Using a maximum entropy model to build segmentation lattices for MT

Chris Dyer
Laboratory for Computational Linguistics and Information Processing
Department of Linguistics
University of Maryland
College Park, MD 20742, USA
redpony AT umd.edu

Abstract

Recent work has shown that translating segmentation lattices (lattices that encode alternative ways of breaking the input to an MT system into words), rather than text in any particular segmentation, improves translation quality of languages whose orthography does not mark morpheme boundaries. However, much of this work has relied on multiple segmenters that perform differently on the same input to generate sufficiently diverse source segmentation lattices. In this work, we describe a maximum entropy model of compound word splitting that relies on a few general features that can be used to generate segmentation lattices for most languages with productive compounding. Using a model optimized for German translation, we present results showing significant improvements in translation quality in German-English, Hungarian-English, and Turkish-English translation over state-of-the-art baselines.

1 Introduction

Compound words pose significant challenges to the lexicalized models that are currently common in statistical machine translation. This problem has been widely acknowledged, and the conventional solution, which has been shown to work well for many language pairs, is to segment compounds into their constituent morphemes using either morphological analyzers or empirical methods and then to translate from or to this segmented variant (Koehn et al., 2008; Dyer et al., 2008; Yang and Kirchhoff, 2006).

But into what units should a compound word be segmented? Taken as a stand-alone task, the goal of a compound splitter is to produce a segmentation for some input that matches the linguistic intuitions of a native speaker of the language. However, there are often advantages to using elements larger than single morphemes as the minimal lexical unit for MT, since they may correspond more closely to the units of translation. Unfortunately, determining the optimal segmentation is challenging, typically requiring extensive experimentation (Koehn and Knight, 2003; Habash and Sadat, 2006; Chang et al., 2008). Recent work has shown that by combining a variety of segmentations of the input into a *segmentation lattice* and effectively marginalizing over many different segmentations, translations superior to those resulting from any single single segmentation of the input can be obtained (Xu et al., 2005; Dyer et al., 2008; DeNeefe et al., 2008). Unfortunately, this approach is difficult to utilize because it requires multiple segmenters that behave differently on the same input.

In this paper, we describe a maximum entropy word segmentation model that is trained to assign high probability to possibly several segmentations of an input word. This model enables generation of diverse, accurate segmentation lattices from a single model that are appropriate for use in decoders that accept word lattices as input, such as Moses (Koehn et al., 2007). Since our model relies a small number of dense features, its parameters can be tuned using very small amounts of manually created *reference lattices*. Furthermore, since these parameters were chosen to have valid interpretation across a variety of languages, we find that the weights estimated for one apply quite well to another. We show that these lattices significantly improve translation quality when translating into English from three languages exhibiting productive compounding: German, Turkish, and Hungarian.

The paper is structured as follows. In the next sec-

Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the ACL, pages 406–414,
Boulder, Colorado, June 2009.

tion, we describe translation from segmentation lattices and give a motivating example, Section 3 describes our segmentation model and its tuning and how it is used to generate segmentation lattices, Section 5 presents experimental results, Section 6 reviews relevant related work, and in Section 7 we conclude and discuss future work.

2 Segmentation lattice translation

In this section we give a brief overview of lattice translation and then describe the characteristics of segmentation lattices that are appropriate for translation.

2.1 Lattice translation

Word lattices have been used to represent ambiguous input to machine translation systems for a variety of tasks, including translating automatic speech recognition transcriptions and translating from morphologically complex languages (Bertoldi et al., 2007; Dyer et al., 2008). The intuition behind using lattices in both approaches is to avoid the error propagation effects that are found when a one-best guess is used. By carrying a certain amount of uncertainty forward in the processing pipeline, information contained in the translation models can be leveraged to help resolve the upstream ambiguity. In our case, we want to propagate uncertainty about the proper segmentation of a compound forward to the decoder, which can use its full translation model to select proper segmentation for translation. Mathematically, this can be understood as follows: whereas the goal in conventional machine translation is to find the sentence $\hat{e}_1^I$ that maximizes $Pr(e_1^I|f_1^J)$, the lattice adds a latent variable, the path $\bar{f}$ from a designated start start to a designated goal state in the lattice $\mathcal{G}$:

$$\hat{e}_1^I = \arg\max_{e_1^I} Pr(e_1^I|\mathcal{G}) \quad (1)$$

$$= \arg\max_{e_1^I} \sum_{\bar{f}\in\mathcal{G}} Pr(e_1^I|\bar{f})Pr(\bar{f}|\mathcal{G}) \quad (2)$$

$$\approx \arg\max_{e_1^I} \max_{\bar{f}\in\mathcal{G}} Pr(e_1^I|\bar{f})Pr(\bar{f}|\mathcal{G}) \quad (3)$$

If the transduction formalism used is a synchronous probabilistic context free grammar or weighted finite state transducer, the search represented by equation (3) can be carried out efficiently using dynamic programming (Dyer et al., 2008).

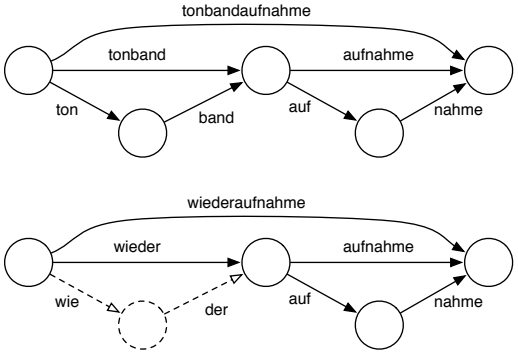

Figure 1: Segmentation lattice examples. The dotted structure indicates linguistically implausible segmentation that might be generated using dictionary-driven approaches.

2.2 Segmentation lattices

Figure 1 shows two lattices that encode the most linguistically plausible ways of segmenting two prototypical German compounds with compositional meanings. However, while these words are structurally quite similar, translating them into English would seem to require different amounts of segmentation. For example, the dictionary fragment shown in Table 1 illustrates that *tonbandaufnahme* can be rendered into English by following 3 different paths in the lattice, *ton*/audio *band*/tape *aufnahme*/recording, *tonband*/tape *aufnahme*/recording, and *tonbandaufnahme*/tape recording. In contrast, *wiederaufnahme* can only be translated correctly using the unsegmented form, even though in German the meaning of the full form is a composition of the meaning of the individual morphemes.[1]

It should be noted that phrase-based models can translate multiple words as a unit, and therefore capture non-compositional meaning. Thus, by default if the training data is processed such that, for example, *aufnahme*, in its sense of *recording*, is segmented into two words, then more paths in the lattices be-

[1]The English word *resumption* is likewise composed of two morphemes, the prefix *re-* and a kind of bound morpheme that never appears in other contexts (sometimes called a 'cranberry' morpheme), but the meaning of the whole is idiosyncratic enough that it cannot be called compositional.

German	English
auf	*on, up, in, at, ...*
aufnahme	*recording, entry*
band	*reel, tape, band*
der	*the, of the*
nahme	*took* (3P-SG-PST)
ton	*sound, audio, clay*
tonband	*tape, audio tape*
tonbandaufnahme	*tape recording*
wie	*how, like, as*
wieder	*again*
wiederaufnahme	*resumption*

Table 1: German-English dictionary fragment for words present in Figure 1.

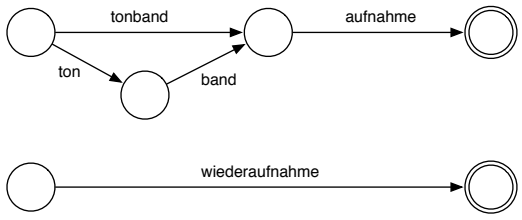

Figure 2: Manually created reference lattices for the two words from Figure 1. Although only a subset of all linguistically plausible segmentations, each path corresponds to a plausible segmentation for word-for-word German-English translation.

come plausible translations. However, using a strategy of "over segmentation" and relying on phrase models to learn the non-compositional translations has been shown to degrade translation quality significantly on several tasks (Xu et al., 2004; Habash and Sadat, 2006). We thus desire lattices containing as little oversegmentation as possible.

We have now have a concept of a "gold standard" segmentation lattice for translation: it should contain all linguistically motivated segmentations that also correspond to plausible word-for-word translations into English. Figure 2 shows an example of the reference lattice for the two words we just discussed. For the experiments in this paper, we generated a development and test set by randomly choosing 19 German newspaper articles, identifying all words greater than 6 characters is length, and segmenting each word so that the resulting units could be translated compositionally into English. This resulted in 489 training sentences corresponding to 564 paths for the dev set (which was drawn from 15 articles), and 279 words (302 paths) for the test set (drawn from the remaining 4 articles).

3 A maximum entropy segmentation model

We now turn to the problem of modeling word segmentation in a way that facilitates lattice construction. As a starting point, we consider the work of Koehn and Knight (2003) who observe that in most languages that exhibit compounding, the morphemes used to construct compounds frequently also appear as individual tokens. Based on this observation, they propose a model of word segmentation that splits compound words into pieces found in the dictionary based on a variety heuristic scoring criteria. While these models have been reasonably successful (Koehn et al., 2008), they are problematic for two reasons. First, there is no principled way to incorporate additional features (such as phonotactics) which might be useful to determining whether a word break should occur. Second, the heuristic scoring offers little insight into which segmentations should be included in a lattice.

We would like our model to consider a wide variety of segmentations of any word (including perhaps hypothesized morphemes that are not in the dictionary), to make use of a rich set of features, and to have a probabilistic interpretation of each hypothesized split (to incorporate into the downstream decoder). We decided to use the class of maximum entropy models, which are probabilistically sound, can make use of possibly many overlapping features, and can be trained efficiently (Berger et al., 1996). We thus define a model of the conditional probability distribution $Pr(s_1^N|w)$, where w is a surface form and s_1^N is the segmented form consisting of N segments as:

$$Pr(s_1^N|w) = \frac{\exp \sum_i \lambda_i h_i(s_1^N, w)}{\sum_{s'} \exp \sum_i \lambda_i h_i(s', w)} \quad (4)$$

To simplify inference and to make the lattice representation more natural, we only make use of local feature functions that depend on properties of each segment:

$$Pr(s_1^N|w) \propto \exp \sum_i \lambda_i \sum_j^N h_i(s_j, w) \quad (5)$$

3.1 From model to segmentation lattice

The segmentation model just introduced is equivalent to a lattice where each vertex corresponds to a particular coverage (in terms of letters consumed from left to right) of the input word. Since we only make use of local features, the number of vertices in a lattice for word w is $|w| - m$, where m is the minimum segment length permitted. In all experiments reported in this paper, we use $m = 3$. Each edge is labeled with a morpheme s (corresponding to the morpheme associated with characters delimited by the start and end nodes of the edge) as well as a weight, $\sum_i \lambda_i h_i(s, w)$. The cost of any path from the start to the goal vertex will be equal to the numerator in equation (4). The value of the denominator can be computed using the forward algorithm.

In most of our experiments, s will be identical to the substring of w that the edge is designated to cover. However, this is not a requirement. For example, German compounds frequently have so-called *Fugenelemente*, one or two characters that "glue together" the primary morphemes in a compound. Since we permit these characters to be deleted, then an edge where they are deleted will have fewer characters than the coverage indicated by the edge's starting and ending vertices.

3.2 Lattice pruning

Except for the minimum segment length restriction, our model defines probabilities for all segmentations of an input word, making the resulting segmentation lattices are quite large. Since large lattices are costly to deal with during translation (and may lead to worse translations because poor segmentations are passed to the decoder), we prune them using forward-backward pruning so as to contain just the highest probability paths (Sixtus and Ortmanns, 1999). This works by computing the score of the best path passing through every edge in the lattice using the forward-backward algorithm. By finding the best score overall, we can then prune edges using a threshold criterion; i.e., edges whose score is some factor α away from the global best edge score.

3.3 Maximum likelihood training

Our model defines a conditional probability distribution over virtually all segmentations of a word w. To train our model, we wish to maximize the likelihood of the segmentations contained in the reference lattices by moving probability mass away from the segmentations that are *not* in the reference lattice. Thus, we wish to minimize the following objective (which can be computed using the forward algorithm over the unpruned hypothesis lattices):

$$\mathcal{L} = -\log \sum_i \sum_{\mathbf{s} \in \mathcal{R}_i} p(\mathbf{s}|w_i) \quad (6)$$

The gradient with respect to the feature weights for a log linear model is simply:

$$\frac{\partial \mathcal{L}}{\partial \lambda_k} = \sum_i \mathbb{E}_{p(\mathbf{s}|w_i)}[h_k] - \mathbb{E}_{p(\mathbf{s}|w_i, \mathcal{R}_i)}[h_k] \quad (7)$$

To compute these values, the first expectation is computed using forward-backward inference over the full lattice. To compute the second expectation, the full lattice is intersected with the reference lattice $\mathcal{R}_i$, and then forward-backward inference is redone.[2] We use the standard quasi-Newtonian method L-BFGS to optimize the model (Liu et al., 1989). Training generally converged in only a few hundred iterations.

3.3.1 Training to minimize 1-best error

In some cases, such as when performing word alignment for translation model construction, lattices cannot be used easily. In these cases, a 1-best segmentation (which can be determined from the lattice using the Viterbi algorithm) may be desired. To train the parameters of the model for this condition (which is arguably slightly different from the lattice generation case we just considered), we used the minimum error training (MERT) algorithm on the segmentation lattices to find the parameters that minimized the error on our dev set (Macherey

[2]The second expectation corresponds to the empirical feature observations in a standard maximum entropy model. Because this is an expectation and not an invariant observation, the log likelihood function is not guaranteed to be concave and the objective surface may have local minima. However, experimentation revealed the optimization performance was largely invariant with respect to its starting point.

et al., 2008). The error function we used was WER (the minimum number of insertions, substitutions, and deletions along any path in the reference lattice, normalized by the length of this path). The WER on the held-out test set for a system tuned using MERT is 9.9%, compared to 11.1% for maximum likelihood training.

3.4 Features

We remark that since we did not have the resources to generate training data in all the languages we wished to generate segmentation lattices for, we have confined ourselves to features that we expect to be reasonably informative for a broad class of languages. A secondary advantage of this is that we used denser features than are often used in maximum entropy modeling, meaning that we could train our model with relatively less training data than might otherwise be required.

The features we used in our compound segmentation model for the experiments reported below are shown in Table 2. Building on the prior work that relied heavily on the frequency of the hypothesized constituent morphemes in a monolingual corpus, we included features that depend on this value, $f(s_i)$. $|s_i|$ refers to the number of letters in the ith hypothesized segment. Binary predicates evaluate to 1 when true and 0 otherwise. $f(s_i)$ is the frequency of the token s_i as an independent word in a monolingual corpus. $p(\#|s_{i1} \cdots s_{i4})$ is the probability of a word start preceding the letters $s_{i1} \cdots s_{i4}$. We found it beneficial to include a feature that was the probability of a certain string of characters beginning a word, for which we used a reverse 5-gram character model and predicted the word boundary given the first five letters of the hypothesized word split.[3] Since we did have expertise in German morphology, we did build a special German model. For this, we permitted the strings *s*, *n*, and *es* to be deleted between words. Each deletion fired a count feature (listed as *fugen* in the table). Analysis of errors indicated that the segmenter would periodically propose an incorrect segmentation where a single word could be divided into a word and a nonword consisting of common inflectional suffixes. To address this, an additional feature was added that fired when a proposed segment was one of a set $\mathcal{N}$ of 30 nonwords that we saw quite frequently. The weights shown in Table 2 are those learned by maximum likelihood training on models both with and without the special German features, which are indicated with †.

Feature	de-only	neutral
$^{\dagger}s_i \in \mathcal{N}$	-3.55	–
$f(s_i) > 0.005$	-3.13	-3.31
$f(s_i) > 0$	3.06	3.64
$\log p(\#\|s_{i1}s_{i2}s_{i3}s_{i4})$	-1.58	-2.11
segment penalty	1.18	2.04
$\|s_i\| \geq 12$	-0.9	-0.79
oov	-0.88	-1.09
†*fugen*	-0.76	–
$\|s_i\| \leq 4$	-0.66	-1.18
$\|s_i\| \leq 10,\ f(s_i) > 2^{-10}$	-0.51	-0.82
$\log f(s_i)$	-0.32	-0.36
$2^{-10} < f(s_i) < 0.005$	-0.26	-0.45

Table 2: Features and weights learned by maximum likelihood training, sorted by weight magnitude.

4 Model evalatuion

To give some sense of the performance of the model in terms of its ability to generate lattices independently of a translation task, we present precision and recall of segmentations for pruning parameters (cf. Section 3.2) ranging from $\alpha = 0$ to $\alpha = 5$. Precision measures the number of paths in the hypothesized lattice that correspond to paths in the reference lattice; recall measures the number of paths in the reference lattices that are found in the hypothesis lattice. Figure 3 shows the effect of manipulating the density parameter on the precision and recall of the German lattices. Note that very high recall is possible; however, the German-only features have a significant impact, especially on recall, because the reference lattices include paths where *Fugenelemente* have been deleted.

5 Translation experiments

We now review experiments using segmentation lattices produced by the segmentation model we just introduced in German-English, Hungarian-English,

[3] In general, this helped avoid situations where a word may be segemented into a frequent word and then a non-word string of characters since the non-word typically violated the phonotactics of the language in some way.

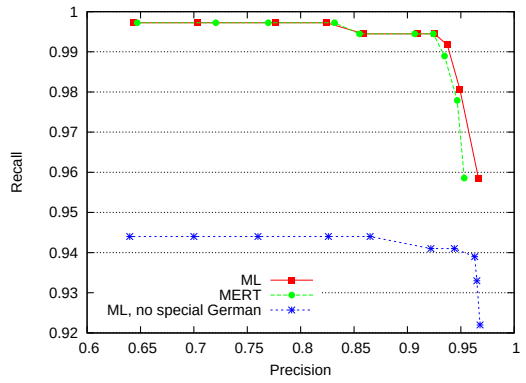

Figure 3: The effect of the lattice density parameter on precision and recall.

and Turkish-English translation tasks and then show results elucidating the effect of the lattice density parameter. We begin with a description of our MT system.

5.1 Data preparation and system description

For all experiments, we used a 5-gram English language model trained on the AFP and Xinua portions of the Gigaword v3 corpus (Graff et al., 2007) with modified Kneser-Ney smoothing (Kneser and Ney, 1995). The training, development, and test data for German-English and Hungarian-English systems used were distributed as part of the 2009 EACL Workshop on Machine Translation,[4] and the Turkish-English data corresponds to the training and test sets used in the work of Oflazer and Durgar El-Kahlout (2007). Corpus statistics for all language pairs are summarized in Table 3. We note that in all language pairs, the 1BEST segmentation variant of the training data results in a significant reduction in types.

Word alignment was carried out by running Giza++ implementation of IBM Model 4 initialized with 5 iterations of Model 1, 5 of the HMM aligner, and 3 iterations of Model 4 (Och and Ney, 2003) in both directions and then symmetrizing using the `grow-diag-final-and` heuristic (Koehn et al., 2003). For each language pair, the corpus was aligned twice, once in its non-segmented variant and once using the single-best segmentation variant.

For translation, we used a bottom-up parsing decoder that uses cube pruning to intersect the language model with the target side of the synchronous grammar. The grammar rules were extracted from the word aligned parallel corpus and scored as described in Chiang (2007). The features used by the decoder were the English language model log probability, $\log f(\bar{e}|\bar{f})$, the 'lexical translation' log probabilities in both directions (Koehn et al., 2003), and a word count feature. For the lattice systems, we also included the unnormalized $\log p(\bar{f}|\mathcal{G})$, as it is defined in Section 3, as well as an *input* word count feature. The feature weights were tuned on a held-out development set so as to maximize an equally weighted linear combination of BLEU and 1-TER (Papineni et al., 2002; Snover et al., 2006) using the minimum error training algorithm on a packed forest representation of the decoder's hypothesis space (Macherey et al., 2008). The weights were independently optimized for each language pair and each experimental condition.

5.2 Segmentation lattice results

In this section, we report the results of an experiment to see if the compound lattices constructed using our maximum entropy model yield better translations than either an unsegmented baseline or a baseline consisting of a single-best segmentation.

For each language pair, we define three conditions: BASELINE, 1BEST, and LATTICE. In the BASELINE condition, a lowercased and tokenized (but not segmented) version of the test data is translated using the grammar derived from a non-segmented training data. In the 1BEST condition, the single best segmentation $\hat{s}_1^N$ that maximizes $Pr(s_1^N|w)$ is chosen for each word using the MERT-trained model (the German model for German, and the language-neutral model for Hungarian and Turkish). This variant is translated using a grammar induced from a parallel corpus that has also been segmented according to the same decision rule. In the LATTICE condition, we constructed segmentation lattices using the technique described in Section 3.1. For all languages pairs, we used $d = 2$ as the pruning density parameter (which corresponds to the highest F-score on the held out test set). Additionally, if the unsegmented form of the word was removed from the lattice during pruning, it was restored to the lattice with zero weight.

Table 4 summarizes the results of the translation

[4] http://www.statmt.org/wmt09

	f-tokens	f-types	e-tokens.	e-types
DE-BASELINE	38M	307k	40M	96k
DE-1BEST	40M	136k	”	”
HU-BASELINE	25M	646k	29M	158k
HU-1BEST	27M	334k	”	”
TR-BASELINE	1.0M	56k	1.3M	23k
TR-1BEST	1.1M	41k	”	”

Table 3: Training corpus statistics.

	BLEU	TER
DE-BASELINE	21.0	60.6
DE-1BEST	20.7	60.1
DE-LATTICE	**21.6**	**59.8**
HU-BASELINE	11.0	71.1
HU-1BEST	10.7	70.4
HU-LATTICE	**12.3**	**69.1**
TR-BASELINE	26.9	61.0
TR-1BEST	27.8	61.2
TR-LATTICE	**28.7**	**59.6**

Table 4: Translation results for German (DE)-English, Hungarian (HU)-English, and Turkish (TR)-English. Scores were computed using a single reference and are case insensitive.

experiments comparing the three input variants. For all language pairs, we see significant improvements in both BLEU and TER when segmentation lattices are used.[5] Additionally, we also confirmed previous findings that showed that when a large amount of training data is available, moving to a one-best segmentation does not yield substantial improvements (Yang and Kirchhoff, 2006). Perhaps most surprisingly, the improvements observed when using lattices with the Hungarian and Turkish systems were *larger* than the corresponding improvement in the German system, but German was the only language for which we had segmentation training data. The smaller effect in German is probably due to there being more in-domain training data in the German system than in the (otherwise comparably sized) Hungarian system.

Targeted analysis of the translation output shows that while both the 1BEST and LATTICE systems generally produce adequate translations of compound words that are out of vocabulary in the BASELINE system, the LATTICE system performs better since it recovers from infelicitous splits that the one-best segmenter makes. For example, one class of error we frequently observe is that the one-best segmenter splits an OOV proper name into two pieces when a portion of the name corresponds to a known word in the source language (e.g. *tom tancredo→tom tan credo* which is then translated as *tom tan belief*).[6]

5.3 The effect of the density parameter

Figure 4 shows the effect of manipulating the density parameter (cf. Section 3.2) on the performance and decoding time of the Turkish-English translation system. It further confirms the hypothesis that increased diversity of segmentations encoded in a segmentation lattice can improve translation performance; however, it also shows that once the density becomes too great, and too many implausible segmentations are included in the lattice, translation quality will be harmed.

6 Related work

Aside from improving the vocabulary coverage of machine translation systems (Koehn et al., 2008; Yang and Kirchhoff, 2006; Habash and Sadat, 2006), compound word segmentation (also referred to as *decompounding*) has been shown to be helpful in a variety of NLP tasks including mono- and

[5]Using bootstrap resampling (Koehn, 2004), the improvements in BLEU, TER, as well as the linear combination used in tuning are statistically significant at at least $p < .05$.

[6]We note that our maximum entropy segmentation model could easily address this problem by incorporating information about whether a word is likely to be a named entity as a feature.

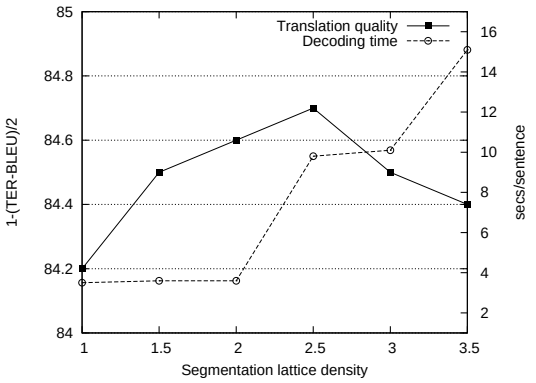

Figure 4: The effect of the lattice density parameter on translation quality and decoding time.

crosslingual IR (Airio, 2006) and speech recognition (Hessen and Jong, 2003). A number of researchers have demonstrated the value of using lattices to encode segmentation alternatives as input to a machine translation system (Dyer et al., 2008; DeNeefe et al., 2008; Xu et al., 2004), but this is the first work to do so using a single segmentation model. Another strand of inquiry that is closely related is the work on adjusting the source language segmentation to match the granularity of the target language as a way of improving translation. The approaches suggested thus far have been mostly of a heuristic nature tailored to Chinese-English translation (Bai et al., 2008; Ma et al., 2007).

7 Conclusions and future work

In this paper, we have presented a maximum entropy model for compound word segmentation and used it to generate segmentation lattices for input into a statistical machine translation system. These segmentation lattices improve translation quality (over an already strong baseline) in three typologically distinct languages (German, Hungarian, Turkish) when translating into English. Previous approaches to generating segmentation lattices have been quite laborious, relying either on the existence of multiple segmenters (Dyer et al., 2008; Xu et al., 2005) or hand-crafted rules (DeNeefe et al., 2008). Although the segmentation model we propose is discriminative, we have shown that it can be trained using a minimal amount of annotated training data. Furthermore, when even this minimal data cannot be acquired for a particular language (as was the situation we faced with Hungarian and Turkish), we have demonstrated that the parameters obtained in one language work surprisingly well for others. Thus, with virtually no cost, this model can be used with a variety of diverse languages.

While these results are already quite satisfying, there are a number of compelling extensions to this work that we intend to explore in the future. First, unsupervised segmentation approaches offer a very compelling alternative to the manually crafted segmentation lattices that we created. Recent work suggests that unsupervised segmentation of inflectional affixal morphology works quite well (Poon et al., 2009), and extending this work to compounding morphology should be feasible, obviating the need for expensive hand-crafted reference lattices. Second, incorporating target language information into a segmentation model holds considerable promise for inducing more effective translation models that perform especially well for segmentation lattice inputs.

Acknowledgments

Special thanks to Kemal Oflazar and Reyyan Yeniterzi of Sabancı University for providing the Turkish-English corpus and to Philip Resnik, Adam Lopez, Trevor Cohn, and especially Phil Blunsom for their helpful suggestions. This research was supported by the Army Research Laboratory. Any opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the view of the sponsors.

References

Eija Airio. 2006. Word normalization and decompounding in mono- and bilingual IR. *Information Retrieval*, 9:249–271.

Ming-Hong Bai, Keh-Jiann Chen, and Jason S. Chang. 2008. Improving word alignment by adjusting Chinese word segmentation. In *Proceedings of the Third International Joint Conference on Natural Language Processing*.

A.L. Berger, V.J. Della Pietra, and S.A. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

N. Bertoldi, R. Zens, and M. Federico. 2007. Speech

translation by confusion network decoding. In *Proceeding of ICASSP 2007*, Honolulu, Hawaii, April.

Pi-Chuan Chang, Dan Jurafsky, and Christopher D. Manning. 2008. Optimizing Chinese word segmentation for machine translation performance. In *Proceedings of the Third Workshop on Statistical Machine Translation*, Prague, Czech Republic, June.

D. Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

S. DeNeefe, U. Hermjakob, and K. Knight. 2008. Overcoming vocabulary sparsity in mt using lattices. In *Proceedings of AMTA*, Honolulu, HI.

C. Dyer, S. Muresan, and P. Resnik. 2008. Generalizing word lattice translation. In *Proceedings of HLT-ACL*.

D. Graff, J. Kong, K. Chen, and K. Maeda. 2007. English gigaword third edition.

N. Habash and F. Sadat. 2006. Arabic preprocessing schemes for statistical machine translation. In *Proc. of NAACL*, New York.

Arjan Van Hessen and Franciska De Jong. 2003. Compound decomposition in dutch large vocabulary speech recognition. In *Proceedings of Eurospeech 2003, Geneve*, pages 225–228.

R. Kneser and H. Ney. 1995. Improved backing-off for m-gram language modeling. In *Proceedings of IEEE Internation Conference on Acoustics, Speech, and Signal Processing*, pages 181–184.

P. Koehn and K. Knight. 2003. Empirical methods for compound splitting. In *Proc. of the EACL 2003*.

P. Koehn, F.J. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *Proceedings of NAACL 2003*, pages 48–54, Morristown, NJ, USA. Association for Computational Linguistics.

P. Koehn, H. Hoang, A. Birch Mayne, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Annual Meeting of the Association for Computation Linguistics (ACL), Demonstration Session*, pages 177–180, June.

Philipp Koehn, Abhishek Arun, and Hieu Hoang. 2008. Towards better machine translation quality for the German-English language pairs. In *ACL Workshop on Statistical Machine Translation*.

P. Koehn. 2004. Statistical signficiance tests for machine translation evluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 388–395.

Dong C. Liu, Jorge Nocedal, Dong C. Liu, and Jorge Nocedal. 1989. On the limited memory BFGS method for large scale optimization. *Mathematical Programming B*, 45(3):503–528.

Yanjun Ma, Nicolas Stroppa, and Andy Way. 2007. Bootstrapping word alignment via word packing. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 304–311, Prague, Czech Republic, June. Association for Computational Linguistics.

Wolfgang Macherey, Franz Josef Och, Ignacio Thayer, and Jakob Uszkoreit. 2008. Lattice-based minimum error rate training for statistical machine translation. In *Proceedings of EMNLP*, Honolulu, HI.

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

Kemal Oflazer and Ilknur Durgar El-Kahlout. 2007. Exploring different representational units in English-to-Turkish statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 25–32, Prague, Czech Republic, June. Association for Computational Linguistics.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the ACL*, pages 311–318.

Hoifung Poon, Colin Cherry, and Kristina Toutanova. 2009. Unsupervised morphological segmentation with log-linear models. In *Proc. of NAACL 2009*.

S. Sixtus and S. Ortmanns. 1999. High quality word graphs using forward-backward pruning. In *Proceedings of ICASSP*, Phoenix, AZ.

Matthew Snover, Bonnie J. Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*.

J. Xu, R. Zens, and H. Ney. 2004. Do we need Chinese word segmentation for statistical machine translation? In *Proceedings of the Third SIGHAN Workshop on Chinese Language Learning*, pages 122–128, Barcelona, Spain.

J. Xu, E. Matusov, R. Zens, and H. Ney. 2005. Integrated Chinese word segmentation in statistical machine translation. In *Proc. of IWSLT 2005*, Pittsburgh.

M. Yang and K. Kirchhoff. 2006. Phrase-based backoff models for machine translation of highly inflected languages. In *Proceedings of the EACL 2006*, pages 41–48.