal clitics. Additionally, MR (morphemes) splits words into stem and affix morphemes, and EN (English-like) aims to minimize the difference between English and Arabic. This scheme is similar to D2,but it uses lexemes and English-like part-of-speech tags instead of regenerated words. It also indicates pro-drop explicitly with a separate token.

Habash and Sadat made use of the Portage phrase-based SMT system in their experiments, which uses a log-linear model to combine the translation model, language model, distortion model, and word-length feature. This system uses IBM word alignment models 1 and 2 and trains in both directions to extract the phrase table. The max phrase size was 8 in the experiments, and the authors used the SRILM toolkit to implement a 3-gram language model. They then optimized the decoding weights with minimum error rate training (MERT) using 200 sentences from the 2003 NIST MT evaluation test set. This optimization lead to the weights for the language model, phrase translation model, distortion model, and word-length feature in the log-linear model. Portage uses the Canoe decoder, which utilizes a dynamic programming beam search algorithm. The training data came from the Linguistic Data Consortium (LDC), and a parallel corpus of about 5 million words was used to train the translation model. The English language model was trained on the English side of the corpus, as well as 116 million words from the English Gigaword Corpus (LDC2005T12) and 128 million words from the UN Parallel corpus (LDC2004E13). English preprocessing involved converting all words to lowercase, separating punctuation from words, and splitting off "'s." Habash and Sadat used two test sets: the 2004 NIST MT evaluation set (MT04), and the 2005 NIST MT evaluation set (MT05). MT04 was a mix of news, editorials, and speeches, whereas MT05 was only news (like the training data). They tested all schemes on different training set sizes of 1%, 10%, and 100% to look at the effect of amount of data on translation.

The EN scheme performed best with less training data (scarce resources), and D2 performed best with more training data (large resources). Also, MADA performed better than BAMA, which performed better than REGEX with scarce resources, but the differences between the three techniques were statistically insignificant with a larger amount of training data. Unsurprisingly, MT05 had better performance than MT04 because MT04 included out of domain items such as editorials and speeches (the training data only consisted of news). To test the effect of choice of preprocessing technique further, the authors tested the sentences in MT04 that were not news using similar experiments as with the entire corpus. Under such conditions, the choice of preprocessing technique becomes even more

important. For example, MADA+D2 with 100% training achieved 12% improvement over the baseline but only 2.4% improvement for news only. Finally, the experiments show that the schemes may be complementary, and the combination of the output of all six schemes may lead to a potential large improvement. For example, selecting the output sentence with the highest sentence-level BLEU score for each input sentence lead to 19% improvement in BLEU score using MT04, MADA, and 100% of the training data. The best score under MADA and using 100% of the training data was 37.1 with MADA+D2, but the best score under the same conditions and combining the six schemes was 44.3.

2.2 English to Turkish Statistical Machine Translation

English to Turkish SMT is representative of translating in the opposite direction. The goal here is to once again to create representations of the source and target language so they more closely resemble each other. The main issues are overcoming the data sparseness problem created by the rich morphology of the target language and creating the rich morphological distinctions in the target language, even though the source language often does not represent such distinctions.

Oflazer and El-Kahlout attempt to tackle such problems in their paper. The authors make use of a lexical-morpheme representation to conflate the statistics for seemingly different suffixes. Typically segmentation of a word is not unique, so the authors generated a representation containing both the lexical segments and morphological features for all possible segmentations. They then used a statistical disambiguator with 94% accuracy to disambiguate. The morphological features of each parse were then removed, leaving only the lexical morphemes.

The authors worked with parallel Turkish-English data and preprocessing involved segmenting words in the Turkish corpus into lexical morphemes in order to abstract out different surface forms because of word-internal phenomena. The goal was to improve the statistics when aligning words. Then Oflazer and El-Kahlout tagged the English side of the corpus with TreeTagger (Schmid 1994) producing a lemma and part-of-speech tag. Tags, such as cat+NN, not implying morphemes or exceptional forms were removed. Finally, the authors extracted a sequence of roots for open class content words for both English and Turkish. For English, this entailed removing all the closed class words and tags signaling a morpheme on an open class word, e.g., VVG. For Turkish, this meant removing all morphemes and roots for closed classes. The plan was to use the removed roots and open class content words to expand the training corpus and bias content word alignment. The authors hoped that the roots would then be able to align without the extra noise from morphemes and function words.

The first set of experiments used the phrase-based SMT framework, Mosses toolkit, and SRILM language modeling toolkit. Results were evaluated using BLEU and one reference translation. The baseline model was trained with an unlimited distortion limit and distortion weight of 0.1 (allowing for long-range distortions), and adding the content word training data hurt the baseline system's performance. The fully-morphologically segmented model made use of a 5-gram morpheme-based language model, which attempted to capture local morphotactic constraints. The decoder produced 1000-best lists, which were converted to words by concatenating morphemes, and rescored using a 4-gram word-based language model (to enforce more-distant word sequence constraints). The optimal linear combination of the word-based language model and translation model was found on the tune set. Rescoring with the 4-gram word-based language model was found to result in large improvements for the best model (the model that included the expanded training data): 22.18 BLEU versus 21.47 without rescoring versus 20.16 for the baseline system.

After the initial experiments, the authors noticed that some morphemes on the Turkish side did not align with anything on the English side. This may have been because derivational morphological analysis was only performed on the Turkish side, meaning that verbal nominalizations on the English side only aligned to verb roots on the Turkish side. Even without derivational morphology, the nominal and agreement markers on the Turkish side were mostly unaligned. For these cases, Oflazer and El-

Kahlout attached the unaligned morphemes to the root on the Turkish side. Retraining the models with the selectively attached morphemes lead to a 2.43 point BLEU improvement over the previous best model.

The experiments in this paper demonstrated that using a language-pair specific representation between full word forms and full morphological segmentation lead to significant improvements. Additionally, adding content word as additional training data and re-scoring the MT system output with a word-based language model lead to even further improvements.

3. Modifying the Language Model

In addition to preprocessing the data, adaptation of the language model is another way to incorporate morphological information into SMT. Two possible adaptations to the language model include factored language models and using the general statistical framework to combine word and stem-based SMT.

3.1 Factored Language Models

Bojar used a factored language model to translate from English-Czech. Czech is a very rich morphological language and has a relatively free word order. This results in very complex morphological tags with an alleged 4000 different tags possible. In factored SMT, the source and target words are represented as tuples of factors, and a log-linear model combines the independent feature functions. Most features are phrase-based and operate synchronously (on the same segmentation) and independently of the neighboring segments. Translation involves decoding, which is made up of a mapping step and generation step. The mapping step maps a subset of source factors to a subset of target factors, and the phrase is restricted to certain factors. The generation step maps a subset of target factors to a disjoint subset of target factors, and this is restricted to word-to-word correspondences. It is possible to include an arbitrary number of target language models over the subsets of target features, including the standard n-gram language model. This language model enforces the sequential coherence of the output. Decoding is then a stack-based beam search, which builds all possible translations with the mapping and generation steps, prunes low-scoring options, and produces output in a left-to-right order and scores the output with the language models.

Bojar experimented with the News Commentary corpus, tagging and lemmatizing the Czech side with a tool by Hajic and Hladka, and tagging the English side with MXPOST and lemmatizing the English side with the Morpha tool (Minnen, et al 2001). The corpus had 55,676 pairs of sentences, and Bojar used 1023 sentences for tuning and 964 sentences for evaluation. Word alignment was done with GIZA++, and to reduce sparseness, the English data was lower-cased, and the Czech data was lemmatized. The language models were based on the target side of the parallel corpus only, and the model parameters were optimized on the tune set with MERT. Bojar created four separate systems to evaluate. The first system was the baseline system, involved a single factor, translated lowercase English word forms to lowercase Czech forms, and used a 3-gram language model to check the output word forms. The second system was called T+C (translate and check), used a single generation step, checked the output for word-level coherence using a 3-gram language model and morphological coherence using a 7-gram language model, and translated an input word form to an output word form and used the output word form to generate its morphological tag. The next system was T+T+C, which built output words based on the output word form and the input morphology to reinforce the correct translation of morphological features such as number. The final system, T+T+G, was the most linguistically motivated, and generated output lemmas and morphological tags from the input in two independent translation steps and combined them in a single generation step to produce the output. The English text was not lemmatized, so Bojar used the English word forms to produce the Czech lemmas. As above, a 3-gram language model was used to check word coherence and a 7-gram for

morphological coherence. The multi-factored models always outperformed the baseline model, but the more complex multi-factored models showed little improvements over T+C (13.9 versus 13.6). This may have been because the more complex models produced more hypotheses with similar scores, thereby making it harder to predict future scores. The more complex models also involved more steps, meaning that there were more model weights to tune, so perhaps there was not enough data to properly tune these weights.

Czech morphological tags are very complicated, so Bojar experimented with simplifying these tags and seeing if this simplification had any effect on the systems. He chose the T+T+C because it performed the best in the previous experiments, and the goal was to balance the richness and robustness of the statistics available in the corpus. The five scenarios ran the gamut from full tags (1200 unique tags seen in 56000 word corpus), which included 15 different morphological properties such as number and case to POS+case (184 unique seen), which only included part-of-speech and sub-part-of-speech for all words and case for nouns, pronominals, adjectives, prepositions. The best scenario in the experiments was CNG03 (1017 unique seen), which matched appropriate tags with the most relevant morphological information for that tag (e.g., case, number, and gender for nouns, pronominals, and adjectives). Using this set of tags lead to an improvement of 0.3 in BLEU score (13.9 to 14.2).

The final set of experiments involved scaling up the amount of data and mixing domains. The new, scaled-up corpus included newswire, e-books, stories, and data from the European Parliament. Bojar noted two findings: The experiments netted better results if individual language models were optimized for different domains separately rather than using a single, mixed-domain language model. More importantly, with a large enough training set, the full tags outperformed the stripped down tags, as data sparsity is much less of an issue.

3.2 Use of general statistical framework to combine word and stem-based SMT

Karageorgakis, Potamianos, and Klasinas demonstrate another way to incorporate morphological information into SMT in their paper. They use an unsupervised morphological learning algorithm, which is suitable because SMT systems train on untagged corpora. The morphological analysis of this corpus provides information about the morphology of the source and target languages, which can then be incorporated into the SMT system.

The authors used the Linguistica system (Goldsmith) on both the source (English) and target (Greek) languages, and this system uses a set of heuristics to provide the initial morphological analysis. The first heuristic takes all the possible splits of a word and uses entropy and probability of a split (frequency of the number of times a stem/suffix appears in the corpus) as a metrics. For each word, the best split is selected using maximum likelihood, and this selection is used to bootstrap the heuristic, and the metric is optimized globally over all words, stems, and suffixes in the corpus. The second heuristic computes the counts of all sequences of characters with a length between 2 and 6 and computes the mutual information metric for each n-gram. The top-100 n-grams are kept and used to parse each word into a stem and suffix, and if more than 1 split is possible, the first heuristic chooses the best one. The Linguistica algorithm creates a signatures, which are stems and a list of corresponding suffixes, stems with the same suffix signature are merged, and signatures with more than 1 stem and affix are regular signatures. This allows further generalization of the morphological rules.

The system incorporates morphological information from both the source and target languages, which is represented as a deterministic mapping from a sequence of words to a sequence of stems. A statistical morphological analyzer (stemmer) extracts stems and computes the probability of a stem given a word: P(S|W). A morphological generator performs the exact opposite process and computes the probability of a word given a stem: P(W|S). The model is similar to a basic word-to-word translation model, but the word-to-word translation is performed by a stem-to-stem system: Ws --> Ss --> St --> Wt, where s represents source, t represents target, W represents word, and S represents stem.

The goal is to find the word maximizing the probability of a target word given a target stem times the probability of a target stem given a source stem: P(Wt|St) * P(St|Ss), which is known as the stem-based SMT system. The authors combined this stem-based SMT system with a traditional lexical-based SMT system, assuming independence between the two systems (each system computes the probabilities separately). The authors placed weights on the two systems, with w0 representing the confidence in the lexical SMT model, and w1 representing the confidence in the morphological SMT model.

Combining the lexical and morphological systems involved building the lexical SMT system and computing P(Wt|Ws), stemming the training corpus using the Linguistica system, using the stemmed corpus to create the morphological SMT system and computing P(St|Ss), decoding each sentence in the evaluation corpus using the lexical SMT system and producing a lattice of possible word-level translation, which was represented as a finite state acceptor Fw, stemming and decoding each sentence in the evaluation corpus using the morphological SMT system and producing a lattice of all possible stem translation, which was represented as a finite state acceptor Fs, building a stem to word model P(Wt|St) by running Linguistica on the target language corpus, which is represented as an unweighted finite state acceptor Tsw and getting the morphological signature, composing Fs and Tsw to get the stem to word mapping and projecting the transducer to the output symbols to get a finite state acceptor Fw', reweighting Fw and Fs by w0 and w1, and intersecting the weighted acceptors Fw and Fw'. The final step is using Viterbi decoding to find the best path T,' which is the translated sentence from the combined morphological and lexical SMT systems. The authors used the AT&T FSM Library to represent the finite state machines.

Experiments involved testing both the lexical-based SMT system and the lexical-morphological SMT system. Both systems were trained on the Europarl corpus (Koehn) with two different training set sizes: 1 million words and 4 million words. The authors used the rest of the Europarl corpus for the development and testing sets. The baseline system was the lexical SMT system, and the authors used GIZA++ to obtain word alignments and trained the phrase-based SMT models and target language models on these alignments. They used the Pharaoh decoder to compute the best translation of a sentence. The lexical-morphological SMT system performed morphological analysis on the English and Greek words and extracted stems but ignored affixes. The Linguistica morphological analyzer automatically derived the morphological rules for both English and Greek using a 5 million word parallel corpus. The initial precision was 85.9% and recall was 90.4%, but the authors wanted to increase precision at the expense of recall, so they only stemmed words if a word's length was greater than 6 and if the ratio of the length of a suffix to the length of the whole word was greater than 0.3. On a set of 2000 distinct words, precision increased from 79.5% to 93.8% using the above heuristics. The corpus was then stemmed using the rules obtained from Linguistica, and the phrase-based models and target language language model were trained on this stemmed corpus. The authors then combined the lexical and morphological SMT systems using finite state machines as described above.

The test data consisted of 26,000 sentences, and with the 1 million word training corpus, the morphological system produced 11,000 different sentences than the lexical system. With the 4 million word training corpus, the morphological system produced 6,000 different sentences than the lexical system. The authors only compared the systems on the different sentences, and the best improvement for the morphological system over the lexical system was with the 1 million training word corpus and was 14.3% for NIST and 14.74% for BLEU. Also, a smaller w1 weight lead to better improvements, meaning that the finite state machine with the morphological information should be weighted more heavily than the finite state machine with the lexical information, and that the statistics of the word stems were better trained than the statistics of the words for training sets of the same size. In addition, use of the morphological information showed greater improvements for systems trained with the smaller training set. Using bootstrap resampling (Efron and Tibshirani, 1994), the authors showed that the improvements using the morphological information were significant at the 95% confidence level.

4. Post-processing

Post-processing involves manipulating the output of the SMT system. It is complementary to the previously mentioned techniques to incorporate morphology into SMT and can be used in conjunction with any of the previously discussed techniques. One particular interesting way of post-processing is translating stems from both the source and target languages and using statistical models to generate the necessary morphology in the post-processing phase.

Minkov, Toutanova, and Suzuki do just that in their paper. Their goals were to resolve the data sparsity issue by allowing generation over morphology and improving this generation of morphologically rich languages using morphological agreement in the target language. The authors aimed to build a model that predicted inflected forms of a sequence of word stems in the target sentence given a source sentence. This model would make use of word-to-word alignment information and lexical resources to provide information about morphological information about words on the source and target sides. It would also use a sentence pair to get syntactic analysis information for both the source and translated sentences and generate inflected forms in the target sentence using all available information and a log-linear model to learn the relevant mapping functions. The approach is general, as it uses only limited morphological resources and training data. This problem is challenging, as the model must handle complex agreement phenomena along multiple morphological dimensions.

The authors assumed that the target and source lexicons were available for the target and source languages. They then defined three operations. Stemming produces a set of possible morphological stems of a word according to the lexicon. Inflections produces a set of surface words having the same stem as the word, and Morphological Analysis produces a set of possible morphological analyses for a word, and a morphological analysis is a vector of category values, each having a dimension and possible values. The task is to take the output of a MT system in the target language (a sequence of words) and convert this sequence of words into a stem set by applying the Stemming operation. The goal is to then given a stem, predict the inflection, where the prediction should reflect the meaning of the source sentence and follow the agreement rules of the target language. The models for inflection prediction use a Maximum Entropy Markov model to breakdown the overall probability of a predicted sequence into a product of local probabilities for individual word predictions. The local prediction is conditioned on the previous k conditions, and model is second order, so the conditional probability distribution is over the labels of the previous two predictions. Features pair predicates on the context and target labels, and they can easily encode the morphological properties of a word and its surface inflected form. Context features describing the morphological properties of the two previous predictions can be derived because the model is second order. The model shares features across multiple target labels, enabling generalization and inflections to apply to many forms. This is a structured-prediction model, which defines structure by the morphological properties of the target predictions and the break down of the word sequence.

There are two types of features. Monolingual features cover only the context and predicted label of the target language. Bilingual features, on the other hand, cover the above information and information about the source sentences. Information about the source sentences is obtained by traversing word-alignment links between target words to a (set of) source words. Both features have three classes. Lexical features talk about surface word forms and stems, and the model is second order, so such features include the features of a standard 3-gram language model. The model is discriminative (it predicts words from a given stem), so the monolingual lexical model uses stems and the predicted words from the left-hand-side of the current position, the current position, and stems from the right-hand-side context. The second class of feature is morphological, and this feature includes part-of-speech, person, number, gender, and various other morphological features (relevant to the source and target languages). The morphological features describe the target label and context, and the aim is to capture morphological generalizations. The third class of feature is syntactic, and this feature type

makes use of syntactic analyses of the source and target sentences using dependency parsing. The aim is to capture morphological agreement phenomena, such as gender in Russian.

Monolingual-lexical features look at stems of the predicted words and the immediately adjacent words, as well as traditional 2-gram and 3-gram features, whereas bilingual-lexical features look at words aligned to the current word and words aligned to its immediate neighbors. Monolingual-morphological features look at morphological attributes of the two previously predicted words and the current prediction, whereas bilingual-morphological features look at features in the source language that are expected to be useful in predicting the morphology of the target language. For example, such a feature would look at the source dependency tree, and if a child of a word aligned to a target word is a particular part-of-speech, the feature value is assigned to its surface word form. Monolingual-syntactic features look at the stem of the parent node.

To evaluate this method with reduced noise, the authors performed reference experiments, which used aligned sentence pairs of reference translations. The corpora came from the technical software manual domain and involved 1 million aligned English-Russian sentence pairs. The authors also made use of 0.5 million pairs of English-Arabic aligned sentences. They used 1000 sentence pairs for both development and testing, aligned words using GIZA++, and used a Treelet-based MT system (Quirk, et al 2005). This type of MT system uses the word dependency structure of the source language and projects the word dependency structure to the target language.

The Russian lexicon was limited to word types seen in training, resulting in 14,000 stems. The Arabic lexicon was created by considering all the different stems returned by BAMA, resulting in 12,670 distinct stems and 89,360 inflected forms. For the word features, the authors looked at the only dominant analysis for any given surface word. If a tie occurred in Russian, they looked at the first arbitrary analysis, and they used the most frequent analysis estimated from the Arabic Treebank for Arabic. The baseline system was a 3-gram language model trained using the CMU language modeling toolkit with default settings.

The authors tested four models: Monolingual-Word used language model-like features and stem n-grams, Bilingual-Word used the above features and bilingual lexical features, Monolingual-All had access to all the information available in the target language including morphological and syntactic features, and Bilingual-All used all possible feature types. They performed feature selection by using a greedy forward stepwise feature selection algorithm to maximize the development set accuracy. Features were represented as templates, which generated a set of binary features corresponding to different values of the template, e.g., POS=NOUN. They also considered combinations of up to 3 features: 1 predicate on the prediction and up to 2 on the context. After feature selection, the authors manually inspected the selected templates and wound up using 11 for Monolingual-All Russian, 36 for Bilingual-All Russian, 27 for Monolingual-All Arabic, and 39 for Bilingual-All Arabic.

All the suggested models handily outperformed the baseline model, and the best was Bilingual-All, which scored 91.5% accuracy for Russian versus 77.6% accuracy for the baseline, and 73.3% accuracy for Arabic versus 31.7% accuracy for the baseline. The bilingual and non-lexical features made large contributions, resulting in about 1.5-2% absolute gain both monolingual and bilingual non-lexical features and about 2% gain for bilingual features. The models also did a good job of resolving the data sparsity problem in terms of generating morphology, as when they were trained on as few as 5000 sentence pairs, they had much greater accuracy than a language model trained on a much larger dataset. The learning curve became less steep with more training data, possible showing that the models were learning morphological generalizations. One final experiment tested this framework on the results of an English-Russian MT system. The authors trained an English-Russian MT system on a stemmed version of aligned data, and used the system to generate stemmed word sequences as outputs. They then inflected these stem sequences with the suggested framework and noticed BLEU score improvements of 1.7 compared to a typical phrase-based system.

5. Conclusion

The above papers demonstrate that there are a wide number of ways of incorporating morphology into SMT, and the best technique depends largely on the amount of training data available and the direction of translation (rich-morphology to poor or vice versa). Regardless, all of the above solutions helped to resolve the data sparsity issue and produce better word alignments. Splitting target or source or both words into morphemes helped reduce the data sparsity problem by conflating the statistics of two or more base forms (that did not look the same on the surface but have the same form after stripping off morphology), and this had a positive effect on the word alignment and translation processes. The better word alignments resulted in better overall translations, as more accurate phrase tables were able to be extracted. The use of morphological information worked better with smaller datasets because the sparsity problem is more of an issue with smaller datasets, as less forms are present, and it is harder to reliably estimate probabilities. On the other hand, with a large dataset, the effect of morphological information is less pronounced, as the issue of data sparsity is less of an issue.

The first issue to note is that just because sparse data is an issue, that does not mean that sparsity is an issue for each and every word form. It is quite likely that some forms in the dataset occur frequently enough to produce reliable probability estimates in training. However, every paper that I read did not really take this into account and used morphological information in the translation process for each word. Assuming a word occurs frequently enough to produce reliable probability estimates, splitting the word using morphological information may not be justified. Such a splitting may be error prone, as is reassembling the word after translation. I believe a kind of back-off model may be more appropriate here. The MT system can make use of a factored language model, and one language model can be morphologically based, and the other can be word-based. Before translation, the system would then iterate over the training set and produce a count of each type present in the data. During translation, if a word had a frequency greater than some arbitrary, pre-defined threshold, the MT system would make use of the word-based language model. If not, the system would use the morphologically-based language model. Such a system would still maintain the advantages of including morphological information such as reduction of data sparseness, but it would also include the advantages of word-based translation: no need for possibly error-prone word segmentation and reassembly.

Moreover, the above papers noted that a segmentation in between full-segmentation and full word forms was ideal. Leaving word forms un-segmented does not aid in resolving the data sparsity issue. Fully-segmenting words results in affixes that do not align with anything on the target side, creating noise, worsening the alignment, and thereby worsening the translation results. Some affixes are helpful in the translation process, whereas other are not and should remain attached to the root. Which affixes to re-attach is highly dependent on the language pairs being translated. The question is how to determine how much to split based on which affixes are relevant and which ones are not. The Arabic, Czech, and Turkish translation systems from above all used linguistic knowledge or looked at which affixes did not align with anything to determine which affixes to re-attach to stems. Linguistic knowledge requires an expert on the language, and such a person may or may not be available for a translation project. Inspecting alignments works well, but this method requires manual re-attachment of affixes aligning with nothing to their stems. Ideally, the entire MT process is fully automatic and even more ideally, unsupervised. One possible solution may be present in a paper by Talbot and Osbourne. They attempt to formulate a framework that minimizes the amount of lexical redundancy in a pair of languages. Redundancy means that distinctions made between lexical types in one language are not made in another language. For example, such redundancy may come about if one language encodes gender on adjectives, while another does not. The goal is to optimize the source lexicon for a given target language by selecting a model over a set of cluster-based translation models. Such clusters group words by similarity, e.g, part-of-speech or distribution in training data. The authors define a prior over these models using a Markov random field. The framework then learns features about sets of

monolingual types that predict lexical redundancy (Talbot, et al 2006). The intuition is that the less redundancy present in two language pairs, the less re-attachment necessary, as the two languages have highly similar morphological structures. On a similar token, the more redundancy present in a language pair, the more re-attachment necessary, as the two languages have highly dissimilar morphological structures. Use of this algorithm lessens the dependence on a linguist to provide knowledge about linguistic information about the source and target languages. However, re-attachment of affixes that do not align still must be done manually. The Karageorgakis, et al paper provides another possible solution, in that they used the Linguistica system, which automatically derived morphological rules using the Minimum Description Length principal and other heuristics to determine an optimal splitting for a word. They got good results using this system, but it would have been interesting to see comparisons of a system using a morphological analyzer designed for English and one designed for Greek versus the results from the Linguistica morphological analyzer. The Linguistica morphological analyzer was close to fully automatic, but the authors still had to tune a couple of ad-hoc parameters (minimum length of word to stem and minimum ratio of suffix length to word length on a tuning set.

Integration of morphology into SMT helps achieve better local agreement, but long-distance agreement is still an issue. Oflazer, et al made use of a decoder and distortion model set to unlimited to try to allow for long-distance movement. However, this lead to mixed results. For example, their best model achieved 65% root accuracy but only 52% word 1-gram accuracy. Another examples is in English-Czech MT, the verb-modifier relation is hard to translate correctly. Bojar found that 56% of the time the verb and noun were both lexically correct, but the relation between them was not. Both of the above issues may result from a sparsity problem, as there may not be enough evidence in the data to reliably produce the correct forms and the correct agreement. Instead of relying on the distortion model to produce long-distance movement, a better solution may be incorporating syntax into SMT. A syntax-based language model could be trained on dependency parses between verbs and modifiers or stems and morphemes, and this language model could then be an additional language model in a factored SMT system. Using a factored system also enables the use of other features and clues that may help identify the correct agreement, e.g., context to the left or right of the relation.

The final point is that using BLEU to evaluate is always an issue. With morphologically-rich languages, using BLEU is even more of a problem. This is because BLEU is an 'all or none' evaluation, and morphologically-rich languages have fewer tokens in the output. A system could get 100% of the root words correct and 90% of the affixes but still receive a low BLEU score because one affix is wrong in each of the words in the output. Using a more lenient and linguistically-based scoring metric makes a lot more sense for morphologically rich languages. For example, using Meteor (Lavie, et al) could result in improved scores. A possibility is using WordNet to provide synonyms for each root and affix. A system would then receive a score based on how closely each root and affix matched the reference translation. An exact match of a stem or affix results in a perfect score, whereas an affix or stem that matched a synonym in WordNet would result in a score that was slightly discounted from the perfect score depending on how distant the synonym is from the reference translation. The resulting score would still be good and accurately reflect how well the translation system produced stems and affixes, and it would resolve the 'all or none' scoring of BLEU. Hence, integrating morphological information into SMT systems can be done in a variety of ways, reduces the data sparsity issue, aids alignment, and ultimately results in improved translation scores compared to word-based SMT systems.

Works Cited

Bojar, Ondrej. English-to-Czech Factored Machine Translation. ACL, 2007.

Creutz, Mathias, and Krista Lagus. Morfessor in the Morpho Challenge. MC, 2006.

Goldwater, Sharon and David McClosky. Improving Statistical MT through Morphological Analysis. HLT, 2005.

Habash, Nizar and Fatiha Sadat. Arabic Preprocessing Schemes for Statistical Machine Translation. ACL, 2006.

Karageorgakis, Panagiotis, Potamianos, Alexandros, and Ioannis Klasinas. Towards Incorporating Language Morphology into Statistical Machine Translation Systems. Automatic Speech Recognition and Understanding Workshop, 2005.

Lee, Young-Suk. Morphological Analysis for Statistical Machine Translation. HLT-NAACL, 2004.

Minkov, Einat, Toutanova, Kristina, and Hisami Suzuki. Generating Complex Morphology for Machine Translation. ACL, 2007.

Oflazer, Kemal and Ilknur Durgar El-Kahlout. Exploring Different Representational Units in English-to-Turkish Statistical Machine Translation. ACL, 2007.

Popovic, Maja and Hermann Ney. Towards the Use of Word Stems and Suffixes for Statistical Machine Translation. LREC, 2004.

Sarikaya, Ruhi and Yonggang Deng. Joint Morphological-Lexical Language Modeling for Machine Translation. HLT-NAACL, 2007.

Talbot, David and Miles Osbourne. Modelling Lexical Redundancy for Machine Translation. ACL, 2006.

Zollman, Andreas, Venugopal, Ashish, and Stephan Vogel. Bridging the Inflection Morphology Gap for Arabic Statistical Machine Translation. ACL, 2006.