Decision Trees for Lexical Smoothing in Statistical Machine Translation

Rabih Zbib[†] and **Spyros Matsoukas** and **Richard Schwartz** and **John Makhoul**
BBN Technologies, 10 Moulton Street, Cambridge, MA 02138, USA
† Massachusetts Institute of Technology, 77 Massachusetts Ave., Cambridge, MA 02139, USA

Abstract

We present a method for incorporating arbitrary context-informed word attributes into statistical machine translation by clustering attribute-qualified source words, and smoothing their word translation probabilities using binary decision trees. We describe two ways in which the decision trees are used in machine translation: by using the attribute-qualified source word clusters directly, or by using attribute-dependent lexical translation probabilities that are obtained from the trees, as a lexical smoothing feature in the decoder model. We present experiments using Arabic-to-English newswire data, and using Arabic diacritics and part-of-speech as source word attributes, and show that the proposed method improves on a state-of-the-art translation system.

1 Introduction

Modern statistical machine translation (SMT) models, such as phrase-based SMT or hierarchical SMT, implicitly incorporate source language context. It has been shown, however, that such systems can still benefit from the explicit addition of lexical, syntactic or other kinds of context-informed word features (Vickrey et al., 2005; Gimpel and Smith, 2008; Brunning et al., 2009; Devlin, 2009). But the benefit obtained from the addition of attribute information is in general countered by the increase in the model complexity, which in turn results in a sparser translation model when estimated from the same corpus of data. The increase in model sparsity usually results in a deterioration of translation quality.

In this paper, we present a method for using arbitrary types of source-side context-informed word attributes, using binary decision trees to deal with the sparsity side-effect. The decision trees cluster attribute-dependent source words by reducing the entropy of the lexical translation probabilities. We also present another method where, instead of clustering the attribute-dependent source words, the decision trees are used to interpolate attribute-dependent lexical translation probability models, and use those probabilities to compute a feature in the decoder log-linear model.

The experiments we present in this paper were conducted on the translation of Arabic-to-English newswire data using a hierarchical system based on (Shen et al., 2008), and using Arabic diacritics (see section 2.3) and part-of-speech (POS) as source word attributes. Previous work that attempts to use Arabic diacritics in machine translation runs against the sparsity problem, and appears to lose most of the useful information contained in the diacritics when using partial diacritization (Diab et al., 2007). Using the methods proposed in this paper, we manage to obtain consistent improvements from diacritics against a strong baseline. The methods we propose, though, are not restrictive to Arabic-to-English translation. The same techniques can also be used with other language pairs and arbitrary word attribute types. The attributes we use in the described experiments are local; but long distance features can also be used.

In the next section, we review relevant previous work in three areas: Lexical smoothing and lexical disambiguation techniques in machine translation; using decision trees in natural language processing, and especially machine translation; and Arabic diacritics. We present a brief exposition of Arabic orthography, and refer to previous work on automatic diacritization of Arabic text. Section 3 describes the procedure for constructing the decision trees, and the two methods for using them in machine translation. In section 4 we describe the experimental setup and present experimental results. Finally, section 5 concludes the paper and discusses future directions.

2 Previous Work

2.1 Lexical Disambiguation and Lexical Smoothing

Various ways have been proposed to improve the lexical translation choices of SMT systems. These approaches typically incorporate local context information, either directly or indirectly.

The use of Word Sense Disambiguation (WSD) has been proposed to enhance machine translation by disambiguating the source words (Cabezas and Resnick, 2005; Carpuat and Wu, 2007; Chan et al., 2007). WSD usually requires that the training data be labeled with senses, which might not be available for many languages. Also, WSD is traditionally formulated as a classification problem, and therefore does not naturally lend itself to be integrated into the generative framework of machine translation. Carpuat and Wu (2007) formulate the SMT lexical disambiguation problem as a WSD task. Instead of learning from word sense corpora, they use the SMT training data, and use local context features to enhance the lexical disambiguation of phrase-based SMT.

Sarikaya et al. (2007) incorporate context more directly by using POS tags on the target side to model word context. They augmented the target words with POS tags of the word itself and its surrounding words, and used the augmented words in decoding and for language model rescoring. They reported gains on Iraqi-Arabic-to-English translation.

Finally, using word-to-word context-free lexical translation probabilities has been shown to improve the performance of machine translation systems, even those using much more sophisticated models. This feature, usually called lexical smoothing, has been used in phrase-based systems (Koehn et al., 2003). Och et al. (2004) also found that including IBM Model 1 (Brown et al., 1993) word probabilities in their log-linear model works better than most other higher-level syntactic features at improving the baseline. The incorporation of context on the source or target side enhances the gain obtained from lexical smoothing. Gimpel and Smith (2008) proposed using source-side lexical features in phrase-based SMT by conditioning the phrase probabilities on those features. They used word context, syntactic features or positional features. The features were added as components into the log-linear decoder model, each with a tunable weight. Devlin (2009) used context lexical features in a hierarchical SMT system, interpolating lexical counts based on multiple contexts. It also used target-side lexical features.

The work in the paper incorporates context information based on the reduction of the translation probability entropy.

2.2 Decision Trees

Decision trees have been used extensively in various areas of machine learning, typically as a way to cluster patterns in order to improve classification (Duda et al., 2000). They have, for instance, been long used successfully in speech recognition to cluster context-dependent phoneme model states (Young et al., 1994).

Decision trees have also been used in machine translation, although to a lesser extent. In this respect, our work is most similar to (Brunning et al., 2009), where the authors extended word alignment models for IBM Model 1 and Hidden Markov Model (HMM) alignments. They used decision trees to cluster the context-dependent source words. Contexts belonging to the same cluster were grouped together during Expectation Maximization (EM) training, thus providing a more robust probability estimate. While Brunning et al. (2009) used the source context clusters for word alignments, we use the attribute-dependent source words directly in decoding. The approach we propose can be readily used with any alignment model.

Stroppa et al. (2007) presented a generalization of phrase-based SMT (Koehn et al., 2003) that also takes into account source-side context information. They conditioned the target phrase probability on the source phrase as well as source phrase context, such as bordering words, or part-of-speech of bordering words. They built a decision tree for each source phrase extracted from the training data. The branching of the tree nodes was based on the different context features, branching on the most class-discriminative features first. Each node is associated with the set of aligned target phrases and corresponding context-conditioned probabilities. The decision tree thus smoothes the phrase probabilities based on the different features, allowing the model to back off to less context, or no context at all depending on the presence of that context-dependent source phrase in the training data. The model, however, did not provide for a back-off mechanism if the phrase pair was not found in the extracted phrase table. The method presented in this paper differs in various aspects. We use context-dependent information at the source word level, rather than the phrase level, thus making it readily applicable to any translation model and not just phrase-based translation. By incorporating context at the word level, we can decode directly with attribute-augmented source data (see section 3.2).

2.3 Arabic Diacritics

Since an important part of the experiments described in this paper use diacritized Arabic source, we present a brief description of Arabic orthography, and specifically diacritics.

The Arabic script, like that of most other Semitic languages, only represents consonants and long vowels using letters [1]. Short vowels can be written as small marks written above or below the preceding consonant, called diacritics. The diacritics are, however, omitted from written text, except in special cases, thus creating an additional level of lexical ambiguity. Readers can usually guess the correct pronunciation of words in non-diacritized text from the sentence and discourse context. Grammatical case on nouns and adjectives are also marked using diacritics at the end of words. Arabic MT systems use undiacritized text, since most available Arabic data is undiacritized.

Automatic diacritization of Arabic has been done with high accuracy, using various generative and discriminative modeling techniques. For example, Ananthakrishnan et al. (2005) used a generative model that incorporates word level n-grams, sub-word level n-grams and part-of-speech information to perform diacritization. Nelken and Shieber (2005) modeled the generative process of dropping diacritics using weighted transducers, then used Viterbi decoding to find the most likely generator. Zitouni et al. (2006) presented a method based on maximum entropy classifiers, using features like character n-grams, word n-grams, POS and morphological segmentation. Habash and Rambow (2007) determined various morpho-syntactic features of the word using SVM classifiers, then chose the corresponding diacritization. The experiments in this paper use the automatic diacritizer by Sakhr Software. The diacritizer determines word diacritics through rule-based morphological and syntactic analysis. It outputs a diacritization for both the internal stem and case ending markers of the word, with an accuracy of 97% for stem diacritization and 91% for full diacritization (i.e., including case endings).

There has been work done on using diacritics in Automatic Speech Recognition, e.g. (Vergyri and Kirchhoff, 2004). However, the only previous work on using diacritization for MT is (Diab et al., 2007), which used the diacritization system described in (Habash and Rambow, 2007). It investigated the effect of using full diacritization as well as partial diacritization on MT results. The authors found that using full diacritics deteriorates MT performance. They used partial diacritization schemes, such as diacritizing only passive verbs, keeping the case endings diacritics, or only gemination diacritics. They also saw no gain in most configurations. The authors argued that the deterioration in performance is caused by the increase in the size of the vocabulary, which in turn makes the translation model sparser; as well as by errors during the automatic diacritization process.

[1] Such writing systems are sometimes referred to as *Abjads* (See Daniels, Peter T., et al. eds. The World's Writing Systems Oxford. (1996), p.4.)

3 Decision Trees for Source Word Attributes

3.1 Growing the Decision Tree

In this section, we describe the procedure for growing the decision trees using context-informed source word attributes.

The attribute-qualified source-side of the parallel training data is first aligned to the target-side data. If S is the set of attribute-dependent forms of source word s, and t_j is a target word aligned to $s_i \in S$, then we define:

$$p\left(t_j|s_i\right) \;=\; \frac{\text{count}(s_i,t_j)}{\text{count}(s_i)} \tag{1}$$

where $\text{count}(s_i, t_j)$ is the count of alignment links between s_i and t_j.

A separate binary decision tree is grown for each source word. We start by including all the attribute-dependent forms of the source word at the root of the tree. We split the set of attributes at each node into two child nodes, by choosing the splitting that maximizes the reduction in weighted entropy of the probability distribution in (1). In other words, at node $\mathbf{n}$, we choose the partition $(S_1^\star, S_2^\star)$ such that:

$$\begin{aligned}(S_1^\star, S_2^\star) &= \\ &\underset{\substack{(S_1,S_2)\\ S_1\cup S_2=S}}{\text{argmax}}\ \{h(S) - (h(S_1) + h(S_2))\}\end{aligned} \tag{2}$$

where $h(S)$ is the entropy of the probability distribution $p(t_j|s_i \in S)$, weighted by the number of samples in the training data of the source words in S. We only split a node if the entropy is reduced by more than a threshold θ_h. This step is repeated recursively until the tree cannot be grown anymore.

Weighting the entropy by the source word counts gives more weight to the context-dependent source words with a higher number of samples in the training data, sine the lexical translation probability estimates for frequent words can be trusted better. The rationale behind the splitting criterion used is that the split that reduces the entropy of the lexical translation probability distribution the most is also the split that best separates the list of forms of the source word in terms of the target word translation. For a source word that has multiple meanings, depending on its context, the decision tree will tend to implicitly separate those meanings using the information in the lexical translation probabilities.

Although we describe this method as growing one decision tree for each word, and using one attribute type at a time, a decision tree can clearly be constructed for multiple words, and more than one attribute type can be used in the same decision tree.

3.2 Trees for Source Word Clustering

The source words could be augmented to explicitly incorporate the word attributes (diacritics or other attribute types). The augmented source will be less ambiguous if the attributes do in fact contain disambiguating information. This, in principle, helps machine translation performance. The flip side is that the resulting increase in vocabulary size increases the translation model sparsity, usually with a detrimental effect on translation.

To mitigate the effect of the increase in vocabulary, decision trees can be use to cluster the attribute-augmented source words. More specifically, a decision tree is grown for each source word as described in the previous section, using a predefined entropy threshold θ_h. When the tree cannot be expanded anymore, its leaf nodes will contain a multi-set partitioning of the list of attribute-dependent forms of that source word. Each of the clusters is treated as an equivalence class, and all forms in that class are mapped to a unique form (e.g. an arbitrarily chosen member of the cluster). The mappings are used to map the tokens in the parallel training data before alignment is run on the mapped data. The test data is also mapped consistently. This clustering procedure will only keep the attribute-dependent forms of the source words that decrease the uncertainty in the translation probabilities, and are thus useful for translation.

The experiments we report on use diacritics as an attribute type. The various diacritized forms of a source word are thus used to train the decision trees. The resulting clusters are used to map the data into a subset of the vocabulary that is used in translation training and decoding (see section 4.2 for results). Diacritics are obviously specific to Arabic. But this method can be used with other attribute types, by first appending the source words with

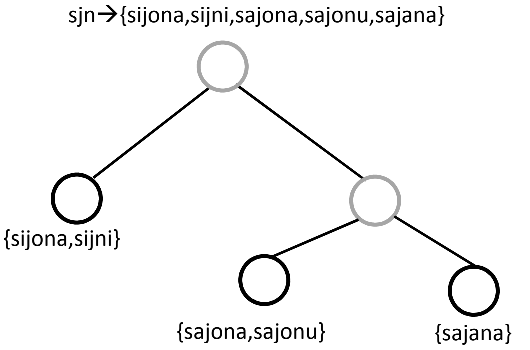

Figure 1: Decision tree for source word *sjn* using diacritics as an attribute.

their context (e.g. attach to each source word its part-of-speech tag or context), and then training decision trees and mapping the source side of the data.

Figure 1 shows an example of a decision tree for the Arabic word *sjn*[2] using diacritics as a source attribute. The root contains the various diacritized forms (*sijona 'prison ACCUSATIVE', sijoni 'prison DATIVE', sajona 'imprisonment ACCUSATIVE.', sajoni 'imprisonment ACCUSATIVE.', sajana 'he imprisoned'*). The leaf nodes contain the attribute-dependent clusters.

3.3 Trees for Lexical Smoothing

As mentioned in section 2.1, lexical smoothing, computed from word-to-word translation probabilities, is a useful feature, even in SMT systems that use sophisticated translation models. This is likely due to the robustness of context-free word-to-word translation probability estimates compared to the probabilities of more complicated models. In those models, the rules and probabilities are estimated from much larger sample spaces.

In our system, the lexical smoothing feature is computed as follows:

$$f(\mathbf{U})=\prod_{t_j\in T(\mathbf{U})}\left(1-\prod_{s_i\in\{S(\mathbf{U})\cup\mathrm{NULL}\}}(1-\bar{p}(t_j|s_i))\right) \quad (3)$$

where $\mathbf{U}$ is the modeling unit specific to the translation model used. For a phrase-based system, $\mathbf{U}$ is the phrase pair, and for a hierarchical system $\mathbf{U}$ is the translation rule. $S(\mathbf{U})$ is the set of terminals on the source side of $\mathbf{U}$, and $T(\mathbf{U})$ is the set of terminals on its target. The NULL term in the equation above accounts for unaligned target words, which we found in our experiments to be beneficial. One way of interpreting equation (3) is that $f(\mathbf{U})$ is the probability that for each target word t_j in $\mathbf{U}$, t_j is a likely translation of at least one word s_i on the source side. The feature value is then used as a component in the log-linear model, with a tunable weight.

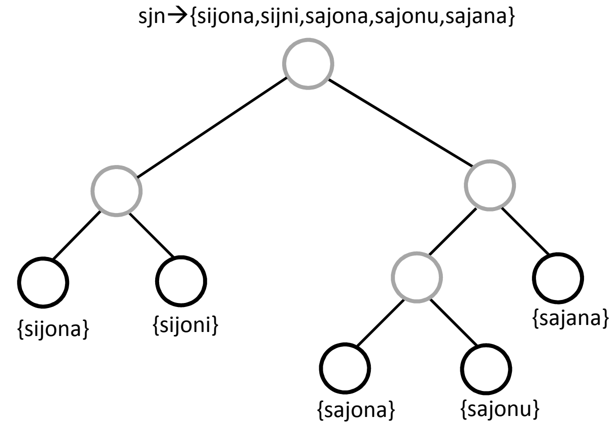

Figure 2: Decision tree for source word *sjn* grown fully using diacritics.

In this work, we generalize the lexical smoothing feature to incorporate the source word attributes. A tree is grown for each source word as described in section 3.1, but using an entropy threshold $\theta_h = 0$. In other words, the tree is grown all the way until each leaf node contains one attribute-dependent form of the source word. Each node in the tree contains a cluster of attribute-dependent forms of the source word, and a corresponding attribute-dependent lexical translation probability distribution. The lexical translation probability models at the root nodes are those of the regular attribute-independent lexical translation probabilities. The models at the leaf nodes are the most fine-grained, since they are conditioned on only one attribute value. Figure 2 shows a fully grown decision tree for the same source word as the example in Figure 1.

The lexical probability distribution at the leafs are from sparser data than the original distributions, and are therefore less robust. To address this, the attribute-dependent lexical

[2] Examples are written using Buckwalter transliteration.

smoothing feature is estimated by recursively interpolating the lexical translation probabilities up the tree. The probability distribution $p_{\mathbf{n}}$ at each node $\mathbf{n}$ is interpolated with the probability of its parent node as follows:

$$\overline{p}_{\mathbf{n}} = \begin{cases} p_{\mathbf{n}} & \text{if } \mathbf{n} \text{ is root,} \\ w_{\mathbf{n}} p_{\mathbf{n}} + (1 - w_{\mathbf{n}}) \overline{p}_{\mathbf{m}} & \text{otherwise} \end{cases}$$

$$\text{where } \mathbf{m} \text{ is the parent of } \mathbf{n} \tag{4}$$

A fraction of the parent probability mass is thus given to the probability of the child node. If the probability estimate of an attribute-dependent form of a source word with a certain target word t is not reliable, or if the probability estimate is 0 (because the source word in this context is not aligned with t), then the model gracefully backs off by using the probability estimates from other attribute-dependent lexical translation probability models of the source word.

The interpolation weight is a logistic regression function of the source word count at a node $\mathbf{n}$:

$$w_{\mathbf{n}} = \frac{1}{1 + e^{-\alpha - \beta \log(\text{count}(S_{\mathbf{n}}))}} \tag{5}$$

The weight varies depending on the count of the attribute-qualified source word in each node, thus reflecting the confidence in the estimates of each node's distribution. The two global parameters of the function, a bias α and a scale β are tuned to maximize the likelihood of a set of alignment counts from a heldout data set of 179K sentences. The tuning is done using Powell's method (Brent, 1973).

During decoding, we use the probability distribution at the leaves to compute the feature value $f(\mathbf{R})$ for each hierarchical rule $\mathbf{R}$. We train and decode using the regular, attribute-independent source. The source word attributes are used in the decoder only to index the interpolated probability distribution needed to compute $f(\mathbf{R})$.

4 Experiments

4.1 Experimental Setup

As mentioned before, the experiments we report on use a string-to-dependency-tree hierarchical translation system based on the model described in (Shen et al., 2008). Forward and backward context-free lexical smoothing are used as decoder features in all the experiments. Other features such as rule probabilities and dependency tree language model (Shen et al., 2008) are also used. We use GIZA++ (Och and Ney, 2003) for word alignments. The decoder model parameters are tuned using Minimum Error Rate training (Och, 2003) to maximize the IBM BLEU score (Papineni et al., 2002).

	Likelihood	**%**
baseline	-1.29	-
Diacs. dec. trees	-1.25	+2.98%
POS dec. trees	-1.24	+3.41%

Table 1: Normalized likelihood of the test set alignments without decision trees, then with decision trees using diacritics and part-of-speech respectively.

For training the alignments, we use 27M words from the Sakhr Arabic-English Parallel Corpus (SSUSAC27). The language model uses 7B words from the English Gigaword and from data collected from the web. A 3-gram language model is used during decoding. The decoder produces an N-best list that is reranked using a 5-gram language model.

We tune and test on two separate data sets consisting of documents from the following collections: the newswire portion of NIST MT04, MT05, MT06, and MT08 evaluation sets, the GALE Phase 1 (P1) and Phase 2 (P2) evaluation sets, and the GALE P2 and P3 development sets. The tuning set contains 1994 sentences and the test set contains 3149 sentences. The average length of sentences is 36 words. Most of the documents in the two data sets have 4 reference translations, but some have only one. The average number of reference translations per sentence is 3.94 for the tuning set and 3.67 for the test set.

In the next section, we report on measurements of the likelihood of test data, and describe the translation experiments in detail.

4.2 Results

In order to assess whether the decision trees are in fact helpful in decreasing the uncertainty in the lexical translation probabilities

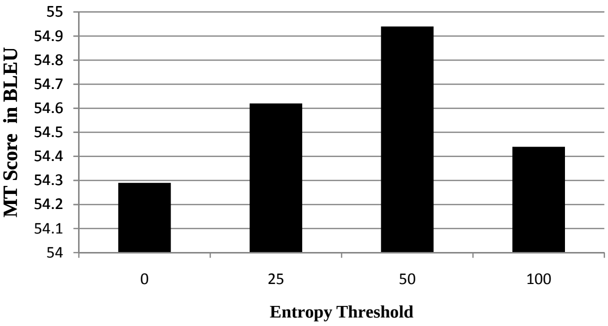

Figure 3: BLEU scores of the clustering experiments as a function of the entropy threshold on tuning set.

on unseen data, we compute the likelihood of the test data with respect to these probabilities with and without the decision tree splitting. We align the test set with its reference using GIZA++, and then obtain the link count $l\_count(s_i, t_j)$ for each alignment link $i = (s_{i,}t_i)$ in the set of alignment links I. We calculate the normalized likelihood of the alignments:

$$\begin{aligned} L &= \log\left[\left(\prod_i \bar{p}(t_i \mid s_i)^{l\_count(s_i,t_i)}\right)^{\frac{1}{|I|}}\right] \\ &= \frac{1}{|I|}\sum_{i\in I} l\_count(s_i,t_i)\log \bar{p}\,(t_i \mid s_i) \quad (6) \end{aligned}$$

where $\bar{p}\,(t_i \mid s_i)$ is the probability for the word pair (t_i, s_i) in equation (4). If the same instance of source word s_i is aligned to two target words t_i and t_j, then these two links are counted separately. If a source in the test set is out-of-vocabulary, or if a word pair (t_i, s_i) is aligned in the test alignment but not in the training alignments (and thus has no probability estimate), then it is ignored in the calculation of the log-likelihood.

Table 1 shows the likelihood for the baseline case, where one lexical translation probability distribution is used per source word. It also shows the likelihoods calculated using the lexical distributions in the leaf nodes of the decision trees, when either diacritics or part-of-speech are used as an attribute type. The table shows an increase in the likelihood of 2.98% using diacritics, and 3.41% using part-of-speech.

The translation result tables present MT scores in two different metrics: Translation Edit Rate (Snover et al., 2006) and IBM BLEU. The reader is reminded that a higher BLEU score and a lower TER are desired. The tables also show the difference in scores between the baseline and each experiment. It is worth noting that the gains reported are relative to a strong baseline that uses a state-of-the-art system with many features, and a fairly large training corpus.

	TER	**BLEU**
	Test	
baseline	**40.14**	**52.05**
full diacritics	40.31	52.39
	+0.17	+0.34
dec. trees, diac ($\theta_h = 50$)	39.75	52.60
	-0.39	+0.55

Table 2: Results of experiments using decision trees to cluster source words.

The decision tree clustering experiment as described in section **3.2** depends on a global parameter, namely the threshold in entropy reduction θ_h. We tune this parameter manually on a tuning set. Figure 3 shows the BLEU scores as a function of the threshold value, with diacritics as an attribute type. The most gain is obtained for an entropy threshold of 50.

The fully diacritized data has an average of 1.78 diacritized forms per source word. The average weighted by the number of occurrences is 6.28, which indicates that words with more diacritized forms tend to occur more frequently. After clustering using a value of $\theta_h = 50$, the average number of diacritized forms becomes 1.11, and the occurrence weighted average becomes 3.69. The clustering procedure thus seems to eliminate most diacritized forms, which likely do not contain helpful disambiguating information.

Table 2 lists the detailed results of experiments using diacritics. In the first experiment, we show that using full diacritization results in a small gain on the BLEU score and no gain on TER, which is somewhat consistent with the result obtained by Diab et al. (2007). The next experiment shows the results of clustering the diacritized source words using decision trees for the entropy threshold of 50. The TER loss of the full diacritics becomes a gain, and the BLEU gain increases. This confirms our speculation that the use of fully diacritized data in-

	TER	**BLEU**
	Test	
baseline	**40.14**	**52.05**
dec. trees, diacs	39.75	52.55
	-0.39	+0.50
dec. trees, POS	40.05	52.40
	-0.09	+0.35
dec. trees, diacs, no interpolation	39.98	52.09
	-0.16	+0.04

Table 3: Results of experiments using the word attribute-dependent lexical smoothing feature.

creases the model sparsity, which undoes most of the benefit obtained from the disambiguating information that the diacritics contain. Using the decision trees to cluster the diacritized source data prunes diacritized forms that do not decrease the entropy of the lexical translation probability distributions. It thus finds a sweet-spot between the negative effect of increasing the vocabulary size and the positive effect of disambiguation.

In our experiments, using diacritics with case endings gave consistently better score than using diacritics with no case endings, despite the fact that they result in a higher vocabulary size. One possible explanation is that diacritics not only help in lexical disambiguation, but they might also be indirectly helping in phrase reordering, since the diacritics on the final letter indicate the word's grammatical function.

The results from using decision trees to interpolate attribute-dependent lexical smoothing features are summarized in table 3. In the first experiment, we show the results of using diacritics to estimate the interpolated lexical translation probabilities. The results show a gain of +0.5 BLEU points and 0.39 TER points. The gain is statistically significant with a 95% confidence level. Using part-of-speech as an attribute gives a smaller, but still statistically significant gain. We also ran a control experiment, where we used diacritic-dependent lexical translation probabilities obtained from the decision trees, but did not perform the probability interpolation of equation (4). The gains mostly disappear, especially on BLEU, showing the importance of the interpolation step for the proper estimation of the lexical smoothing feature.

5 Conclusion and Future Directions

We presented in this paper a new method for incorporating explicit context-informed word attributes into SMT using binary decision trees. We reported on experiments on Arabic-to-English translation using diacritized Arabic and part-of-speech as word attributes, and showed that the use of these attributes increases the likelihood of source-target word pairs of unseen data. We proposed two specific ways in which the results of the decision tree training process are used in machine translation, and showed that they result in better translation results.

For future work, we plan on using multiple source-side attributes at the same time. Different attributes could have different disambiguating information, which could provide more benefit than using any of the attributes alone. We also plan on investigating the use of multi-word trees; trees for word clusters can for instance be grown instead of growing a separate tree for each source word. Although the experiments presented in this paper use local word attributes, nothing in principle prevents this method from being used with long-distance sentence context, or even with document-level or discourse-level features. Our future plans include the investigation of using such features as well.

Acknowledgment

This work was supported by DARPA/IPTO Contract No. HR0011-06-C-0022 under the GALE program.

References

S. Ananthakrishnan, S. Narayanan, and S. Bangalore. 2005. Automatic diacritization of arabic transcripts for automatic speech recognition. Kanpur, India.

R. Brent. 1973. *Algorithms for Minimization Without Derivatives*. Prentice-Hall.

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

J. Brunning, A. de Gispert, and W. Byrne. 2009. Context-dependent alignment models for statistical machine translation. In *NAACL '09: Proceedings of the 2009 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 110–118.

C. Cabezas and P. Resnick. 2005. Using WSD techniques for lexical selection in statistical machine translation. In *Technical report, Institute for Advanced Computer Studies (CS-TR-4736, LAMP-TR-124, UMIACS-TR-2005-42)*, College Park, MD.

M. Carpuat and D. Wu. 2007. Improving statistical machine translation using word sense disambiguation. In *EMNLP-CoNLL-2007: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Prague, Czech Republic.

Y. Chan, H. Ng, and D. Chiang. 2007. Word sense disambiguation improves statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*.

J. Devlin. 2009. Lexical features for statistical machine translation. Master's thesis, University of Maryland, December 2009.

M. Diab, M. Ghoneim, and N. Habash. 2007. Arabic diacritization in the context of statistical machine translation. In *MT Summit XI*, pages 143–149, Copenhagen, Denmark.

R. O. Duda, P. E. Hart, and D. G. Stork. 2000. *Pattern Classification*. Wiley-Interscience Publication.

K. Gimpel and N. A. Smith. 2008. Rich source-side context for statistical machine translation. In *StatMT '08: Proceedings of the Third Workshop on Statistical Machine Translation*, pages 9–17, Columbus, Ohio.

N. Habash and O. Rambow. 2007. Arabic diacritization through full morphological tagging. In *Proceedings of the 2007 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 53–56, Rochester, New York.

P. Koehn, F. J. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 48–54, Edmonton, Canada.

R. Nelken and S. M. Shieber. 2005. Arabic diacritization using weighted finite-state transducers. In *Proceedings of the 2005 ACL Workshop on Computational Approaches to Semitic Languages*, Ann Arbor, Michigan.

F. J. Och and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

F. J. Och, D. Gildea, S. Khudanpur, A. Sarkar, K. Yamada, A. Fraser, S. Kumar, L. Shen, D. Smith, K. Eng, V. Jain, Z. Jin, and D. R. Radev. 2004. A smorgasbord of features for statistical machine translation. In *HLT-NAACL*, pages 161–168.

F. J. Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, Sapporo, Japan.

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

Ruhi Sarikaya, Yonggang Deng, and Yuqing Gao. 2007. Context dependent word modeling for statistical machine translation using part-of-speech tags. In *Proceedings of INTERSPEECH 2007fs*, Antwerp, Belgium.

L. Shen, J. Xu, and R. Weischedel. 2008. A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL)*, Columbus, Ohio.

M. Snover, B. Dorr, R. Schwartz, J. Makhoul, and L. Micciulla. 2006. A study of translation error rate with targeted human annotation. In *Proceedings of the 7th Conf. of the Association for Machine Translation in the Americas (AMTA 2006)*, pages 223–231, Cambridge, MA.

N. Stroppa, A. van den Bosch, and A Way. 2007. Exploiting source similarity for SMT using context-informed features. In *Proceedings of* *the 11th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-07)*, pages 231–240.

D. Vergyri and K. Kirchhoff. 2004. Automatic diacritization of arabic for acoustic modeling in speech recognition. In *Semitic '04: Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages*, pages 66–73, Geneva, Switzerland.

D. Vickrey, L. Biewald, M. Teyssier, and D. Koller. 2005. Word-sense disambiguation for machine translation. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, Vancouser, BC, Canada.

S.J. Young, J.J. Odell, and P.C. Woodland. 1994. Tree-based state tying for high accuracy acoustic modelling. In *HLT'94: Proceedings of the Workshop on Human Language Technology*, pages 307–312.

I. Zitouni, J. S. Sorensen, and Ruhi Sarikaya. 2006. Maximum entropy based restoration of arabic diacritics. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 577–584, Sydney, Australia.